

Research Article

A Modified Column Generation to Solve the Heterogeneous Fixed Fleet Open Vehicle Routing Problem

Majid Yousefikhoshbakht,¹ Azam Dolatnejad,² Farzad Didehvar,³ and Farhad Rahmati³

¹Young Researchers & Elite Club, Islamic Azad University, Hamedan Branch, Hamedan 65181-15743, Iran

²Young Researchers & Elite Club, Islamic Azad University, Tehran North Branch, Tehran 19136-74711, Iran

³Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran 15875-4413, Iran

Correspondence should be addressed to Majid Yousefikhoshbakht; yousefikhoshbakht@gmail.com

Received 18 November 2015; Revised 4 March 2016; Accepted 30 March 2016

Academic Editor: YoungSu Yun

Copyright © 2016 Majid Yousefikhoshbakht et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the heterogeneous fixed fleet open vehicle routing problem (HFFOVRP), several different types of vehicles can be used to service the customers. The types of vehicles are different in terms of capacity, fixed cost, and variable cost. In this problem, the vehicles are not required to return to the depot after completing a service and the number of vehicles of each type is fixed and limited. Since this problem belongs to NP-hard problems, in this paper a compound heuristic algorithm called SISEC which includes sweep algorithm, insert, swap, and 2-opt moves, modified elite ant system (EAS), and column generation (CG) is applied to solve the HFFOVRP. We report computational results on 22 problems and solve each problem by using our SISEC. The results which were compared to the results of exact algorithms and the classic CG confirm that the proposed algorithm produces high quality solutions within an acceptable computation time.

1. Introduction

The open vehicle routing problem (OVRP) is a variant of vehicle routing problem (VRP) which has many applications in industry and service providing businesses [1]. The description of the OVRP appeared in the literature over 30 years ago but it has attracted the attention of scientists and researchers in recent years. In the OVRP, the objective is to minimize the total traveled distance (or time) by the number of vehicles so that each customer is visited only once by a single vehicle. In general, the number of vehicles used is not imposed but it is a decision variable. Furthermore, the demand of each customer must be completely fulfilled by a single vehicle and the total demand served by each vehicle must not exceed vehicle capacity; that is, split deliveries are not allowed. Moreover, in one variant of the problem, the travel time of each vehicle should not exceed an upper limit. It should be noted that the OVRP has a unique character in that it has an open path form. This means that the vehicles are not required to return to the depot after completing a service

and each vehicle ends its trip at a customer [2]. This problem is defined as follows.

Let $G = (V, E)$ be a complete undirected connected graph with $V = \{0, 1, \dots, n\}$ as the set of vertices and the set of arcs $E = \{(i, j) : 0 \leq i, j \leq n\}$ (if the graph is not complete, we can instead replace each arc with an arc which has an infinite size). Vertex 0 denotes the depot, and each vertex $i \in \{1, \dots, n\}$ is a customer with a nonnegative demand $0 < q_i < Q$. The cost of travel from vertex i to vertex j is denoted by c_{ij} (if $i = j$, $c_{ii} = 0$ for $0 \leq i \leq n$), and it is assumed that costs are symmetric (i.e., $c_{ij} = c_{ji}$). The available fleet consists of K vehicles located at the depot. Furthermore, each customer must be serviced by a single vehicle and no vehicle may serve a set of customers whose total demand exceeds its capacity. Each vehicle route must start at the depot and end at the last customer it serves. The objective is to define a set of vehicle routes which minimizes the total costs [3].

In the standard version of the VRP and OVRP, a fleet of *homogeneous* vehicles is used and exists at a single depot [4]. A sequence of customers must be visited by each vehicle

so that all customers are serviced exactly once and the total traveled distance by all the vehicles is minimized. In contrast to other customers, some customers may have such large demands that they can be serviced only by large capacity vehicles. Therefore, several different vehicle types are applied in the *heterogeneous* version of VRP [5]. There are n_i vehicles (unlimited or limited numbers) of type i which can be used to serve the customers. When the number of vehicles of type t is *unlimited*, the best composition of the fleet is determined. This problem involves two kinds of decisions including

- (i) the choice of a mix of vehicles among the available vehicle types,
- (ii) the routing for each selected vehicle.

In the fixed fleet version of the *heterogeneous* VRP (HFFVRP), the values of n_i are fixed and the number of vehicles for type t is *limited*. If the fleet composition is known in advance, we decide how to make the best use of heterogeneous vehicles. The HFFVRP can be defined on an undirected complete graph with one depot and n customers indexed from 1 to n . A fleet of identical heterogeneous vehicles with limited capacity Q_k for each k is based at the depot. Furthermore, the vehicles do not necessarily have the same capacity, vehicle fixed cost, and unit variable cost in the HFFVRP. In other words, a capacity Q_k , a fixed cost f_k , and a variable cost α_k are associated with each type of vehicle k so that 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 customers i to j with a vehicle of type k . The HFFVRP is a further generalization with a limited availability n_k for each vehicle type k . This problem deals with finding the minimum total transportation cost including the fixed and variable cost for a fleet of vehicles so that the following constraints are taken into account:

- (i) Each vehicle starts at the depot and returns to the depot.
- (ii) The capacity of a vehicle cannot exceed the capacity of the corresponding vehicle type.
- (iii) The used number of vehicles of type k cannot exceed n_k .
- (iv) The demand of each customer is satisfied by exactly one vehicle in only one visit.

The aim of this paper is to study the open version of HFFVRP. In residential and commercial waste collection, there are several applications of the HFFVRP [6]. In this problem, vehicles commonly have different capacities and a specific vehicle is used when the actual service at a location must be completed such as a side loader with mechanical arms which picks up the trash cans. As a result, in logistic operations, decisions relating to routing heterogeneous fleets of vehicles are frequently taken into consideration [7]. The difference between the open HFFVRP and the HFFVRP is the fact that a solution of the former consists of a set of Hamiltonian paths, rather than Hamiltonian cycles [8, 9]. At first sight, having open routes instead of closed ones looks like a minor modification. Indeed, if travel costs are asymmetric, there

is essentially no difference between the open and closed versions. To transform the open version into the closed one, it suffices to set the cost to zero for traveling from any customer to the depot. However, if travel costs are symmetric, things are more subtle. Indeed, it is proved that, somewhat surprisingly, the open version turns out to be more general than the closed one in the sense that any closed HFFVRP on n customers can be transformed into an open HFFVRP on n customers, but there is no transformation in the reverse direction.

