J Intell Manuf (2011) 22:789–799
DOI 10.1007/s10845-009-0311-9

An exact algorithm for vehicle routing and scheduling problem of free pickup and delivery service in flight ticket sales companies based on set-partitioning model

Gang Dong · Jiafu Tang · Kin Keung Lai · Yuan Kong

Received: 5 December 2008 / Accepted: 21 August 2009 / Published online: 16 September 2009

Abstract This paper addresses a vehicle routing and scheduling problem arising in Flight Ticket Sales Companies for the service of free pickup and delivery of airline passengers to the airport. The problem is formulated under the framework of Vehicle Routing Problem with Time Windows (VRPTW), with the objective of minimizing the total operational costs, i.e. fixed start-up costs and variable traveling costs. A 0–1 mixed integer programming model is presented, in which service quality is factored in constraints by introducing passenger satisfaction degree functions that limit time deviations between actual and desired delivery times. The problem addressed in this paper has two distinctive characteristics—small vehicle capacities and tight delivery time windows. An exact algorithm based on the set-partitioning model, concerning both characteristics, is developed. In the first phase of the algorithm the entire candidate set of best feasible routes is generated, and then the optimal solution is obtained by solving the set-partitioning model in the second phase. Finally, we use four actual instances to illustrate application of the proposed algorithm. Moreover, the proposed algorithm is applied to a random instance containing more orders to verify the general effectiveness of the proposed algorithm even if the number of passengers increases in future.

G. Dong (✉) · K. K. Lai
Department of Management Sciences,
City University of HongKong, Kowloon, Hong Kong
e-mail: gdongms@cityu.edu.hk

J. Tang · Y. Kong
Institute of Systems Engineering, Northeastern University,
Shenyang, China

Keywords Routing and scheduling · Free pickup and delivery service · Flight ticket sales company · Exact Algorithm · Set-partitioning model

Introduction

High economic growth has resulted in greater attention being paid to the services industry, particularly transportation services. Success of a transportation service provider depends on many factors, such as operational costs and service quality. Optimization of operational costs is always a major concern of service providers when dealing with transportation problems. However, service quality, which is often neglected, is equally critical to success in earning market share and enhancing competitiveness. Unfortunately, high service quality is incompatible with low operational costs more often than not. Therefore, operating a transportation service with both low operational costs and high service quality has become an important issue for most decision makers, especially those who have to make daily decisions.

Flight Ticket Sales Companies (FTSCs) operate as typical service companies in China's aviation industry. Their major functions include ticket sales, flight line designs, and free delivery of tickets to passengers. There is fierce competition among airline companies (China Southern Airlines Co., Ltd., China Eastern Airlines Co., Ltd. and others), as well as among flight ticket sales agencies, i.e. FTSCs, in mainland China. To compete successfully in the market, some value-added services are provided by FTSCs to passengers, such as *Free Pickup and Delivery of Passengers to the Airport Service* (FPDS hereinafter), which is a relatively new phenomenon. Under FPDS, passengers reserving tickets from FTSCs are picked up at conveniently designated points and are delivered at the airport at the desired time.

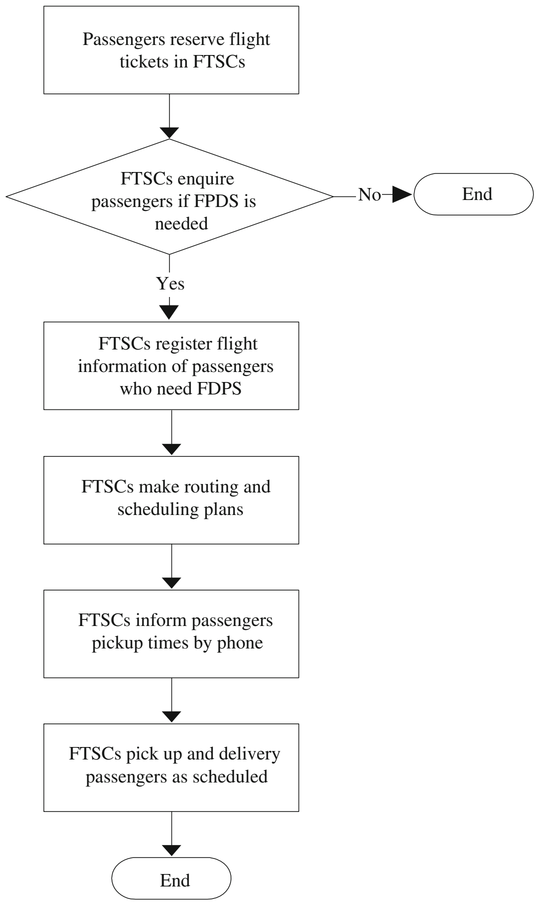

Fig. 1 Basic operation of FPDS

The basic operation of FPDS is illustrated in Fig. 1. First, passengers make ticket reservations from FTSCs through the Internet, phone or at their counters. If the passenger needs FPDS, his/her flight information, including flight number, date, departure time, required pickup point and desired delivery time are recorded. Normally, the pickup points are determined by negotiation between FTSCs and passengers. However, a compromise is needed sometimes because passengers may indicate a pickup point where car parking is forbidden, or finding the point will be too time-consuming for the vehicle. The desired delivery time could be stipulated by passengers; it could be any time of the day. However, in general cases, the delivery time is fixed according to the flight departure time, unless a passenger has some special request. Passengers of domestic flights would be delivered at the airport about 1 h before flight departure time, while passengers of international flights would be delivered about one-and-a-half hours before departure. According to the above information, FTSCs make routing and scheduling plans to transport all the passengers who need FPDS, for each day. Passengers are informed of their pickup time a couple of hours before their flight departure time by phone. Finally, passengers are picked up and delivered at the airport as scheduled.

