International Journal of Production Management and Engineering

https://doi.org/10.4995/ijpme.2017.5916

Received 2016-06-11 Accepted: 2017-06-09

A Column Generation for the Heterogeneous Fixed Fleet Open Vehicle Routing Problem

Majid Yousefikhoshbakht [a] **and Azam Dolatnejad** [b*]

[a]Department of Mathematics, Faculty of Science, Bu-Ali Sina University, Hamedan, Iran.

[b]Young Researchers & Elite Club, Tehran North Branch, Islamic Azad University, Tehran, Iran.

[*] *a_dolatnejad@aut.ac.ir*

Abstract: This paper addressed the heterogeneous fixed fleet open vehicle routing problem (HFFOVRP), in which the vehicles are not required to return to the depot after completing a service. In this new problem, the demands of customers are fulfilled by a heterogeneous fixed fleet of vehicles having various capacities, fixed costs and variable costs. This problem is an important variant of the open vehicle routing problem (OVRP) and can cover more practical situations in transportation and logistics. Since this problem belongs to NP-hard Problems, An approach based on column generation (CG) is applied to solve the HFFOVRP. A tight integer programming model is presented and the linear programming relaxation of which is solved by the CG technique. Since there have been no existing benchmarks, this study generated 19 test problems and the results of the proposed CG algorithm is compared to the results of exact algorithm. Computational experience confirms that the proposed algorithm can provide better solutions within a comparatively shorter period of time.

Key words: Open Vehicle Routing Problem, Heterogeneous Fixed Fleet, NP-hard Problems, Column Generation.

1. Introduction

While the travelling salesman problem (TSP) is perhaps the most famous single-vehicle problem, the vehicle routing problem (VRP) is an important variant of TSP which has many applications in industrial and service firms (YousefiKhoshbakht and Khorram, 2012). This problem plays a vital role in supply chains where in the first transportation step to collect agricultural products, for instance or in the final distribution phase to deliver goods to customers. The VRP was first defined by Dantzig and Ramser more than 50 years ago (1959) and can be defined on an undirected complete graph with one depot (node 0) and *n* customers indexed from 1 to *n* (if the graph is not complete, we can instead lack of each arc with the arc that has infinite size) (Saadati Eskandari and YousefiKhoshbakht, 2012). A fleet of same vehicles is based at the depot in which each vehicle has a fixed capacity and perhaps a route-length restriction which limits the maximum distance it can travel. Furthermore, each customer has a known demand and a defined distance is associated with each edge. This problem involves routing a fleet of vehicles that start to move simultaneously from the depot and come back to the depot after visiting customers. In other words, each route in the VRP is a Hamiltonian cycle over the subset of customers visited on the route. The objective is to design a set of minimum cost routes to serve all customers so that:

The load on a vehicle is below vehicle capacity at each point on the route.

To cite this article: Yousefikhoshbakht, M., Dolatnejad, A. (2017). A Column Generation for the Heterogeneous Fixed Fleet Open Vehicle Routing Problem. *International Journal of Production Management and Engineering*, *5*(2), 55-71. https://doi.org/10.4995/raet.2017.5916

Each customer is serviced by only one visit of a single vehicle, i.e., split deliveries are not allowed.

The minimum number of vehicles is also required to service all customers. In other words, the number of trips or vehicles used is not imposed, it is a decision variable.

The VRP is not very realistic. To make VRP models more realistic and applicable, there are many varieties of the VRP obtained by adding constraints to the basic model. For example, Several variations and specializations of the VRP are the load along each route must not exceed the service of the customers and must occur within given time windows (vehicle routing problem with time windows, VRPTW) (Wang and Chen, 2012), the customer has pick-up and delivery demand (vehicle routing problem with pickup and delivery, VRPPD) (Çatay, 2010), the customer demands may not be completely known in advance (stochastic vehicle routing problem, SVRP) (Lei, Laporte and Guo, 2011), the service of a customer may be split among different vehicles (split delivery vehicle routing problem, SDVRP) (Aleman and Hill, 2010), precedence relations may exist between the customers (vehicle routing problem with backhauls, VRPB) (Anbuudayasankar *et al.,* 2012), and the demands or the travel times may vary dynamically (dynamic vehicle routing problem, DVRP) (Gendreau *et al.,* 2006).

Another version of the VRP is heterogeneous fleet vehicle routing problem (HFVRP) in which the fleet may contain heterogeneous vehicles like most of the companies in the real-life context (Li, Tian and Aneja, 2010). In this problem, the fleet is composed of various vehicle types. Each type *k* is defined by a capacity Q_k, a fixed cost f_k and a cost per distance unit α_k often called variable cost. It should be noted that a cycle of length L done by a vehicle of type *k* has a cost $f_k + \alpha_k \cdot L$. As in the VRP, each customer must be visited by one vehicle only, each vehicle must start and finish its travel at the depot and the capacity of a vehicle and maybe the maximum length of a route must not be violated. The objective of the HFVRP is to compute a set of cycles and to assign vehicles to cycle to minimize total cost which includes both the vehicle variable and fixed costs. The idea is not only to consider the routing of the vehicles, but also the composition of the vehicle fleet (Gendreau *et al.,* 1999).

Using a heterogeneous fleet of vehicles has several advantages such as the scheduler can revise the fleet composition to better suit customer needs and vehicles of different carrying capacities give the flexibility to assign capacity according to the customer's varying demand by deploying the suitable vehicle types to areas with the analogous concentration of customers. It is also possible to facility customers requiring small vehicles because of accessibility restrictions in urban areas, environmental concerns or physical restrictions on the vehicle size and weight. If we assume that the number of vehicles of each type is unlimited, we get a type of problem known as the fleet size and mix vehicle routing problem (FSMVRP) (Brandão, 2009). Furthermore, The Heterogeneous Fixed Fleet Vehicle Routing Problem (HFFVRP) is a variant of the FSMVRP in which there is a limited number of homogeneous and heterogeneous vehicles respectively. In other words, in contrast to the FSMVRP, the number of vehicles of each type is limited. Although the FSMVRP and HFFVRP are very similar, these two types of problems are used in rather different situations. The FSMVRP is more appropriate for tactical decisions (selecting the acquired number of vehicles) and operational ones (computing the routes and the vehicles assigned to them). In these strategic decisions, a company wants to buy a vehicle fleet and needs to define its size and composition, but the HFFVRP represents better the operational decisions of defining the vehicles that should be used in order to serve the customers among those available.

The HFFVRP intensifies from a practical perspective when the vehicle fleet is hired, that is, vehicles do not constitute company assets. In such cases, effective planning is a critical success factor for the operational efficiency and the resulting service level, since non-company resources are responsible for the physical interface with the final customer. The open vehicle routing operational framework is faced by a company which either does not own a vehicle fleet at all, or its fleet is inappropriate or inadequate to satisfy the demand of its customers. Thus, the company has to contract all or part of its distribution activities to external carriers. These contractors have their own vehicles, they pay their own vehicle costs (e.g. capital, operating, maintenance and depreciation), and they usually consider a compensation model based on mileage. Whenever the company does not need the contractor or the vehicle back at the depot, the paths followed by the vehicles must not include the vehicle trip after the last delivery (i.e. the return trip to the depot) that will add extra mileage to the compensation model. This problem, called Heterogeneous Fixed Fleet

Open Vehicle Routing Problem (HFFOVRP) (Li, Leung and Tian, 2012). This reference addressed this problem for the first one. Besides, a multistart adaptive memory programming metaheuristic with modified tabu search algorithm was proposed to solve this new version of VRP. Finally, the efficiency and effectiveness of the proposed algorithm are experimentally evaluated on a set of generated instances. Figure 1 presents a feasible solution for the HFFOVRP.