On the other hand, the OVRP is not very realistic since most companies have heterogeneous fleets of vehicles. If using heterogeneous fixed fleet as constraints are assumed for the OVRP, we call the problem a heterogeneous fixed fleet open vehicle routing problem (HFFOVRP) which is a relaxation of the standard OVRP. In the HFFOVRP, the fleet is composed of several vehicle types. Unlike the VRP, the number of vehicles of each type is limited [10]. Moreover, the vehicles have different capacities, fixed costs, and variable costs per unit distance. The idea of this problem is not only to consider the routing of the vehicles but also to take into account the composition of the vehicle fleet. This problem is more practical in distribution management and transportation [11]. In more detail, the HFFOVRP appears in situations where the fleet is purchased: it combines tactical decisions (selecting the number of vehicles to be acquired) and operational ones (computing the trips and the vehicles assigned to them).

For the first time, the HFFOVRP was addressed in [12]. In this paper, each arc (i, j) is associated with a traveling distance d_{ij} , a service time δ_i , and the maximum allowed traveling distance L_k . They proposed a multistart adaptive memory programming metaheuristic called MAMP with modified tabu search algorithm for solving this problem. This algorithm is based on the framework of the adaptive memory programming procedure constructed by Taillard et al. [13]. The major mechanism considered is to produce a high quality HFFOVRP by using the search memory and experience. The algorithmic efficiency and effectiveness are experimentally evaluated on a set of generated instances including 4, 15, and 20 solutions. The maximum number of iterations in all MAMPs is set to be 2500. Because the HFFOVRP is a new variant of the classical OVRP in that time, and there has been no related work done in the literature, the proposed algorithm is only compared with a tabu search algorithm because it is similar to the solution improvement procedure in MAMP.

Huachi et al. deal with both versions of the OVRP including homogeneous and heterogeneous fleet (HFFOVRP) in which their goals are to minimize traveled distance and the total costs [14]. In contrast to the HFFOVRP, the number of vehicles must be minimized when the fleet is homogeneous. The objective of this work is to present a heuristic algorithm based on the iterated local search (ILS) which uses a variable neighborhood descent (VNS) procedure in the local search phase. In this phase, the VNS is combined with random variable neighborhood ordering (RVND). The developed algorithm was tested in benchmark instances with up to 480 customers and the results obtained showed that the algorithm is quite competitive with those found in the literature. This approach is an extension of

[15] for the Heterogeneous Fleet Vehicle Routing Problem (HFVRP).

Yousefikhoshbakht et al. proposed a bone route algorithm (BRMTS) which uses the tabu search (TS) as an improved procedure to solve the HFFOVRP [9]. This population-based algorithm was first proposed for solving the classic VRP by Tarantilis and Kiranoudis [16] which constructs a new solution using components of routes in the adaptive memory. In that work, they present a version of this method which 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 (GEROCA) which was first presented in [17], for generating initial diversified solutions and modified TS improvement procedure [15]. They have made two main modifications to the TS:

- (i) To improve the TS further, the amount of tabu tenure (TT) is considered variable for intensification and diversification policies. The TT is considered a minimum value for the intensification policy and a maximum value for the diversification policy.
- (ii) The proposed TS comprises three types of neighborhood moves including 2-opt, 0-1, and 1-1 exchanges. These moves are distinguished in terms of exchanges performed to convert one tour into another. These moves are not equally performed in each iteration and probability of them is changed gradually during the run time.

Furthermore, some test problems were derived from the well-known Taillard's benchmark for HFFVRP and the results of BRMTS algorithm are compared with the results of several metaheuristic algorithms. The comparison showed that the proposed algorithm can provide better solutions within a comparatively shorter period of time.

The HFFOVRP plays an important role in supply chains specially in the first transportation step such as collecting agricultural products or in the final distribution phase toward customers. Besides, this problem is obviously NP-hard since it includes the OVRP as a special case. For such problems, the use of heuristics such as column generation (CG) is considered a reasonable approach in finding solutions. Although CG has been successfully applied to several combinatorial optimization problems, it has some shortcomings like its slow computing speed and local-convergence. Besides, despite the developments in modern heuristics, this algorithm has its own strengths and weaknesses. Therefore, many research studies have tried to develop hybrid algorithms, expecting to achieve the effectiveness and efficiency. In this paper, we have proposed an efficient hybrid heuristic called SISEC in order to improve both the performance of the algorithm and the quality of the solutions. The proposed algorithm took advantage of various versions of sweep algorithm with insert, swap, and 2-opt moves for producing a feasible solution. Furthermore, a tight integer programming model is presented. The linear programming relaxation is solved by the CG technique and an elite ant system (EAS) and several local search algorithms are used for solving generated Hamiltonian paths with some

modifications to generate feasible columns efficiently. The computational results are provided on a new set of generated instances derived from HFFVRP's problems. The results show that the hybrid algorithm explores different parts of the solution space and the search method is not trapped at the local optimum. Besides, the proposed approach is able to find high quality solutions in a very short time compared to exact and heuristic algorithms.

This paper is structured as follows. In Section 2, in addition to clarifying the intricate history of HFFOVRP and providing bibliographic references, we review three algorithms which have been developed to solve this problem and its extensions. Components for CG which are able to tackle the HFFOVRP are described in Section 3. In Section 4, we first generate 22 new test problems which have 10–100 customers and the results of the proposed algorithm are reported for these instances. Moreover, the proposed algorithm is compared to exact and CG algorithms. Finally, some concluding remarks are given in Section 5.