Obviously, making the routing and scheduling plan is the most difficult part of the process. Generally, there would be about 80 FPDS orders in a mid-size FTSC on each service day. The passengers are located in different areas and expect different pickup and delivery times. Passengers with the same expected pickup point and the same desired delivery time are regarded as one single order. It is not difficult for FTSCs to make routing and scheduling plans to serve all the orders but making a plan with both low operational cost and high level of passenger satisfaction is the issue. Since the FPDS is totally free to passengers, low operational costs are imperative for its successful operation by FTSCs. Operational costs considered in this paper are composed of fixed vehicle start-up costs and variable vehicle traveling costs. Passenger satisfaction degree is used as the only criterion for evaluating service quality; it is measured by the deviation between actual and desired delivery time at the airport. High degree of passenger satisfaction generally implies high operational costs. For example, serving each order with a single car could ensure total satisfaction for all passengers but it will also be the most expensive way. On the other hand, one vehicle serving too many passengers may result in some being picked up (and delivered) too early may result in many passengers being dissatisfied with the service. Therefore, FTSCs have to strive for the least cost FPDS plans that can provide a satisfactory level of passenger satisfaction degree, instead of pursuing very low operational costs or very high passenger satisfaction degree. In order to achieve this goal, FTSCs need to solve the following three problems:

(1) Which orders would be transported together?
(2) What is the sequence of orders in each route?
(3) What is the starting time of each vehicle?

After taking into consideration the composition of desired delivery times at the airport, these three problems can be viewed as a typical operations problem—Vehicle Routing Problem with Time Windows (VRPTW).

VRPTW is generalization of the Vehicle Routing Problem (VRP), with the added complexity of time windows. Besides time windows, VRP can be added with other complexities such as backhauls and inventory which can be found in Liu and Chung (2009). In addition, VRPTW is a special case of Pickup and Delivery Problem with Time Windows (PDPTW), with all origins or all destinations located at the same depot. In general PDPTW, the origins and destinations of different transportation requests served by one vehicle are always different. Dial-a-Ride Problem (DARP) is another special case of PDPTW in which the loads to be transported are people. More details about PDPTW and DARP are available in Savelsbergh and Sol (1995). In FPDS, all destinations are the same (the airport). Therefore, we can easily model FPDS under the framework of VRPTW.

In VRPTW, one has to design a set of minimum cost routes that start and end at a central depot. A fleet of vehicles serves a set of customers with known demands. Each customer must be assigned exactly once to the vehicles and vehicle capacities can not be exceeded. Service to a customer must begin within the time window, defined by the earliest and the latest time specified by the customer.

In VRPTW, time windows can be hard or soft. In case of a hard time window, if a vehicle arrives too early to pickup a customer, it is permitted to wait until the customer is ready to board the vehicle. However, the service is not permitted to begin after the latest time. In contrast, in case of a soft time window, the time window can be violated at a cost. Time windows are always used to constrain the vehicle arrival time for picking up a customer.

A relatively early survey of heuristic solutions for VRPTW can be found in Solomon (1986) where he proposes a route-first cluster-second scheme using a giant tour heuristic. Solomon (1987) further describes several heuristics for VRPTW, including an extension of the savings algorithm of Clarke and Wright (1964), a time-oriented nearest-neighbor heuristic, insertion heuristic, and a time-oriented sweep heuristic. Potvin and Rousseau (1993) introduce a parallel version of Solomon's insertion heuristic. An asymptotically optimal heuristic is proposed by Bramel and Simchi-Levi (1996). Heuristics used to improve the constructed routes focus mainly on local search algorithm and they can be found in Koskosidis et al. (1992), Potvin and Rousseau (1995); Taillard et al. (1997), and Cordone and Wolfler-Calvo (2001), etc.

Metaheuristics are also widely used to solve VRPTW, including tabu search, genetic algorithm, simulated annealing, etc. Garcia et al. 1994 first applied tabu search for VRPTW. Other applications of tabu search algorithm for VRPTW can be found in Chiang and Russell (1997), Cordeau et al. (2001), Landrieu et al. (2001), and Lau et al. (2003) etc. Thangiah et al. (1991) first attempted to apply genetic algorithm to VRPTW. During the past few years, genetic algorithm applications to VRPTW have been the subject of intensive research. More details about genetic algorithm for VRPTW can be found in Bräysy et al. (2004). Chiang and Russell (1996) developed a simulated annealing algorithm and Li et al. (2003) proposed tabu-embedded simulated annealing for VRPTW. For a detailed survey of heuristics and metaheuristics for VRPTW, we refer to Bräysy and Gendreau (2005a) and Bräysy and Gendreau (2005b).

An optimal approach, using dynamic programming, column generation and Lagrangian relaxation methods, is proposed for VRPTW. These methods are, in many aspects, inherited from work done on the Traveling Salesman Problem (TSP). There are two main lines of development in relation to exact algorithms. One concerns the general decomposition approach and the solution to a certain dual problem associated with the primal VRPTW. The other concerns analysis of the polyhedral structure of the VRPTW. In 1987, Kolen et al. introduced the first method for finding the exact solution for the VRPTW. There is rich literature available on exact algorithms for VRPTW, e.g. Fisher et al. (1997) and Bard et al. (2002). The latest survey of exact algorithms for VRPTW can be found in Kallehauge (2008).

Although there is a large body of literature on VRPTW, the problem addressed in this paper is slightly different from the standard VRPTW. First, in FPDS, all vehicles would pass a common point, the airport, before returning to the vehicle base. Second, passengers' time demands focus on the time of delivery at the airport, rather than the time of picking up the passengers. These two distinctions make the VRPTW of FPDS slightly different from the standard VRPTW, for modeling. Furthermore, the VRPTW of FPDS has two distinctive characteristics:

a. Small vehicle capacity is the first characteristic because the most common vehicles used to transport passengers in FPDS are always cars.
b. The other characteristic of the VRPTW of FPDS is the demand for more exact delivery time at the airport because delivering either too early or too late is unacceptable to passengers.

Due to the above two distinctions, and the passenger satisfaction degrees considered in FPDS, a new 0–1 mixed integer programming model is proposed for VRPTW of FPDS. Taking into consideration the characteristics of the problem, during the algorithm development process, we propose an exact algorithm based on the set-partitioning model for VRPTW of FPDS. Computational results indicate that this algorithm is particularly suitable for VRPTW of FPDS.