The HFFOVRP compared to HFFVRP has a unique character in that the vehicles are not required to come back to the depot after completing a service. The HFFOVRP is utilized in practice in delivering packages and newspapers to homes. In this important variant of the OVRP, contractors who are not employees of the delivery company use their own vehicles and do not return to the depot. Furthermore, companies which use contractors to deliver newspapers to residential customers do not require the contractors and their vehicles to return to the depot. As a result, interest of researchers in the OVRP and its variations has increased dramatically and a wide variety of new algorithms have been developed over the last ten years to solve the problem.

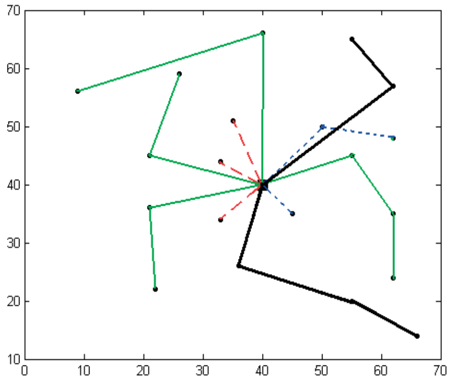


Figure 1. Feasible solution for the HFFOVRP.

However, to the best of our knowledge, a first done work to address HFFOVRP, which is a relaxation of the standard VRP, was the paper of Li, Leung and Tian (2012). This problem is more practical in distribution management and in which the fleet consists of vehicles with different attributes and the number of available vehicles is fixed like HFFVRP. In contrast to HFFVRP, the vehicles do not return to the starting point after completing the service of the last customers. As we know, most works on OVRP has aimed at minimizing the number of vehicles, but this objective was not considered in the HFFOVRP proposed in this work. To solve the problem, a multi-start adaptive memory programming metaheuristic (MAMP) was proposed as a new memory-based algorithm. Then, the parameters tuning is conducted systematically by an automatic procedure and the advantage of considering a multi-start strategy in the MAMP is verified with the obtained best configuration. Finally, the results are reported on the generated instances. After that, Penna *et al.* presented a heuristic algorithm based on the iterated local search metaheuristic and on the randomized variable neighborhood descent (Subramanian *et al.,* 2010) for solving the HFFOVRP. This work is an extension of the one proposed by themselves for the HFVRP (Penna, Subramanian and Ochi, 2013). This paper dealt with the OVRP and the HFFOVR which often arises in distribution management and transportation. They solved both variants by a multi-start algorithm based on the iterated local search metaheuristic. The proposed algorithm uses a variable neighborhood descent procedure, with random neighborhood, ordering in the local search phase.

Yousefikhoshbakht *et al.* (2014) proposed an efficient adaptive memory based algorithm equipped with diversification and intensification to solve the HFFOVRP. This algorithm called BRMTS is a bone route algorithm combined with a modified tabu search (BRMTS) which is effective for solving HFFOVRP problems in a reasonable computing time. Furthermore, the proposed algorithm directly produces a new solution from a component of the other solutions while using new diversification and intensification mechanisms. The BRMTS employs the generalized route for constructive algorithm which was first presented in (Tarantilis and Kiranoudis, 2014) for generating initial diversified solutions and modified TS improvement procedure. Computational results on some test-generated instances demonstrate that the proposed BRMTS finds high quality solutions within a reasonable time. Besides, these authors proposed a column generation technique and an elite ant system for the HFFOVRP. This efficient hybrid heuristic used versions algorithms, including sweep, insert, swap, and 2-Opt moves in order to solve generated Hamiltonian paths with some modifications to generate feasible columns efficiently. These modifications lead to improve both the performance of the algorithm and the quality of the solutions (Yousefikhoshbakht *et al.,* 2016). The results show that the hybrid algorithm explores different parts of the solution space and try to be not trapped at the local optimum points.

The HFFOVRP is an NP-hard combinatorial problem, since it reduces to the OVRP if the number of types of vehicles is just one, i.e., the vehicle fleet is homogeneous, and the number of vehicles is unlimited. From the point of view of graph theory, the difference between the OVRP and the VRP is that a solution of the former consists of a set of Hamiltonian paths rather than Hamiltonian cycles. The problem of finding the best Hamiltonian path for each set of customers assigned to a vehicle is NP-hard (Syslo, Deo and Kowalik, 1983). Hence HFFOVRP is also NP-hard. Therefore, most of the practical examples of this problem cannot be solved by exact algorithms to optimality within reasonable time. Furthermore, because there is no known polynomial algorithm that will find the optimal solution in every instance, the use of heuristics and metaheuristic are considered as a reasonable approach in finding solutions. In this paper, we have proposed a heuristic column generation algorithm (CG) in order to improve both the performance of the algorithm and the quality of the solutions of the exact algorithm. The algorithm explores different parts of the solution space, and the search method is not trapped at the local optimum. The experimental results have shown that the proposed algorithm is to be very efficient and competitive in terms of solution quality compared to exact algorithm.

The structure of the remainder of the paper is as follows. In section 2, a proposed mixed integer model is described and in Section 3, the proposed idea based on column generation is explained in great detail. In Section 4, the proposed algorithm is compared with an exact algorithm on generated or standard problems belonging to HFFOVRP library. Finally, some concluding remarks are given in the section 5.

2. Problem description and formulation

From a graph theoretical point of view, we can define the HFFOVRP as follows. Let $G=(V, E)$ be an undirected connected graph with $V=\{0,1,\ldots,n\}$ as the set of vertexes and the set of arcs $E=\{(i,j):0 \leq i, j \leq n\}$ (if the graph is not complete, we can instead lack of each arc with the arc that has infinite size). Node 0 is the depot and the customer set C consists of n customers, i.e., $C=\{1,2,...,n\}$. A nonnegative cost c_{ij} ($c_{ii}=0$, $0 \leq i \leq n$) associated with each arc $(i,j) \in E$ and each vertex $i \in C$ is a customer with a non-negative demand p_i. The available fleet consists of K different type vehicles located at the depot and the number of available vehicles of each type is fixed and equal to n_k. A capacity Q_k, a fixed cost f_k and a variable cost α_k are associated with each type of vehicle k in which α_k is cost per unit of distance corresponding to each vehicle type k. Hence, $c_{ij}^k = c_{ij} \times \alpha_k$ represents the cost of the travel from customer i to j with a vehicle of type k. The HFFOVRP deals with finding the minimum total transportation cost, including the fixed and variable cost for a fleet of vehicles which start and end at the depot so that the following constraints are taken into account:

- The total load of each vehicle cannot exceed the capacity of the corresponding vehicle type.

The used number of vehicles of type k cannot exceed n_k.

