Transportation Research Part E 54 (2013) 67–78

Contents lists available at SciVerse ScienceDirect

Transportation Research Part E

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

A hybrid population heuristic for the heterogeneous vehicle routing problems

Shuguang Liu

School of Business, State University of New York at New Paltz, 1 Hawk Drive, New Paltz, NY 12561, United States

ARTICLE INFO

Article history:
Received 25 September 2012
Received in revised form 29 January 2013
Accepted 16 March 2013

Keywords:
Population heuristic
Local search
Chromosome
Heterogeneous vehicle routing problem

ABSTRACT

The heterogeneous vehicle routing problem (HVRP) plays an important role in supply chain logistics. Two variants of HVRP are treated in this paper: one with fixed and variable costs (HVRPFD), and the other with only variable cost (HVRPD). A hybrid population heuristic that is able to solve both variants is proposed, in which a population of solutions are progressively evolved by crossovers and local searches. Computational results on a set of eight benchmark test problems from literature show that the proposed heuristic produces excellent solutions in short computing times.

1. Introduction

We propose a genetic algorithm based population heuristic for a variant of the capacitated vehicle routing problem (CVRP): the heterogeneous vehicle routing problem (HVRP). This problem arises in the practical contexts where the strategic decision of vehicle fleet composition has been made, and one has to make the best possible use of a given fleet. Formally, the problem can be defined as follows. Let the set of nodes of a given complete network be {0, 1, ..., N}, where N is a positive integer. Node 0 denotes the depot and $\mathcal{N} := \{1, 2, \ldots, N\}$ is the set of customers. Distances $d_{i,j}$ on the arcs between any two nodes i and j are known, symmetric and satisfy the triangle inequality: $d_{i,j} < d_{i,l} + d_{l,j}$, with l an additional node.

A fleet of heterogeneous vehicles located at the depot delivers order quantities $q_i,\ i \in \mathcal{N}$ of goods to the customers. Let the set of different vehicle types be $\mathcal{K} := \{1, 2, \ldots, K\}$, where K is a positive integer. The number of available vehicles for each vehicle type $k \in \mathcal{K}$ is fixed at a constant n_k. A vehicle of type k has a carrying capacity Q_k, a fixed cost f_k and a variable cost per unit distance v_k. It is assumed that between any two vehicle types u and v, we have $f_u < f_v$ if $Q_u < Q_v$. A distance $d_{i,j}$ travelled by a vehicle of type k implies a variable cost $v_k d_{i,j}$, and the use of a vehicle of type k incurs a fixed cost f_k.

A solution to the problem is a set of routes and vehicle assignment. Denoted by $\mathcal{R} := \{1, 2, \ldots, R\}$ the set of routes. Let $x_{r,k} = 1$ if a vehicle of type k is assigned to a route $r \in \mathcal{R}$. Each route r has a total travel distance L_r. The total cost function is of two components: the fixed vehicle cost $\sum f_k x_{r,k}$ and the variable traveling cost $\sum v_k L_r x_{r,k}$.

The HVRP studied in this paper consists of determining the number and type of vehicles to use and the route of each vehicle so that the total cost of delivering is minimized while each route starts and ends at the depot, each customer of $\mathcal{N}$ is visited exactly once, customer demands are satisfied, vehicle capacity is not violated, and the number of vehicles used does not exceed the given constant. Formally, the constraints are defined as follows:

E-mail address: liush@newpaltz.edu

1366-5545/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.tre.2013.03.010

$$\sum_r x_{r,k}\delta_{i,r} = 1 \quad \forall i \tag{1}$$

$$\sum_k x_{r,k} \leqslant 1 \quad \forall r \tag{2}$$

$$\sum_r x_{r,k} \leqslant n_k \quad \forall k \tag{3}$$

where $\delta_{i,r}$ is an indicator of value 1 if customer i is in route r, 0 otherwise.