Analysis and formulation of VRPTW of FPDS

Analysis of current operational practices of FPDS

Currently, FTSCs in most large and middle size cities of mainland China (e.g. Beijing and Shenyang, etc.), are providing FPDS to passengers who reserve tickets through them. But in all FTSCs, routing and scheduling plans are made in the morning of the service day only. The whole process of planning is manual. The planners first put together a list of passengers who will depart on the service day. Then passengers with the same expected pickup point and the same desired delivery time are grouped into one order. If the number of passengers in an order exceeds the vehicle capacity, the order would be further divided into more than one order, such that each has fewer passengers than a car's capacity. Next, these orders are divided into different geographical areas, according to the locations of expected pickup points. One or more cars are allotted and scheduled to serve orders in each area. The planners determine a tentative picking up sequence for orders in each area. Then more cars are allotted and scheduled to serve the remaining orders in the same or other areas. Passengers in different areas are generally not transported together, unless there are some emergencies like traffic jams, or if there are several single passenger to be pickedup in different areas. In the latter case, in order to save operational costs, planners always transport these single passengers together, even though their desired delivery times are not close to each other.

Apparently, a service operated in such a manner would probably lead to high operational costs and low passenger satisfaction degrees. It is necessary to develop an effective method that can optimize the routing and scheduling plans of FPDS in terms of both total operational costs and passenger satisfaction degrees.

Mathematical formulation of VRPTW of FPDS

In this section, VRPTW of FPDS is formulated mathematically, which could help us to understand the problem better and work out a better solution to solve it. The problem is considered in planning horizon H, which is generally one day. First we assume there are N orders that need to be transported. These orders are indexed by i. Since FTSCs could rent cars from taxi companies, we assume the number of available cars is unlimited. The total number of available cars is represented by K and the cars are indexed by k. The number of cars actually used is determined while firming up the best set of routes and schedules. The main objective we pursue is to obtain the least operational costs at a satisfactory level of passenger satisfaction degree. As discussed above, operational costs are composed of vehicle start-up costs and vehicle traveling costs. Single vehicle start-up cost and unit vehicle traveling cost are denoted by f and c respectively. The objective of minimizing total operational costs can be written as:

$$\text{Min}\, c\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=1}^{N+1} d_{i,j}x_{k,i,j} + f\sum_{k=1}^{K} z_k, \tag{1}$$

where $d_{i,j}$ denotes the direct traveling distance between order i and j, $x_{k,i,j}$ is a binary variable which equals to 1 when order j is picked up directly after order i by vehicle k and 0 otherwise, z_k is also a binary variable which equals to 1 when vehicle k is used and 0 otherwise, and "0" and "$N+1$" denote the vehicle base and the airport, respectively. It is worth mentioning that for each vehicle the cost of returning to vehicle base is included in the start-up cost as the return to the vehicle base is always from the airport only.

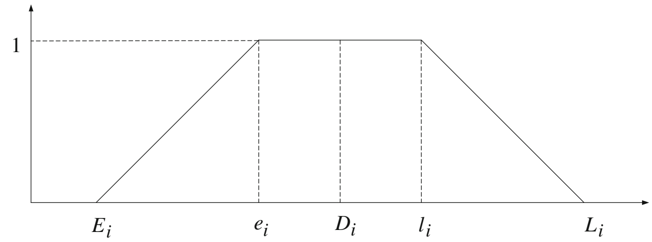

Fig. 2 Passenger satisfaction degree function

As mentioned above, the passenger satisfaction degree is measured by the deviation between actual and desired delivery time. A simple piecewise linear function is introduced to compute passenger satisfaction degree. We assume passengers would feel totally satisfied with the service if the time deviation (both early and late delivery) does not exceed a given value U_1. However, if the time deviation is larger than U_1 but smaller than U_2, which is also a given value, the service is not satisfactory but acceptable. When time deviation exceeds U_2, the service becomes unacceptable. As discussed above, the desired delivery times are generally fixed, based on flight departure times, which are definite time points. The desired delivery time of each order i, that is a time point, is denoted by D_i. To express the passenger satisfaction degree function more clearly, e_i is used to denote the value of expression $D_i - U_1$ and l_i is used to denote the value of expression $D_i + U_1$. Accordingly, E_i is used to denote the value of expression $D_i - U_2$ and L_i is used to denote the value of expression $D_i + U_2$. From the above discussion, we use a simple piecewise linear function $S(\cdot)$ to represent passenger satisfaction degree as follows:

$$S(\tau_i) = \begin{cases} 1 & e_i \le \tau_i \le l_i \\ \frac{\tau_i - E_i}{e_i - E_i} & E_i \le \tau_i \le e_i \\ \frac{L_i - \tau_i}{L_i - l_i} & l_i \le \tau_i \le L_i \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where τ_i represents the actual delivery time of order i. The piecewise linear function is shown in Fig. 2.

By introducing the passenger satisfaction degree function, passenger satisfaction degree can be measured quantitatively, and treated as a constraint in the mathematical model, which could guarantee a satisfactory level of the passenger satisfaction degree. Passenger satisfaction degree constraint can be written as follows:

$$S(\tau_i) \ge \alpha \quad i = 1, 2, \ldots, N, \tag{3}$$

where α is the satisfactory level of passenger satisfaction degree.