The demand of each customer is satisfied by exactly one vehicle in only one visit.

We present following mathematical formulation for HFFOVRP using variables and y_{ij} where, x_{ij}^k take the value 1 if a vehicle of type k travels directly from customer i to customer j, and 0 otherwise; denotes the route. The flow variables y_{ij} specify the quantity of goods that a vehicle k is carrying when leaves customer i to service customer j.

$$Min \sum_{k=1}^{K} f_k \sum_{j=1}^{n} x_{0j}^k + \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij}^k x_{ij}^k \qquad (1)$$

subject to

$$\sum_{k=1}^{K} \sum_{i=0}^{n} x_{ij}^k = 1 \forall j = 1,2,...,n \qquad (2)$$

$$\sum_{k=1}^{K} \sum_{j=1}^{n} x_{ij}^k \leq 1 \forall i = 1,2,...,n \qquad (3)$$

$$0 \leq \sum_{i=1}^{n} x_{ij}^k - \sum_{i=1}^{n} x_{ji}^k \leq 1 \quad \forall j = 1,2,\ldots,n \quad \forall k = 1,2,\ldots,K \qquad (4)$$

$$\sum_{j=1}^{n} x_{0j}^k \leq n_k, \quad \forall k = 1,2,\ldots,K \qquad (5)$$

$$\sum_{k=1}^{K} \sum_{i=0}^{n} y_{ij}^k - \sum_{k=1}^{K} \sum_{i=0}^{n} y_{ji}^k = p_j \quad \forall j = 1,2,...,n \qquad (6)$$

$$p_j x_{ij}^k \leq y_{ij}^k \leq (Q_k - p_i) x_{ij}^k \ \forall i,j = 0,1,...,n, i \neq j, \quad \forall k = 1,2,...,K \tag{7}$$

$$\sum_{j=1}^{n} x_{i0}^k = 0, \forall k = 1,2,...,K \tag{8}$$

$$x_{ij}^k \in \{0,1\} \ \ \forall i,j = 0,1,...,n, i \neq j, \quad \forall k = 1,2,...,K \tag{9}$$

$$y_{ij}^k \geq 0 \forall i,j = 0,1,...,n, \quad \forall k = 1,2,...,K \tag{10}$$

The objective function (1) gives the sum of the total fixed cost of the vehicles used plus the total variable routing cost. Constraints (2) mean that only one arc can be entered for each customer; however, constraints (3) show that almost one arc can be exited from each customer. Constraints (4) states that if a vehicle visits a customer, it can remain there or depart from it. The maximum number of vehicles available for each vehicle type is guaranteed by constraints (5). Equality equations (6) insure that the demands of all customers are fully satisfied. Constraints (7) state that the vehicle capacity is never exceeded. Constraints (8) guarantee that there is not any arc from each customer to the depot. Constraints (9) describe that each arc in the network has the value 1 if it is used and 0 otherwise. Finally, Restrictions (10) force the flow to remain non-negative.

3. The proposed algorithm

A route-vehicle pair in open vehicle routing, (*r*,*k*), is feasible when route r starts at the depot and the sum of demand of customers on route r is not larger than the capacity of the vehicle type *k*. The reformulation of the HFFOVRP is required the following additional notation:

K The set of vehicle type, $k \in \{1,2,...,K\}$.

R_k The set of feasible route r for vehicle type *k*.

R The set of all of feasible route, $r \in R$, $R = \bigcup_{k \in K} R_k$

c_r^k Cost of rout *r* for vehicle type *k*, $(r, k) \in R_k$, $c_r^k = \sum_{(i,j) \in r} c_{ij}$

a_{ir}^k Binary parameter of equal to 1 if the customer *i* is serviced by route $r \in R$ and vehicle type $k \in K$, 0 otherwise.

x_r^k Binary variable of equal to 1 if route $r \in R_k$ is used in the solution, 0 otherwise.

n_k The number of vehicle type $k \in K$.

Now, HFFOVRP can be formulated as follows:

$$Min \sum_{k \in K} \sum_{r \in R_k} c_r^k x_r^k \tag{11}$$

Subject to

$$\sum_{k \in K} \sum_{r \in R_k} a_{ir}^k x_r^k = 1 \quad \forall i = 1,\ldots,n \tag{12}$$

$$\sum_{r \in R_k} x_r^k \leq n_k \qquad \forall k \in K \tag{13}$$

$$x_r^k \in \{0,1\} \qquad \forall k \in K, r \in R_k \tag{14}$$

In this formulation, the objective function (11) minimizes the total costs. Constraints (12) indicate that each customer is assigned to once and only once feasible route-vehicle pair. Constraints (13) ensure that that maximum of the number of used vehicles of each type does not exceed the number of available vehicles of same type. Finally, constraints (14) describe that each arc in the network has the value 1 if it is used and 0 otherwise. In this type of formulation, each column corresponds to one route-vehicle pair. As number of columns is extremely large, this problem cannot be solved directly, so we use a column generation approach to solve this problem.

3.1. Column Generation approach

Column Generation approach deals with two problems, the Master-Problem (MP) and the Pricing Sub-Problem (PSP). The MP is linear programming problem that the goal of this problem is to find the minimal cost and to produce the shadow prices of the temporary optimal solution to be used in the Pricing Sub-Problem.

Since the number of columns of MP is arbitrarily large, a Restricted MP is used to initiate the computations. The goal of the Pricing Sub-Problem is to generate additional columns for MP. In this section, the proposed sweep-based algorithm is first described and then the formulations of the Master Problem and Pricing Sub-Problem are presented.

3.1.1. The proposed sweep-based algorithm

In this section, the sweep-based algorithm is described. This heuristic belongs to the class of heuristic called cluster first-route second in which a set of initial feasible open routes is generated. To apply the heuristic, we assume that the location of each customer is known in terms of an (x,y) coordinate. We compute the polar coordinates of each customer with respect to the depot and then order the customers by increasing polar angle and generate a list of customer. Note that if customer i and customer j have same polar angle, we put the customer that has lower distance respect to the depot earlier in the list. Then do the following steps:

Step 1: Sort the type of vehicles randomly.

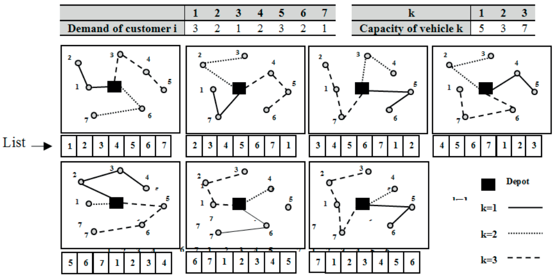
Step 2: For each of the type of vehicles (k), perform the following step, until all of the customers to be swept or all vehicles of this type are loaded:

Step 2.1: For the unused vehicle, we gradually add customers in vehicle route according to list of customers until the capacity constraint is attained. Now, we add pair (r,k) to the set of Initial feasible routes.

Step 3: Delete the first customer in the list and add it to the end of list.

Step 4: If updated list was the initial list, stop and the algorithm is terminated, otherwise go to step 1.

In Figure 2, we illustrate the heuristic described by means of an example involving 7 customers and 3 vehicles. The customer demands and the vehicle capacity are given below.