A recent survey on routing heterogeneous fleet by Baldacci et al. (2008) classifies the HVRP into two variants: HVRP with fixed costs and vehicle depent routing costs (HVRPFD), and HVRP with vehicle dependent routing costs (HVRPD). The latter does not consider the fixed costs associated with using different vehicles.

Thus, the objective function for HVRPFD is

$$min\sum_{r,k}(v_kL_r + f_k)x_{r,k} \tag{4}$$

And the objective function for HVRPD is

$$min\sum_{r,k}v_kL_rx_{r,k} \tag{5}$$

This problem is NP-hard, as it includes the capacitated vehicle routing problem in which there is an unlimited number of identical vehicles. In this paper we propose a population based heuristic for the HVRPD and HVRPFD problems, in which a population of initial solutions are continually improved. Comparisons with known best solutions on the eight benchmark problems from Taillard (1999) whose sizes vary from 50 to 100 customers indicate that our population heuristic produces seven new best solutions for HVRPFD and reproduces four best-known solutions for HVRPD. Although it is premature to claim on the performance of the population heuristic on real-life problems of even larger sizes, our results clearly show that the population heuristic is attractive for the HVRP.

The remainder of this paper is organized as follows. We survey the literature in Section 2 and outlines the population heuristic we have developed for the HVRP in Section 3. Section 4 reports our computational results. The paper concludes in Section 5.

2. Literature review

The HVRP studied in this paper has received relatively less attention than some of other variants of CVRP. The HVRP was introduced by Taillard (1999), and later studied by Prins (2002), Tarantilis et al. (2003), Tarantilis et al. (2004), Gencer et al. (2006), Li et al. (2007), Prins (2009), Baldacci and Mingozzi (2009), Li et al. (2010), and more recently by Penna et al. (2011) and Subramanian et al. (2012).

Taillard (1999) designed a heuristic column generation (HCG) procedure for the HVRPD. Tabu search is used to generate a set of initial solutions to the homogeneous VRP. Then, a set partitioning problem – where each column is a route from the initial solution – is solved to obtain the final solution. In the article, Taillard (1999) also presented a set of benchmark problems to assess the quality of solutions produced by different heuristics. The data of all instances can be found at the Internet address http://mistic.heig-vd.ch/taillard/problemes.dir/vrp.dir/vrp.html. Prins (2002) proposed a merger heuristic that revised the Clark-Wright savings algorithm for a special case of HVRPD, where the variable cost is vehicle independent (i.e., $v_k = 1$, $\forall k \in \Psi$, i.e., only traveling time/distance is considered). Tarantilis et al. (2003), Tarantilis et al. (2004) developed a listed based threshold accepting (LBTA) algorithm and a backtracking adaptive threshold accepting (BATA) algorithm. Computationally, BATA obtains better solutions but with longer computing time than LBTA. Li et al. (2007) applied a record-to-record algorithm to the same problem and achieved the best results on the benchmark problems in the literature so far. Prins (2009) designed two memetic algorithms for the HVRPD, where a giant tour is splitted into feasible trips by a pseudo-polynomial procedure in $O(mKN^K)$ time, m being the number of arcs in the graph. No exact algorithm has been developed for the HVRPD. All of the algorithms proposed have been tested on the Taillard (1999) benchmark instances. The record-to-record algorithm showed best performance, while the memetic algorithms are a close second.

Gencer et al. (2006) proposed a passenger pickup algorithm (PPA) for the HVRPFD. PPA is a cluster first – route second heuristic, in which clusters are formed and then a sequence is determined within each cluster by means of solving traveling salesman problems. Baldacci and Mingozzi (2009) presented an exact algorithm for the HVRPFD based on the set partitioning formulation and used three types of bounding procedures based on the LP-relaxation and the Lagrangean relaxation. Another study by Li et al. (2010) presents a multistart adaptive memory programming metaheuristic (MAMP) for HVRPFD, in which several provisional solutions are constructed at each iteration and are improved by a modified tabu search. When tested on the Taillard (1999) instances, the exact algorithm by Baldacci and Mingozzi (2009) provides the best solution quality.