When the value of α is given, the passenger satisfaction degree constraints could in fact be converted to hard time window constraints:

$$\overline{E_i} \le \tau_i \le \overline{L_i} \quad i = 1, 2, \ldots, N, \tag{4}$$

where $\overline{E_i}$ and $\overline{L_i}$ are lower and upper bounds of the hard time window of order i. In this paper, the hard time windows are called delivery time windows, and $\overline{E_i}$ and $\overline{L_i}$ can be easily obtained by the equation $S(\tau_i) = \alpha$. For modeling convenience, constraint (4) is expressed in another way. Let $y_{k,i}$ be a binary variable which equals to 1 when the passengers of order i are transported by vehicle k, otherwise it equals to 0. If $y_{k,i}$ equals to 1, the actual delivery time of order i could be represented by the arrival time of vehicle k at the airport. Therefore, constraint (4) can be transformed into constraints (5) and (6):

$$(\overline{E_i} - T_k^{N+1})y_{k,i} \le 0 \quad i = 1, 2, \ldots, N \quad k = 1, 2, \ldots, K \tag{5}$$

$$(T_k^{N+1} - \overline{L_i})y_{k,i} \le 0 \quad i = 1, 2, \ldots, N \quad k = 1, 2, \ldots, K, \tag{6}$$

where T_k^{N+1} is the arrival time of vehicle k at the airport.

0–1 mixed integer programming model of VRPTW of FPDS

Assumptions

1) There is only one airport.
2) Passengers of one order share the same satisfaction degree function.
3) Unloading and loading times are negligible.
4) Both traveling distance and time between two points are symmetric.
5) The number of available vehicles is unlimited.

Model parameters

N = total number of orders
K = total number of vehicles
i = index of orders
k = index of vehicles
0 = vehicle base
$N+1$ = airport
c = vehicle traveling cost per unit distance
f = start-up cost of a single vehicle
Q = vehicle capacity
q_i = number of passengers of order i
$d_{i,j}$ = direct traveling distance between point of pickup of order i and order j
$t_{i,j}$ = direct traveling time between point of pickup of order i and order j
α = satisfactory passenger satisfaction degree

Decision variables

$z_k = 1$ if vehicle k is used, 0 otherwise ($k = 1, 2, \ldots, K$)
$y_{k,i} = 1$ if order i is transported by vehicle k, 0 otherwise ($k = 1, 2, \ldots, K, i = 1, 2, \ldots, N$)
$x_{k,i,j} = 1$ if vehicle k travels directly from order i to order j, 0 otherwise ($k = 1, 2, \ldots, K, i = 0, 1, \ldots, N, j = 1, 2, \ldots, N+1$)
T_k^i = arrival time of vehicle k at point i ($k = 1, 2, \ldots, K, i = 0, 2, \ldots, N+1$)

Model

$$\text{Min } c\sum_{k=1}^{K}\sum_{i=0}^{N}\sum_{j=1}^{N+1} d_{i,j}x_{k,i,j} + f\sum_{k=1}^{K} z_k \tag{1}$$

$$(\overline{E_i} - T_k^{N+1})y_{k,i} \le 0 \quad i = 1, 2, \ldots, N \quad k = 1, 2, \ldots, K \tag{5}$$

$$(T_k^{N+1} - \overline{L_i})y_{k,i} \le 0 \quad i = 1, 2, \ldots, N \quad k = 1, 2, \ldots, K \tag{6}$$

$$\sum_{k=1}^{K} y_{k,i} = 1 \quad i = 1, 2, \ldots, N \tag{7}$$

$$\sum_{k=1}^{K}\sum_{j=1}^{N+1} x_{k,i,j} = 1 \quad i = 1, 2, \ldots, N \tag{8}$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} x_{k,j,i} = 1 \quad i = 1, 2, \ldots, N \tag{9}$$

$$\sum_{i=1}^{N} q_i y_{k,i} \le Q \quad k = 1, 2, \ldots, K \tag{10}$$

$$(T_k^j - T_k^i - t_{i,j})x_{k,i,j} = 0 \quad k = 1, 2, \ldots, K \quad i = 0, 1, \ldots, N \quad j = 1, 2, \ldots, N+1 \tag{11}$$

$$0 \le T_k^i \le H k = 1, 2, \ldots, K \quad i = 0, 1, \ldots, N+1 \tag{12}$$

$$x_{k,i,j} \le y_{k,i} \le z_k \quad i = 0, 2, \ldots, N \quad j = 1, 2, \ldots, N+1 \quad k = 1, 2, \ldots, K \tag{13}$$

$$x_{k,i,j} \le y_{k,j} \le z_k \quad i = 0, 2, \ldots, N \quad j = 1, 2, \ldots, N+1 \quad k = 1, 2, \ldots, K \tag{14}$$

$$x_{k,i,j},\ y_{k,i},\ z_k \in \{0, 1\} \tag{15}$$

Objective Function (1) minimizes the total operational cost, including vehicle start-up cost and vehicle traveling cost. Constraints (5) and (6) guarantee a satisfactory level of passenger satisfaction degree. Constraint (7) ensures that each order is served exactly once. Constraints (8) and (9) ensure that whether before or after a pickup point, there is only one point which the vehicle has to travel from or to, and this point could be a pickup point, the vehicle house, or the airport. Constraint (10) ensures that the total number of passengers transported by a vehicle does not exceed its capacity. For each vehicle, constraint (11) ensures correctness of the vehicle arrival time at pickup points of the orders that the vehicle serves. Constraint (12) guarantees all the arrival times are within the planning horizon. Constraints (13) and (14) are the relation constraints of 0–1 decision variables. Constraint (15) is a binary constraint.

Exact algorithm based on set-partitioning model

The set-partitioning approach was originally proposed by Balinski and Quandt (1964). It is widely applied in a variety of routing and scheduling problems. The basic procedures of the set-partitioning approach for VRP are that the candidate set of feasible routes is generated in the first phase. Then the set-partition model is solved to obtain the final solution in the second phase. The quality of the final solution obtained by this algorithm depends, to a great extent, on the completeness of the candidate set of feasible routes. A totally complete candidate set always contains an exponential number of feasible routes, which leads to unacceptable computing time. Without exception, for general VRPTW, the size of the entire candidate set of feasible routes is also exponentially large for almost all instances. Solving the set-partitioning formulation of VRPTW, even of a moderate size, directly, is extremely time consuming and even impractical. However, relaxation of set-partitioning formulation can always be used to generate excellent lower bounds for Capacitated Vehicle Routing Problem (CVRP). Such examples can be found in Christofides et al. (1981), Fukasawa et al. (2006) and Baldacci et al. (2008). Moreover, based on the resource-constrained path formulation of the VRPTW, Desrochers et al. (1992) proposed a method using column generation to solve the linear programming relaxation of the master set-partitioning problem.