	1	2	3	4	5	6	7
Demand of customer i	3	2	1	2	3	2	1

k	1	2	3
Capacity of vehicle k	5	3	7

Figure 2. Sweep Algorithm.

3.1.2. Master-Problem

The MP can be formulated as a set covering formulation. This formulation is linear programming problem that each customer is assigned to at least one feasible route–vehicle pair, since the arc costs satisfy the triangle inequality and each customer will be visited exactly once to minimize the costs. Also in constraint (14), integrality of variable x_r^k is eliminated.

$$(MP)\, Min \sum_{k\in K}\sum_{r\in R_k} c_r^k x_r^k \tag{15}$$

Subject to

$$\sum_{k\in K}\sum_{r\in R_k} a_{ir}^k x_r^k \ge 1 \qquad \forall i = 1,\dots,n \tag{16}$$

$$\sum_{r\in R_k} k_r^k \le n_k \qquad \forall k \in K \tag{17}$$

$$0 \le x_r^k \le 1 \qquad \forall k \in R_k \tag{18}$$

Creating all feasible routes is considered as a NP-hard problem. The main idea of column generation approach is to use only a small number of feasible routes in order to find the optimal solution out of a large set of possible feasible routes and additional routes are added only when needed. Therefore, at first, we considered the Restricted Master Problem (RMP) containing only the routes that have been generated by sweep algorithm. Let generated routes for vehicle type k indicated by $R'_k \subset R_k$, then the RMP created by replaced R_k in the MP by R'_k. The RMP is solved from this solution and we are able to obtain shadow prices for each of the constraints in the RMP. This information is then utilized in the objective function of the Pricing Sub-Problem. Each time the Pricing Sub-Problem is being solved, new routes are being generated as needed and inserted in the set $R' = \bigcup_{k\in K} R'_k$.

We solve the RMP by solver CPLEX 12.1 and vector $\pi^*=(\pi^*_{i1},\dots,\pi^*_{in},\pi^*_1,\dots,\pi^*_m)$ is the values of the dual variables (shadow price) corresponding to constraints (16) and (17).

Pricing Sub-Problem

The Pricing Sub-Problem (PSP) is a new problem created to identify a new route–vehicle pair with the minimum reduced cost. The PSP uses the shadow price information from RMP to generate promising a route–vehicle pair that is also feasible. We formulated a Pricing Sub-Problem for vehicle type k (PSP(k)) and consequently have m problems. We consider vehicle type k and $r \in R_k$. This open route is as follows:

$r=(i_0,i_1,\ldots,i_H,i_{H+1});\ i_0=,\ i_{H+1}\neq 0$

The reduced cost $\bar{c}_r^k$ of $r \in R_k$ is defined as:

$$\bar{c}_r^k = c_r^k - \sum_{i\in I} \pi_i^* a_{ir}^k - \pi_k^* = (\sum_{h=1}^{H} c_{i_{h-1},i_h}^k + f_k) - \sum_{h=1}^{H} \pi_{i_h}^* - \pi_k^* =$$
$$= \sum_{h=1}^{H+1} (c_{i_{h-1},i_h}^k - \pi_{i_h}^*) + f_k \qquad (19)$$

Where $\pi_{i0}^*=\pi_k^*$. Therefore, reduced cost $\bar{c}_r^k$ is equal to sum of the fixed cost and the cost of route r on the directed sub-graph $G_k=(V,A_k)$ for vehicle type k with arc modified cost defined as follows:

$$\bar{c}_{ij}^k = \begin{cases} c_{ij}^k - \pi_j^* - \pi_k^* & i = 0 \\ c_{ij}^k - \pi_j^* & i \neq 0 \end{cases} \qquad (20)$$

Now, the PSP formulation can be written as follows:

$$PSP(k){:}Min\_\bar{c}_r^k = f_k + \sum_{i=0}^{n}\sum_{j=0}^{n} \bar{c}_{ij}^k x_{ij}^k \qquad (\ 21)$$

Subject to

$$\sum_{i=0}^{n} x_{ij}^k \leq 1 \qquad \forall j = 1,2,\ldots,n \qquad (22)$$

$$\sum_{j=1}^{n} x_{ij}^k \leq 1 \qquad \forall i = 1,\ldots,n \qquad (23)$$

$$0 \leq \sum_{i=0}^{n} x_{ij}^k - \sum_{i=0}^{n} x_{ji}^k \leq 1 \qquad \forall j = 1,2,\ldots,n \qquad (24)$$

$$\sum_{i=1}^{n} x_{i0}^k = 0 \qquad (25)$$

$$\sum_{i=0}^{n} y_{ij}^k - \sum_{i=0}^{n} y_{ji}^k = q_j \sum_{i=0}^{n} x_{ij}^k \quad \forall j = 1,2,\ldots,n \qquad (26)$$

$$q_j\, x_{ij}^k \leq y_{ij}^k \leq (Q_k - q_i) x_{ij}^k \quad \forall i,j = 0,1,2,\ldots,n\ i \neq j \qquad (27)$$

$$y_{ij}^k \geq 0 \ \ \forall i,j = 0,1,2,\ldots,n \qquad (28)$$

$$x_{ij}^k \in \{0,1\} \ \ \forall i,j = 0,1,2,\ldots,n, i \neq j \qquad (29)$$

Therefore, the result of PSP(k) will be one route for vehicle type k, if the optimal objective value of PSP(k) is negative, the route–vehicle pair having the minimum reduced cost $\bar{c}_r^k$ will be added to the set R'_k in RMP. Then the RMP will be solved again and obtain shadow prices for each of the constraints in RMP. For vehicle type k, arc modified cost $\bar{c}_{ij}^k$ is obtained and the process is repeated until no routes with negative reduced cost are identified. The CG approach will be described as follows:

1. Find initial sets R' of routes by sweep algorithm for the MP.

2. Solve the MP by using Solver CPLEX 12.1 and obtain the shadow prices of the optimal solution.

3. For vehicle type k, produce the modified costs $\bar{c}_{ij}^k$, as equation (20).

4. For vehicle type k, solve PSP(k) by using Solver CPLEX 12.1 and Find feasible route–vehicle pairs with negative reduced costs and add them to the sets R'_k.

5. If no new route–vehicle pair was found go to Step 6, otherwise continue with step 2.

6. If solution obtained from the last RMP is integer, the approach terminates, otherwise replace R_k in HFFOVRP formulation by R'_k, then solve it and the approach terminates.

Figure 3 presents the column generation approach flowchart.