2. Related Works

The heterogeneous open vehicle routing problem (HOVRP) can be studied in two different versions. Some researchers consider an assumption that there is an unlimited number of vehicles of each type and try to find the optimal set of vehicles to be scheduled in the problem. This is called the fleet size and mix OVRP (FSMOVRP). However, to the best of our knowledge, no work has addressed this version of VRP which is more practical in distribution management and transportation. Other researchers study the case where there is a fixed vehicle fleet and try to schedule this fleet of vehicles to the customers in an optimal way. This problem is called heterogeneous fixed fleet OVRP (HFFOVRP) as mentioned before. This HFFOVRP is more realistic than FSMOVRP and has attracted more attention in literature. In the fixed fleet version of the OVRP, the values of n_k are fixed. More specifically, the number of vehicles of type k is limited and the fleet composition is known in advance. We must decide how to make the best use of a fixed fleet of heterogeneous vehicles.

The overarching idea is that many linear programs like HFFOVRP are too large to consider all the variables explicitly [18]. Since most of the variables will be nonbasic and assume a value of zero in the optimal solution, only a subset of variables need to be considered in theory when solving the problem. Column generation leverages this idea to generate only the variables which have the potential to improve the objective function—that is, to find variables with negative reduced cost (assuming without loss of generality that the problem is a minimization problem). So, the column generation is one of the effective algorithms in order to solve an NP-hard problems.

The HFFOVRP is defined on an undirected connected graph $G = (V, E)$, where $V = \{0, 1, \dots, n\}$ is a vertex set and $E = \{(i, j): i, j \in V\}$ is an edge set. Vertex 0 represents the depot and each other vertex $i \in V \setminus \{0\}$ is a customer with a demand q_i . A distance c_{ij} is associated with each edge $(i, j) \in E$. There is a fixed fleet of K different types of vehicles located at the depot and the number of vehicles of each type

is n_k ($k \in \{1, 2, \dots, K\}$). It should be noted that there is no guarantee that a feasible solution can be found because the number of vehicles of each type is fixed. It means that if this happens in real-life problems, other vehicles have to be hired in order to serve the customers. A capacity Q_k , a fixed cost f_k , and a variable cost α_k are associated with each type of vehicle k . In this problem, a sequence of deliveries is generated for a heterogeneous fleet of vehicles, in order to fulfill a known client requirement such that

- (1) each route starts at the depot and ends at an arbitrary customer,
- (2) each client is serviced by exactly one visit of a single vehicle,
- (3) the total demand of each route does not exceed vehicle's fixed capacity,
- (4) the total routing cost is minimized.

Furthermore, R_k is the set of feasible routes for vehicle type k , R is the set of all of feasible routes ($r \in R$, $R = \bigcup_{k \in K} R_k$), c_r^k is the cost of route r for vehicle type k ($(r, k) \in R_k$). $c_r^k = \sum_{(i,j) \in r} c_{ij}$, and y_{ij} is the flow variables specifying the quantity of goods which vehicle k is carrying when it leaves customer i to service customer j . The traveling cost of each edge $(i, j) \in E$ by a vehicle of type k is $c_{ij}^k = f_k + a_k \cdot c_{ij}$. The HFFOVRP consists of defining a set of routes and vehicles as assigned to each route so that no more vehicles than those available are used; customers' demands are satisfied; each customer is visited exactly once; each vehicle route starts at the depot and does not finish at the depot; and the capacity of each kind of vehicle is not exceeded. The objective of the HFFOVRP is to minimize the sum of the costs of all the routes subject to the mentioned constraints. The formulation of the HFFOVRP requires the following additional notation:

$$a_{ir}^k = \begin{cases} 1 & \text{if the customer } i \text{ is serviced by route } r \in R_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$x_r^k = \begin{cases} 1 & \text{if route } r \in R_k \text{ is used in the solution} \\ 0 & \text{otherwise.} \end{cases}$$

Now, HFFOVRP can be formulated as follows:

$$\min \sum_{k \in K} \sum_{r \in R_k} c_r^k x_r^k \quad (2)$$

$$\text{Subject to } \sum_{k \in K} \sum_{r \in R_k} a_{ir}^k x_r^k = 1 \quad \forall i = 1, \dots, n \quad (3)$$

$$\sum_{r \in R_k} x_r^k \leq n_k \quad \forall k \in K \quad (4)$$

$$x_r^k \in \{0, 1\} \quad \forall k \in K, r \in R_k. \quad (5)$$

In this formulation, the objective function (2) minimizes the total costs. Constraints (3) indicate that each customer

is assigned only once to a feasible route-vehicle pair. Constraints (4) ensure that a maximum number of used vehicles of each type does not exceed the number of available vehicles of the same type. Finally, constraints (5) 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 and consequently a CG approach is used to solve this problem.

3. The Proposed Algorithm

Now the proposed algorithm which uses the EAS as an improved procedure is presented in this section. In our approach, we have made three main modifications to the CG:

- (1) To produce a high quality initial solution, the modified sweep algorithm, swap, insert, and 2-opt move are used.
- (2) 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 . In the proposed algorithm, each (r, k) is solved by the modified elite ant system (MEAS) proposed by Yousefikhoshbakht et al. [19].
- (3) To improve the initial solution, the CG and several local search algorithms are applied.

In this section, first, the sweep, insert, swap, and 2-opt algorithms are explained and then the MEAS and the proposed CG are described in more detail.