Fortunately, the set-partitioning model in this paper can be directly solved by optimization software such as CPLEX because of the small size of the entire candidate set and because of small vehicle capacities and tight time windows. The core of the proposed algorithm is that, in the first phase, the entire candidate set of best feasible routes is generated. The entire candidate set is composed of N route subsets, i.e. each order corresponds to a *route subset*. The route subset of order i is constructed by all the best feasible routes generated, for the orders in the corresponding *order subset* of order i. The order subset of order i is constructed by order i and all other orders whose delivery time windows overlap with that of order i. For each order, the orders in its order subset would be combined with it to generate all the feasible order combinations. A feasible order combination is defined to be the one where the total number of passengers of combined orders does not exceed the car capacity. Each feasible order combination could generate several order sequences. For each order combination, traveling costs of all order sequences are computed, and the least cost sequence is selected as the best feasible route. Hence, each feasible order combination would correspond to several order sequences but there will be only one best feasible route. In the second phase, the set-partitioning model is directly solved using the CPLEX optimization software package. Each solution obtained by solving the set-partitioning model is a set of some of the best feasible routes and the optimal solution is the set associated with the lowest total operational cost, including fixed vehicle start-up and variable vehicle traveling costs. In each solution, each order is served exactly once.

An obvious drawback of the set-partitioning approach is that the time used to solve the set-partitioning model largely depends on the size of the candidate set of feasible routes, whereas the proposed set-partitioning model based algorithm is particularly suitable for VRPTW of FPDS because small vehicle capacities and tight time windows imply that the number of ways the orders could be combined to form vehicle routes would be small.

Phase I: Generating all best feasible routes

This section focuses mainly on procedures of generating the entire candidate set of best feasible routes. The basic idea of the first phase goes as follows. First, an order subset is constructed for each order, which is called the seed order in the subset. Second, all feasible routes containing the seed order within each route are enumerated for each order subset. Finally, the enumerated routes for each seed order are put together in a route subset. Before proceeding to the details of Phase I, a useful property can be derived for time windows placed on delivery times.

Property 1 *Delivery time windows of the orders to be transported by one vehicle overlap*

This property holds just because the time windows are placed on the delivery times. Suppose there are two orders i and j whose delivery time windows do not overlap, and the desired delivery time of order i is earlier than that of order j. That means the following inequality holds:

$$\overline{E_i} < \overline{L_i} < \overline{E_j} < \overline{L_j} \tag{16}$$

Because the actual delivery time at the airport is a time point, it is obvious that we can never find a common feasible route for order i and j, to transport both by one vehicle. Property 1, however, does not hold if the time windows are placed on pickup times.

The above property can guarantee that for each order i, order subset P_i can be constructed using order i, and all other orders whose delivery time windows overlap with that of order i. Furthermore, to reduce computation time, fewer orders are used to construct the order subset, without affecting the final solution, which is shown in Corollary 1.

Corollary 1 *For each order i, orders used to generate set P_i can be selected only from orders after i in sequence S.*

Let us take order i as an example to explain Corollary 1. Suppose order subset P_i of order i is constructed by $i-2$, $i-1, i, i+1$ and $i+2$. Among them, we select $i-2$, i and $i+2$ to construct a route, assuming the total number of passengers of these three orders does not exceed vehicle capacity. Based on Property 1, delivery time windows of the three selected orders overlap. That is, routes constructed by these orders are feasible. However, these feasible routes must have been included in route set R_{i-2} of $i-2$. Above analysis shows that order $i-2$ can be deleted from set P_i. The same conclusion can be derived for order $i-1$. Based on the above analysis, we can easily get Corollary 1.

Next, methods of enumerating all feasible routes containing seed orders within each route are demonstrated, and route subset is constructed for each seed order. Based on Property 1, no combination of orders in one order subset violates constraints (5) and (6). Thus, any combination of orders within each order subset can be used to construct feasible routes, if the total number of passengers of the combined orders does not exceed the vehicle capacity. As mentioned above, in each order subset P_i, order i is called the *seed order*. For each order i, all orders in set P_i, except the seed order, are selected and combined with the seed order to generate order combinations. If the number of passengers of orders of the generated combination does not exceed vehicle capacity, the combination is then used to construct feasible routes. Each feasible route is a permutation of the combination, and the route with the least traveling cost is selected as the best feasible route corresponding to the combination. In this manner, the best feasible route of each combination generated by orders of set P_i is constructed and put in the route subset R_i.

In FPDS, vehicles used to transport passengers are cars, and each single car can carry at the most four passengers in one trip. The number of passengers in one order does not exceed four because an order containing more than four passengers would be split into several orders with maximum of four passengers each. Hence, there would be four cases when constructing route subset R_i of order i, using the orders in order subset P_i, as follows:

Case 1: The seed order contains four passengers;
Case 2: The seed order contains three passengers;
Case 3: The seed order contains two passengers;
Case 4: The seed order contains only one passenger.

Let us take order i as an example to illustrate how route subset R_i is constructed using the orders in order subset P_i.

In case 1, the number of passengers of order i equals to the maximum capacity of a car. Passengers in other orders can not be transported together with those of order i by one car. Therefore, in this case, there is only one best feasible route in set R_i. In this route, the car starts from the vehicle base, and then picks up four passengers of the order and delivers them to the airport.

In case 2, the number of passengers of order i is one less than the car capacity. Another order in set P_i, containing only one passenger, can be transported together with order i. However, a problem arises—of the two orders, which one should be picked up first? This combination corresponds to two transporting sequences. Traveling costs of the two different sequences are computed and compared. The one with less traveling cost is selected as the best feasible route and put in set R_i. Moreover, it is feasible that the three passengers of order i are transported exclusively. Thus the route transporting only the seed order is constructed and put in set R_i. In case 2, the number of feasible routes in set R_i totally depends on the number of orders containing only one passenger in set P_i.