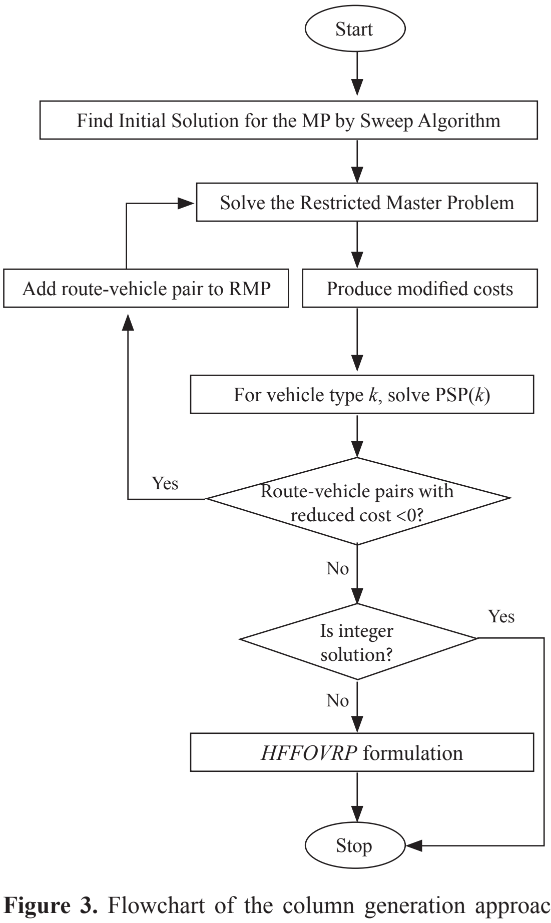

Start
Find Initial Solution for the MP by Sweep Algorithm
Solve the Restricted Master Problem
Add route-vehicle pair to RMP
Produce modified costs
For vehicle type k, solve PSP(k)
Yes
Route-vehicle pairs with reduced cost <0?
No
Yes
Is integer solution?
No
HFFOVRP formulation
Stop

Figure 3. Flowchart of the column generation approach for the HFFOVRP.

4. Results

The proposed algorithm was coded in Aimms with solver GUROBI 4.5 and all the experiments were implemented on a PC with Pentium 4 at 2.3GHZ and 4GB RAM running Windows XP Home Basic Operating system. AIMMS is an advanced development environment to build advanced planning systems and optimizing the problems in applied research studies. The numerical experiment is performed using two sets of problem instances. These problems are built of 10-85 nodes including the depot that all randomly located over a square with no service time. They have fixed fleet with capacity restrictions, without route length. Euclidean distances are used in the all problems. The first one consists 14 generated by author instances from 10 to 85 customers and the second one consists 5 test problems available in the literature from the well-known Taillard's benchmark for Heterogeneous fixed fleet Vehicle Routing Problem (HFFVRP) provided by Taillard (1999). It is noted that the first data set is also derived from Taillard's benchmark and the second set consists of instances 9, 10, 11, 12 and 17 with sizes ranging 50, 50, 50, 50 and 75 respectively without the depot. For more information regarding the provided examples visit:

http://mistic.heig-vd.ch/taillard/problemes.dir/vrp.dir/vrp.html

In this section, we first introduce the benchmark problems and then the detailed computational results obtained. The specifications of these twenty two problems are reported in Table 1.

Table 1. Data for problems.

Instance	Number of customers	Number of different type vehicles	Capacity	Fixed cost	Variable cost	Available number of kind k^{th} of vehicle
1	10	1	20	20	1	1
		2	30	40	1.1	1
		3	40	70	1.3	2
		4	70	200	1.7	1
2	15	1	30	60	1	1
		2	60	100	1.1	1
		3	80	250	1.5	1
		4	150	300	2	1
3	20	1	20	70	1	1
		2	35	120	1.1	2
		3	50	200	1.2	2
		4	120	250	2	3
4	25	1	25	50	1	2
		2	35	80	1.1	2
		3	50	200	1.2	3
		4	120	250	1.7	3
5	30	1	25	35	1	3
		2	35	50	1.1	2
		3	50	75	1.2	4
		4	120	150	1.7	4
6	35	1	50	60	0.7	1
		2	120	75	1	2
		3	160	200	1.1	3
7	40	1	60	20	1	3
		2	140	50	1.7	2
		3	200	120	2	1
8	45	1	50	20	1	2
		2	150	35	1.4	1
		3	200	50	1.4	1
		4	300	120	1.7	1
9	50	1	20	20	1	4
		2	30	35	1.1	2
		3	40	50	1.2	4
		4	70	120	1.7	4
		5	120	225	2.5	2
		6	200	400	3.2	1

Instance	Number of customers	Number of different type vehicles	Capacity	Fixed cost	Variable cost	Available number of kind k^{th} of vehicle
10	50	1	120	100	1	4
		2	160	1500	1.1	2
		3	300	3500	1.4	1
11	50	1	50	100	1	4
		2	100	250	1.6	3
		3	160	450	2	2
12	50	1	40	100	1	2
		2	80	200	1.6	4
		3	140	400	2.1	3
13	55	1	20	10	1	2
		2	50	35	1.3	2
		3	100	100	1.9	2
		4	150	180	2.4	1
		5	250	400	2.9	1
		6	400	800	3.2	1
14	60	1	20	10	1	2
		2	50	35	1.3	2
		3	100	100	1.9	2
		4	150	180	2.4	1
		5	250	400	2.9	1
		6	400	800	3.2	1
15	65	1	20	10	1	1
		2	50	35	1.3	2
		3	100	100	1.9	2
		4	150	180	2.4	2
		5	250	400	2.9	1
		6	400	800	3.2	1
16	70	1	20	10	1	3
		2	50	35	1.1	3
		3	100	100	1.2	2
		4	150	180	1.4	2
		5	250	400	1.9	1
		6	400	800	2.2	1
17	75	1	20	10	1	4
		2	50	35	1.3	4
		3	100	100	1.9	2
		4	150	180	2.4	2
		5	250	400	2.9	1
		6	400	800	3.2	1
18	80	1	20	10	1	5
		2	50	35	1.3	6
		3	100	100	1.9	2
		4	150	180	2.4	2
		5	250	400	2.9	1
		6	400	800	3.2	1
19	85	1	20	10	1	5
		2	50	35	1.3	6
		3	100	100	1.9	3
		4	150	180	2.4	2
		5	250	400	2.9	1
		6	400	800	3.2	1

The proposed compound heuristic algorithm is compared to the solver Cplex 12.1 in AIMMS as an exact algorithm in the Table 2. In this table, the first column includes the instance name, the second column shows the number of customers n, and the third and fourth columns present the results obtained by exact algorithm and its CPU time. It is noted that in these results shown in the third column, the exact algorithm continues until the software finishes the search. Number of generated pair (r,k) for each instance is presented in fifth column. Besides, the sixth and seventh columns of Table 2 show the results of the proposed algorithm and running time of the algorithm. The best results of the exact algorithm for the presented time in the seventh column are shown in the eighth column. In other word, the exact algorithm continues until the presented time shown in seventh column is over. The best solutions found by sweep algorithms is displayed in the ninth column and finally, to show the method's performance more clearly, we present the best known solutions (BKS) in the last column of Table 2.

A simple criterion to measure the efficiency of two algorithms CG and Sweep algorithm is to compute the relation percentage deviation of their solutions on specific benchmark instances. These values are calculated by below formula.

$$\text{Gap} = 100 \times \frac{(\text{value of the Sweep algorithm - value of the CG algorithm})}{\text{value of the CG algorithm}}$$

Figure 4 presents the gap between these two algorithms for the all the instances. In this figure, the horizontal axis shows the name of instances and the vertical axis indicates the percentage of sweep algorithm compared to the CG. Form this figure, we conclude that the sweep algorithm is high quality algorithm in order to produce an initial solution because this algorithm able to find very good solution with average almost 41% gap in comparison with the CG. Moreover, the lower gap in all instances is 1 with almost 17% gap and the upper gap is instance 19 with almost 70% gap. Finally, we see that there is not any relationship between size of the instance and value of gap in this figure.