Penna et al. (2011) and Subramanian et al. (2012) studied both HVRPD and HVRPFD. Their approaches are based on iterated local search. The former utilizes random neighborhood ordering (RVND) in the search phase, while the latter uses both

Table 1
A summary of the related work on HVRP.

Publications	Methodology
HVRPD	
Taillard (1999)	Column generation based Tabu search
Tarantilis et al. (2003) and Tarantilis et al. (2004)	Threshold accepting procedures
Li et al. (2007)	Record-to-record travel algorithm
Prins (2009)	Memetic algorithms
Penna et al. (2012)	Iterated local search
Subramanian et al. (2012)	Iterated local search with set partitioning
HVRPFD	
Gencer et al. (2006)	Passenger pickup algorithm
Baldacci and Mingozzi (2009)	Set partitioning based algorithm with Lagrangian relaxation
Li et al. (2010)	Multi-start adaptive memory procedure
Penna et al. (2012)	Iterated local search
Subramanian et al. (2012)	Iterated local search with set partitioning

RVND and a set partitioning formulation. These methods have been tested on the benchmark instances and proved to be effective. A summary of above related research is presented in Table 1.

HVRP contains as a special case the fleet size and mix vehicle routing problem (FSMVRP), in which the available number of vehicles of each type is unlimited $n_k = \infty, \forall k \in \Psi$. FSMVRP was introduced by Lenstra and Rinnooy Kan (1981), who proved that the fixed cost version of the problem is NP-hard. Classical heuristics for the problem have been designed by Golden et al. (1984), Desrochers and Verhoog (1991), Salhi and Rand (1993), and Osman and Salhi (1996). Several mathematical programming based methods have been developed by Taillard (1999), Renaud and Boctor (2002), Yaman (2006), and Choi and Tcha (2007). Meta-heuristics, such as tabu search algorithms and evolutionary algorithms, have been applied tot he problem. Readers interested in tabu search are directed to Osman and Salhi (1996), Gendreau et al. (1999), Wassan and Osman (2002), and Brandão (2009), while for evolutionary algorithms see the work by Ochi et al. (1998), Lima et al. (2004), and Liu et al. (2009).

In this paper, we design a population heuristic to tackle both HVRPD and HVRPFD problems. Population heuristics are population based probabilistic search and optimization algorithms. Their framework follows the principles of Darwinian evolution. A population heuristic simulates the evolution processes by starting with an initial population of candidate solutions to the optimization problem. “Individuals” in the population are encoded as chromosomes, and the chromosomes can be decoded to solutions to the optimization problem. Each individual or solution has its “fitness” value that is evaluated using an objective function.

An offspring population of individuals is generated through genetic operators such as mutation and crossover. Mutation changes each single parent randomly, and crossover recombines genes from more than one parent into an offspring. Both mutation and crossover introduces diversity in the population. To evolve the population in the direction of optimum, a selection operator is needed which selects highly fit individuals to reproduce for the next generation.

Depending on the manner in which the offsprings are placed, we have generational algorithms (where the offspring replaces the whole population) or incremental algorithm (where the offspring replaces less fit individuals). The process is repeated until a termination condition has been reached. Common terminating conditions are a fixed number of generations reached; an allocated computation time reached; and the successive iterations no longer produce better results etc.

In recent year population heuristics have been applied with great success to the VRP and its variants. In the case of capacitated VRP, the hybrid genetic algorithm of Prins (2004) appears to be among the best. In recent development, the simple mutation has been replaced by problem specific local searches to improve the fitness of an offspring. See, for example, Prins (2004), Sorensen and Sevaux (2006), Wang et al. (2008), Liu et al. (2009).