In case 3, there are two vacant seats in the car transporting passengers of order i. The two vacant seats could be taken by passengers of two different orders containing only one passenger each, or they could be taken by passengers of a single order containing two passengers. In the former instance, all pairs of orders containing only one passenger are selected and combined with seed order i. For each combination, the sequence with the least traveling cost is selected as the best feasible route and put in set R_i. In the latter instance, every order containing two passengers in set P_i is selected and combined with seed order i. For each combination, the least cost sequence is selected as the best feasible route and put in set R_i. Finally, the route transporting only the seed order is constructed and put in set R_i.

In case 4, the combination and sequencing problems become more complicated because there is only one passenger in the seed order, such that more orders could be combined with the seed order. There are seven possible combinations of the seed order.

(1) Three different orders separately containing only one passenger each are combined with the seed order;
(2) Two different orders separately containing only one passenger each are combined with the seed order;
(3) One order containing only one passenger is combined with the seed order;
(4) One order containing two passengers and one order containing only one passenger are combined with the seed order;
(5) One order containing two passengers is combined with the seed order;

(6) One order containing three passengers is combined with the seed order;
(7) No other orders are combined with the seed order.

For each of the above combinations, all feasible order combinations are enumerated, and for each order combination, the least cost sequence is selected as the best feasible route and put in set R_i, as in three previous cases.

Phase II: Assembling routes by solving set-partitioning model

In this section, by solving the set-partitioning model, the minimal cost set of best feasible routes is achieved as the optimal solution. By far, the entire candidate set of best feasible routes has been generated. Let R be the entire candidate set; then it could be written as:

$$R = R_1 \cup R_2 \cdots R_{N-1} \cup R_N \tag{17}$$

Routes in the entire candidate set are indexed by $r, r \in R$. Define c_r to be the cost of candidate route r. Candidate route cost c_r consists of vehicle start-up cost f and vehicle traveling cost in route r. All routes start from the vehicle base and end at the airport. The total traveling cost of route r is defined as the sum of the cost of arcs of the route. Let $a_{i,r}$ be a binary variable equaling to 1 if passenger i is included on route r. y_r is used as a binary variable equaling to 1 if candidate route r is used, 0 otherwise. VRPTW of FPDS can be formulated as the following set-partitioning model.

$$\min \sum_{r=1}^{R} c_r\, y_r \tag{18}$$

$$\sum_{r=1}^{R} a_{i,r}\, y_r = 1 \quad i = 1, 2, \ldots, N \tag{19}$$

$$y_r = 0 \text{ or } 1 \quad r = 1, 2, \ldots, R \tag{20}$$

The above model is quite simple and it can easily be solved by the CPLEX optimization software package.

Computational results

Computational results are reported in this section. First, the proposed algorithm is illustrated by solving four actual FPDS instances provided by Zhongshan Flight Ticket Sales Company (Zhongshan here in after). Then, the algorithm is tested on a set of 200 orders generated randomly, which would verify the general effectiveness of the proposed algorithm for VRPTW of FPDS, even if the number of passengers increases in future. The whole algorithm is calculated on a Pentium IV PC computer running at 3.06 GHz with 1 Gbyte of RAM memory. The first phase of the algorithm is coded with the programming language of Visual C++ 6.0.

Table 1 Parameters of model and algorithm

H	f	c	U_1	U_2
10 h	10 RMB	1.2 RMB	10 min	30 min

Moreover, it is worth mentioning that prior to using the proposed algorithm, we have solved the proposed model directly, using the CPLEX software package, whereas the computing time is disappointingly long. For a random instance containing 10 orders, it would take at least 2 h to obtain the optimal solution, which is unacceptable to FTSCs in practical situations.

Zhongshan is one of the largest FTSCs in mainland China and it is also the first one that began to provide FPDS for passengers reserving flight tickets through it. The set of the actual instances is provided by its branch located in the city of Shenyang. According to statistics covering the past three years, on each service day, the number of FPDS orders in Shenyang branch varies from 75 to 110, and the number of FPDS passengers ranges between 100 and 145. In our experiment, we select four instances containing 80 orders, 90 orders, 100 orders and 110 orders, with 103 passengers, 116 passengers, 127 passengers and 142 passengers, respectively. For every instance, passengers are located within a territory of about $14 \times 8\,\text{km}^2$. The airport is on the edge of this area. Service cars start from the vehicle base of the company, located at the centre of the area. Distances between different points are computed by an electronic map, and traveling times are computed by multiplying traveling distances with a coefficient varying from 1.7 to 2.0, according to traffic conditions. The planning horizon is 10 h, from 8:00 a.m. to 18:00 p.m. Due to space limitations, some detailed information such as passenger locations and flight departure times are not provided. Parameters of the model and the algorithm are presented in Table 1. Computational results obtained by applying the proposed algorithm, showing the cost of achieving passenger satisfaction degrees varying from 10% to 100%, are presented in Tables 2, 3, 4, and 5.

By observing the results in Tables 2, 3, 4, and 5, we can easily draw some apparent conclusions. First, with the levels of passenger satisfaction degree increasing, the number of best feasible routes decreases, while both total operational costs and average costs increase. Second, at any level of passenger satisfaction degree, optimal solutions can be obtained within a few seconds. Moreover, both ANP and ACP are very important criteria for evaluating the quality of routing and scheduling plans in FTSCs because either low ANP or high ACP means high operational costs. For plans made by Zhongshan, ANP generally falls between 3.2 and 3.4 and ACP equals to 10 on average, but there are always some complaints about

Table 2 Computational results of 80-orders instance

PSD (%)	BFR	TOC	AUV	ANP	ACC	ACP	CT1 (s)	CT2 (s)
10	2963	911.7	27	3.8	33.8	8.9	1	2
20	2721	918.0	27	3.8	34.0	8.9	1	3
30	2057	931.4	27	3.8	34.5	9.0	1	1
40	2029	931.4	27	3.8	34.5	9.0	1	1
50	1261	951.6	27	3.8	35.2	9.2	1	1
60	1261	951.6	27	3.8	35.2	9.2	1	1
70	1090	971.3	27	3.8	36.0	9.4	1	1
80	649	1001.3	30	3.4	33.4	9.7	1	1
90	649	1001.3	30	3.4	33.4	9.7	1	1
100	297	1193.7	33	3.1	36.2	11.6	1	1