Generally, the exact algorithm fails to find optimal solutions for most of the problems especially instances with more than 35 customers and is not able to be used as an efficient and applicant algorithm. As a result, the performance comparison of results between Cplex and the CG shows that the proposed algorithm clearly yields better CPU time than the other algorithms in Table 2. In more detail, the CG not only can find optimal solution for four inctance including 1, 2, 3 and 4 but also this algorithm capale to find near optimal in two other instance 5 and 6. The ratio of CPU time exact algorithm to CG until instances with 35 customers is shown in Figure 5. In this figure, the horizontal axis shows the customer's number of instance 1 to instance 6 and the vertical axis indicates the ratio of these algorithms.

As mention above, the result of CG and exact algorithm in the same CPU time are shown in the column 6 and 8 respectively in Table 2. Column 7 in Table 2 shows this CPU time in second. It is found in table 2 that CPLEX can reach an optimal solution for only one small-scale instance out of 5 instances. Furthermore, we find that in other instances the exact algorithm lower bound is far away from its best solution

Table 2. Comparison results between the proposed algorithm and Aimms.

Instance	*n*	CPLEX12.3	Time (Sec)	**#(*r*,*k*)**	CG	Time (Sec)	CPLEX12.3	Sweep	BKS
1	10	191.10	1.09	107	191.10	0.36	191.10	224.22	191.10
2	15	282	171.02	168	282	4.28	525.56	345.14	282
3	20	379.63	16.55	277	379.63	5.44	390.25	465.01	379.63
4	25	437.79	22.78	397	437.79	9.78	532.18	534.09	437.79
5	30	472.76	61.78	585	473.31	16.46	497.48	615.69	472.76
6	35	346.26	55567.75	788	349.95	370.08	Na	487.61	346.26
7	40	Na	-	629	600.99	767.42	Na	997.13	600.99
8	45	Na	-	787	676.04	2408.40	Na	950.46	676.04
9	50	Na	-	1263	907.3	256.72	Na	1308.23	907.3
10	50	Na	-	867	507.58	2253.26	Na	737.18	507.58
11	50	Na	-	855	826.19	840.18	Na	1061.61	826.19
12	50	Na	-	886	947.81	698.96	Na	1261.77	947.81
13	55	Na	-	1110	1074.91	1940.82	Na	1801.45	1074.91
14	60	Na	-	1274	1937.03	8807.25	Na	2239.13	1937.03
15	65	Na	-	1363	1563.33	4203.14	Na	2047.70	1563.33
16	70	Na	-	1638	962.57	6357.70	Na	1452.27	962.57
17	75	Na	-	1873	1356.67	6021.97	Na	2274.05	1356.67
18	80	Na	-	2222	1285.74	8331.64	Na	2162.70	1285.74
19	85	Na	-	2470	1295.58	13731.35	Na	2199.34	1295.58

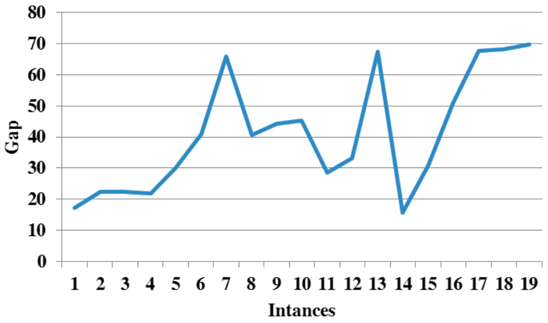


Figure 4. Gap between CG and Sweep algorithms.

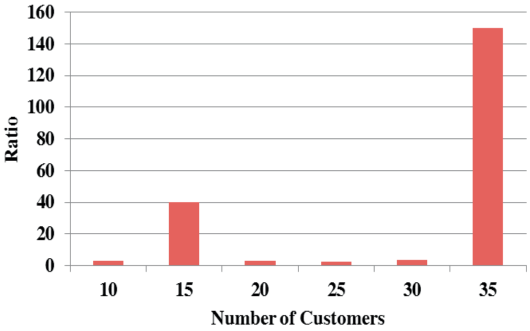


Figure 5. CPU comparison between CG and Cplex.

result of the BKS. Besides, the CG has been able to find the best solutions in four instances including 1, 2, 3 and 4 of the 5 examples. Therefore, the CG has been able to find the better solutions than exact algorithm in four out of five examples including 2, 3, 4 and 5. In other words, in 4 examples except instance 1 among remaining 5 examples, the proposed algorithm has been capable of improving the solutions gained from exact algorithm. Furthermore, CG has failed in improving the solutions in only instance 1 and has come up with solutions similar to the ones found by exact algorithm. Hence, it can be concluded that CG is more efficient than exact algorithm in finding good solutions at the same CPU time.

5. Conclusion and Future Works

This paper investigates HFFOVRP in the transportation system and presents a new combined heuristic algorithm base on CG, which has allowed us to obtain high quality solutions which cannot obtain by exact algorithm. The HFFOVRP is a variant of the classical OVRP in which customers are served by a given hired heterogeneous fixed fleet of vehicles with various capacities and variable costs. This problem has significant applications in the transportation system, especially when a company used hired vehicles to serve customers. Computational results generally have shown that the proposed CG gives better results compared to the exact algorithm in terms of the solution quality in the same CPU time. It seems that using effective heuristic algorithms can lead to gain better initial solutions than the sweep algorithm. Furthermore, powerful metaheuristic can be used for solving this problem and other versions of HFFOVRP. Future projects will focus on working on such ideas and making them operational.

References

Aleman, R. E., Hill, R. R. (2010). A tabu search with vocabulary building approach for the vehicle routing problem with split demands, *International Journal of Metaheuristics, 1*(1), 55-80. https://doi.org/10.1504/IJMHEUR.2010.033123

Anbuudayasankar S.P., Ganesh b, K., Lenny Koh S.C., Ducq, Y. (2012). Modified savings heuristics and genetic algorithm for bi-objective vehicle routing problem with forced backhauls. *Expert Systems with Applications*, *39*, 2296-2305. https://doi.org/10.1016/j.eswa.2011.08.009

Brandão J. (2009). A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research, 195*, 716-728. https://doi.org/10.1016/j.ejor.2007.05.059

Çatay, B. (2010). A new saving-based ant algorithm for the Vehicle Routing Problem with Simultaneous Pickup and Delivery. *Expert Systems with Applications, 37*(10), 6809-6817. https://doi.org/10.1016/j.eswa.2010.03.045

Dantzig, G., Ramser, J. (1959). The truck dispatching problem. *Management Science, 6*, 80-91. https://doi.org/10.1287/mnsc.6.1.80

Gendreau, M., Guertin, F., Potvin, J. Y., Séguin, R. (2006). Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries. *Transportation Research Part C, 14*, 157-174. https://doi.org/10.1016/j.trc.2006.03.002

Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers and Operations Research, 26*, 1153-1173. https://doi.org/10.1016/S0305-0548(98)00100-2

Lei, H., Laporte, G., Guo, B., (2011). The capacitated vehicle routing problem with stochastic demands and time windows. *Computers & Operations Research, 38*(12), 1775-1783. https://doi.org/10.1016/j.cor.2011.02.007

Li, X., Leung, S. C. H., Tian, P. (2012). A multi start adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. *Expert Systems with Applications, 39,* 365-374. https://doi.org/10.1016/j.eswa.2011.07.025

Li, X., Tian, P., Aneja, Y. P. (2010). An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Transportation Research Part E, 46*, 1111-1127. https://doi.org/10.1016/j.tre.2010.02.004

Penna, P. H. V., Subramanian, A., Ochi, L. S. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics, 19*(2), 201-232. https://doi.org/10.1007/s10732-011-9186-y

Saadati Eskandari, Z., YousefiKhoshbakht, M. (2012). Solving the Vehicle Routing Problem by an Effective Reactive Bone Route Algorithm, *Transportation Research Journal, 1*(2), 51-69.

