CIRRELT
Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport
Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation

A Hybrid Evolutionary Algorithm for Heterogeneous Fleet Vehicle Routing Problems with Time Windows

Çağri Koç
Tolga Bektaş
Ola Jabali
Gilbert Laporte

March 2014

CIRRELT-2014-16

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Hybrid Evolutionary Algorithm for Heterogeneous Fleet Vehicle Routing Problems with Time Windows

Çağri Koç[1,2,*], Tolga Bektaş[2], Ola Jabali[1,3], Gilbert Laporte[1,4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] School of Management,Centre for Operational Research, Management Science and Information systems (CORMSIS), University of Southampton, Southampton, SO17 1BJ, United Kingdom

[3] Department of Logistics and Operations Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[4] Department of Management Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

Abstract. This paper presents a hybrid evolutionary algorithm (HEA) to solve heterogeneous fleet vehicle routing problems with time windows. There are two main types of such problems, namely the Fleet Size and Mix Vehicle Routing Problem with Time Windows (F) and the Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows (H), where the latter, in contrast to the former, assumes a limited availability of vehicles. The main objective is to minimize the fixed vehicle cost and the distribution cost, where the latter can be defined with respect to en-route time (T) or distance (D). The proposed algorithm is able to solve the four variants of heterogeneous fleet routing problem, called FT, FD, HT and HD, where the last variant is new. The HEA successfully combines several metaheuristics and offers a number of new advanced efficient procedures tailored to handle the heterogeneous fleet dimension. Extensive computational experiments on benchmark instances have shown that the HEA is highly effective on FT, FD and HT. In particular, 149 of the 360 best known solution values for the three previously studied variants have been retrieved or improved within reasonable computational times. New benchmark results on HD are also presented.

Keywords: Vehicle routing, time windows, heterogeneous fleet, genetic algorithm, neighborhood search.

Acknowledgements. The authors gratefully acknowledge funding provided by the Southampton Management School at the University of Southampton and by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grants 39682-10 and 436014-2013.

---

* Corresponding author: Cagri.Koc@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2014

1. Introduction

In heterogeneous fleet vehicle routing problems with time windows, one considers a fleet of vehicles with various capacities and vehicle-related costs, as well as a set of customers with known demands and time windows. These problems consist of determining a set of vehicle routes such that each customer is visited exactly once by a vehicle within prespecified time window, all vehicles start and end their routes at a depot, and the load of each vehicle does not exceed its capacity. There are two main categories of such problems, namely the Fleet Size and Mix Vehicle Routing Problem with Time Windows (F) and the Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows (H). In category F, there is no limit in the number of available vehicles of each type, whereas such a limit exists in category H. Two measures are used to compute the total cost to be minimized. The first is based on the *en-route time* (T), which is the sum of the fixed vehicle cost and the trip duration without the service time. In this case, service times are only used to check feasibility and for performing adjustments to the departure time from the depot in order to minimize pre-service waiting times. The second cost measure is based on *distance* (D) and consists of the fixed vehicle cost and the distance traveled by the vehicle, as is the case in the standard vehicle routing problem with time windows (VRPTW) (Solomon 1987).

We differentiate between four variants defined with respect to the problem category and to the way in which the objective function is defined, namely FT, FD, HT and HD. The first variant is FT, described by Liu and Shen (1999b) and the second is FD, introduced by Bräysy et al. (2008). The third variant HT was defined and solved by Paraskevopoulos et al. (2008). Finally, HD is a new variant which we introduce in this paper.

Several versions of these problems without time windows arise in distribution problems such as grocery store deliveries and mail collection, and have, to a large extent, been addressed in the vehicle routing literature. Hoff et al. (2010) and Belfiore and Yoshizaki (2009) describe several industrial aspects and practical applications of these problems. The most studied versions are the fleet size and mix vehicle routing problem, described by Golden et al. (1984), which considers an unlimited heterogeneous fleet, and the heterogeneous fixed fleet vehicle routing problem, proposed by Taillard (1999). For further details, the reader is referred to the surveys of Baldacci et al. (2008) and of Baldacci and Mingozzi (2009).

The FT variant has several extensions, e.g., multiple depots (Dondo et al. 2007, Bettinelli et al. 2011), overloads (Kritikos and Ioannou 2013), and split deliveries (Belfiore and Yoshizaki 2009, 2013). There exist several exact algorithms for the capacitated vehicle routing problem (VRP) (Toth and Vigo 2002, Baldacci et al. 2010), and for the heterogeneous VRP (Baldacci and Mingozzi 2009). However, to the best of our knowledge, no exact algorithm has been proposed for the heterogeneous VRP with time windows, i.e., FT, FD and HT. The existing heuristic algorithms for these three variants are briefly described below.

Liu and Shen (1999b) proposed a heuristic for FT which starts by determining an initial solution through an adaptation of the Clarke and Wright (1964) savings algorithm, previously presented by Golden et al. (1984). The second stage improves the initial solution by moving customers by means of parallel insertions. The algorithm was tested on a set of 168 benchmark instances derived from the set of Solomon (1987) for the VRPTW. Dullaert et al. (2002) described a sequential construction algorithm for FT, which is an extension of the insertion heuristic of Golden et al. (1984). Dell'Amico et al. (2007) described a multi-start parallel regret construction heuristic for FT, which is embedded into a ruin and recreate metaheuristic. Bräysy et al. (2008) presented a deterministic annealing metaheuristic for FT and FD. In a later study, Bräysy et al. (2009) described a hybrid metaheuristic algorithm for large scale FD instances. Their algorithm combines the well-known threshold acceptance heuristic with a guided local search metaheuristic having several search limitation strategies. An adaptive memory programming algorithm was proposed by Repoussis and Tarantilis (2010) for FT, which combines a probabilistic semi-parallel construction heuristic, a reconstruction mechanism and a tabu search algorithm. Computational results indicate that their method is highly successful and improves many best known solutions. In a recent study, Vidal et al. (2014) introduced a genetic algorithm based on a unified solution framework for different variants of the VRPs, including FT and FD. To our knowledge, Paraskevopoulos et al. (2008) are the only authors who have studied HT. Their two-phase solution methodology is based on a hybridized tabu search algorithm capable of solving both FT and HT.

This brief review shows that the two problem categories F and H have already been solved independently through different methodologies. We believe there exists merit for the development of a unified algorithm capable of efficiently solving the two problem categories. This is the main motivation behind this paper.

This paper makes three main scientific contributions. First, we develop a unified hybrid evolutionary algorithm (HEA) capable of handling the four variants of the problem. The HEA builds on a state-of-the art metaheuristic of Vidal et al. (2012, 2014) which combines population based search with local search procedures for education and has proved highly successful for several variants of the VRP. The second contribution is the introduction of several algorithmic improvements to the procedures developed by Prins (2009) and Vidal et al. (2012). Namely, we use Adaptive Large Neighborhood Search (ALNS) equipped with a range of operators as the main EDUCATION procedure within the search. We also propose an advanced version of the SPLIT algorithm of Prins (2009) capable of handling infeasibilities. Finally, we introduce an innovative aggressive INTENSIFICATION procedure on elite solutions, as well as a new diversification scheme through the REGENERATION and the MUTATION procedures of solutions. The third contribution is to introduce HD as a new problem variant.

The remainder of this paper is structured as follows. Section 2 presents a detailed description of the HEA. Computational experiments are presented in Section 3, and conclusions are provided in Section 4.

2. Description of the Hybrid Evolutionary Algorithm

We start by introducing the notation related to FT, FD, HT and HD. All problems are defined on a complete graph $G=(N,A)$, where $N=\{0,\dots,n\}$ is the set of nodes, and node 0 corresponds to the depot. Let $A=\{(i,\ j):0\leq i,\ j\leq n, i\neq j\}$ denote the set of arcs. The distance from i to j is denoted by d_{ij}. The customer set is $N\backslash\{0\}$ in which each customer i has a demand q_i and a service time s_i, which must start within time window $[a_i,\ b_i]$. If a vehicle arrives at customer i before a_i, it then waits until a_i. Let $K=\{1,\dots,k\}$ be the set of available vehicle types. Let e_k and Q_k denote the fixed vehicle cost and the capacity of vehicle type k, respectively. The travel cost from i to j associated with a vehicle of type k is c_{ij}^k. The travel time from i to j is denoted by t_{ij} and is independent of the vehicle type. In HT and HD, the available number of vehicles of type $k\in K$ is n_k, whereas no such constant applies to FT and FD.

The remainder of this section introduces the main components of the HEA. A general overview of the HEA is given in Section 2.1. More specifically, Section 2.2 presents the initialization of the population. The selection of parent solutions, the ordered crossover operator and the advanced algorithm SPLIT are described in Sections 2.3, 2.4 and 2.5, respectively. Section 2.6 presents the offspring EDUCATION procedure and Section 2.7 presents the INTENSIFICATION procedure. The survivor selection mechanism is detailed in Section 2.8. Finally, the diversification stage, including the REGENERATION and MUTATION procedures, is described in Section 2.9.

2.1. Overview of the Hybrid Evolutionary Algorithm

The general structure of the HEA is presented in Algorithm 1. The modified version of the classical Clarke and Wright savings algorithm and the ALNS operators are combined to generate the initial population (Line 1). Two parents are selected (Line 3) through a binary tournament, following which the crossover operation (Line 4) generates a new offspring C. The advanced SPLIT algorithm is applied to the offspring C (Line 5), which optimally segments the giant tour by choosing the vehicle type for each route. The EDUCATION procedure (Line 6) uses the ALNS operators to educate offspring C and inserts it back into the population. If C is infeasible, the EDUCATION procedure is iteratively applied until a modified version of C is feasible, which is then inserted into the population.

The probabilities associated with the operators used in the EDUCATION procedure and the penalty parameters are updated by means of an adaptive weight adjustment procedure (AWAP) (Line 7). Elite solutions are put through an aggressive INTENSIFICATION procedure, based on the ALNS algorithm (Line 8) in order to improve their quality. The population size, shown by n_a, changes during the algorithm as new offsprings are added, but is limited by $n_p + n_o$, where n_p is a constant denoting the size of the population initialized at the beginning of the algorithm and n_o is a constant showing the maximum allowable number of offsprings that can be inserted into the population. If, at any iteration, the populations size n_a reaches $n_p + n_o$, then a survivor selection mechanism is applied (Line 9). At each iteration of the algorithm, MUTATION is applied to a randomly selected individual from the population with probability p_m. If there are no improvements in the best known solution for a number of consecutive iterations it_r, the entire population undergoes a REGENERATION (Line 10). The HEA terminates when the number of iterations without improvement it_t is reached (Line 11).

Algorithm 1 The general framework of the HEA

1: *Initialization*: initialize a population with size n_p
2: **while** number of iterations without improvement $< it_t$ **do**
3:     *Parent selection*: select parent solutions P_1 and P_2
4:     *Crossover*: generate offspring C from P_1 and P_2
5:     SPLIT: partition C into routes
6:     EDUCATION: educate C with ALNS and insert into population
7:     *AWAP*: update probabilities of the ALNS operators
8:     INTENSIFICATION: intensify elite solution with ALNS
9:     *Survivor selection*: if the population size n_a reaches $n_p + n_o$, then select survivors
10:     *Diversification*: diversify the population with MUTATION or REGENERATION procedures
11: **end while**
12: Return best feasible solution

2.2. Initialization

The procedure used to generate the initial population is based on a modified version of the Clarke and Wright and ALNS algorithms. An initial individual solution is obtained by applying Clarke and Wright algorithm and by selecting the largest vehicle type for each route. Then, until the initial population size reaches n_p, new individuals are created by applying to the initial solution operators based on random removals and greedy insertions with a noise function (see Section 2.6). We have selected these two operators in order to diversify the initial population. In the destroy phase, a removal operator is used to remove a subset of nodes from the solution. The number of nodes removed is randomly chosen from the initialization interval $[b_l^i, b_u^i]$, which is defined by a lower and an upper bound calculated as a percentage of the total number of nodes in an instance. In a subsequent repair phase, an insertion operator is applied to the destroyed solution.

2.3. Parent Selection

In evolutionary algorithms, the evaluation function of individuals is often based on the solution cost. However, this kind of evaluation, does not take into account other important factors such as the diversity of the population, which plays a critical role in evolutionary algorithms. Vidal et al. (2012) proposed a new method to tackle this issue. The first step of this method is an extended evaluation function, named *biased fitness* $bf(C)$, which considers the cost of an individual C, as well as its *diversity contribution* $dc(C)$ to the population. This function is an adaptive mechanism which is continuously updated and is used to measure the quality of individuals during selection phases. The second part of this function is defined as

$$dc(C) = \frac{1}{n_c} \sum_{C_2 \in N_c} \beta(C, C_2), \tag{1}$$

where N_c is the set of the n_c closest neighbours of C in the population. Thus, $dc(C)$ calculates the average distance between C and its neighbours in N_c. The distance between two parents $\beta(C, C_2)$ is the number of pairs of adjacent requests in C which are no longer adjacent, (called broken), in C_2. For example, let $C = \{4,5,6,7,8,9,10\}$ and $C_2 = \{10,7,8,9,5,6,4\}$, in C_2 the pairs $\{4,5\}$, $\{6,7\}$ and $\{9,10\}$ are broken and $\beta(C, C_2) = 3$. The algorithm selects the broken pairs distance (see Prins 2009) to compute the distance β. The main idea behind $dc(C)$ is to assess the differences between individuals.

The evaluation function of an individual C in a population is

$$bf(C) = r_c(C) + (1 - \frac{n_e}{n_a}) r_{dc}(C), \tag{2}$$

which is based on the rank $r_c(C)$ of solution cost, and on the rank $r_{dc}(C)$ of the *diversity contribution*. In (2), n_e is the number of elite individuals and n_a is the current number of individuals.

The HEA selects two parents through a binary tournament to yield an offspring C. The selection process randomly chooses two individuals from the population and keeps the one having the best biased fitness.

2.4. Crossover

Following the parent selection phase, two parents undergo the classical Ordered Crossover or OX without trip delimiters. The OX operator is well suited for cyclic permutations, and the giant tour encoding allows recycling crossovers designed for the traveling salesman problem (TSP) (see

Prins 2004, 2009). Initially, two positions i and j are randomly selected in the first parent P_1. Subsequently, the substring $(i, ..., j)$ is copied into the first offspring O_1, at the same positions. The second parent P_2 is then swept cyclically from position $j+1$ onwards to fill the empty positions in O_1. While avoiding repetitions, in order to complete a circuit over all nodes. The second offspring O_2 is generated likewise by exchanging the roles of P_1 and P_2. In the original version of OX, two offsprings are obtained. However in the HEA, we only randomly select one offspring.