3. A population heuristic for the HVRP

In this section, we will describe the elements of the population heuristic we have developed for the HVRPD and HVRPFD problems. The main elements to be discussed include:

- Solution representation;
- Fitness evaluation;
- Population initialization;
- Selecting parents;
- Crossover;
- Local search as mutation;
- Population placement.

A flow chart for our heuristic is presented in Fig. 1.

3.1. Coding and decoding of chromosomes

A chromosome represents a permutation of the N customer nodes. Each chromosome is an individual in the population and can be evaluated to a solution to the problem. The implementation and evaluation of the objective function is an important factor in the speed and efficiency of the algorithm.

To decode a given chromosome X as a permutation of $\mathcal{N}$ to a solution is to partition customers in X into routes and assigning vehicles to the routes so that each route is served by a vehicle, the total demand of each route does not exceed the capacity of the vehicle assigned, and the total cost required to deliver all customer orders is minimized.

When the available number of vehicles of each type is unlimited, i.e., $n_k = \infty,\ \forall k \in \mathcal{K}$, the partitioning of the a chromosome can be done in strongly polynomial time. In fact, one simply needs to apply a straightforward extension of the known Optimal Partitioning Procedure (OPP) (Beasley, 1983; Altinkemer and Gavish, 1987; Li and Simchi-Levi, 1990). The OPP deals with a homogeneous fleet and needs to be extended to consider a heterogenous fleet, which has been done in Liu et al. (2009). For the completeness of the paper, we present this Extended Optimal Partitioning (EOP) procedure below.

Construct an acyclic graph as follows: let $G(X)$ be a directed acyclic graph with vertex $V(G) = \{i|0 \leqslant i \leqslant N\}$, and $E(G)$ be the set of directed arcs on $G(X)$. Each arc (i, j) has a vehicle of type k. $(i, j) \in E(G)$ iff $\sum_{m=i+1}^{j} d_m \leqslant Q_k$, where k is the vehicle type with the smallest capacity Q_k that meet the above condition. Hence, each arc (i, j) represents a feasible route, where the vehicle for route (i, j) departs from node 0 (depot) and visits nodes $i + 1, i + 2, i + 3, \ldots, j - 1$, and j, consecutively. The total load for route (i, j) is equal to $\sum_{m=i+1}^{j} q_m$. For HVRPFD the cost of arc (i, j) is equal to the variable cost plus the fixed cost, that is $c_{i,j} = v_t(d_{0,i+1} + \sum_{h=i+1}^{j-1} d_{h,h+1} + d_{j,0}) + f_k$; for HVRPD the last term f_k is dropped.

Proposition 3.1. *The shortest path on G(X) defines the optimal partition of the chromosome X and the optimal partitioning can be found in O(N^3) time.*

Proof. The shortest path on $G(X)$ minimizes the total cost while satisfying the demand and vehicle capacity constraints. Since $G(X)$ is acyclic and contains at most $O(N^2)$ arcs and $N + 1$ nodes, the shortest path on $G(X)$ can be determined in $O(N^3)$ using Bellman's algorithm. □

The above extended OPP ensures that the capacity constraints of heterogenous vehicles are satisfied. However, since it does not take into account of the number of available vehicles of each type $n_k\ \forall k \in \Psi$, the resulted routes may use more vehicles of a certain type than that available.

One way to ensure that constraint on the available number of vehicles is to employ a resource constrained shortest path algorithm as have been done by Prins (2009), where the resource being the number of vehicles and the algorithm running in pseudo-polynomial time.

In this paper, we adopt a simple, yet efficient approach – where a penalty will be added where excessive use of vehicles will be penalized (i.e., a penalty will be added to the cost function whenever $\sum x_{k,r} > n_k,\ \forall k \in \Psi$). We determine the penalty cost using the larger cost between (a) the upper bound of total variable cost and (b) the highest vehicle fixed cost, that is $p = \max\{2\max_k v_k \sum_i d_{0,i}, \max_k f_k\}$. The objective value with penalty cost is $\sum_{r,k} f_k x_{r,k} + \sum_{r,k} v_k x_{r,k} L_r + p \sum_k \max(\sum_r x_{r,k} - n_k, 0)$ for HVRPFD, and $\sum_{r,k} v_k x_{r,k} L_r + p \sum_k \max(\sum_r x_{r,k} - n_k, 0)$ for HVRPD.