Subramanian, A., Drummond, L., Bentes, C., Ochi, L.S., Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research, 37*(11), 1899-1911. https://doi.org/10.1016/j.cor.2009.10.011

Syslo, M., Deo, N., Kowalik, J. (1983). *Discrete Optimization Algorithms with Pascal Programs*, Prentice Hall.

Taillard, E. D. (1999). A heuristic column generation method for the heterogeneous fleet VRP, *RAIRO Operations Research, 33*, 1-14. https://doi.org/10.1051/ro:1999101

Tarantilis, C. D., Kiranoudis, C. T. (2007). A Flexible Adaptive Memory-based Algorithm for Real-life Transportation Operations: Two Case Studies from Dairy and Construction Sector. *European Journal of Operational Research, 179*, 806-822. https://doi.org/10.1016/j.ejor.2005.03.059

Wang, H. F., Chen, Y. Y. (2012). A genetic algorithm for the simultaneous delivery and pickup problems with time window. *Computers & Industrial Engineering, 62*, 84-95. https://doi.org/10.1016/j.cie.2011.08.018

YousefiKhoshbakht, M., Didehvar, F. and Rahmati, F. (2014). Solving the Heterogeneous Fixed Fleet Open Vehicle Routing Problem by a Combined Meta-heuristic Algorithm. *International Journal of Production Research, 52*(9), 2565-2575. https://doi.org/10.1080/00207543.2013.855337

Yousefikhoshbakht, M., Dolatnejad, A., Didehvar, F., Rahmati, F. (2016). A Modified Column Generation to Solve the Heterogeneous Fixed Fleet Open Vehicle Routing Problem, *Journal of Engineering*, https://doi.org/10.1155/2016/5692792

YousefiKhoshbakht, M. and Khorram, E. (2012). Solving the vehicle routing problem by a hybrid meta-heuristic algorithm. *Journal of Industrial Engineering International, 8*(11), 1-9. https://doi.org/10.1186/2251-712X-8-11

Appendix A.

Best solutions obtained by CG

The best solutions found by the proposed algorithm for the problems are presented in Tables 1-19. All the calculations have been performed with a precision of 64 bits and the total solution cost is presented with three or four decimal places.

Instance 1	Routes	Kind of vehicle	Cost
1	1-7-8	3	36.262
2	1-6-10-11	3	50.119
3	1-5	1	14.866
4	1-3-4	4	61.745
5	1-2-9	2	28.110
Sum	-	-	191.102

Instance 2	Routes	Kind of vehicle	Cost
1	1-10-11-16	1	50.93977564
2	1-13-6-12-3-4	4	108.7786929
3	1-7-15-5-14	3	78.72884492
4	1-2-9-8	2	43.54901829
Sum	-	-	281.9963

Instance 3	Routes	Kind of vehicle	Cost
1	1-3-2	3	34.94452747
2	1-17-4-19	3	53.63437012
3	1-7	1	9.219544457
4	1-14-21	2	48.33244534
5	1-6-16	2	36.36848352
6	1-18-13-10-11	4	86.07287612
7	1-8-9-20-5-12	4	96.91824471
8	1-5	4	14.14213562
Sum	-	-	379.6326

Instance 4	Routes	Kind of vehicle	Cost
1	1-6-22	3	45.64608577
2	1-11	2	28.600
3	1-3	2	16.01624176
4	1-14-16-21	4	53.19762955
5	1-17-24	1	32.22273631
6	1-18-13-10-26-19-25	3	105.6486204
7	1-7-2	3	42.41666743
8	1-8-9-20-15-12	4	82.380508
9	1-3	1	19.6468827
10	1-5	4	12.02081528
Sum	-	-	437.7962

Instance 5	Routes	Kind of vehicle	Cost
1	1-4-17-24	2	45.5984443
2	1-28-14	1	23.22656223
3	1-7-3-29-22	4	63.41043657
4	1-2-23	1	35.30470192
5	1-8-9-20-15-12	4	82.380508
6	1-5-31-30-6-16-21	4	86.41338606
7	1-27-11	3	31.3292963
8	1-18-13-10-26-19-25	4	105.6486204
Sum	-	-	473.312

Instance 6	Routes	Kind of vehicle	Cost
1	1-31-22	1	19.14935811
2	1-3-29-23-2-24-25	2	71.99992496
3	1-27-8-36-20-15-12	3	52.62606364
4	1-5-35-9-14-28-30-6-16-21	3	82.03318409
5	1-7-34-17-4-33-10-26-19	3	79.53758332
6	1-18-13-11-32	2	44.60737999
Sum	-	-	349.9535

Instance 7	Routes	Kind of vehicle	Cost
1	1-41-23-3-16	1	52.1982838
2	1-14-7-6-18-17-38-15-39	2	139.0471265
3	1-19-9-8-12-20-37	1	73.52931824
4	1-28-32-11-33-31-21-10-36	2	125.7796298
5	1-2-34-4-25-30-35	1	69.13258122
6	1-29-13-27-22-5-26-40-24	3	141.302948
Sum	-	-	600.9899

Instance 8	Routes	Kind of vehicle	Cost
1	1-37-8-20-12	1	80.24315107
2	1-3-16-44-43-14-15-38-39-45-17-7	3	273.1300937
3	1-2-34-4-30-25	1	51.43887377
4	1-29-13-27-41-22-5-26-40-24-23-42	4	153.3682033
5	1-28-32-11-33-31-21-10-36-35	2	117.8604791
Sum	-	-	676.0408

Instance 9	Routes	Kind of vehicle	Cost
1	1-15	1	27.80287755
2	1-39	1	26.92582404
3	1-47	2	12.29837388
4	1-25	1	33.13608305
5	1-12	1	29.15475947
6	1-4-19-26	3	56.6872941
7	1-8-36-20	3	31.70669462
8	1-7-34-2-44-42-43	5	103.267770
9	1-18-41-45-33-51	5	91.76199689
2	1-31-49-22	4	51.56637333
10	1-5-46-30-6-48-37-38-21	6	168.6069753
11	1-28-14-16	3	41.07187468
12	1-35-9	3	20.4852813
13	1-27-11-32	4	67.2543306
14	1-13-40-10	4	45.05302896
15	1-3-29-23	4	57.96477825
16	1-17-50-24	2	42.55449871
Sum	-	-	907.2988