2.5. Split Algorithm

This algorithm is a tour splitting procedure which optimally partitions a solution into feasible routes. Each solution is a permutation of customers without trip delimiters and can therefore be viewed as a giant TSP tour for a vehicle with a large enough capacity. This algorithm was successfully applied in evolutionary based algorithms for several routing problems (Prins 2004, 2009, Vidal et al. 2012, 2013).

We propose an advanced tour splitting procedure, denoted by Split, which is embedded in the HEA to segment a giant tour and to determine the fleet mix composition. This is achieved through a controlled exploration of infeasible solutions (see Cordeau et al. 2001 and Nagata et al. 2010), by relaxing the limits on time windows and vehicle capacities. Violations of these limits are penalized through an objective function containing extra terms to acccount for infeasibilities. This is in contrast to Prins (2009) who does not allow infeasibilities, and in turn solves a resource-constrained shortest path problem using dynamic programming to determine the best fleet mix on a given solution. Our implementation also differs from those of Vidal et al. (2013) since it allows for infeasibilities that are not just related to time windows or load, but also to the fleet size in the case of HT and HD. When this algorithm is applied to FT and FD, there always exists a feasible assignment of vehicles to routes. However, in the case of HT and HD there may not be enough vehicles of a given type and infeasibilities can therefore occur.

We now describe the Split algorithm. Let $\Re$ be the set of all routes in individual C, and let R be a route. While formally R is a vector, for convenience we denote the number of its components by $|R|$. Therefore, $R = (i_0 = 0, i_1, i_2, \ldots, i_{|R|-1}, i_{|R|} = 0)$, we also write $i \in R$ if i is a component of R, and $(i, j) \in R$ if i and j appear in succession in R. Let z_t be the arrival time at the t^{th} customer in R, thus the time window violation of route R is $\sum_{t=1}^{|R|-1} \max\{z_t - b_{i_t}, 0\}$. The total load for route R is $\sum_{t=1}^{|R|-1} q_{i_t}$, and we consider solutions with a total load not exceeding twice the capacity of the largest vehicle given by Q_{max} (Vidal et al. 2013). Furthermore, for route R and for each vehicle type k we compute $y(k)$, which is the number of vehicles of type k used in the solution.

Let λ_t, λ_l and λ_f represent the penalty values for any violations of the time windows, the vehicle capacity and the fleet size, respectively. The variable x_{ij}^k is equal to 1 if customer i immediately precedes customer j visited by vehicle k. For each route $R \in \Re$ traversed by vehicle $k \in K$, the total cost including penalties is

$$\nu(R,k) = \sum_{(i,j)\in R} c_{ij}^k x_{ij}^k + e_k + \lambda_t \sum_{t=1}^{|R|-1} \max\{z_t - b_{i_t}, 0\} + \lambda_l \max\{\sum_{t=1}^{|R|-1} q_{i_t} - Q_{max}, 0\}. \tag{3}$$

Thus, the total cost including penalty cost of individual C is

$$\Delta(C) = \sum_{R\in\Re} \nu(R,k) + \lambda_f \sum_{k\in K} \max\{0, y(k) - n_k\}. \tag{4}$$

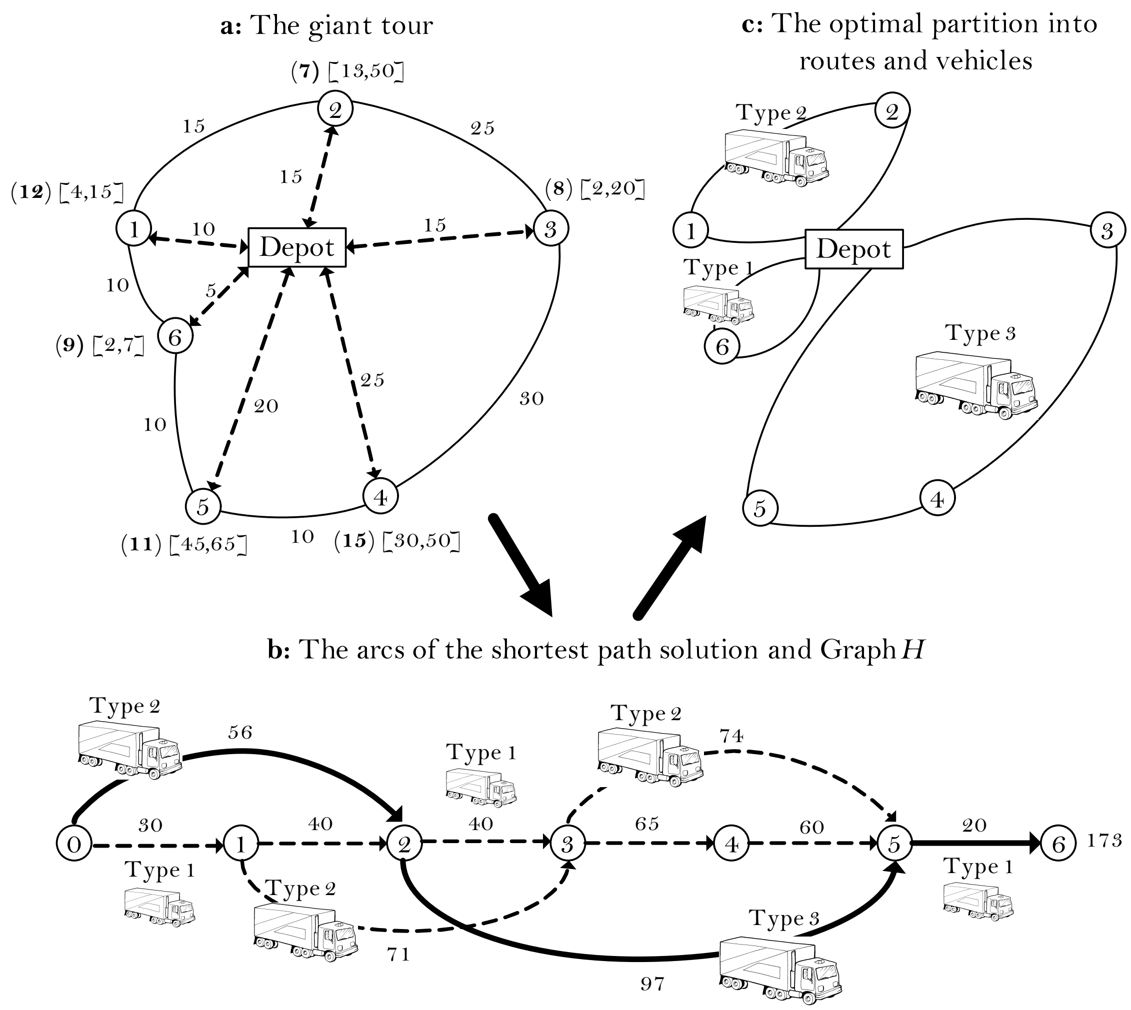

Figure 1 **Illustration of procedure** Split

Figure 1 shows the steps of this advanced procedure using on an FD instance. The arc costs, demands and time windows are given in Figure 1a. In particular, the number in bold within the parentheses associated with each node is the demand for that customer; the two numbers within brackets define the time window. Service times are identical and equal to 4 for each customer, and three different types of vehicles are available. The capacity q_k and fixed cost e_k of vehicles of type {1,2,3} are $q_1 = 10$, $q_2 = 20$, $q_3 = 30$ and $e_1 = 6$, $e_2 = 8$, $e_3 = 10$, respectively. The algorithm starts with a giant TSP tour which includes six customers and uses one vehicle with unlimited capacity. The SPLIT algorithm computes an optimal compound segmentation in three routes corresponding to three sequences of customers {1,2}, {3,4,5} and {6} with three vehicle choices, Type 2, Type 3 and Type 1, respectively, as shown in Figure 1b. The resulting solution is shown in Figure 1c. An optimal partitioning of the giant tour into routes for offspring C corresponds to a minimum-cost path.

The penalty parameters of the SPLIT algorithm are initially set to an initial value and are dynamically adjusted during the algorithm. If an individual is still infeasible after the first EDUCATION procedure, then the penalty parameters are multiplied by λ_m and the EDUCATION procedure restarts. When this solution becomes feasible, the parameters are reset to their initial values.

2.6. EDUCATION

The EDUCATION procedure is systematically applied to each offspring in order to improve its quality. The ALNS algorithm is used as a way of educating the solutions in the HEA. This is achieved by applying both the destroy and repair operators, and a number of removable nodes are modified in each iteration. The operators used within the HEA are either adapted or inspired from those employed by various authors (Ropke and Pisinger 2006a,b, Pisinger and Ropke 2007, Demir et al. 2012, Paraskevopoulos et al. 2008). The EDUCATION procedure is detailed in Algorithm 2.

The removal procedure (Line 4 of Algorithm 2) runs for n' iterations and removes n' customers from the solution, where n' is in the interval of removable nodes $[b_l^e, b_u^e]$ for the first eight operators and is equal to the number of customers in the chosen route for the last operator.

The removal set of customers are added to the removal list L_r. An insertion operator is then selected to iteratively insert the nodes of L_r into the partially destroyed solution until L_r is empty (Line 5). Feasibility conditions in terms of capacity and time windows are always respected during the insertion process. The removal and insertion operators are selected according to their past performance (see Section 2.5.3). The cost of an individual C before the removal is denoted by $\omega(C)$, and its cost after the insertion is denoted by $\omega(C^*)$. An example of the removal and insertion phases is illustrated in Figure 2.

2.6.1. Removal Operators

Nine removal operators are used in the destroy phase of the EDUCATION procedure and are described in detail below.

1. *Random removal* (RR): The RR operator randomly selects a node $j \in N\backslash\{0\}\backslash L_r$, removes it from the solution. The worst-case time complexity of the RR operator is $O(n)$.

Algorithm 2 EDUCATION

$\omega(C^*) = 0$, iteration $= 0$
while there is no improvement and C is feasible **do**
    $L_r = \varnothing$ and select a removal operator $d \in D$
    Apply a removal operator d to the individual C to remove a set of nodes and add them to L_r
    Select an insertion operator $m \in M$ and apply it to the partially destroyed individual C to insert the nodes of L_r
    Let C^* be the new solution obtained by applying operator m
    if iteration $= 1$ **then**
        insert C^* into the population
    end if
    if $\omega(C^*) < \omega(C)$ and C^* is feasible **then**
        $\omega(C) \leftarrow \omega(C^*)$
    end if
    iteration $\leftarrow$ iteration $+ 1$
end while
Return educated feasible solution

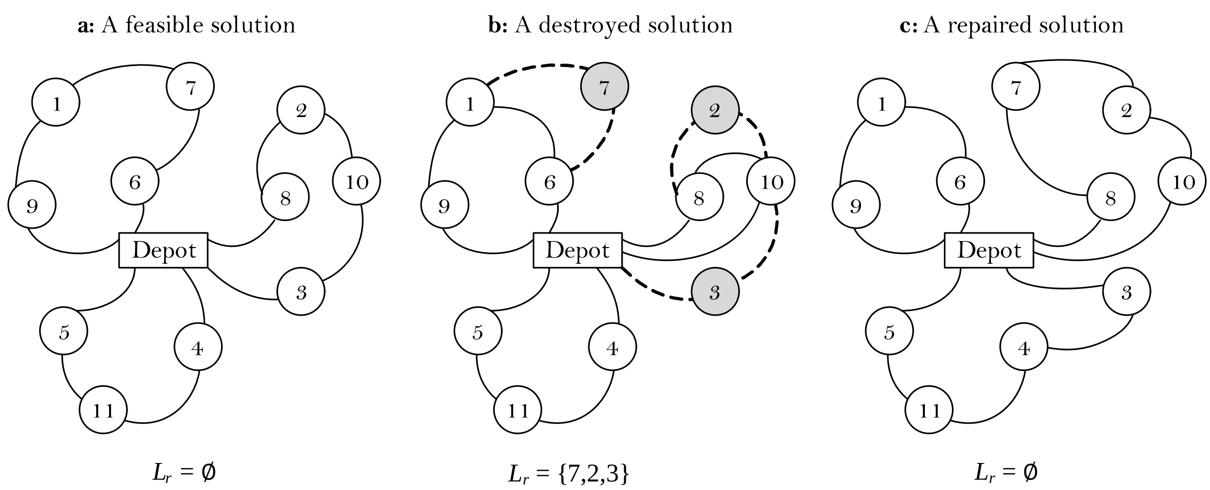

Figure 2 **Illustration of the** EDUCATION **procedure**

2. *Worst distance removal* (WDR): The purpose of the WDR operator is to choose a number of expensive nodes according to their distance based cost. The cost of a node $j \in N\backslash\{0\}\backslash L_r$ is the distance from its predecessor i and its distance to its successor j. The WDR operator iteratively

removes nodes j^* from the solution where $j^* = \arg\max_{j\in N\setminus\{0\}\setminus L_r}\{|d_{ij}+d_{jk}|\}$. The time complexity of this operator is $O(n^2)$.

3. *Worst time removal* (WTR): The WTR operator is a variant of the WDR operator. For each node $j \in N\setminus\{0\}\setminus L_r$ costs are calculated, depending on the deviation between the arrival time z_j and the beginning of the time window a_j. The WTR operator iteratively removes customers from the solution, where $j^* = \arg\max_{j\in N\setminus\{0\}\setminus L_r}\{|z_j - a_j|\}$. The ALNS iteratively applies this process to the solution after each removal. The WTR operator can be implemented in $O(n^2)$ time.

4. *Neighborhood removal* (NR): In a given solution with a set $\Re$ of routes, the NR operator calculates an average distance $\bar{d}(R) = \sum_{(i,j)\in R} d_{ij}/|R|$ for each route $R \in \Re$, and selects a node $j^* = \arg\max_{(R\in\Re;j\in R)}\{\bar{d}(R) - d_{R\setminus\{j\}}\}$, where $d_{R\setminus\{j\}}$ denotes the distance of route R excluding node j. The time complexity of this operator is $O(n^2)$.

5. *Shaw removal* (SR): The general idea behind the SR operator is to remove a set of n' similar customers. The similarity between two customers i and j is defined by the relatedness measure $\delta(i,j)$. This includes four terms: a distance term d_{ij}, a time term $|a_i - a_j|$, a relation term l_{ij}, which is equal to -1 if i and j are in the same route, and 1 otherwise, and a demand term $|q_i - q_j|$. The relatedness measure is given by

$$\delta(i,j) = \varphi_1 d_{ij} + \varphi_2|a_i - a_j| + \varphi_3 l_{ij} + \varphi_4|q_i - q_j|, \tag{5}$$

where φ_1 to φ_4 are weights that are normalized to find the best candidate solution. The operator starts by randomly selecting a node $i \in N\setminus\{0\}\setminus L_r$, and selects the node j^* to remove where $j^* = \arg\min_{j\in N\setminus\{0\}\setminus L_r}\{\delta(i,j)\}$. The operator is iteratively applied to select a node which is most similar to the one last added to L_r. The time complexity of this operator is $O(n^2)$.