The evaluation of the penalty cost for all routes takes $O(N)$ time, thus the evaluation of a chromosome for the HVRP takes $O(N^3)$.

Beside the chromosome to represent an individual in the population, the corresponding solution (including routes, and the type of vehicle used for each route) to the chromosome is also stored for later manipulation in the local search procedure. Note that while a chromosome can be decoded into a solution using the procedure outlined above, a solution can be converted into a chromosome by concatenating the routes into a single sequence.

3.2. Selection and crossover

We implement the fitness proportionate selection, also known as roulette-wheel selection, to choose the individuals to be mated. In fitness proportionate selection, as in all selection methods, the fitness function assigns a fitness to possible solutions or chromosomes. This fitness level is used to associate a probability of selection with each individual chromosome. We rank each individual in the population by their total costs. The individual with smallest cost is ranked first, and then each is assigned a fitness according to he following procedure.

Algorithm 1. Fitness evaluation

```
1: Given popSize; // number of individuals in the population
2: $fit[0] = 2.0/popSize$; // the best individual
3: for $i = 1$ to popSize do
4:     $fit[i] = fit[i-1] + 2.0 * (popSize - i - 1)/(popSize * (popSize - 1))$;
5: end for
```

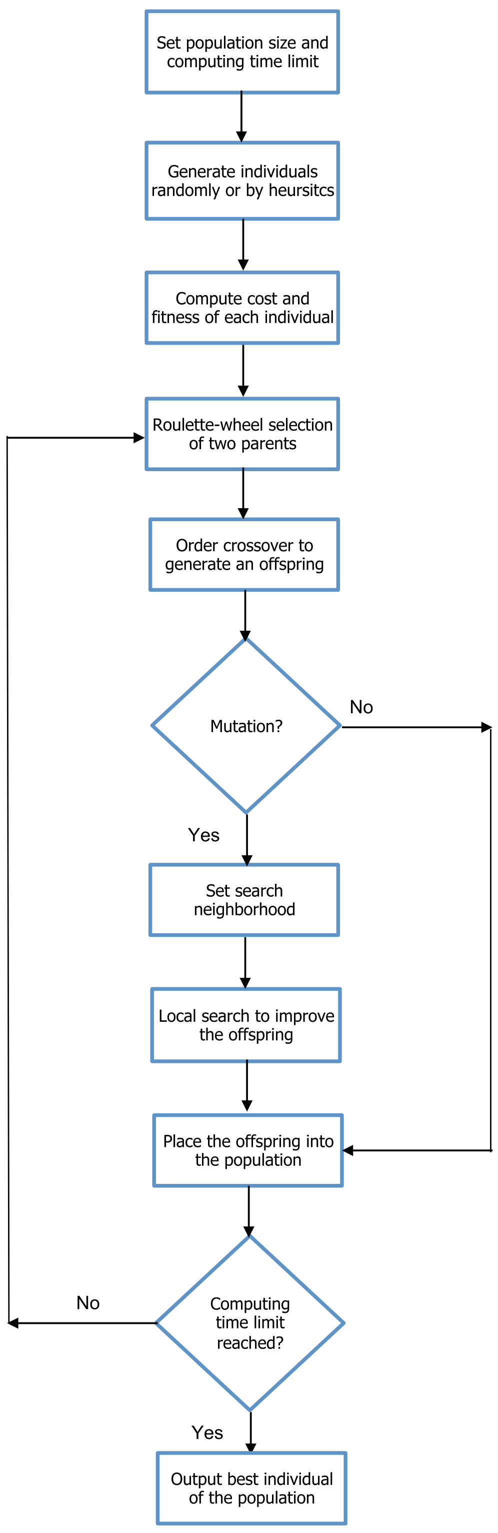
Set population size and computing time limit
Generate individuals randomly or by heursitcs
Compute cost and fitness of each individual
Roulette-wheel selection of two parents
Order crossover to generate an offspring
Mutation?
No
Yes
Set search neighborhood
Local search to improve the offspring
Place the offspring into the population
Computing time limit reached?
No
Yes
Output best individual of the population

Fig. 1. A flow chart of proposed heuristic.