3.1. The Proposed Modified Sweep Algorithm. In order to be applied, the proposed algorithm requires a high quality initial solution. Here, a modified sweep algorithm combined with insert, swap, and 2-opt moves is used to obtain initial solutions. The classic powerful sweep algorithm is relatively simple and was introduced by Gillett and Miller in 1974 [20]. It perhaps is one of the most greedy construction algorithms for the traveling salesman problem (TSP) and VRP. The sweeping clockwise or counterclockwise of this algorithm starts from customer i which has not been visited, from the unvisited customer with the smallest angle to the unvisited customer with the largest angle until all customers are included. In other words, this algorithm has the simple idea of inserting the unvisited customer with the nearest angle of the last inserted customer in the route. This method can produce a set of initial feasible open routes solution for HFFOVRP in order to use as an initial solution for other algorithms. This algorithm ranks and links demand points by their polar coordinate angle. To apply the proposed modified sweep algorithm (PMSA), we assume that the depot is node 0 and each customer has an (x, y) coordinate as its location. Therefore, polar coordinate angle for each customer i is calculated by formula (6). It should be noted that if $y(i) - y(0) < 0$ and $y(i) - y(0) \geq 0$ then $-\pi < An(i) < 0$ and $0 \leq An(i) \leq \pi$, respectively. Besides, if customer i and customer j have the same polar angle, we put a customer

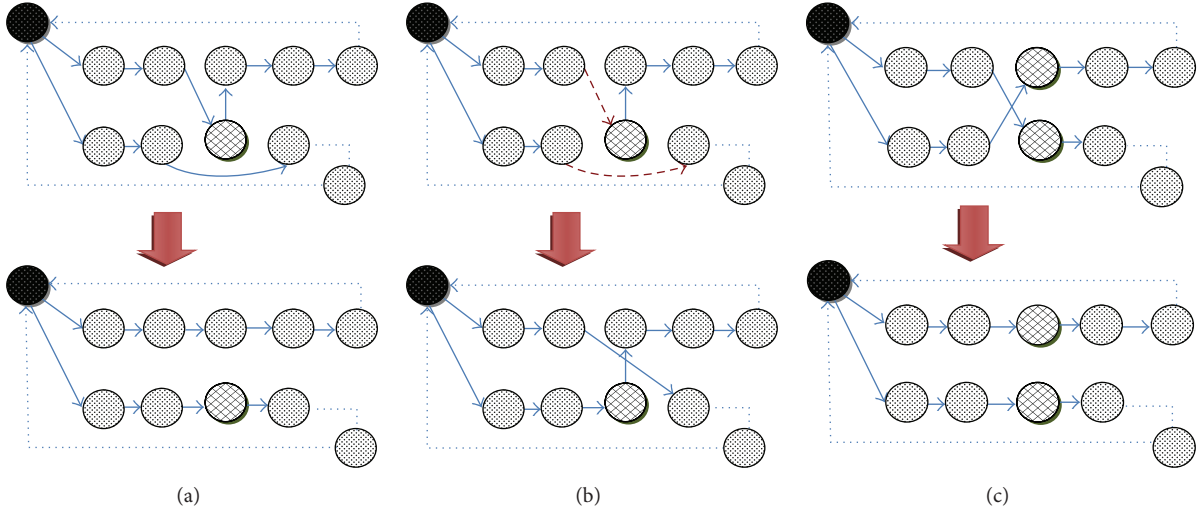


FIGURE 1: Insert (a), swap (b), and 2-opt moves (c).

which has a shorter distance with respect to the depot earlier in the list:

$$An(i) = \arctan \left\{ \frac{(y(i) - y(0))}{(x(i) - x(0))} \right\}. \quad (6)$$

In the proposed algorithm, first, the polar coordinates of all n customers are calculated and ordered as $An(1) \leq An(2) \leq \dots \leq An(n)$. Then, the route assigned to the chosen vehicle k starts with the unrouted customer farthest from the depot such as i . The next customer j to be inserted in the route will be the one who has not been served yet and who is the nearest polar coordinates to the customers of the route. If u_i^k is load carrying capacity of vehicle k in customer i , that customer j also has to be admissible in $u_i^k + p_j \leq Q^k$. This process is repeated until no customer is admissible in the current route. When this happens, a new vehicle is selected and the whole process is repeated until all the customers are routed. In the proposed algorithm we assume a vehicle with infinite capacity and fixed cost. If the modified sweep algorithm cannot produce a feasible solution, all the remaining customers are allocated to this artificial vehicle.

After producing a feasible or unfeasible solution, the insert and sweep algorithm is used to obtain a feasible solution with high quality. These neighborhood structures are an important key feature in the performance of most of effective algorithms because it determines the extent and the quality of the solution space explored. These moves are distinguished in terms of exchanges performed to convert one tour into another and they can be described as follows.

In insert exchange, a node from its position in one route is moved to another position in either the same or a different route. Consequently, while the initial tour is $(0, \dots, i, i+1, \dots, j-2, j-1, j, j+1, \dots, 0)$, the improved one is constructed as $(0, \dots, i, j, i+1, \dots, j-2, j-1, j+1, \dots, 0)$. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 1(a).

Besides, in the swap algorithm two nodes from different routes are exchanged [4]. Consequently, if it is supposed that the initial tour consists of the set of nodes $(0, \dots, i-1, i, i+1, \dots, j-1, j, j+1, \dots, 0)$, the improved one is constructed as $(0, \dots, i-1, j, i+1, \dots, j-1, i, j+1, \dots, 0)$. The same procedure is conducted in the case of multiple routes. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objective and constraints. This move is demonstrated in Figure 1(b). The most commonly encountered move is the 2-opt. In multiple routes, edges $(i, i+1)$ and $(j, j+1)$ belong to different routes but they form a criss-cross again. The 2-opt move is applied exactly in the same way as is the case in multiple routes. The move is allowed when it is considered favorable for the performance of the entire algorithm in terms of objectives and constraints. This move is demonstrated in Figure 1(c).

3.2. Column Generation Approach. Two problems are in CG algorithm including the Master Problem (MP) and the Pricing Subproblem (PSP). The MP is linear programming problem whose aim is to find the minimal cost and to produce the shadow prices of the temporary optimal solution to be used in the Pricing Subproblem. Since the number of columns of MP might be large, a Restricted MP is used to initiate the computations. The goal of the Pricing Subproblem is to generate additional columns for MP. In this section, the formulations of the Master Problem and Pricing Subproblem are presented.