6. *Proximity-based removal* (PBR): This operator is a second variant of the classical Shaw removal operator. The selection criterion of a set of routes is solely based on the distance. Therefore, the weights are $\varphi_1 = 1$ and $\varphi_2 = \varphi_3 = \varphi_4 = 0$. The PBR operator can be implemented in $O(n^2)$ time.

7. *Time-based removal* (TBR): The TBR operator removes a set of nodes that are related in terms of time. It is a special case of the Shaw removal operator where $\varphi_2 = 1$ and $\varphi_1 = \varphi_3 = \varphi_4 = 0$. Its time complexity is $O(n^2)$.

8. *Demand-based removal* (DBR): The DBR operator is yet another variant of the Shaw removal operator with $\varphi_4 = 1$ and $\varphi_1 = \varphi_2 = \varphi_3 = 0$. It can be implemented in $O(n^2)$ time.

9. *Average cost per unit removal* (ACUTR): The average cost per unit (ACUT) is described by Paraskevopoulos et al. (2008) to measure the utilization efficiency of a vehicle $\Pi(R)$ on a given route R. $\Pi(R)$ is expressed as the ratio of the total travel cost and fixed vehicle cost over the total demand carried by a vehicle k traversing route R:

$$\Pi(R) = \frac{\sum_{(i,j)\in A} c_{ij}x_{ij}^k + e^k}{\sum_{i\in N\setminus\{0\}} q_i x_{ij}^k}. \tag{6}$$

The aim of the ACUTR operator is to calculate the cost of each route and remove the one with the least $\Pi(R)$ value from the solution. The ACUTR operator can be implemented in $O(n^2)$ time.

2.6.2. Insertion Operators Three insertion operators are used in the repair phase of the EDUCATION procedure.

1. *Greedy insertion* (GI): The aim of this operator is to find the best possible insertion position for all nodes in L_r. For node $i \in N \backslash L_r$ succeeded in the destroyed solution by $k \in N \backslash \{0\} \backslash L_r$, and node $j \in L_r$ we define $\gamma(i,j) = d_{ij} + d_{jk} - d_{ik}$. We find the least-cost insertion position for $j \in L_r$ by $i^* = \arg\min_{i \in N \backslash L_r} \{\gamma(i,j)\}$. This process is iteratively applied to all nodes in L_r. The time complexity of this operator is $O(n^2)$.

2. *Greedy insertion with noise function* (GINF): The GINF operator is based on the GI operator but extends it by allowing a degree of freedom in selecting the best place for a node. This is done by calculating the noise cost $\upsilon(i,j) = \gamma(i,j) + d_{max} p_n \epsilon$ where d_{max} is the maximum distance between all nodes, p_n is a noise parameter used for diversification and is set equal to 0.1, and ϵ is a random number in $[-1, 1]$. The time complexity of this operator is $O(n^3)$.

3. *Greedy insertion with en-route time* (GIET): This operator calculates the *en-route* time difference $\eta(i,j)$ between before and after inserting the customer $j \in L_r$. For node $i \in N \backslash L_r$ succeeded in the destroyed solution by $k \in N \backslash \{0\} \backslash L_r$, and node $j \in L_r$, we define $\eta(i,j) = \tau_{ij} + \tau_{jk} - \tau_{ik}$ where τ_{ij} is the *en-route* time from node i to node j. We find the least-cost insertion position for $j \in L_r$ by $i^* = \arg\min_{i \in N \backslash L_r} \{\eta(i,j)\}$. The GIET operator can be implemented in $O(n^2)$ time.

2.6.3. Adaptive Weight Adjustment Procedure Each removal and insertion operator has a certain probability of being chosen in every iteration. The selection criterion is based on the historical performance of every operator and is calibrated by a roulette-wheel mechanism (Ropke and Pisinger 2006a, Demir et al. 2012). After it_w iterations of the roulette wheel segmentation, the probability of each operator is recalculated according to its total score. Initially, the probabilities of each removal and insertion operator are equal. Let p_i^t be the probability of operator i in the last it_w iterations, $p_i^{t+1} = p_i^t(1 - r_p) + r_p \pi_i / \tau_i$, where r_p is the roulette wheel probability, for operator i; π_i is its score and τ_i is the number times it was used during the last segment. At the start of each segment, the scores of all operators are set to zero. The scores are changed by σ_1 if a new best solution is found, by σ_2 if the new solution is better than the current solution and by σ_3 if the new solution is worse than the current solution.

2.7. INTENSIFICATION

We introduce a two-phase aggressive INTENSIFICATION procedure to improve the quality of elite individuals. This procedure intensifies the search within promising regions of the solution space. The detailed pseude-code of this method is shown in Algorithm 3. The algorithm starts with an elite list of solutions L_e, which takes the best n_e individuals from the main population as measured by equation (2). Step 1 is similar to the main EDUCATION procedure (Section 2.5). Step 2 attempts to explore different regions of the search space with the RR operator, intensifies this area by applying the GI operator for FD and HD, and GIET for FT and HT, to a partially the destroyed solution. Steps 1 and 2 terminate when there is no improvement to the solution and the main loop terminates when n_e successive iterations have been performed.

2.8. Survivor Selection

In population-based metaheuristics, avoiding premature convergence is a key challenge. Ensuring the diversity of the population, in other words to search a different location in the solution space during the algorithm, in the hope of being closer to the best known or optimal solutions constitutes a major trade-off between solutions in a population. The method of Vidal et al. (2012), aims to ensure the diversity of the population and preserve the elite solutions. The second part of this method is the survivor selection process (the first part was discussed in Section 2.3). The algorithm starts with an initial population of size n_p, and after each iteration an offspring is added to the population. The maximum number of allowable offsprings in the population is shown by n_o. When the current population size n_a reaches the maximum allowable size $n_p + n_o$, the survivor selection mechanism is put in place. This mechanism then selects n_p and discards n_o individuals from the population. The removal of n_o individuals is based on their biased fitness. In this way, elite individuals are protected.

2.9. Diversification

The efficient management of feasible solutions plays a significant role in population diversity. The performance of the HEA is improved by applying a MUTATION after the EDUCATION procedure. Over the iterations, individuals tend to become more similar, making it difficult to avoid premature convergence. To overcome this difficulty, we introduce a new scheme in order to increase the population diversity. The diversification stage includes two procedures, namely REGENERATION and MUTATION, representations of which are shown in Figure 3.

A REGENERATION procedure (Figure 3a) takes place when the maximum allowable iterations for REGENERATION it_r is reached without an improvement in the best solution value. In this procedure, the n_e elite individuals are preserved and are transferred to the next generation. The remaining $n_p - n_e$ individuals, which are ranked according to their biased fitness, are subjected to the RR and GINF operators, to create new individuals. At the end of this procedure, new individuals having various compositions are placed in the population.

The MUTATION procedure is applied with probability p_m. Figure 3b illustrates the MUTATION procedure. In this procedure, an individual C different from the best solution is randomly selected.

Algorithm 3 INTENSIFICATION

Initialize $L_e = \{O_1, \ldots, O_n\}$, $i \leftarrow 1$
while all elite solutions are intensified **do**
    $O \leftarrow O_i$
    Step 1
    while there is no improvement and elite solution O is feasible **do**
        $L_r = \varnothing$ and select a removal operator $d \in D$
        Apply $d \in D$ to the elite solution O to remove nodes and add them to L_r
        Select an insertion operator $m \in M$ and apply it to the destroyed elite solution O by inserting the node of L_r
        Let O^* be the new solution obtained by applying operator m
        if $\omega(O^*) < \omega(O)$ **then**
            $\omega(O) \leftarrow \omega(O^*)$
        end if
    end while
    Step 2
    while there is no improvement and O is feasible **do**
        $L_r = \varnothing$ and apply RR operator to the elite solution O to remove nodes and add them to L_r
        Apply GI or GIET operator to the partially destroyed elite solution O by inserting the node of L_r
        Let O^* be the new elite solution obtained by applying insertion operator
        if $\omega(O^*) < \omega(O)$ **then**
            $\omega(O) \leftarrow \omega(O^*)$
        end if
    end while
    $i \leftarrow i + 1$
end while
Return elite solutions

Two randomized structure based ALNS operators, the RR and the GINF, are then used to change the positions of a specific number of nodes, which are chosen from the interval $[b_l^m, b_u^m]$ of removable nodes in the MUTATION procedure.

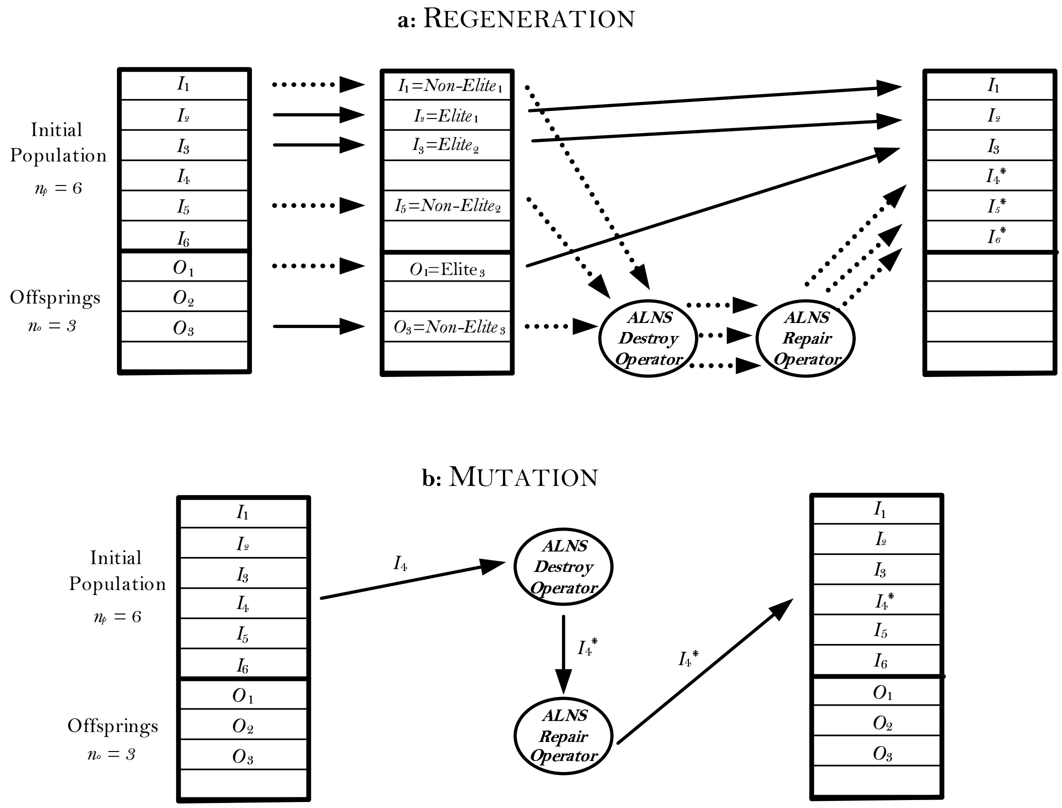

Figure 3 Illustration of the diversification stage

3. Computational Experiments

This section presents the results of computational experiments performed in order to assess the performance of the HEA. The HEA was implemented in C++ and run on a computer with one gigabyte RAM and Intel Xeon 2.6 GHz processor. We first describe the benchmark instances and the parameters used within the algorithm. This is followed by a presentation of the results.

3.1. Data Sets and Experimental Settings

The benchmark data sets of Liu and Shen (1999b), derived from the classical Solomon (1987) VRPTW instances with 100 nodes, are used as the test-bed. These sets include 56 instances, split into a random data set R, a clustered data set C and a semi-clustered data set RC. Sets shown by R1, C1 and RC1 have a short scheduling horizon and small vehicle capacities, in contrast to sets denoted R2, C2 and RC2 with a long scheduling horizon and large vehicle capacities. Liu and Shen (1999b) also introduced three cost structures, namely A, B and C, and several vehicle types with different capacities and fixed vehicle costs for each of the 56 instances. This results in a total of 168 benchmark instances for FT or FD.

The benchmark set used by Paraskevopoulos et al. (2008) for HT is a subset of the FT instances, in which the fleet size is set equal to that found in the best known solutions of Liu and Shen

(1999a). In total, there are 24 benchmark instances derived from Liu and Shen (1999a) for HT. We use the same set for HD, with the new objective.

Evolutionary algorithms use a set of correlated parameters and configuration decisions. In our implementation, we initially used the parameters suggested by Vidal et al. (2012, 2013) for the genetic algorithm, and by Demir et al. (2012) for the ALNS, but we have conducted several experiments to further fine-tune these parameters on instances C101A, C203A, R101A, R211A, RC105A and RC207A. Following these tests, the following parameter values were used in our experiments: $it_t = 5000, it_r = 2000, it_w = 500, n_p = 25, n_o = 25, n_e = 10, n_c = 3, p_m \in [0.4, 0.6], [b_l^i, b_u^i] = [0.3, 0.8], [b_l^e, b_u^e] = [0.1, 0.16], [b_l^m, b_u^m] = [0.1, 0.16], r_p = 0.1, \varphi_1 = 0.5, \varphi_2 = 0.25, \varphi_3 = 0.15, \varphi_4 = 0.25, \sigma_1 = 3, \sigma_2 = 1, \sigma_3 = 0, \lambda_t = \lambda_l = \lambda_f = 3, \lambda_m = 10$. These settings are identical for all four considered problems.

Table 1 presents the results of a fine-tuning experiment on parameters n_p and n_o, and to test the effect of these parameters on the solution quality.

Table 1 Average percentage deviations of the solution values found by the HEA from best-known solution values with varying n_p and n_o

	n_o				
n_p	10	25	50	75	100
10	0.42	0.26	0.38	0.56	0.69
25	0.19	**0.11**	0.26	0.37	0.49
50	0.39	0.29	0.30	0.45	0.57
75	0.56	0.42	0.51	0.61	0.68
100	0.67	0.53	0.61	0.72	0.78

The table shows the percent gap between the solution value obtained by the HEA and the best-known solution (BKS) value, averaged over the six chosen instances. The maximum population size is dependent on n_p and n_o, both of which have a significant impact on the solution quality, where the best setting is obtained with $n_p = n_o = 25$.

3.2. Comparative Analysis

We now present a comparative analysis of the results of the HEA with those reported in the literature. In particular, we compare ourselves against LSa (Liu and Shen 1999a), LSb (Liu and Shen 1999b), T-RR-TW (Dell'Amico et al. 2007), ReVNTS (Paraskevopoulos et al. 2008), MDA (Bräysy et al. 2008), BPDRT (Bräysy et al. 2009), AMP (Repoussis and Tarantilis 2010) and UHGS (Vidal et al. 2014). The comparisons are presented in Tables 2–4, where the columns show the total cost (TC), and percent deviations (Dev) of the values of solutions found by each method with respect to the HEA. The first column displays the instance sets and the number of instances in each set in parentheses. The rows named Avg (%), Min (%) and Max (%) show the average, minimum and maximum deviations across all benchmark instances, respectively. A negative deviation shows that the solution found by the HEA is of better quality. In the column labeled BKS, "=" shows the total number of matches and "<" shows the number of new BKS found for each instance set.