Instance 10	Routes	Kind of vehicle	Cost
1	1-21-46-30-38-6-37	1	94.59151952
2	1-47-35-14-28-16-5	1	65.94248802
3	1-18-41-33-45-4-17-50-25	3	62.37450693
4	1-13-40-10-26-51-19	1	51.04362313
5	1-7-34-2-44-43-42-24	1	52.97935262
6	1-27-8-36-9-20-15-12-39-11-32	2	114.3707895
7	1-3-31-49-48-22-29-23	3	66.27947061
Sum	-	-	507.5818

Instance 11	Routes	Kind of vehicle	Cost
1	1-49-27-28	1	49.18808315
2	1-9-32-29	1	37.56681532
3	1-15-25-7-44-24-8	2	136.4414249
4	1-11-6-34-46-16-13	2	150.2146344
5	1-12-39-50-10-17-51-35-31-40	3	143.1569457
6	1-48-19-26-14-42-20-41	3	133.7873043
7	1-5-18-38-45-43	1	47.49518318
8	1-33-2-23-4-21-36-37	2	90.11914556
9	1-47-3-30-22	1	38.22563859
Sum	-	-	826.1952

Instance 12	Routes	Kind of vehicle	Cost
1	1-44-24-8-49-27	2	125.1892307
2	1-6-34-46-16-13-38-45-18-43	3	221.9402168
3	1-14-19-26-15-25-7	3	180.3042078
4	1-39-10-22-30	1	43.75622479
5	1-50-11-40	1	39.24908076
6	1-47-12-3-17-51-35-31	3	105.7905219
7	1-28-9-32-294	2	75.30656257
8	1-48-5-42-20-41	2	74.12480107
9	1-33-2-23-21-36-37	2	82.1455313
Sum	-		947.8064

Instance 13	Routes	Kind of vehicle	Cost
1	1-14-16	6	33.561
2	1-27-13-40-10-26-56	3	90.871
3	1-5-31-49-6-30-46-28-53-35-47-9-36-54-15-12-39-11-32	1	390.379
4	1-8-20-55	5	45.383
5	1-52-24-50-25	5	64.040
6	1-7-34-2-44-42-43-23-29-22-48-37-38-21	2	284.046
7	1-18-41-4-45-33-51-19	4	112.274
8	1-3	4	34.945
9	1-17	6	19.416
Sum	-	-	1074.915

Instance 14	Routes	Kind of vehicle	Cost
1	1-20-9-47-55-35-53-14-28-58-16-5-21-46-30	2	529.739
2	1-38-6-61	5	74.677
3	1-4-45-19-18-51-33-41-56-26-10-40-13-32-11-59-27-39-12-54	1	830.922
4	1-15	6	27.802
5	1-57-24-17-52-50-25	4	168.043
6	1-22	6	27.294
7	1-7-34-2-44-42-43-23-29	3	148.414
8	1-8-36-60	5	51.558
9	1-3-31-49-48-37	4	78.582
Sum	-	-	1937.031

Instance 15	Routes	Kind of vehicle	Cost
1	1-10-40-13-32-11-59-27-39-66-12-54-8-60-15-36-20-9-47-55	1	727.202
2	1-64-57-24	5	55.975
3	1-46-30-49-48-37-61	3	93.873
4	1-35-53-28-14-58-16-6-38-21	4	143.206
5	1-52-50-25-19-51-26-56	3	136.353
6	1-7-63-23	5	41.080
7	1-5-31-3-29-22-62	4	128.298
8	1-18-41-33-45-4-17-34-2-44-42-43-65	2	236.353
Sum	-	-	1562.34

Instance 16	Routes	Kind of vehicle	Cost
1	1-18	6	8.062
2	1-34-64-57-24-17	4	68.869
3	1-5	6	7.071
4	1-63-23-65-43-2-44-7-42	3	168.074
5	1-32	6	37.108
6	1-52-50-25-19-51-26-56	4	86.117
7	1-27-13-41-4-45-33-10-40-59-11-39-66-67-12-54-15-60	1	257.205
8	1-68-35-47-9-53-28-46-30-6-38-21-71-61	2	133.779
9	1-31-49	5	23.528
10	1-55-14-58-16	5	52.377
11	1-69-3-29-62-22-48-37-70	3	91.320
12	1-8-36-20	5	29.064
Sum	-	-	962.574

Instance 17	Routes	Kind of vehicle	Cost
1	1-18-51	2	38.60423502
2	1-32	1	37.10795063
3	1-5	2	9.192388155
4	1-22	1	27.29468813
5	1-48-37-72-62	2	82.44826844
6	1-7-34-74-2-44-42-43-65	4	126.5496831
7	1-27-23-41-4-45-33-10-40-73-59-11-39-66-67-12-54-15-60	6	374.1160165
8	1-76-69-3-31-75-29-63-23	4	118.3894944
9	1-64-24-57	2	48.63452897
10	1-68-35-47-53-28-46-30-49-6-38-21-71-61	5	203.3042731
11	1-52-17-50-25-19-26-56	3	132.1757352
12	1-8-36-20-55-14-58-16	3	105.6807833
13	1-9	1	15.8113883
14	1-70	1	37.36308338
Sum	-	-	1356.673

Instance 18	Routes	Kind of vehicle	Cost
1	1-41	6	14.142
2	1-31	6	14.318
3	1-75-70	6	38.868
4	1-28-14-55	5	40.348
5	1-52-17-50-25-24-57	4	108.747
6	1-74-23	6	31.661
7	1-27-13-40-10-33-51-19	3	117.933
8	1-59-73-32	5	53.940
9	1-8-36-60	5	51.559
10	1-7-34-64-2-44-42-43-65	3	135.777
11	1-53-81-58-16	5	44.297
12	1-4-45-26-56	5	62.867
13	1-76-5-46-30-49-6-38-21-71-61-72-37-48-22	2	237.586
14	1-69-3-63-29-62	4	75.047
15	1-68-77-35-47-78-9-80-20-79-15-54-12-67-66-39-11	1	250.591
16	1-18	6	8.062
Sum	-	-	1285.743

Instance 19	Routes	Kind of vehicle	Cost
1	1-51	6	29.68
2	1-27	6	6.083
3	1-49	6	20.616
4	1-3	6	14.560
5	1-29	6	24.515
6	1-83-19-26-56	5	74.505
7	1-28-14-55	5	40.348
8	1-8-36-20-79-15-60	4	87.922
9	1-53-81-58-16	5	44.297
10	1-59-73-32	5	53.940
11	1-13-41-40-10-33-45-4	3	111.872
12	1-68-77-35-47-78-9-80-82-54-12-67-66-39-11	1	212.519
13	1-76-69-7-34-74-2-86-44-42-43-65	2	169.280
14	1-31-75-22-48-37-70	4	93.151
15	1-64-85-24-57	5	52.321
16	1-18-52-17-84-50-25	4	74.836
17	1-63-23-62	5	55.418
18	1-5-46-30-6-38-21-71-61-72	3	129.714
Sum	-	-	1295.577