3.2.1. Master Problem. The MP can be formulated as a set covering formulation. This formulation is a linear programming problem to which each customer is assigned at least one feasible route-vehicle pair since the arc costs satisfy the triangle inequality and each customer is visited exactly once to minimize the costs. In addition, in constraint (5), the integrality of variable x_r^k is eliminated:

$$(MP) \min \sum_{k \in K} \sum_{r \in R_k} c_r^k x_r^k \quad (7)$$

$$\text{Subject to } \sum_{k \in K} \sum_{r \in R_k} a_{ir}^k x_r^k \geq 1 \quad \forall i = 1, \dots, n \quad (8)$$

$$\sum_{r \in R_k} x_r^k \leq n_k \quad \forall k \in K \quad (9)$$

$$0 \leq x_r^k \leq 1 \quad \forall k \in K, r \in R_k. \quad (10)$$

Creating all feasible routes is considered as an 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 which have been generated by the proposed modified sweep algorithm. If the generated routes for vehicle type k are indicated by $R'_k \subset R_k$, then the RMP is created by replacing R_k with R'_k in the MP. 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 Subproblem. Each time the Pricing Subproblem 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 by modified elite ant system and vector $\pi^* = (\pi_{i_1}^*, \dots, \pi_{i_n}^*, \pi_1^*, \dots, \pi_m^*)$ is the values of the dual variables (shadow price) corresponding to constraints (8) and (9).

3.2.2. Pricing Subproblem. The Pricing Subproblem (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 Subproblem 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:

$$r = (i_0, i_1, \dots, i_H, i_{H+1}); \quad i_0 = 0, \quad i_{H+1} \neq 0. \quad (11)$$

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

$$\begin{aligned} \bar{c}_r^k &= c_r^k - \sum_{i \in I} \pi_i^* a_{ir}^k - \pi_k^* \\ &= \left(\sum_{h=1}^H c_{i_{h-1}, i_h}^k + f_k \right) - \sum_{h=1}^H \pi_{i_h}^* - \pi_k^* \\ &= \sum_{h=1}^{H+1} (c_{i_{h-1}, i_h}^k - \pi_{i_h}^*) + f_k, \end{aligned} \quad (12)$$

where $\pi_{i_0}^* = \pi_k^*$. Therefore, the reduced cost \bar{c}_r^k is equal to the sum of the fixed cost and the cost of route r on the directed subgraph $G_k = (V, A_k)$ for vehicle type k with arc modified cost defined as follows:

$$\bar{c}_{ij}^k = \begin{cases} c_{ij}^k - \pi_j^* - \pi_k^* & i = 0 \\ c_{ij}^k - \pi_j^* & i \neq 0. \end{cases} \quad (13)$$

Now, the PSP formulation can be solved by modifying the elite ant system which is one of the most important versions of the ant colony optimization (ACO). The ACO is used to solve the traveling salesman problem (TSP) in 1991 [19]. This algorithm simulates the natural ant behavior for food finding and applies it for solving the NP-hard problems for which no effective algorithm has been found yet. Studies on real ants show that, despite the fact that they do not enjoy a complete sense of vision, they can find the shortest path from the food sources to the nest by some evaporated material called pheromone [4]. This material is secreted by the ants when they move from one place to another to find the shortest path. The first version of ACO family was the ant system (AS) used for solving small scale TSP instances. Although it gained good solutions for small scale problems of TSP type, the AS algorithm failed to reach an acceptable efficiency in large scale problems of TSP in comparison with the famous algorithms of that time. Therefore, much effort was made to solve the problem efficiently [4]. The first modification applied on the AS algorithm was the usage of elitist strategy called EAS published in 1996. In EAS algorithm like the AS, the probability of movement from node i to node j which is not visited yet by ant " k " is presented as follows:

$$P_{ij}^k(t) = \begin{cases} \frac{\tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)}{\sum_{r \in J_i^k} \tau_{ij}^\alpha(t) \eta_{ij}^\beta(t)} & j \in J_i^k \\ 0 & j \notin J_i^k, \end{cases} \quad (14)$$

where J_i^k is the collection of nodes which have not been visited by ant " k " yet, $\tau_{ij}(t)$ is the value of pheromone on the arc joining i to j , $\eta_{ij}(t)$ is the heuristic information for the ant visibility measure (e.g., defined as the reciprocal of the distance between node i and node j for the TSP), and α, β are the controlling parameters by user which determine the ratio of importance of ant's visibility measure compared to the value of pheromone released on arc(i, j) [4].

This process further highlights the arcs belonging to the best route in any iteration and causes them to be updated according to the value of the best route L^{gb} . Note that the above operator indicates that the less the value of L^{gb} is, the more pheromone is released on the arcs.

Updating the pheromone simulates the changes in values of pheromone in any iteration and mainly it is one of the reasons that algorithms are different. Generally, two operations motivate this updating procedure in EAS algorithm [4]:

- (i) 1st, releasing new pheromone on the arcs, locally and globally. This operation leads to an increase in pheromone secretion on the arcs. Based on this algorithm, in addition to the local releasing of pheromone on the arcs through which the ants have passed, pheromone is released on the arcs belonging to the best route T^* and they are encouraged with the constant coefficient e in the following way [4]:

$$\Delta \tau_{ij}^{gb}(t) = \begin{cases} \frac{e}{L^{gb}(t)} & (i, j) \in T^* \\ 0 & (i, j) \notin T^*. \end{cases} \quad (15)$$

- (ii) 2nd, the evaporation of pheromone. At the end of each iteration of the algorithm, the value of pheromone left on the arcs is decreased by the constant coefficient $0 \leq \rho \leq 1$. Thus, the new footprint of pheromone has an average weight between the value of the pheromone left on the arcs, and the value of new pheromone released in the arcs. Thus the formula of updating pheromone in the EAS algorithm is

$$\tau_{ij}(t+1) = (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta\tau_{ij}^k(t) + \Delta\tau_{ij}^{gb}(t), \quad (16)$$

where $\Delta\tau_{ij}^k(t)$ is the formula of locally updating the pheromone [4]. Thus ants passing over the arc between nodes i and j release pheromone on the respective edge which is equal to the inverse of the cost of the tour ($L^k(t)$) taken by ants by formula (17). In this formula, T^k is the collection of arcs passed over by the ant “ k ”:

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)} & (i, j) \in T^k \\ 0 & (i, j) \notin T^k. \end{cases} \quad (17)$$