Ten separate runs are performed for each instance, the best one of which is reported. For each instance, a boldface refers to match with current BKS, where as a boldface with a "*" indicates new BKS. For detailed results, the reader is referred to Appendix A. Tables A.1-A.6 present the fixed vehicle cost (VC), the distribution cost (DC), the computational time in minutes (Time) and the actual number of vehicles used (Mix) for the HEA.

Tables 2 and 3 summarize the average comparison results of the current state-of-the-art solution methods for FT and FD, compared with the HEA. According to Tables 2 and 3, the HEA is highly competitive, with average deviations ranging from −5.63% to 0.13% and a worst-case performance of 0.98% for FT. As for FD, average cost reductions range from −0.78% to 0.10% and the worst case performance is 0.94%. The average performance of our HEA is better than that of all the competition, except for the algorithm of Vidal et al. (2014) which is slightly better on average. However, the HEA seems to outperforms this algorithm on to the second type of FT instances, which are less tight in terms of vehicle capacity.

Table 2 Average results for FT

Instance set	T-RR-TW		ReVNTS		MDA		AMP		UHGS		HEA	BKS	
	TC	Dev	TC	Dev	TC	Dev	TC	Dev	TC	Dev	TC	=	<
R1A (12)	4180.83	−1.51	4128.48	−0.24	4131.31	−0.31	4113.89	0.12	4103.16	0.38	4118.70	0	0
R1B (12)	1927.57	−1.55	1902.19	−0.22	1898.88	−0.04	1896.83	0.07	1891.63	0.34	1898.10	0	**1***
R1C (12)	1615.44	−2.43	1582.18	−0.33	1579.17	−0.13	1578.12	−0.07	1574.32	0.17	1577.04	0	0
C1A (9)	7229.02	−1.20	7143.16	0.01	7141.15	0.03	7139.96	0.05	7138.93	0.06	7143.35	**2**	0
C1B (9)	2384.77	−0.95	2361.78	0.02	2365.49	−0.13	2359.82	0.11	2359.63	0.12	2362.38	**1**	0
C1C (9)	1629.70	−0.62	1621.09	−0.09	1621.83	−0.14	1618.91	0.04	1619.18	0.03	1619.63	**6**	0
RC1A (8)	5117.96	−3.35	4961.69	−0.19	4948.53	0.07	4948.02	0.08	4915.10	0.75	4952.16	0	0
RC1B (8)	2163.51	−1.13	2142.65	−0.16	2129.60	0.45	2136.73	0.12	2129.04	0.48	2139.23	0	**1***
RC1C (8)	1784.51	−1.25	1769.93	−0.42	1758.29	0.24	1762.34	0.01	1752.19	0.59	1762.56	0	0
R2A (11)	3568.97	−9.06	3304.57	−0.98	3310.70	−1.17	3287.80	−0.47	3267.31	0.16	3272.48	**2**	**1***
R2B (11)	1727.04	−17.4	1498.97	−1.88	1495.37	−1.63	1487.09	−1.07	1480.30	−0.60	**1471.33***	**1**	**7***
R2C (11)	1436.22	−15.3	1281.31	−2.84	1257.65	−0.94	1260.97	−1.20	1237.79	0.66	1245.97	0	0
C2A (8)	6267.75	−9.07	5759.02	−0.22	5797.38	−0.88	5749.98	−0.06	5760.29	−0.20	**5746.53***	**4**	0
C2B (8)	1897.62	−8.53	1754.07	−0.32	1756.08	−0.43	1748.99	−0.03	1750.37	−0.10	**1748.52***	**3**	0
C2C (8)	1276.29	−4.77	1232.98	−1.21	1223.86	−0.47	1224.08	−0.48	1221.17	−0.20	**1218.18***	**4**	**2***
RC2A (8)	4752.95	−8.24	4406.28	−0.34	4399.12	−0.18	4388.88	0.05	4381.73	0.21	4391.16	0	0
RC2B (8)	2156.11	−15.4	1888.83	−1.13	1899.20	−1.68	1874.86	−0.38	1877.84	−0.50	**1867.80***	0	**1***
RC2C (8)	1828.95	−19.5	1567.22	−2.43	1562.19	−2.10	1541.13	−0.72	1545.29	−1.00	**1530.08***	0	0
Avg (%)		−5.63		−0.50		−0.56		−0.12		0.13			
Min (%)		–		−6.64		−10.2		−3.26		−4.19			
Max (%)		–		0.96		0.93		0.97		0.98			
Total												**23**	**13***
Runs	1		1		3		1		10		10		
Processor	P 600M		PIV 1.5GHz		Ath 2.6GHz		PIV 3.4GHz		Opt 2.2GHz		Xe 2.6GHz		
Avg Time	14.15		20		10.97		16.67		5.08		4.83		

Table 4 presents the comparison results for each HT instance against LSa and ReVNTS. We note that LSa originally solved FT, which was the basis for setting the number of available vehicles in ReVNTS. The results show that the HEA outperforms both methods and yields higher quality solutions within short computation times. On average, the total cost reductions obtained were

Table 3 Average results for FD

Instance set	MDA		BPDRT		UHGS		HEA	BKS	
	TC	Dev	TC	Dev	TC	Dev	TC	=	<
R1A (12)	4068.59	−0.53	4060.96	−0.34	4031.28	0.39	4047.06	0	0
R1B (12)	1854.60	−0.42	–	–	1841.43	0.29	1846.79	0	**1***
R1C (12)	1539.48	−0.65	1539.90	−0.67	1530.25	−0.04	**1529.61***	0	**5***
C1A (9)	7085.56	−0.04	7085.91	−0.04	7082.98	0.00	**7082.98**	**9**	0
C1B (9)	2335.11	−0.09	–	–	2332.90	0.00	**2332.90**	**9**	0
C1C (9)	1615.75	−0.02	1615.40	−0.01	1615.49	−0.01	**1615.39***	**8**	**1***
RC1A (8)	4944.48	−0.57	4935.52	−0.39	4891.25	0.51	4916.41	0	0
RC1B (8)	2121.62	−0.45	–	–	2107.08	0.24	2112.12	**1**	**1***
RC1C (8)	1741.78	−0.48	1749.66	−0.94	1734.36	−0.06	**1733.40***	**2**	**4***
R2A (11)	3193.41	−1.33	3180.59	−0.92	3151.95	−0.01	**3151.54***	**5**	**3***
R2B (11)	1392.92	−3.06	–	–	1351.91	−0.03	**1351.52***	**4**	**2***
R2C (11)	1149.65	−2.06	1149.11	−2.01	1128.71	−0.20	**1126.42***	**5**	**4***
C2A (8)	5690.87	−0.07	5689.40	−0.05	5686.75	0.00	**5686.75**	**8**	0
C2B (8)	1698.51	−0.70	–	–	1686.75	0.00	**1686.75**	**8**	0
C2C (8)	1186.03	−0.07	1185.70	−0.04	1185.19	0.00	**1185.19**	**8**	0
RC2A (8)	4241.33	−0.73	4231.25	−0.49	4210.10	0.00	**4210.10**	**3**	**1***
RC2B (8)	1704.13	−0.72	–	–	1686.63	0.31	1691.94	0	**2***
RC2C (8)	1374.55	−0.85	1385.32	−1.64	1358.24	0.34	1362.90	**1**	**1***
Avg (%)		−0.78		−0.65		0.10			
Min (%)		−4.30		−7.67		−1.26			
Max (%)		0.41		0.10		0.94			
Total								**71**	**25***
Runs	3		1		10		10		
Processor	Ath 2.6GHz		Duo 2.4GHz		Opt 2.2GHz		Xe 2.6GHz		
Avg Time	3.56		–		4.72		4.43		

−12.68% and −0.34% compared to LSa and ReVNTS, with minimum deviations of −29.47% and −2.01% and maximum deviations of −1.26% and 0.35%, respectively. Finally, Table 5 shows the results obtained on the newly introduced HD.

In summary, the HEA was able to find 36 BKS for 168 FT instances, where 14 are strictly better than those obtained by competing heuristics. As for FD, the algorithm has identified 96 BKS out of the 168 instances, 25 of which are strictly better. The results are even more striking for HT, with 17 BKS on the 24 instances, 14 of which are strictly better than those reported earlier. Overall, the HEA improves 52 BKS and matches 97 BKS out of 360 benchmark instances.

4. Conclusions

We have proposed a unified heuristic for four types of heterogeneous fleet vehicle routing problems with time windows. The first two are the Fleet Size and Mix Vehicle Routing Problem with Time Windows (F) and the Heterogeneous Fixed Fleet Vehicle Routing Problem with Time Windows (H). Each of these two problems was solved under a time and a distance objective, yielding the four variants FT, FD, HT and HD. We have developed a unified hybrid evolutionary algorithm (HEA) capable of solving all variants without any modification. This heuristic combines state-of-the-art metaheuristic principles such as adaptive large scale neighborhood search and population search. We have integrated within our HEA an innovative Intensification strategy on elite solutions and we have developed a new diversification scheme based on the Regeneration and the Mutation

Table 4 Results for HT

Instance set	LSa			ReVNTS			HEA					BKS	
	Mix	TC	Dev	Mix	TC	Dev	DC	VC	Mix	TC	Time	=	<
R101	$A^{1}B^{11}C^{11}D^{1}$	5061	−10.29	$B^{10}C^{11}D^{1}$	4583.99	0.10	1998.76	2590	$B^{10}C^{11}D^{1}$	4588.76	5.49	0	0
R102	$A^{1}B^{4}C^{14}D^{2}$	5013	−13.25	$B^{3}C^{14}D^{2}$	4420.68	0.13	1736.54	2640	$A^{1}B^{4}C^{13}D^{2}$	**4376.54***	6.78	0	**1***
R103	$B^{7}C^{15}$	4772	−13.57	$B^{6}C^{15}$	4195.05	0.16	1621.71	2580	$B^{6}C^{15}$	4201.71	7.45	0	0
R104	$B^{9}C^{14}$	4455	−10.61	$B^{8}C^{14}$	4065.52	−0.94	1487.69	2540	$B^{9}C^{13}$	**4027.69***	6.14	0	**1***
C101	$A^{1}B^{10}$	9272	−5.02	B^{10}	8828.93	0.00	828.93	8000	B^{10}	**8828.93**	3.67	**1**	0
C102	A^{19}	8433	−17.89	A^{19}	7137.79	0.21	1453.13	5700	A^{19}	7153.13	4.12	0	0
C103	A^{19}	8033	−12.78	A^{19}	7143.88	−0.30	1422.57	5700	A^{19}	**7122.57***	3.45	0	**1***
C104	A^{19}	7384	−4.25	A^{19}	7104.96	−0.30	1383.74	5700	A^{19}	**7083.74***	3.13	0	**1***
RC101	$A^{7}B^{7}C^{7}$	5687	−7.99	$A^{4}B^{7}C^{7}$	5279.92	−0.26	1876.36	3390	$A^{4}B^{7}C^{7}$	**5266.36***	5.73	0	**1***
RC102	$A^{5}B^{6}C^{8}$	5649	−10.77	$A^{4}B^{5}C^{8}$	5149.95	−0.99	1709.55	3390	$A^{4}B^{5}C^{8}$	**5099.55***	5.14	0	**1***
RC103	$A^{11}B^{2}C^{8}$	5419	−8.58	$A^{10}B^{2}C^{8}$	5002.41	−0.22	1691.29	3300	$A^{10}B^{2}C^{8}$	**4991.29***	4.90	0	**1***
RC104	$A^{2}B^{13}C^{3}D^{1}$	5189	−3.43	$A^{2}B^{13}C^{3}D^{1}$	5024.25	−0.15	1596.97	3420	$A^{2}B^{13}C^{3}D^{1}$	**5016.97***	5.21	0	**1***
R201	A^{5}	4593	−21.43	A^{5}	3779.12	0.09	1532.49	2250	A^{5}	3782.49	7.45	0	0
R202	A^{5}	4331	−20.85	A^{5}	3578.91	0.14	1333.92	2250	A^{5}	3583.92	8.45	0	0
R203	$A^{4}B^{1}$	4220	−18.74	$A^{4}B^{1}$	3582.54	−0.81	1053.92	2500	$A^{4}B^{1}$	**3553.92***	7.12	0	**1***
R204	A^{5}	3849	−24.89	A^{5}	3143.68	−2.01	831.80	2250	A^{5}	**3081.80***	6.99	0	**1***
C201	$A^{4}B^{1}$	6711	−9.29	$A^{4}B^{1}$	6140.64	0.00	740.64	5400	$A^{4}B^{1}$	**6140.64**	4.89	**1**	0
C202	$A^{1}C^{3}$	7720	−1.26	$A^{1}C^{3}$	7752.88	−1.69	623.96	7000	$A^{1}C^{3}$	**7623.96***	4.26	0	**1***
C203	$C^{2}D^{1}$	7466	−2.23	$C^{2}D^{1}$	7303.37	0.00	603.37	6700	$C^{2}D^{1}$	**7303.37**	4.37	**1**	0
C204	A^{5}	6744	−18.72	A^{5}	5721.09	−0.72	680.46	5000	A^{5}	**5680.46***	5.29	0	**1***
RC201	$C^{1}E^{3}$	5871	−6.08	$C^{1}E^{3}$	5523.15	0.21	1684.59	3850	$C^{1}E^{3}$	5534.59	6.47	0	0
RC202	$A^{1}C^{1}D^{1}E^{2}$	5945	−15.43	$A^{1}C^{1}D^{1}E^{2}$	5132.08	0.35	1450.23	3700	$A^{1}C^{1}D^{1}E^{2}$	5150.23	6.35	0	0
RC203	$A^{1}B^{1}C^{5}$	5790	−29.47	$A^{1}B^{1}C^{5}$	4508.27	−0.81	1221.92	3250	$A^{1}B^{1}C^{5}$	**4471.92***	6.01	0	**1***
RC204	$A^{14}B^{2}$	4983	−17.47	$A^{14}B^{2}$	4252.87	−0.26	1441.83	2800	$A^{14}B^{2}$	**4241.83***	5.87	0	**1***
Avg (%)			−12.68			−0.34							
Min (%)			−29.47			−2.01							
Max (%)			−1.26			0.35							
Total												**3**	**14***
Runs	3			1			10						
Processor	P 233M			PIV 1.5GHz			Xe 2.6GHz						
Avg Time	–			20.00			5.61						