Meanings of abbreviations in Tables 2, 3, 4 and 5 are as follows: *PSD* passenger satisfaction degree, *BFR* the number of all the best feasible routes, *TOC* total operational costs, *AUV* total number of actually used vehicles, *ANP* average number of passengers per car, *ACC* average cost per car, *ACP* average cost per passenger, *CT1* computing time of Phase I, *CT2* computing time of Phase II

Table 3 Computational results of 90-orders instance

PSD (%)	BFR	TOC	AUV	ANP	ACC	ACP	CT1 (s)	CT2 (s)
10	6684	1048.1	30	3.9	34.9	9.0	2	2
20	6473	1063.8	31	3.7	34.3	9.2	2	2
30	4287	1083.6	31	3.7	35.0	9.3	2	2
40	4204	1089.0	31	3.7	35.1	9.4	2	5
50	2575	1112.4	31	3.7	35.9	9.6	1	1
60	2575	1112.4	31	3.7	35.9	9.6	1	1
70	2510	1136.1	32	3.6	35.5	9.8	1	1
80	1076	1183.4	33	3.5	35.9	10.2	1	1
90	1076	1183.4	33	3.5	35.9	10.2	1	1
100	525	1321.1	37	3.1	35.7	11.4	1	1

for footnote see Table 2

Table 4 Computational results of 100-orders instance

PSD (%)	BFR	TOC	AUV	ANP	ACC	ACP	CT1 (s)	CT2 (s)
10	8751	1054.1	32	4.0	32.9	8.3	2	3
20	8258	1060.5	32	4.0	33.1	8.4	2	3
30	5736	1099.6	33	3.8	33.3	8.7	2	11
40	5592	1101.6	33	3.8	33.4	8.7	2	5
50	3305	1127.2	34	3.7	33.2	8.9	1	7
60	3305	1127.2	34	3.7	33.2	8.9	1	7
70	3044	1141.0	34	3.7	33.6	9.0	1	2
80	1375	1183.2	34	3.7	34.8	9.3	1	1
90	1375	1183.2	34	3.7	34.8	9.3	1	1
100	706	1321.4	39	3.3	33.9	10.4	1	1

for footnote see Table 2

Table 5 Computational results of 110-orders instance

PSD (%)	BFR	TOC	AUV	ANP	ACC	ACP	CT1 (s)	CT2 (s)
10	11693	1203.5	36	3.9	33.4	8.5	2	4
20	11400	1212.4	36	3.9	33.7	8.5	2	7
30	7639	1230.9	36	3.9	34.2	8.7	2	2
40	7505	1244.4	36	3.9	34.6	8.8	2	3
50	4262	1279.3	37	3.8	34.6	9.0	1	3
60	4262	1279.3	37	3.8	34.6	9.0	1	3
70	4205	1291.4	37	3.8	34.9	9.1	1	1
80	1719	1333.1	38	3.7	35.1	9.4	1	1
90	1719	1333.1	38	3.7	35.1	9.4	1	1
100	880	1510.4	44	3.2	34.3	10.6	1	1

for footnote see Table 2

Table 6 Computational results of 200-orders instance

PSD (%)	BFR	TOC	AUV	ANP	ACC	ACP	CT1 (s)	CT2 (s)
10	72062	2244.6	67	4.0	33.5	8.4	4	555
20	66659	2247.1	67	4.0	33.5	8.4	4	1349
30	45139	2268.2	67	4.0	33.9	8.5	3	989
40	43954	2270.2	67	4.0	33.9	8.5	3	906
50	21550	2329.4	68	3.9	34.3	8.7	2	16
60	21550	2329.4	68	3.9	34.3	8.7	2	16
70	19786	2335.1	68	3.9	34.3	8.7	2	95
80	13300	2379.4	68	3.9	35.0	8.9	2	252
90	9365	2427.4	69	3.9	35.2	9.1	2	38
100	3978	2598.3	74	3.6	35.1	9.7	1	6

too early or too late delivery. In some extreme situations, passengers are delivered more than 2 h before boarding at the airport. Under the satisfaction degree function (2) proposed in this paper, actual satisfaction degrees can vary from 100 to 0%. That means high satisfaction degrees of some passengers are obtained by sacrificing other passengers' satisfaction degrees. Our computational results show that all the passenger satisfaction degrees can be guaranteed at the level of at least 90% when ANP reaches 3.2 or ACP reaches 10.

As operations of FTSCs grow, the number of FPDS orders would probably increase in future. To verify the general effectiveness of our proposed algorithm for VRPTW of FPDS, the algorithm is applied to a 200 orders instance generated randomly. This random instance contains 200 orders and 268 passengers. Other information and parameters are the same as those of the previous actual instances. Computational results of the random instance are presented in Table 6.

From Table 6, it is observed that the first conclusion drawn from Tables 2, 3, 4, and 5 is also applicable to the 200 orders instance. But the 200 orders instance would need much more computation time, and in an extreme case, the time could be nearly half an hour. However, it is acceptable to daily operations. In addition, from Table 2, 3, 4, 5, and 6 we can easily find ANP increases with the increase of the number of orders, while ACP decreases with the increase of the number of orders. Actually, this conclusion is intuitive. When the number of orders increases, the passenger density increases, and the vehicle can serve more passengers each time.