Therefore, the result of $PSP(k)$ will be one route for vehicle type k . If the best 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. After this step, the primal model shown in Section 3 is solved by an exact algorithm and the best solution is considered s . For this solution, the insert and swap algorithms are used to further improve the best solutions and then the new produced route-vehicle pairs are added to the set R'_k in RMP. In this level, the RMP will be solved again and shadow prices are obtained 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 proposed algorithm will be described as follows:

Pseudo Code of the Proposed Approach

- (1) Find initial sets R' of routes by sweep, insert, swap, and 2-opt algorithms for the MP.
- (2) Solve the MP by using Solver CPLEX 12.3 and obtain the shadow prices of the optimal solution.
- (3) For vehicle type k , produce the modified costs \bar{c}_{ij}^k , as (13).
- (4) For vehicle type k , solve $PSP(k)$ by using the MEAS and find feasible route-vehicle pairs with negative reduced costs and add them to the sets R'_k .
- (5) Solve the primal model in Section 3 by exact algorithm and produce s . Then, the insert and swap algorithms are used to further improve s .
- (6) Add route-vehicle pairs to the sets R'_k .
- (7) If the 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.

4. Computational Results

To evaluate the validity of our proposed SISEC algorithm compared to other algorithms including exact algorithm (EA), EAS, genetic algorithm (GA), and CG for the HFFOVRP, the performance of our algorithms was tested on a set of 22 benchmark instances derived from well-known Taillard's benchmarks in this section. These algorithms can be downloaded from the website [21]. The proposed algorithm was coded in AIMMS with solver GUROBI 4.5. One of the most advanced development environments for building optimization-based operations research applications and advanced planning systems is AIMMS. This commercial linear programming software could find optimal solutions for the small scale of the problems, like the HFFOVRP, and hence can be used to evaluate the accuracy of the proposed model. Based on this information, AIMMS obtained the optimal solution only for a few instances and in the other instance automatically terminated before reaching to optimal solution. Also in the large problems, this algorithm failed to obtain a feasible solution.

Besides, all the experiments were implemented on a PC with Pentium 4 at 3.5 GHZ and 8 GB RAM running Windows 7 Home Basic Operating system. First, the benchmark problems are described and then some numerical results of comparison between the proposed algorithm and some algorithms are presented on a set of HFFOVRP benchmark problems. Table 1 shows these problems are built of 10–100 nodes including the depot which all randomly located over a square with no service time. Euclidean distances are used in all the problems. Besides, they have a fixed fleet with capacity restrictions and with no route length restrictions. The proposed compound heuristic algorithm is compared to mentioned algorithms in the Table 2. The information of this table includes the number of customers (n), the best solution costs of EA, EAS, GA, CG, and SISEC, and their CPU time, respectively, the best known solutions (BKSs), and gap between the best known solution (BKS) and our hybrid algorithm. The gap is a simple criterion to measure the efficiency of an algorithm and to compute the relative percentage deviation of its solution from the BKSs on specific benchmark. These values are calculated by the following formula. A zero gap indicates that best known solutions should be obtained:

$$\text{Gap} = 100 \times \frac{(\text{value of the SISEC} - \text{Value of the BKS})}{\text{Value of the BKS}}. \quad (18)$$

It should be noted that while EA, CG, and SISEC have been run until the software stops automatically, the stop condition of EAS and GA is the best know solution iterated ten times. The results of Table 2 show that the EA fails to find optimal solutions for most of the problems especially

TABLE 1: Data for problems.

Instance	Number of customers	Number of different type vehicles	Capacity	Fixed cost	Variable cost	Available number of vehicles of kind k th
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
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

TABLE I: Continued.

Instance	Number of customers	Number of different type vehicles	Capacity	Fixed cost	Variable cost	Available number of vehicles of kind k th
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
20	90	1	20	10	1	5
		2	40	25	1.2	5
		3	80	45	1.5	2
		4	100	80	2.0	2
21	95	1	20	10	1	5
		2	30	35	1.3	5
		3	70	55	1.7	3
		4	100	85	1.9	2
		5	150	135	2.3	2
22	100	1	100	500	1	4
		2	150	1200	1.4	4
		3	300	1900	1.7	5

TABLE 2: Comparison results of the SISEC with some heuristic and metaheuristic algorithms.

Instance	n	EA	Time (Sec)	EAS	Time (Sec)	GA	Time (Sec)	CG	Time (Sec)	SISEC	Time (Sec)	BKS	Gap
1	10	191.10	1.09	191.10	0.55	191.10	1.55	191.10	0.36	191.10	0.21	191.10	0
2	15	282	171.02	282	3.61	282	3.69	282	4.28	282	2.34	282	0
3	20	379.63	16.55	379.63	3.12	379.63	4.62	379.63	5.44	379.63	3.51	379.63	0
4	25	437.79	22.78	439.12	15.53	441.56	16.73	437.79	9.78	437.79	7.54	437.79	0
5	30	472.76	61.78	473.31	39.65	475.61	45.87	473.31	16.46	471.81	10.26	472.76	-0.20
6	35	346.26	55567.75	349.95	56.31	351.73	63.81	349.95	370.08	345.81	165.45	346.26	-0.13
7	40	NA	—	607.99	61.68	611.67	72.39	600.99	767.42	589.41	457.94	600.99	-1.93
8	45	NA	—	676.04	62.64	687.46	77.67	676.04	2408.40	671.91	1157.81	676.04	-0.61
9	50	NA	—	915.63	67.27	924.73	76.72	907.3	256.72	899.51	98.64	907.3	-0.86
10	50	NA	—	537.18	69.76	537.18	85.62	507.58	2253.26	451.21	1784.54	507.58	-11.11
11	50	NA	—	829.24	64.31	829.24	86.11	826.19	840.18	798.61	621.39	826.19	-3.34
12	50	NA	—	952.41	64.51	963.61	89.12	947.81	698.96	924.18	584.74	947.81	-2.50
13	55	NA	—	1086.65	65.98	1084.16	88.52	1074.91	1940.82	998.81	1254.93	1074.91	-7.08
14	60	NA	—	1984.09	73.71	1984.09	98.62	1937.03	8807.25	1842.81	5694.87	1937.03	-4.86
15	65	NA	—	1598.23	78.61	1599.18	106.67	1563.33	4203.14	1498.28	3549.14	1563.33	-4.16
16	70	NA	—	974.27	76.73	983.52	156.27	962.57	6357.70	925.48	3614.87	962.57	-3.85
17	75	NA	—	1399.17	87.72	1421.60	154.72	1356.67	6021.97	1326.36	4987.31	1356.67	-2.23
18	80	NA	—	1321.44	94.93	1387.29	167.13	1285.74	8331.64	1248.21	5927.98	1285.74	-2.92
19	85	NA	—	1295.58	92.72	1341.52	189.32	1295.58	13731.35	1254.33	8642.29	1295.58	-3.18
20	90	NA	—	1685.51	98.83	1783.29	223.69	1645.21	62748.94	1512.84	14621.62	1645.21	-8.05
21	95	NA	—	1998.62	107.72	2210.27	289.73	1951.45	124548.98	1789.34	36124.87	1951.45	-8.31
22	100	NA	—	2198.56	112.82	2287.39	332.27	2154.36	195648.84	2054.11	32548.18	2154.36	-4.65