Table 5 Results for HD

Instance set	HEA				
	DC	VC	Mix	TC	Time
R101A	1765.41	2590	$B^{10}C^{11}D^{1}$	4355.41	5.19
R102A	1716.44	2640	$B^{4}C^{13}D^{2}$	4356.44	6.24
R103A	1500.16	2580	$B^{6}C^{15}$	4080.16	6.57
R104A	1434.72	2520	$B^{7}C14$	3954.72	5.89
C101A	828.94	8000	B^{10}	8828.94	4.25
C102A	1380.17	5700	A^{19}	7080.17	3.97
C103A	1379.21	5700	A^{19}	7079.21	3.99
C104A	1375.06	5700	A^{19}	7075.06	2.98
RC101A	1772.28	3390	$A^{4}B^{7}C^{7}$	5162.28	6.41
RC102A	1598.05	3420	$A^{2}B^{6}C^{8}$	5018.05	5.24
RC103A	1626.55	3300	$A^{10}B^{2}C^{8}$	4926.55	4.39
RC104A	1575.91	3420	$A^{2}B^{13}C^{3}D^{1}$	4995.91	4.88
R201A	1198.76	2250	A^{5}	3448.76	6.74
R202A	1058.16	2250	A^{5}	3308.16	8.13
R203A	882.39	2500	$A^{4}B^{1}$	3382.39	7.49
R204A	768.14	2250	A^{5}	3018.14	5.47
C201A	682.38	5400	$A^{4}B^{1}$	6082.38	4.21
C202A	618.62	7000	$A^{1}C^{3}$	7618.62	3.69
C203A	603.37	6700	$C^{2}D^{1}$	7303.37	3.67
C204A	677.66	5000	A^{5}	5677.66	5.11
RC201A	1494.47	3850	$C^{1}E^{3}$	5344.47	6.72
RC202A	1156.02	3700	$A^{1}C^{1}D^{1}E^{2}$	4856.02	6.48
RC203A	996.25	3250	$A^{1}B^{1}C^{5}$	4246.25	6.93
RC204A	1395.32	2800	$A^{14}B^{2}$	4195.32	6.17
Average				5224.77	5.45
Runs	10				
Processor	Xe 2.6GHz				
Avg Time	5.45				

of solutions. We have also developed an advanced version of the Split algorithm of Prins (2009) to determine the best fleet mix for a set of routes. Finally, we have introduced the new variant HD. Extensive computational experiments were carried out on benchmark instances. In the case of FT and FD, our HEA clearly outperforms all previous algorithms except that of Vidal et al. (2014). In the latter case, it performs slightly worse on average, but seems to be superior on instances which are less tight in terms of vehicle capacity. Overall, the HEA has identified 132 new best solutions out of 336 on the F instances, 39 of which are strictly better. On the HT instances, our HEA outperforms the two existing algorithms and has identified 17 best known solutions out of 24, 14 of which are strictly better. The HD instances are solved here for the first time. Overall, we have improved 52 solutions out of 360 instances, and we have matched 97 others. All instances were solved within a modest computational effort. Our algorithm is not only highly competitive, but it is also flexible in that it can solve four problem classes with the same parameter settings.

Acknowledgments

The authors gratefully acknowledge funding provided by the Southampton Management School at the University of Southampton and by the Canadian Natural Sciences and Engineering Research Council under grants 39682-10 and 436014-2013.

Appendix

Table A.1 to A.6 present the detailed results on all benchmark instances for FT and FD.

Table A.1 Results for FT for cost structure A

Instance set	ReVNTS		MDA		AMP		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101A	4539.99	0.04	4631.31	−1.97	4536.40	0.12	4608.62	−1.50	1951.70	2590	$A^1B^2C^{17}$	4541.70	5.26
R102A	4375.70	−0.47	4401.31	−1.06	4348.92	0.14	4369.74	−0.30	1775.10	2580	B^6C^{15}	4355.10	5.87
R103A	4120.63	0.26	4182.16	−1.23	4119.04	0.30	4145.68	−0.30	1551.23	2580	B^6C^{15}	4131.23	4.19
R104A	3992.65	−0.01	3981.28	0.27	3986.35	0.14	3961.39	0.77	1302.10	2690	$B^5C^{11}D^3$	3992.10	5.02
R105A	4229.69	0.07	4236.84	−0.10	4229.67	0.07	4209.84	0.54	1672.54	2560	B^4C^{16}	4232.54	4.73
R106A	4137.96	0.01	4118.48	0.48	4130.82	0.18	4109.08	0.71	1538.30	2600	B^1C^{18}	4138.30	5.13
R107A	4061.10	−0.66	4035.96	−0.04	4031.16	0.08	4007.87	0.66	1474.32	2560	B^4C^{16}	4034.32	5.40
R108A	3986.07	−0.50	3970.26	−0.10	3962.20	0.10	3934.48	0.80	1406.10	2560	B^4C^{16}	3966.10	4.78
R109A	4086.72	−0.68	4060.17	−0.03	4052.21	0.17	4020.75	0.94	1429.02	2630	$C^{17}D^1$	4059.02	4.60
R110A	4030.85	−0.86	3995.18	0.03	3999.09	−0.07	3965.88	0.76	1436.31	2560	B^4C^{16}	3996.31	4.17
R111A	4018.80	0.03	4017.81	0.06	4016.19	0.10	3985.68	0.86	1460.10	2560	$B^4C^{13}D^2$	4020.10	4.98
R112A	3961.63	−0.10	3947.30	0.26	3954.65	0.07	3918.88	0.98	1397.60	2560	B^4C^{16}	3957.60	5.78
C101A	7226.51	0.00	7226.51	0.00	7226.51	0.00	7226.51	0.00	1526.51	5700	A^{19}	**7226.51**	2.97
C102A	7137.79	0.11	7119.35	0.37	7137.79	0.11	7119.35	0.37	1445.65	5700	A^{19}	7145.65	3.10
C103A	7143.88	0.00	7107.01	0.52	7141.03	0.04	7102.86	0.57	1443.88	5700	A^{19}	7143.88	2.70
C104A	7104.96	−0.31	7081.50	0.02	7086.70	−0.05	7081.51	0.02	1382.92	5700	A^{19}	7082.92	2.01
C105A	7171.48	0.05	7199.36	−0.34	7169.08	0.08	7196.06	−0.30	1475.00	5700	A^{19}	7175.00	2.45
C106A	7157.13	0.09	7180.03	−0.23	7157.13	0.09	7176.68	−0.20	1463.32	5700	A^{19}	7163.32	3.01
C107A	7135.43	0.07	7149.17	−0.13	7135.38	0.07	7144.49	−0.10	1440.20	5700	A^{19}	7140.20	2.78
C108A	7115.71	0.07	7115.81	0.07	7113.57	0.10	7111.23	0.14	1420.98	5700	A^{19}	7120.98	2.45
C109A	7095.55	−0.05	7094.65	−0.04	7092.49	−0.01	7091.66	0.00	1391.66	5700	A^{19}	**7091.66**	2.37
RC101A	5253.86	−0.17	5253.97	−0.17	5237.19	0.15	5217.90	0.51	1914.89	3330	$A^3B^{11}C^5$	5244.89	4.97
RC102A	5053.48	0.13	5059.58	0.01	5053.62	0.13	5018.47	0.82	1670.20	3390	$A^4B^7C^7$	5060.20	5.64
RC103A	4892.80	−0.47	4868.94	0.02	4885.58	−0.32	4822.21	0.98	1480.00	3390	$A^4B^3C^9$	4870.00	5.14
RC104A	4783.31	−0.29	4762.85	0.14	4761.28	0.17	4737.00	0.68	1289.30	3480	$A^3B^1C^9D^1$	4769.30	4.97
RC105A	5112.91	0.10	5119.8	−0.03	5110.86	0.14	5097.35	0.41	1788.10	3330	$A^3B^{11}C^5$	5118.10	5.32
RC106A	4997.98	−0.47	4960.78	0.27	4966.27	0.16	4935.91	0.77	1584.37	3390	$A^4B^9C^6$	4974.37	6.01
RC107A	4862.67	−0.78	4828.17	−0.06	4819.91	0.11	4783.08	0.87	1405.21	3420	$A^4B^7C^7$	4825.21	5.37
RC108A	4736.50	0.39	4734.15	0.44	4749.44	0.12	4708.85	0.97	1365.20	3390	$A^4B^3C^9$	4755.20	4.71
R201A	3779.12	−0.50	3922.00	−4.30	3753.42	0.19	3782.88	−0.60	1510.43	2250	A^5	3760.43	8.97
R202A	3578.91	−0.70	3610.38	−1.58	3551.12	0.09	3540.03	0.40	1304.20	2250	A^5	3554.20	9.98
R203A	3334.08	−0.56	3350.18	−1.05	3336.60	−0.64	3311.35	0.13	1065.50	2250	A^5	3315.50	8.76
R204A	3143.68	−2.20	3390.14	−10.20	3103.84	−0.91	3075.95	0.00	825.95	2250	A^5	**3075.95**	7.98
R205A	3371.47	−1.12	3465.81	−3.95	3367.90	−1.01	3334.27	0.00	1084.27	2250	A^5	**3334.27**	8.45
R206A	3272.79	−0.29	3268.36	−0.15	3264.70	−0.04	3242.40	0.64	1013.40	2250	A^5	3263.40	8.17
R207A	3213.60	−1.94	3231.26	−2.51	3158.69	−0.20	3145.08	0.23	902.29	2250	A^5	3152.29	9.29
R208A	3064.76	−1.58	3063.10	−1.52	3056.45	−1.30	3017.52	−0.01	767.12	2250	A^5	**3017.12***	8.51
R209A	3191.63	0.08	3192.95	0.04	3194.74	−0.01	3183.36	0.34	944.28	2250	A^5	3194.28	9.37
R210A	3338.75	−0.89	3375.38	−2.00	3325.28	−0.48	3287.66	0.65	1059.26	2250	A^5	3309.26	8.79
R211A	3061.47	−1.35	3042.48	−0.73	3053.08	−1.08	3019.93	0.02	770.56	2250	A^5	3020.56	7.99
C201A	5820.78	0.16	5891.45	−1.05	5820.78	0.16	5878.54	−0.80	830.20	5000	A^5	5830.20	5.00
C202A	5779.59	−0.05	5850.26	−1.27	5783.76	−0.12	5776.88	0.00	776.88	5000	A^5	**5776.88**	5.17
C203A	5750.58	−0.15	5741.90	−0.01	5736.94	0.09	5741.12	0.00	741.12	5000	A^5	**5741.12**	4.76
C204A	5721.09	−0.72	5691.51	−0.19	5718.49	−0.67	5680.46	0.00	680.46	5000	A^5	**5680.46**	4.21
C205A	5750.53	0.02	5786.71	−0.61	5747.67	0.06	5781.15	−0.50	751.40	5000	A^5	5751.40	6.79
C206A	5757.93	−0.29	5795.15	−0.94	5738.09	0.06	5767.70	−0.50	741.30	5000	A^5	5741.30	4.30
C207A	5723.91	0.02	5743.52	−0.32	5721.16	0.07	5731.44	−0.10	725.10	5000	A^5	5725.10	4.17
C208A	5767.78	−0.75	5884.20	−2.78	5732.95	−0.14	5725.03	0.00	725.03	5000	A^5	**5725.03**	5.21
RC201A	4726.22	−0.39	4740.21	−0.69	4701.88	0.13	4737.59	−0.60	2007.80	2700	A^{18}	4707.80	4.50
RC202A	4518.49	0.02	4522.36	−0.07	4509.11	0.23	4487.48	0.71	1619.40	2900	$A^{10}B^4$	4519.40	4.67
RC203A	4327.57	−0.20	4312.52	0.15	4313.42	0.13	4305.49	0.32	1469.10	2850	$A^{12}B^3$	4319.10	5.27
RC204A	4166.73	−0.26	4141.04	0.35	4157.32	−0.04	4137.93	0.43	1005.77	3150	$A^2B^5C^2$	4155.77	5.19
RC205A	4645.41	−1.08	4652.57	−1.24	4585.20	0.23	4615.04	−0.40	1795.67	2800	$A^{14}B^2$	4595.67	6.89
RC206A	4416.41	0.40	4431.64	0.06	4427.73	0.15	4405.16	0.66	1584.30	2850	$A^9B^3C^1$	4434.30	5.03
RC207A	4338.94	−0.53	4310.11	0.13	4313.07	0.07	4290.14	0.60	1215.00	3100	A^4B^7	4315.90	6.27
RC208A	4109.90	−0.70	4091.92	−0.26	4103.31	−0.54	4075.04	0.16	1031.37	3050	$A^5B^5C^1$	4081.37	5.17