Conclusions

In this paper, we have addressed a new routing and scheduling problem faced by Flight Ticket Sales Companies (FTSCs). The problem is formulated under the framework of Vehicle Routing Problem with Time Windows (VRPTW). A 0–1 mixed integer programming model is proposed for VRPTW of Free Pickup and Delivery of Passengers to the Airport Service (FPDS). However, VRPTW of FPDS has two distinctions that make the problem slightly different from the standard VRPTW. The problem is modeled under costs and service quality criteria. Minimizing total operational costs serves as the objective of the model. Service quality, evaluated by passenger satisfaction degree, is addressed through constraints incorporated in the model. Passenger satisfaction degrees are computed through passenger satisfaction degree function, which is a stepwise linear function. After the satisfactory level is set by decision makers, lower bound and upper bound of the hard time window of each order can be obtained from this function.

Moreover, the new routing and scheduling problem has two characteristics—small vehicle capacity and tight delivery time windows. A set-partitioning model based exact algorithm is developed and the two characteristics are taken into consideration during the algorithm development process. The two characteristics make the set-partitioning model based algorithm particularly suitable for our problem because both characteristics imply the entire candidate set of best feasible routes is small. Finally, we test the proposed algorithm on four actual instances and a larger instance generated randomly. For the four actual instances, the optimal solutions, under different satisfaction degrees, are obtained within a few seconds.

To further improve both the operational costs and service quality of FPDS, we are also considering some extensions to the routing and scheduling problem of FPDS. First, in practical situations, there might be two or more airports in a big city. In such a case, both the proposed model and algorithm shall have to be modified or extended. Second, FTSCs are using single transportation mode, i.e. all vehicles that are used to transport passengers are cars. However, FTSCs would probably save more operational costs if they use a mixed transportation mode. For example, a mixed transportation mode of cars and a larger capacity vehicle like a minibus can be adopted. Third, city size and population density should be considered when we choose the transportation mode. For example, use of a minibus is more appropriate for big cities with high population densities as the main transportation vehicle. On the contrary, it is more appropriate for smaller cities with low population densities to use cars as the main transportation vehicle.

In conclusion, our proposed algorithm can solve the basic routing and scheduling problem of FPDS for FTSCs effectively and fast. However, there are still some more issues that need to be considered.

Acknowledgments The authors thank Zhongshan Flight Ticket Sales Company's staff for providing the company data and finding time for discussions. Research for this paper is financially supported by a grant from the Natural Science Foundation of China (70625001 and 70721001).

References

Baldacci, R., Christofides, N., & Mingozzi, A. (2008). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming, 115*(2), 351–385.

Balinski, M., & Quandt, R. (1964). On an integer program for a delivery problem. *Operations Research, 12*(2), 300–304.

Bard, J. F., Kontoravdis, G., & Yu, G. (2002). A branch-and-cut procedure for the vehicle routing problem with time windows. *Transportation Science, 36*(2), 250–269.

Bramel, J., & Simchi-Levi, D. (1996). Probabilistic analyses and practical algorithms for the vehicle routing problem with time windows. *Operations Research, 44*(3), 501–509.

Bräysy, O., Dullaert, W., & Gendreau, M. (2004). Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics, 10*(6), 587–611.

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transportation Science, 39*(1), 104–118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows, part II: metaheuristics. *Transportation Science, 39*(1), 119–139.

Chiang, W. C., & Russell, R. A. (1996). Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research, 63*(1), 3–27.

Chiang, W. C., & Russell, R. A. (1997). A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing, 9*(4), 417–430.

Christofides, N., Mingozzi, A., & Toth, P. (1981). Exact algorithms for the vehicle routing problem based on spanning tree and shortest path relaxations. *Mathematical Programming, 20*(1), 255–282.

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of Operations Research Society, 52*(8), 928–936.

Cordone, R., & Wolfler-Calvo, R. (2001). A heuristic for the vehicle routing problem with time windows. *Journal of Heuristics, 7*(2), 107–129.

Desrochers, M., Desrosiers, J., & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research, 40*(2), 342–354.

Fisher, M. L., Jörnsten, K. O., & Madsen, O. B. G. (1997). Vehicle routing with time windows: two optimization algorithms. *Operations Research, 45*(3), 488–492.
Fukasawa, R., Lysgaard, J., Aragão, M. P., Reis, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming, 106*(3), 491–511.
Garcia, B.-L., Potvin, J.-Y., & Rousseau, J.-M. (1994). A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers & Operations Research, 21*(9), 1025–1033.
Kallehauge, B. (2008). Formulations and exact algorithms for the vehicle routing problem with time windows. *Computers & Operations Research, 35*(7), 2307–2330.
Kolen, A. W. J., Kan, A. H. G. R., & Trienekens, H. W. J. M. (1987). Vehicle routing with time windows. *Operations Research, 35*(2), 266–273.
Koskosidis, Y. A., Powell, W. B., & Solomon, M. M. (1992). An optimization-based heuristic for vehicle routing and scheduling with soft time window constraints. *Transportation Science, 26*(2), 69–85.
Landrieu, A., Mati, Y., & Binder, Z. (2001). A tabu search heuristic for the single vehicle pickup and delivery problem with time windows. *Journal of Intelligent Manufacturing, 12*(5), 497–508.
Lau, H. C., Sim, M., & Teo K., M. (2003). Vehicle routing problem with time windows and a limited number of vehicles. *European Journal of Operational Research, 148*(3), 559–568.
Li, H., Lim, A., & Huang, J. (2003). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research, 150*(1), 115–127.
Liu, S. C., & Chung, C. H. (2009). A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing, 20*(1), 29–42.
Potvin, J.-Y., & Rousseau, J.-M. (1993). A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *European Journal of Operational Research, 66*(3), 331–340.
Potvin, J.-Y., & Rousseau, J.-M. (1995). An exchange heuristic for routing problems with time windows. *Journal of Operations Research Society, 46*(12), 1433–1446.
Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science, 29*(1), 17–29.
Solomon, M. M. (1986). On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints. *Networks, 16*(2), 161–174.
Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research, 35*(2), 254–265.
Taillard, E., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science, 31*(2), 170–186.
Thangiah, S. R., Nygard, K. E., Juell, P. L. (1991). GIDEON: A genetic algorithm system for vehicle routing with time windows. In: *Proceedings of 7th IEEE conference on artificial intelligence applications* (pp. 322–328).