instances with more than 35 customers and it cannot be used as an efficient and practical algorithm. Furthermore, the proposed algorithm has shown to be competitive with the EA and CG in terms of solution quality and CPU time. Moreover, the proposed algorithm can obtain better solution than EAS and GA in 19 out of 22 instances. During this experiment, some solutions are as close as possible to the BKS so far. Generally, the hybrid algorithm performs better, with solutions averaging -3.18% better than BKSs until now. In comparison to the BKSs, SISEC can not only produce BKSs for the four instances including instances 1, 2, 3, and 4 but obtain new BKS for 18 instances as well.

In more detail, the results indicate that SISEC is a competitive approach compared to the EA. For instances 1, 2, 3, and 4, the gap between SISEC and BKS is zero. However, in other instances, the proposed algorithm finds better solution than the EA. Moreover, computational results of the proposed algorithm and CG show that these algorithms have a close competition for three instances including 1, 2, and 3. In addition, the proposed algorithm produces better 19 solutions than CG. In more detail, the SISEC has better solutions with more than 5% of gap in 10, 13, 20, and 21. Moreover, the proposed algorithm obtains nearly the solutions of CG for 4 instances including 5, 6, 8, and 9; that is, the gap is below 1% and for other instances the gap is between 1% and 5%. For overall, the performance of the proposed algorithm is better in reaching the suboptimal solution than the CG.

Figure 2 presents the comparison of CPU time between SISEC and CG algorithms for the 22 instances. In this figure, the horizontal axis shows the name of instances and the

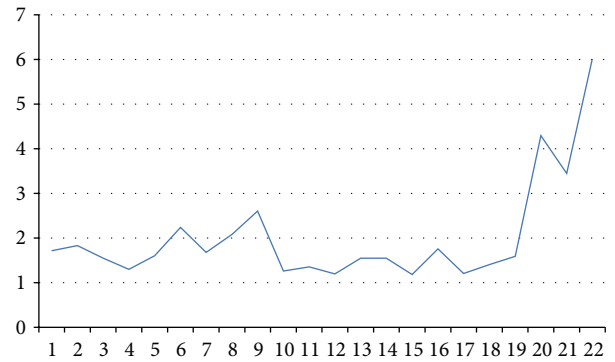


FIGURE 2: The ratio of SISEC CPU time to CG CPU time.

vertical axis indicates the SISEC CPU time divided by CG CPU time.

In Table 3, columns 1 and 2 show the names of instances and the number of customers (n), respectively. Furthermore, columns 3, 4, and 5 contain the EA, CG, and SISEC on the same CPU time which is shown in column 6. Finally, the BKSs for all instances are shown in column 7. From the comparison between exact algorithm and SISEC on five instances including 1, 2, 3, 4, and 5, it can be seen that the proposed algorithm in 4 instances has been able to find better solutions than the exact algorithm. In more detail, the exact algorithm has produced equal solutions only in instance 1. Furthermore, the CG has had a weak performance in general and has not been able to produce better solutions in all instances compared to SISEC. In other words, the CG has obtained

TABLE 3: Comparison results between algorithms in the same run time.

Instance	n	CPLEX 12.3	CG	SISEC	CPU time	BKS
1	10	191.10	191.10	191.10	0.21	191.10
2	15	525.56	282	282	2.34	282
3	20	390.25	394.21	379.63	3.51	379.63
4	25	532.18	498.57	437.79	7.54	437.79
5	30	497.48	497.81	471.81	10.26	472.76
6	35	NA	389.61	345.81	165.45	346.26
7	40	NA	654.18	589.41	457.94	600.99
8	45	NA	671.91	671.91	1157.81	676.04
9	50	NA	958.61	899.51	98.64	907.3
10	50	NA	689.64	451.21	1784.54	507.58
11	50	NA	998.17	798.61	621.39	826.19
12	50	NA	1054.98	924.18	584.74	947.81
13	55	NA	1254.82	998.81	1254.93	1074.91
14	60	NA	2651.84	1642.81	5694.87	1937.03
15	65	NA	2695.39	1498.28	3549.14	1563.33
16	70	NA	1251.84	925.48	3614.87	962.57
17	75	NA	1692.84	1326.36	4987.31	1356.67
18	80	NA	1781.28	1248.21	5927.98	1285.74
19	85	NA	1423.99	1254.33	8642.29	1295.58
20	90	NA	2259.68	1512.84	14621.62	1645.21
21	95	NA	2533.11	1789.34	36124.87	1951.45
22	100	NA	3295.74	2054.11	32548.18	2154.36

equal solutions only on instances 1, 2, and 8. Therefore, the SISEC has been able to escape local optimum points and has caused a satisfactory improvement in the performance of the algorithm.

Figure 3 shows the comparison of our algorithm and CPLEX 12.3 (EA) and CG for five instances including 1, 2, 3, 4, and 5. In this figure, the horizontal axis shows the name of instances and the vertical axis indicates the results of three mentioned algorithms. This figure shows that proposed SISEC clearly yields better solutions than the other algorithms.