Table A.2 Results for FT for cost structure B

Instance set	ReVNTS		MDA		AMP		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101B	2421.19	0.16	2486.76	−2.54	2421.19	0.16	2421.19	0.16	1849.10	576	$A^1B^4C^9D^5$	2425.10	3.78
R102B	2219.03	−0.30	2227.48	−0.68	2209.50	0.13	2209.50	0.13	1608.37	604	$A^2B^1C^6D^8$	2212.37	3.97
R103B	1955.57	0.08	1938.93	0.93	1953.50	0.18	1938.93	0.93	1325.10	632	$A^1C^4D^8E^1$	1957.10	4.28
R104B	1732.26	−0.99	1714.73	0.03	1713.36	0.11	1713.36	0.11	1071.20	644	$A^1C^3D^7E^2$	1715.20	4.01
R105B	2030.83	−0.29	2027.98	−0.15	2030.83	−0.29	2027.98	−0.15	1436.91	588	$B^3C^5D^8$	**2024.91***	3.68
R106B	1924.03	−0.10	1919.03	0.16	1919.02	0.16	1919.02	0.16	1338.10	584	$B^1C^6D^8$	1922.10	4.19
R107B	1781.01	0.12	1789.58	−0.36	1780.52	0.15	1780.52	0.15	1127.20	656	$C^2D^8E^2$	1783.20	5.30
R108B	1667.51	−0.36	1649.24	0.74	1665.78	−0.25	1649.24	0.74	983.58	678	$C^1D^5E^4$	1661.58	4.78
R109B	1844.99	−0.02	1828.63	0.86	1840.54	0.22	1828.63	0.86	1232.58	612	C^4D^{10}	1844.58	4.91
R110B	1792.75	−0.78	1774.46	0.24	1788.18	−0.53	1774.46	0.24	1178.80	600	$B^1C^3D^{10}$	1778.80	5.21
R111B	1780.03	−0.27	1769.71	0.31	1772.51	0.15	1769.71	0.31	1141.24	634	$C^3D^7E^2$	1775.24	4.78
R112B	1677.13	−0.01	1669.78	0.43	1667.00	0.60	1667.00	0.60	1071.00	606	C^2D^{11}	1677.00	6.21
C101B	2417.52	0.00	2417.52	0.00	2417.52	0.00	2417.52	0.00	977.52	1440	A^8B^6	**2417.52**	1.99
C102B	2350.54	0.26	2350.55	0.25	2350.54	0.26	2350.54	0.26	976.56	1380	A^7B^6	2356.56	2.45
C103B	2349.42	−0.01	2353.64	−0.19	2347.99	0.05	2347.99	0.05	969.10	1380	A^7B^6	2349.10	3.47
C104B	2332.94	−0.10	2328.62	0.08	2325.78	0.21	2325.78	0.21	950.59	1380	A^7B^6	2330.59	3.09
C105B	2374.01	0.10	2373.53	0.12	2375.04	0.06	2373.53	0.12	956.45	1420	A^5B^7	2376.45	3.06
C106B	2381.14	0.22	2404.56	−0.76	2381.14	0.22	2381.14	0.22	966.43	1420	A^5B^7	2386.43	2.95
C107B	2357.52	0.06	2370.00	−0.47	2357.67	0.06	2357.52	0.06	939.00	1420	A^5B^7	2359.00	2.45
C108B	2346.38	0.08	2346.38	0.08	2346.38	0.08	2346.38	0.08	968.15	1380	A^7B^6	2348.15	2.79
C109B	2346.58	−0.38	2339.89	−0.10	2336.29	0.06	2336.29	0.06	957.60	1380	A^7B^6	2337.60	2.56
RC101B	2469.50	0.04	2462.60	0.32	2464.66	0.23	2462.60	0.32	1720.41	750	$B^7C^6D^2$	2470.41	4.47
RC102B	2277.79	0.28	2263.45	0.90	2272.68	0.50	2263.45	0.90	1552.08	732	$A^1B^3C^9D^1$	2284.08	4.12
RC103B	2057.55	−0.80	2035.62	0.27	2041.24	−0.01	2035.62	0.27	1291.20	750	$B^1C^9D^2$	2041.20	3.98
RC104B	1914.93	0.38	1905.06	0.90	1916.85	0.28	1905.06	0.90	1172.27	750	$B^1C^6D^4$	1922.27	4.21
RC105B	2337.93	−0.44	2308.59	0.82	2325.99	0.07	2308.59	0.82	1625.70	702	$A^1B^7C^8$	2327.70	4.56
RC106B	2168.44	−0.24	2149.56	0.63	2160.45	0.13	2149.56	0.63	1431.20	732	$A^1B^4C^7D^2$	2163.20	4.21
RC107B	2008.39	−0.62	2000.77	−0.23	2003.26	−0.36	2000.77	−0.23	1264.09	732	$A^1B^2C^5D^4$	**1996.09***	4.19
RC108B	1906.69	0.12	1910.83	−0.10	1908.72	0.01	1906.69	0.12	1176.89	732	$A^1B^1C^7D^3$	1908.89	3.11
R201B	1965.10	−0.45	2002.53	−2.37	1953.42	0.14	1953.42	0.14	1456.21	500	A^4B^1	1956.21	6.21
R202B	1765.09	−0.72	1790.38	−2.17	1751.12	0.07	1751.12	0.07	1302.40	450	A^5	1752.40	8.00
R203B	1535.08	−1.31	1541.19	−1.72	1536.60	−1.41	1535.08	−1.31	1065.17	450	A^5	**1515.17***	5.78
R204B	1306.72	−2.12	1284.33	−0.37	1303.84	−1.90	1284.33	−0.37	829.57	450	A^5	**1279.57***	6.89
R205B	1575.75	−1.70	1563.62	−0.92	1560.07	−0.69	1560.07	−0.69	1099.39	450	A^5	**1549.39***	6.49
R206B	1477.34	−1.86	1464.53	−0.98	1464.70	−0.99	1464.53	−0.98	1000.37	450	A^5	**1450.37***	5.21
R207B	1386.84	−2.04	1380.41	−1.56	1358.69	0.04	1358.69	0.04	909.18	450	A^5	1359.18	6.31
R208B	1261.09	−3.34	1244.74	−2.00	1256.45	−2.96	1244.74	−2.00	770.36	450	A^5	**1220.36***	5.47
R209B	1418.51	−2.37	1431.37	−3.30	1394.74	−0.66	1394.74	−0.66	935.65	450	A^5	**1385.65***	7.14
R210B	1529.04	−2.23	1516.66	−1.40	1525.28	−1.97	1516.66	−1.40	1045.75	450	A^5	**1495.75***	6.93
R211B	1268.14	−3.90	1255.06	−2.83	1253.08	−2.66	1219.93	0.00	770.56	450	A^5	**1219.93**	7.45
C201B	1816.14	0.25	1820.64	0.00	1816.14	0.25	1820.64	0.00	740.64	1080	A^4B^1	1820.64	3.11
C202B	1768.51	0.09	1795.40	−1.43	1768.51	0.09	1768.51	0.09	690.10	1080	$A^2B^1C^1$	1770.10	4.58
C203B	1744.28	−0.61	1733.63	0.00	1734.82	−0.07	1733.63	0.00	653.63	1080	$A^2B^1C^1$	**1733.63**	3.19
C204B	1736.09	−3.31	1708.69	−1.68	1716.18	−2.13	1680.46	0.00	680.46	1000	A^5	**1680.46**	3.17
C205B	1747.68	0.50	1782.74	−1.49	1747.68	0.50	1778.30	−1.24	716.54	1040	A^1B^3	1756.54	5.21
C206B	1756.93	0.92	1772.87	0.02	1756.01	0.97	1767.70	0.31	733.17	1040	A^1B^3	1773.17	3.46
C207B	1732.20	−0.16	1729.49	−0.01	1729.39	0.00	1729.49	−0.01	689.39	1040	A^1B^3	**1729.39**	2.97
C208B	1730.72	−0.38	1724.20	0.00	1723.20	0.06	1724.20	0.00	684.20	1040	A^1B^3	1724.20	3.13
RC201B	2231.69	0.19	2343.79	−4.83	2230.54	0.24	2329.59	−4.19	1615.90	620	$A^4B^4C^2$	2235.90	4.17
RC202B	2002.62	0.96	2091.53	−3.44	2022.54	−0.03	2057.66	−1.76	1392.00	630	$A^3B^3C^3$	**2022.00***	5.47
RC203B	1843.72	−0.18	1852.74	−0.67	1841.26	−0.05	1824.54	0.86	1190.40	650	B^3C^4	1840.40	5.12
RC204B	1611.28	−3.57	1565.31	−0.62	1575.18	−1.25	1555.75	−0.01	885.74	670	$B^1C^4D^1$	**1555.74***	4.98
RC205B	2195.62	−1.23	2195.75	−1.23	2166.62	0.11	2174.74	−0.26	1529.00	640	$A^2B^2C^4$	2169.00	6.47
RC206B	1887.23	0.60	1923.56	−1.31	1893.13	0.29	1883.08	0.82	1218.70	680	$B^5C^1D^1$	1898.70	4.14
RC207B	1780.72	−2.93	1745.85	−0.92	1743.23	−0.76	1714.14	0.92	1080.00	650	B^3C^4	1730.00	5.14
RC208B	1557.74	−4.50	1488.19	0.16	1526.78	−2.42	1483.20	0.50	830.64	660	C^6	1490.64	4.43

Table A.3 Results for FT for cost structure C

Instance set	ReVNTS		MDA		AMP		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101C	2134.90	0.11	2199.78	−2.93	2134.90	0.11	2199.79	−2.93	1840.20	297	$A^1B^2C^9D^6$	2137.20	3.14
R102C	1913.37	0.08	1925.55	−0.56	1913.37	0.08	1925.56	−0.56	1599.87	315	$A^2B^3C^4D^7E^1$	1914.87	6.21
R103C	1633.62	−0.77	1609.94	0.69	1631.47	−0.63	1615.38	0.36	1310.20	311	$A^1C^4D^8E^1$	1621.20	3.24
R104C	1382.82	−0.52	1370.84	0.35	1377.81	−0.16	1363.26	0.90	1025.60	350	D^8E^3	1375.60	4.47
R105C	1729.57	−0.16	1722.05	0.28	1729.57	−0.16	1722.05	0.28	1419.84	307	$B^5C^3D^7E^1$	1726.84	3.17
R106C	1607.96	0.15	1602.87	0.47	1607.96	0.15	1599.04	0.71	1285.40	325	$A^1C^5D^6E^2$	1610.40	4.08
R107C	1455.09	−0.05	1456.02	−0.12	1452.52	0.12	1442.97	0.78	1126.30	328	$C^2D^8E^2$	1454.30	3.51
R108C	1331.54	−0.12	1336.28	−0.48	1330.28	−0.03	1321.68	0.62	979.92	350	D^6E^4	1329.92	5.33
R109C	1525.65	−0.35	1507.77	0.83	1519.37	0.07	1505.59	0.97	1201.40	319	$A^1B^1C^4D^6E^2$	1520.40	4.73
R110C	1463.91	−0.50	1446.41	0.70	1457.43	−0.06	1443.92	0.87	1125.60	331	$C^4D^6E^2$	1456.60	5.46
R111C	1451.92	−1.10	1447.88	−0.82	1443.34	−0.5	1423.47	0.88	1103.10	333	$B^1D^9E^2$	1436.10	6.14
R112C	1355.78	−1.09	1335.41	0.42	1339.44	0.12	1329.07	0.90	988.10	353	C^2D^5E64	1341.10	4.17
C101C	1628.94	0.00	1628.31	0.04	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	1.97
C102C	1610.96	0.00	1610.96	0.00	1610.96	0.00	1610.96	0.00	860.96	750	A^1B^9	**1610.96**	2.53
C103C	1611.14	−0.25	1619.68	−0.78	1607.14	0.00	1607.14	0.00	857.14	750	A^1B^9	**1607.14**	3.79
C104C	1610.07	−0.68	1613.96	−0.92	1598.50	0.04	1599.90	−0.04	869.21	730	A^3B^8	1599.21	2.89
C105C	1628.94	0.00	1628.38	0.03	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	1.97
C106C	1628.94	0.00	1628.94	0.00	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	2.01
C107C	1628.94	0.00	1628.38	0.03	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	1.99
C108C	1622.89	0.13	1622.89	0.13	1622.89	0.13	1622.89	0.13	825.00	800	B^{10}	1625.00	2.45
C109C	1619.02	−0.03	1614.99	0.22	1614.99	0.22	1615.93	0.17	888.61	730	A^3B^8	1618.61	3.54
RC101C	2089.37	0.13	2084.48	0.36	2089.37	0.13	2082.95	0.44	1702.10	390	$B^7C^5D^3$	2092.10	4.54
RC102C	1918.96	−0.58	1895.92	0.63	1906.68	0.06	1895.05	0.67	1541.87	366	$A^1B^4C^7D^2$	1907.87	4.19
RC103C	1674.50	−0.83	1660.62	0.00	1666.24	−0.33	1650.30	0.63	1300.70	360	C^{12}	1660.70	3.56
RC104C	1543.55	−0.19	1537.09	0.23	1540.13	0.03	1526.04	0.95	1159.60	381	$A^1C^5D^5$	1540.60	3.47
RC105C	1972.57	−0.73	1957.52	0.04	1953.99	0.22	1957.14	0.06	1571.30	387	$A^2B^3C^8D^2$	1958.30	4.16
RC106C	1793.12	−0.29	1776.08	0.67	1787.69	0.02	1774.94	0.73	1392.00	396	$A^1B^1C^8D^3$	1788.00	3.49
RC107C	1635.65	−0.95	1614.04	0.39	1622.90	−0.16	1607.11	0.81	1245.30	375	$B^3C^5D^4$	1620.30	3.07
RC108C	1531.69	0.06	1535.14	−0.17	1531.69	0.06	1523.96	0.56	1157.60	375	$B^2C^6D^4$	1532.60	3.56
R201C	1745.39	−0.82	1729.92	0.07	1728.42	0.16	1716.02	0.88	1461.20	270	A^6	1731.20	6.78
R202C	1537.33	−0.50	1537.35	−0.50	1527.92	0.12	1515.96	0.90	1304.70	225	A^5	1529.70	8.14
R203C	1338.42	−3.22	1308.70	−0.92	1311.60	−1.15	1286.35	0.80	1071.72	225	A^5	1296.72	6.50
R204C	1080.66	−2.64	1062.46	−0.91	1085.71	−3.12	1050.95	0.19	802.90	225	A^5	1052.90	7.89
R205C	1350.12	−2.66	1311.84	0.26	1335.07	−1.51	1309.27	0.45	1090.20	225	A^5	1315.20	6.71
R206C	1254.67	−2.26	1251.51	−2.00	1239.70	−1.04	1216.35	0.86	1001.93	225	A^5	1226.93	6.59
R207C	1186.05	−5.38	1149.23	−2.11	1139.61	−1.25	1120.08	0.48	900.50	225	A^5	1125.50	6.98
R208C	1022.31	−2.44	1009.26	−1.13	1022.11	−2.42	992.12	0.59	772.97	225	A^5	997.97	5.87
R209C	1233.07	−5.91	1178.45	−1.21	1171.41	−0.61	1155.79	0.73	939.31	250	A^4B^1	1164.31	7.14
R210C	1284.72	−1.18	1289.35	−1.55	1281.08	−0.90	1257.89	0.93	1019.70	250	A^4B^1	1269.70	6.14
R211C	1061.70	−6.64	1013.84	−1.83	1028.08	−3.26	994.93	0.07	770.58	225	A^5	995.58	6.17
C201C	1269.41	−1.47	1269.41	−1.47	1269.41	−1.47	1269.41	−1.47	650.97	600	A^2C^2	**1250.97***	2.97
C202C	1252.24	−0.92	1242.66	−0.15	1244.54	−0.30	1239.54	0.11	700.86	540	$A^2B^1C^1$	1240.86	3.54
C203C	1228.13	−2.89	1193.63	0.00	1203.42	−0.82	1193.63	0.00	653.63	540	$A^2B^1C^1$	**1193.63**	3.14
C204C	1207.03	−2.59	1176.52	0.00	1188.18	−0.99	1176.52	0.00	636.52	540	$A^2B^1C^1$	**1176.52**	3.67
C205C	1245.51	−0.44	1245.62	−0.45	1239.60	0.04	1238.30	0.15	640.10	600	A^2B^2	1240.10	4.29
C206C	1229.63	0.01	1245.05	−1.25	1229.23	0.00	1238.30	−0.70	629.23	600	A^2C^2	**1229.23**	4.38
C207C	1221.16	−0.97	1215.42	−0.49	1213.07	−0.30	1209.49	−0.01	689.48	520	$A^2B^1C^1$	**1209.48***	3.56
C208C	1210.72	−0.54	1204.20	0.00	1205.18	−0.08	1204.20	0.00	684.20	520	A^1B^3	**1204.20**	3.01
RC201C	1957.60	−2.07	2004.53	−4.52	1915.42	0.13	1996.79	−4.11	1577.90	340	$A^3B^3C^2D^1$	1917.90	4.65
RC202C	1699.48	−1.16	1766.52	−5.15	1677.62	0.14	1732.66	−3.13	1355.00	325	$A^1B^5C^1D^1$	1680.00	6.10
RC203C	1510.13	−0.66	1517.98	−1.19	1504.35	−0.28	1496.11	0.27	1160.20	340	$A^2B^1C^3E^1$	1500.20	6.27
RC204C	1256.91	−2.84	1238.66	−1.35	1241.45	−1.58	1220.75	0.12	887.16	335	$B^1C^4E^1$	1222.16	5.47
RC205C	1901.71	−4.32	1854.22	−1.71	1822.07	0.05	1844.74	−1.19	1453.00	370	$B^2C^4D^1$	1823.00	5.29
RC206C	1598.84	−2.21	1590.22	−1.66	1586.61	−1.43	1553.65	0.68	1224.30	340	$B^5C^1E^1$	1564.30	4.70
RC207C	1431.65	−3.61	1396.16	−1.05	1406.26	−1.78	1377.52	0.30	1026.71	355	$C^3D^1E^1$	1381.71	5.67
RC208C	1181.47	−2.61	1145.84	0.48	1175.23	−2.07	1140.10	0.98	821.40	330	C^6	1151.40	5.17

Table A.4 Results for FD for cost structure A

Instance set	MDA		BPDRT		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101A	4349.80	−0.75	4342.72	−0.58	4314.36	0.07	1787.52	2530	$A^1B^{10}C^{12}$	4317.52	4.14
R102A	4196.46	−0.12	4189.21	0.06	4166.28	0.60	1641.52	2550	$A^1B^5C^{15}$	4191.52	5.98
R103A	4052.85	−0.38	4051.62	−0.35	4027.36	0.25	1437.62	2600	B^1C^{18}	4037.62	5.21
R104A	3978.48	−0.55	3972.65	−0.40	3936.40	0.51	1366.63	2590	$B^3C^{15}D^1$	3956.63	4.12
R105A	4161.72	−0.36	4152.50	−0.14	4122.50	0.58	1596.63	2550	$A61B^5C^{15}$	4146.63	6.01
R106A	4095.20	−0.58	4085.30	−0.34	4048.59	0.56	1511.54	2560	B^4C^{16}	4071.54	5.12
R107A	4006.61	−0.54	3996.74	−0.29	3970.51	0.37	1395.12	2590	$B^3C^{15}D^1$	3985.12	4.78
R108A	3961.38	−0.55	3949.50	−0.25	3928.12	0.29	1349.52	2590	$B^3C^{15}D^1$	3939.52	6.54
R109A	4048.29	−0.58	4035.89	−0.27	4015.71	0.23	1464.83	2560	B^4C^{16}	4024.83	6.12
R110A	3997.88	−0.58	3991.63	−0.42	3961.68	0.33	1374.97	2600	B^1C^{18}	3974.97	5.21
R111A	4011.63	−0.58	4009.61	−0.53	3964.99	0.59	1398.58	2590	$B^3C^{15}D^1$	3988.58	5.12
R112A	3962.73	−0.83	3954.19	−0.61	3918.88	0.29	1300.19	2630	$C^{17}D^1$	3930.19	4.71
C101A	7098.04	−0.06	7097.93	−0.06	7093.45	0.00	1393.45	5700	A^{19}	**7093.45**	2.47
C102A	7086.11	−0.08	7085.47	−0.07	7080.17	0.00	1380.17	5700	A^{19}	**7080.17**	2.65
C103A	7080.35	−0.02	7080.41	−0.02	7079.21	0.00	1379.21	5700	A^{19}	**7079.21**	2.01
C104A	7076.90	−0.03	7075.06	0.00	7075.06	0.00	1375.06	5700	A^{19}	**7075.06**	1.97
C105A	7096.19	−0.04	7096.22	−0.04	7093.45	0.00	1393.45	5700	A^{19}	**7093.45**	2.65
C106A	7086.91	−0.04	7088.35	−0.06	7083.87	0.00	1383.87	5700	A^{19}	**7083.87**	2.17
C107A	7084.92	0.00	7090.91	−0.09	7084.61	0.00	1384.61	5700	A^{19}	**7084.61**	2.39
C108A	7082.49	−0.04	7081.18	−0.02	7079.66	0.00	1379.66	5700	A^{19}	**7079.66**	1.97
C109A	7078.13	−0.01	7077.68	−0.01	7077.30	0.00	1377.30	5700	A^{19}	**7077.30**	2.19
RC101A	5180.74	−0.14	5168.23	0.10	5150.86	0.44	1843.47	3330	$A^3B^{13}C^4$	5173.47	5.14
RC102A	5029.59	−0.21	5025.22	−0.13	4987.24	0.63	1658.83	3360	$A^6B^6C^7$	5018.83	4.26
RC103A	4895.57	−0.94	4888.53	−0.79	4804.61	0.94	1430.20	3420	$A^2B^6C^8$	4850.20	6.47
RC104A	4760.56	−0.74	4747.38	−0.47	4717.63	0.16	1395.40	3330	$A^3B^2C^8D^1$	4725.40	5.29
RC105A	5060.37	−0.23	5068.54	−0.39	5035.35	0.27	1748.86	3300	$A^5B^8C^6$	5048.86	4.78
RC106A	4997.86	−0.68	4972.11	−0.16	4936.74	0.55	1514.13	3450	B^7C^8	4964.13	5.29
RC107A	4865.76	−0.83	4861.04	−0.73	4788.69	0.76	1435.60	3390	$A^4B^5C^8$	4825.60	4.17
RC108A	4765.37	−0.86	4753.12	−0.60	4708.85	0.34	1334.79	3390	$A^4B^2C^8D^1$	4724.79	4.63
R201A	3484.95	−1.08	3530.24	−2.39	3446.78	0.03	1197.76	2250	A^5	3447.76	6.13
R202A	3335.95	−1.08	3335.61	−1.07	3308.16	−0.24	1050.29	2250	A^5	**3300.29***	7.46
R203A	3173.95	−1.05	3164.03	−0.73	3141.09	0.00	891.09	2250	A^5	**3141.09**	6.14
R204A	3065.15	−1.56	3029.83	−0.39	3018.14	0.00	768.14	2250	A^5	**3018.14**	6.28
R205A	3277.69	−1.82	3261.19	−1.31	3218.97	0.00	968.97	2250	A^5	**3218.97**	6.38
R206A	3173.30	−0.86	3165.85	−0.62	3146.34	0.00	896.34	2250	A^5	**3146.34**	8.14
R207A	3136.47	−1.92	3102.79	−0.83	3077.58	−0.01	827.36	2250	A^5	**3077.36***	6.47
R208A	3050.00	−1.71	3009.13	−0.35	2997.24	0.05	748.67	2250	A^5	2998.70	6.34
R209A	3155.73	−0.90	3155.60	−0.89	3122.42	0.17	877.67	2250	A^5	3127.67	4.99
R210A	3219.23	−1.53	3206.23	−1.12	3174.85	−0.13	920.67	2250	A^5	**3170.67***	5.47
R211A	3055.04	−1.16	3026.02	−0.20	3019.93	0.00	769.93	2250	A^5	**3019.93**	7.93
C201A	5701.45	−0.11	5700.87	−0.10	5695.02	0.00	695.02	5000	A^5	**5695.02**	3.46
C202A	5689.70	−0.08	5689.70	−0.08	5685.24	0.00	685.24	5000	A^5	**5685.24**	3.17
C203A	5685.82	−0.08	5681.55	0.00	5681.55	0.00	681.55	5000	A^5	**5681.55**	4.29
C204A	5690.30	−0.22	5677.69	0.00	5677.66	0.00	677.66	5000	A^5	**5677.66**	3.97
C205A	5691.70	−0.01	5691.70	−0.01	5691.36	0.00	691.36	5000	A^5	**5691.36**	3.46
C206A	5691.70	−0.04	5691.70	−0.04	5689.32	0.00	689.32	5000	A^5	**5689.32**	2.97
C207A	5689.82	−0.04	5692.36	−0.09	5687.35	0.00	687.35	5000	A^5	**5687.35**	4.10
C208A	5686.50	0.00	5689.59	−0.05	5686.50	0.00	686.50	5000	A^5	**5686.50**	3.56
RC201A	4407.68	−0.71	4404.07	−0.62	4374.09	0.06	1476.82	2900	$A^{10}B^4$	4376.82	5.14
RC202A	4277.67	−0.78	4266.96	−0.53	4244.63	0.00	1294.63	2950	A^8B^5	**4244.63**	4.26
RC203A	4204.85	−0.79	4189.94	−0.43	4170.17	0.04	1121.90	3050	$A^6B^3C^2$	4171.90	6.14
RC204A	4109.86	−0.52	4098.34	−0.24	4087.11	0.04	988.55	3100	$A^5B^2C^3$	4088.55	5.47
RC205A	4329.96	−0.84	4304.52	−0.25	4291.93	0.04	1343.73	2950	A^8B^5	4293.73	4.19
RC206A	4272.08	−0.48	4272.82	−0.49	4251.88	0.00	1251.88	3000	A^6B^6	**4251.88**	4.27
RC207A	4232.81	−1.20	4219.52	−0.89	4185.98	−0.08	1182.44	3000	A^6B^6	**4182.44***	5.64
RC208A	4095.71	−0.51	4093.83	−0.46	4075.04	0.00	975.04	3100	$A^4B^4C^2$	**4075.04**	5.31

Table A.5 Results for FD for cost structure B

Instance set	MDA		BPDRT		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101B	2226.94	0.11	–	–	2228.67	0.03	1685.34	544	$B^5C^{13}D^2$	2229.34	4.27
R102B	2071.90	−0.55	–	–	2073.63	−0.64	1488.52	572	$A^1B^2C^{10}D^5$	**2060.52***	3.28
R103B	1857.22	0.25	–	–	1853.66	0.44	1255.88	606	$A^1C^7D^6E^1$	1861.88	5.27
R104B	1707.31	−0.96	–	–	1683.33	0.46	1053.08	638	$A^1C^1D^{10}E^1$	1691.08	5.09
R105B	1995.07	−0.28	–	–	1988.86	0.03	1409.46	580	$C^{10}D^6$	1989.46	3.37
R106B	1903.95	−0.26	–	–	1888.31	0.57	1297.06	602	$C^9D^5E^1$	1899.06	4.19
R107B	1766.18	−0.11	–	–	1753.35	0.61	1152.17	612	$C^4D^8E^1$	1764.17	5.26
R108B	1666.89	−0.68	–	–	1647.88	0.47	1011.70	644	$B^1C^1D^8E^1$	1655.70	3.97
R109B	1833.54	−0.68	–	–	1818.15	0.16	1193.15	628	$B^1C^4D^8E^1$	1821.15	3.99
R110B	1781.74	−0.95	–	–	1758.64	0.36	1159.06	606	C^2D^{11}	1765.06	5.47
R111B	1768.74	−0.66	–	–	1740.86	0.93	1145.20	612	$C^4D^8E^1$	1757.20	5.69
R112B	1675.76	−0.53	–	–	1661.85	0.30	1038.90	628	$C^1D^{10}E^1$	1666.90	5.01
C101B	2340.98	−0.04	–	–	2340.15	0.00	960.15	1380	A^7B^6	**2340.15**	2.98
C102B	2326.53	−0.04	–	–	2325.70	0.00	945.70	1380	A^7B^6	**2325.70**	2.73
C103B	2325.61	−0.04	–	–	2324.60	0.00	944.60	1380	A^7B^6	**2324.60**	3.64
C104B	2318.04	0.00	–	–	2318.04	0.00	938.04	1380	A^7B^6	**2318.04**	2.98
C105B	2344.64	−0.19	–	–	2340.15	0.00	960.15	1380	A^7B^6	**2340.15**	2.71
C106B	2345.85	−0.24	–	–	2340.15	0.00	960.15	1380	A^7B^6	**2340.15**	3.19
C107B	2345.60	−0.23	–	–	2340.15	0.00	960.15	1380	A^7B^6	**2340.15**	2.94
C108B	2340.17	−0.07	–	–	2338.58	0.00	958.58	1380	A^7B^6	**2338.58**	3.88
C109B	2328.55	0.00	–	–	2328.55	0.00	948.55	1380	A^7B^6	**2328.55**	3.12
RC101B	2417.16	−0.40	–	–	2412.71	−0.22	1693.43	714	$A^2B^7C^8$	**2407.43***	3.46
RC102B	2234.47	−0.24	–	–	2213.92	0.68	1515.11	714	$A^2B^7C^5D^2$	2229.11	5.14
RC103B	2025.74	−0.21	–	–	2016.28	0.26	1301.46	720	$B^1C^{10}D^1$	2021.46	3.69
RC104B	1912.65	0.09	–	–	1897.04	0.90	1164.30	750	$B^1C^6D^4$	1914.30	4.57
RC105B	2296.16	−0.26	–	–	2287.51	0.12	1558.24	732	$A^1B^6C^6D^2$	2290.24	5.69
RC106B	2157.84	−0.79	–	–	2140.86	0.00	1408.86	732	$A^1B^2C^8D^2$	**2140.86**	3.12
RC107B	2008.02	−0.73	–	–	1989.34	0.21	1261.48	732	$A^1B^2C^5D^1$	1993.48	2.45
RC108B	1920.91	−1.10	–	–	1898.96	0.06	1150.05	750	$B^1C^6D^4$	1900.05	2.67
R201B	1687.44	−2.47	–	–	1646.78	0.00	1196.78	450	A^5	**1646.78**	6.79
R202B	1527.74	−1.73	–	–	1508.16	−0.42	1051.81	450	A^5	**1501.81***	7.23
R203B	1379.15	−2.84	–	–	1341.09	0.00	891.09	450	A^5	**1341.09**	4.56
R204B	1243.56	−2.09	–	–	1218.14	0.00	768.14	450	A^5	**1218.14**	4.11
R205B	1471.97	−3.60	–	–	1418.97	0.13	970.81	450	A^5	1420.81	6.47
R206B	1400.84	−3.97	–	–	1346.34	0.08	897.41	450	A^5	1347.41	6.99
R207B	1333.53	−4.30	–	–	1277.58	0.08	828.57	450	A^5	1278.57	6.78
R208B	1225.37	−2.23	–	–	1197.24	0.12	748.69	450	A^5	1198.70	5.47
R209B	1370.30	−3.62	–	–	1322.42	0.00	872.42	450	A^5	**1322.42**	5.47
R210B	1418.54	−3.51	–	–	1374.31	−0.28	920.41	450	A^5	**1370.41***	5.93
R211B	1263.72	−3.54	–	–	1219.93	0.05	770.57	450	A^5	1220.57	7.81
C201B	1700.87	−0.35	–	–	1695.02	0.00	695.02	1000	A^5	**1695.02**	2.11
C202B	1687.84	−0.15	–	–	1685.24	0.00	685.24	1000	A^5	**1685.24**	2.33
C203B	1696.25	−0.87	–	–	1681.55	0.00	681.55	1000	A^5	**1681.55**	2.57
C204B	1705.94	−1.69	–	–	1677.66	0.00	677.66	1000	A^5	**1677.66**	3.69
C205B	1711.00	−1.16	–	–	1691.36	0.00	691.36	1000	A^5	**1691.36**	3.07
C206B	1691.70	−0.14	–	–	1689.32	0.00	689.32	1000	A^5	**1689.32**	3.19
C207B	1704.88	−1.04	–	–	1687.35	0.00	687.35	1000	A^5	**1687.35**	3.76
C208B	1689.59	−0.18	–	–	1686.50	0.00	686.50	1000	A^5	**1686.50**	2.41
RC201B	1965.31	−1.05	–	–	1938.36	0.34	1314.97	630	$A^4B^1C^4$	1944.97	6.98
RC202B	1771.87	0.01	–	–	1772.81	−0.05	1121.97	650	$A^1B^1C^5$	**1771.97***	6.47
RC203B	1619.55	−0.44	–	–	1604.04	0.52	962.44	650	$A^1B^1C^5$	1612.44	6.15
RC204B	1501.10	−0.79	–	–	1490.25	−0.07	829.27	660	C^6	**1489.27***	3.47
RC205B	1853.58	−0.79	–	–	1832.53	0.35	1208.99	630	$A^1B^7C^1$	1838.99	3.98
RC206B	1761.49	−1.50	–	–	1725.44	0.57	1085.40	650	$A^3B^1C^3D^1$	1735.40	4.54
RC207B	1666.03	−0.51	–	–	1646.37	0.68	1007.60	650	B^3C^4	1657.60	5.01
RC208B	1494.11	−0.62	–	–	1483.20	0.11	824.854	660	C^6	1484.85	4.08