5. Conclusions

This paper investigates the HFFOVRP in the transportation system and presents a new combined heuristic algorithm based on CG called SISEC which has allowed us to improve the quality of several of the solutions obtained by EA, EAS, GA, and classical CG. Computational results generally have shown that the proposed algorithm yields better results compared to the exact and CG algorithms in terms of both the solution quality and the CPU time. It seems that using metaheuristic algorithms like ant colony optimization can lead to gaining better initial solutions. Furthermore, powerful metaheuristic algorithms 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.

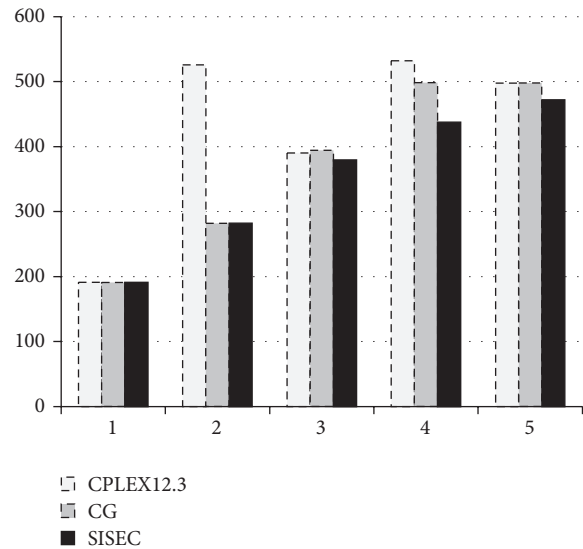


FIGURE 3: Comparison between exact, CG, and SISEC algorithms for 5 instances.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] M. Sedighpour, V. Ahmadi, M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, "Solving the open vehicle routing problem by

- a hybrid ant colony optimization,” *Kuwait Journal of Science*, vol. 41, no. 3, pp. 139–162, 2014.
- [2] S. A. Mirhassani and N. Abolghasemi, “A particle swarm optimization algorithm for open vehicle routing problem,” *Expert Systems with Applications*, vol. 38, no. 9, pp. 11547–11551, 2011.
- [3] L. Sun and B. Wang, “An inverse robust optimisation approach for a class of vehicle routing problems under uncertainty,” *Discrete Dynamics in Nature and Society*, vol. 2016, Article ID 2804525, 12 pages, 2016.
- [4] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, “A combination of modified tabu search and elite ant system to solve the vehicle routing problem with simultaneous pickup and delivery,” *Journal of Industrial and Production Engineering*, vol. 31, no. 2, pp. 65–75, 2014.
- [5] R. M. Chen and Y. M. Shen, “Novel encoding and routing balance insertion based particle swarm optimization with application to optimal CVRP depot location determination,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 743507, 11 pages, 2015.
- [6] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, “A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem,” *European Journal of Operational Research*, vol. 152, no. 1, pp. 148–158, 2004.
- [7] W. Liu and X. Li, “A problem-reduction evolutionary algorithm for solving the capacitated vehicle routing problem,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 165476, 11 pages, 2015.
- [8] P. Kabcome and T. Mouktonglang, “Vehicle routing problem for multiple product types, compartments, and trips with soft time windows,” *International Journal of Mathematics and Mathematical Sciences*, vol. 2015, Article ID 126754, 9 pages, 2015.
- [9] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, “Solving the heterogeneous fixed fleet open vehicle routing problem by a combined metaheuristic algorithm,” *International Journal of Production Research*, vol. 52, no. 9, pp. 2565–2575, 2014.
- [10] M. Yousefikhoshbakht, F. Didehvar, and F. Rahmati, “A mixed integer programming formulation for the heterogeneous fixed fleet open vehicle routing problem,” *Journal of Optimization in Industrial Engineering*, vol. 8, no. 18, pp. 37–46, 2015.
- [11] A. Subramanian, L. M. A. Drummond, C. Bentes, L. S. Ochi, and R. Farias, “A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery,” *Computers and Operations Research*, vol. 37, no. 11, pp. 1899–1911, 2010.
- [12] X. Li, S. C. H. Leung, and P. Tian, “A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 365–374, 2012.
- [13] É. D. Taillard, L. M. Gambardella, M. Gendreau, and J.-Y. Potvin, “Adaptive memory programming: a unified view of metaheuristics,” *European Journal of Operational Research*, vol. 135, no. 1, pp. 1–16, 2001.
- [14] P. Huachi, V. Penna, A. Subramanian, and L. S. Ochi, “An iterated local search heuristic for open vehicle routing problems,” in *Congresso Latino-Iberoamericano de Investigación Operativa*, Rio de Janeiro, Brazil, September 2012.
- [15] P. H. V. Penna, A. Subramanian, and L. S. Ochi, “An iterated local search heuristic for the heterogeneous fleet vehicle routing problem,” *Journal of Heuristics*, vol. 19, no. 2, pp. 201–232, 2013.
- [16] C. D. Tarantilis and C. T. Kiranoudis, “BoneRoute: an adaptive memory-based method for effective fleet management,” *Annals of Operations Research*, vol. 115, no. 1–4, pp. 227–241, 2002.
- [17] C. D. Tarantilis and C. T. Kiranoudis, “A flexible adaptive memory-based algorithm for real-life transportation operations: two case studies from dairy and construction sector,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 806–822, 2007.
- [18] J. Brandão, “A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem,” *European Journal of Operational Research*, vol. 195, no. 3, pp. 716–728, 2009.
- [19] M. YousefiKhoshbakht, M. Sedighpour, and E. Mahmoodabadi, “A modified elite ACO based avoiding premature convergence for traveling salesmen problem,” *Journal of Industrial Engineering International*, vol. 7, no. 15, pp. 68–75, 2011.
- [20] B. E. Gillett and L. R. Miller, “A heuristic algorithm for the vehicle dispatch problem,” *Operations Research*, vol. 22, no. 2, pp. 340–349, 1974.
- [21] <http://mistic.heig-vd.ch/taillard/problemes.dir/vrp.dir/vrp.html>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