Table A.6 Results for FD for cost structure C

Instance set	MDA		BPDRT		UHGS		HEA				
	TC	Dev	TC	Dev	TC	Dev	DC	VC	Mix	TC	Time
R101C	1951.20	−0.36	1951.89	−0.39	1951.20	−0.36	1655.26	289	$A^1B^8C^5D^6$	**1944.26***	4.17
R102C	1770.40	−0.41	1778.29	−0.86	1785.35	−1.26	1474.19	289	$A2C^{11}D^5$	**1763.19***	3.23
R103C	1558.17	−0.40	1555.26	−0.21	1552.34	−0.03	1237.93	314	$A1C^6D^7E^1$	**1551.93***	3.69
R104C	1367.82	−0.62	1372.08	−0.93	1355.15	0.32	1015.45	344	$A1C^1D^5E^4$	1359.45	5.17
R105C	1696.67	−0.47	1698.26	−0.56	1694.56	−0.34	1383.76	305	$B^3C^4D^9$	**1688.76***	4.13
R106C	1589.25	0.03	1590.11	−0.03	1583.17	0.41	1278.70	311	$B^2C^5D^7E1$	1589.70	3.67
R107C	1435.21	−0.67	1439.81	−1.00	1428.08	−0.17	1081.62	344	$A^1C^1D^7E3$	**1425.62***	5.98
R108C	1334.75	−1.04	1334.68	−1.04	1314.88	0.46	976.95	344	$A^1C^1D^5E4$	1320.95	4.78
R109C	1515.22	−0.54	1514.13	−0.47	1506.59	0.03	1185.10	322	$B^1C^1D^{10}E^1$	1507.10	4.11
R110C	1457.42	−0.62	1461.85	−0.92	1443.92	0.32	1126.49	322	$B^1C^1D^{10}E^1$	1448.49	4.78
R111C	1439.43	−0.89	1439.14	−0.87	1420.15	0.46	1088.77	338	$A^1B^1D67E^3$	1426.77	5.14
R112C	1358.17	−2.19	1343.26	−1.07	1327.58	0.11	990.07	339	$C^1D^7E^3$	1329.07	4.67
C101C	1628.94	0.00	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	1.99
C102C	1597.66	0.00	1597.66	0.00	1597.66	0.00	847.66	750	A^1B^9	**1597.66**	2.14
C103C	1596.56	0.00	1596.56	0.00	1596.56	0.00	846.56	750	A^1B^9	**1596.56**	2.65
C104C	1594.06	−0.21	1590.86	−0.01	1590.76	0.00	840.76	750	A^1B^9	**1590.76**	2.11
C105C	1628.94	0.00	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	2.41
C106C	1628.94	0.00	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	1.74
C107C	1628.94	0.00	1628.94	0.00	1628.94	0.00	828.94	800	B^{10}	**1628.94**	2.03
C108C	1622.75	0.00	1622.75	0.00	1622.75	0.00	892.75	730	A^3B^8	**1622.75**	2.56
C109C	1614.99	0.00	1614.99	0.00	1615.93	−0.06	864.99	750	A^1B^9	**1614.99***	2.97
RC101C	2048.44	−0.52	2053.55	−0.77	2043.48	−0.28	1656.85	381	$A^1B^6C^8D^1$	**2037.85***	4.16
RC102C	1860.48	−0.68	1872.49	−1.33	1847.92	0.00	1481.92	366	$A^1B^5C^5D^3$	**1847.92**	4.03
RC103C	1660.81	−0.88	1663.08	−1.02	1646.35	0.00	1271.35	375	C^8D^3	**1646.35**	4.17
RC104C	1536.24	−0.77	1540.61	−1.06	1522.04	0.16	1134.47	390	C^4D^6	1524.47	5.14
RC105C	1913.09	0.41	1929.89	−0.47	1913.06	0.41	1533.90	387	$A^2B^3C^8D^2$	1920.90	4.57
RC106C	1772.05	−0.22	1776.52	−0.47	1770.95	−0.15	1402.21	366	$A^1B^2C^8D^2$	**1768.21***	3.44
RC107C	1615.74	−0.72	1633.29	−1.81	1607.11	−0.18	1229.26	375	$B^1C^6D^4$	**1604.26***	3.47
RC108C	1527.35	−0.67	1527.87	−0.70	1523.96	−0.44	1121.26	396	$A^1C^4D^6$	**1517.26***	3.64
R201C	1441.46	−0.84	1466.13	−2.56	1443.41	−0.97	1204.50	225	A^5	**1429.50***	4.54
R202C	1298.10	−1.96	1296.78	−1.86	1283.16	−0.79	1048.11	225	A^5	**1273.11***	7.12
R203C	1145.38	−2.62	1127.28	−1.00	1116.09	0.00	891.09	225	A^5	**1116.09**	4.58
R204C	1019.77	−2.68	1000.89	−0.78	993.14	0.00	768.14	225	A^5	**993.14**	6.81
R205C	1222.03	−2.19	1240.74	−3.76	1193.97	0.15	970.81	225	A^5	1195.81	6.21
R206C	1138.26	−1.51	1141.13	−1.76	1121.34	0.00	896.34	225	A^5	**1121.34**	5.14
R207C	1086.42	−3.21	1067.97	−1.46	1052.58	0.00	827.58	225	A^5	**1052.58**	5.23
R208C	976.11	−0.25	979.50	−0.60	969.90	0.39	748.69	225	A^5	973.69	5.47
R209C	1140.96	−4.20	1140.96	−4.20	1097.42	−0.22	869.97	225	A^5	**1094.97***	5.64
R210C	1161.87	−1.43	1170.29	−2.17	1149.85	−0.38	920.48	225	A^5	**1145.48***	6.17
R211C	1015.84	−2.10	1008.54	−1.37	994.93	0.00	769.93	225	A^5	**994.93**	6.17
C201C	1194.33	0.00	1194.33	0.00	1194.33	0.00	694.33	500	A^5	**1194.33**	4.50
C202C	1189.35	−0.35	1185.24	0.00	1185.24	0.00	685.24	500	A^5	**1185.24**	2.36
C203C	1176.25	0.00	1176.25	0.00	1176.25	0.00	656.25	520	A^1B^3	**1176.25**	3.07
C204C	1176.55	−0.10	1176.55	−0.10	1175.37	0.00	675.37	500	A^5	**1175.37**	3.09
C205C	1190.36	0.00	1190.36	0.00	1190.36	0.00	690.36	500	A^5	**1190.36**	4.50
C206C	1188.62	0.00	1188.62	0.00	1188.62	0.00	668.62	520	A^1B^3	**1188.62**	3.99
C207C	1184.88	0.00	1187.71	−0.24	1184.88	0.00	684.88	500	A^5	**1184.88**	3.17
C208C	1187.86	−0.11	1186.50	0.00	1186.50	0.00	686.50	500	A^5	**1186.50**	2.87
RC201C	1632.41	−0.05	1630.53	0.07	1623.36	0.51	1316.63	315	$A^1B^7C^1$	1631.63	6.01
RC202C	1459.84	−0.64	1461.44	−0.75	1447.27	0.23	1110.57	340	$A^1B^3C^4$	1450.57	4.12
RC203C	1295.07	−0.77	1292.92	−0.60	1274.04	0.87	960.20	325	B^3C^4	1285.20	3.67
RC204C	1171.26	−1.15	1162.91	−0.43	1159.00	−0.09	807.94	350	C^2D^3	**1157.94***	5.14
RC205C	1525.28	−0.59	1632.67	−7.67	1512.53	0.25	1196.39	320	$A^1B^4C^3$	1516.39	5.01
RC206C	1425.15	−1.51	1420.89	−1.21	1395.18	0.62	1078.90	325	$A^1B^1C^5$	1403.90	3.27
RC207C	1332.40	−1.13	1328.29	−0.82	1314.44	0.23	987.50	330	C^6	1317.50	5.47
RC208C	1155.02	−1.31	1152.92	−1.12	1140.10	0.00	790.10	350	C^2D^3	**1140.10**	5.99

References

Baldacci, R., Battarra, M., D. Vigo. 2008. Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, E. Wasil, eds. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, 1–25.

Baldacci, R., A. Mingozzi. 2009. A unified exact method for solving different classes of vehicle routing problems. *Math. Programming* **120** 347–380.

Baldacci, R., Toth, P., D. Vigo. 2010. Exact algorithms for routing problems under vehicle capacity constraints. *Annals OR* **175** 213–245.

Belfiore, P., H. T. Y. Yoshizaki. 2009. Scatter search for a real-life heterogeneous fleet vehicle routing problem with time windows and split deliveries in Brazil. *Eur. J. Oper. Res.* **199** 750–758.

Belfiore, P., H. T. Y. Yoshizaki. 2013. Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries. *Comput. Ind. Eng.* **64** 589–601.

Bettinelli, A., Ceselli, A., G. Righini. 2011. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. *Transportation Res. Part C* **19** 723–740.

Bräysy, O., Dullaert, W., Hasle, G., Mester, D., M. Gendreau. 2008. An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle routing problem with time windows. *Transportation Sci.* **42** 371–386.

Bräysy, O., Porkka, P. P., Dullaert, W., Repoussis, P. P., C. D. Tarantilis. 2009. A well scalable metaheuristic for the fleet size and mix vehicle-routing problem with time windows. *Expert System with Applications* **36** 8460–8475.

Clarke, G., J. W. Wright. 1964. Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* **12** 568–581.

Cordeau, J.-F., Laporte, G., A. Mercier. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *J. Oper. Res. Soc.* **52** 928–936.

Dell'Amico, M., Monaci, M., Pagani, C., D. Vigo. 2007. Heuristic approaches for the fleet size and mix vehicle routing problem with time windows. *Transportation Sci.* **41** 516–526.

Demir, E., Bektaş, T., G. Laporte. 2012. An adaptive large neighborhood search heuristic for the Pollution-Routing Problem. *Eur. J. Oper. Res.* **223** 346–359.

Dondo, R., J. Cerdá. 2007. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *Eur. J. Oper. Res.* **176** 1478–1507.

Dullaert, W., Janssens, G. K., Sörensen, K., B. Vernimmen. 2002. New heuristics for the fleet size and mix vehicle routing problem with time windows. *J. Oper. Res. Soc.* **53** 1232–1238.

Golden, B. L., Assad, A. A., Levy, L., F. Gheysens. 1984. The fleet size and mix vehicle routing problem. *Comput. Oper. Res.* **11** 49–66.

Hoff, A., Andersson, H., Christiansen, M., Hasle, G., A. Løkketangen. 2010. Industrial aspects and literature survey: fleet composition and routing. *Comput. Oper. Res.* **37** 2041–2061.

Kritikos, M.N., G. Ioannou. 2013. The heterogeneous fleet vehicle routing problem with overloads and time windows. *International J. Production Economics* **144** 68–75.

Liu, F. H., S. Y. Shen. 1999a. A method for vehicle routing problem with multiple vehicle types and times windows. *Proc. Nat. Sci. Counc. Part A Phys. Sci. Eng.* **23** 526–536.

Liu, F. H., S. Y. Shen. 1999b. The fleet size and mix vehicle routing problem with time windows. *J. Oper. Res. Soc.* **50** 721–732.

Nagata, Y., Bräysy, O., W. Dullaert. 2010. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Comput. Oper. Res.* **37** 724–737.

Paraskevopoulos, D. C., Repoussis, P. P., Tarantilis, C. D., Ioannou, G., G. P. Prastacos. 2008. A reactive variable neighbourhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *J. Heuristics* **14** 425–455.

Pisinger, D., S. Ropke. 2007. A general heuristic for vehicle routing problems. *Comput. Oper. Res.* **34** 2403–2435.

Prins, C. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **31** 1985–2002.

Prins, C. 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. *Engineering Applications of Artificial Intelligence* **22** 916–928.

Repoussis, P. P., C. D. Tarantilis. 2010. Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming. *Transportation Res. Part C* **18** 695–712.

Ropke, S., D. Pisinger. 2006a. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Sci.* **40** 455–472.

Ropke, S., D. Pisinger. 2006b. A unified heuristic for a large class of vehicle routing problems with backhauls. *Eur. J. Oper. Res.* **171** 750–775.

Solomon, M. M. 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper. Res.* **35** 254–265.

Taillard, ÉD. 1999. A heuristic column generation method for the heterogeneous fleet vehicle routing problem. *Rairo* **33** 1–14.

Toth, P., D. Vigo. 2002. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Appl. Math.* **123** 487–512.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., W. Rei. 2012. A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Oper. Res.* **60** 611–624.

Vidal, T., Crainic, T. G., Gendreau, M., C. Prins. 2013. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows, *Comput. Oper. Res.* **40** 475–489.

Vidal, T., Crainic, T. G., Gendreau, M., C. Prins. 2014. A unified solution framework for multi-attribute vehicle routing problems. *Eur. J. Oper. Res.* **234** 658–673.