To evolve the population, individuals (children) are generated by mating parents selected using the roulette-wheel selection method. We implemented the classical Order Crossover (OX), where the relative order of the customers in the parent chromosomes are preserved in the child chromosome. For a detail description of OX, see Goldberg (1989). There are other more advanced crossover operators can be used. However, it is our intention to present a relative simple implementation of the population heuristic.

3.3. Local search

Some of the children reproduced will be placed directly into the population, while some will be subject to further manipulation before they are placed. How the placement is done will be discussed in the "population management" section. The manipulation is called mutation, and the probability p_m that a child is to be mutated is mutation probability. Mutation operators change the sequence that the customers are visited. For example insert a customer to a new location, or swap between two customers. Other than these simple operators, we implement some local search operations for the HVRP, which are based on the classical edge exchange schemes for the CVRP. These operations can be single-route and/or multi-route. For details please refer to Liu et al. (2009).

For each child that are mutated, an edge is chosen and the local search operations are performed in a fixed order until an improvement is found. Then the improving operations is applied to change the chromosome. The local search is repeated in the same fixed order, until no improving move related to the edge can be found. A new edge will be picked until all edges in the neighborhood have been tried.

An improving move is an operations that reduces the total cost. To compute the total cost, the vehicle types used must be considered for a multi-route move, since the vehicles used for the routes might alter after the operation. In other words, when different vehicle is used, the fixed cost f_t, and the variable cost v_t for the route will be different.

While we limit our search to the nearest neighborhood, the size of the neighborhood in our implementation is changing as the heuristic progresses (line 5 of the Algorithm 2). The algorithm starts with a small neighborhood and gradually increases the size. To estimate what stage the algorithm is at, we use one of the termination parameters: the total running time of the algorithm. For example, suppose the algorithm terminates in 900 s. Within the first 300 s, the neighborhood searched is one third of the entire neighborhood; the second 300 s, two thirds will be searched; and the last 300 s, the entire neighborhood.

3.4. Population management

The algorithm uses a population of fixed size. To populate, we use both classical heuristics and random sequencing to generate initial chromosomes. Each chromosome is evaluated by the extended OPP algorithm to obtain an solution to the HVRP. The population is sorted in ascending order in terms of solution costs of the individuals.

Our population heuristic is incremental. In our implementation, during each iteration at most one individual is replaced by an offspring. To place an offspring into the population (line 7 in Algorithm 2) and to maintain population diversity, a random solution in the worse half of the population is chosen to be replaced by the offspring. Roulette-wheel selection is used. Then, we require that the offspring does not have the same cost as the existing solutions in the population, other than the one being replaced. If this requirement is not satisfied, the offspring will not be placed into the population.

Algorithm 2. Pseudo code of the population heuristic for the HVRP

```
1. Population initialization $P$
2. **while** Running time $t \leqslant$ timeLimit **do**
3.    $(p_1, p_2) \leftarrow$ Selection $(P)$
4.    $c \leftarrow$ Crossover $(p_1, p_2)$
5.    SizeofNN (t)
6.    With probability $p_m$, $c \leftarrow$ LocalSearch $(c)$
7.    Place $c$ into the population
8. **end while** the best solution $(P)$
```

There are three parameters to be set in the algorithm: population size, mutation probability, and termination criterion. Prins (2004) provided some guidance of setting the first two parameter values when solving the capacitated vehicle routing problem. We follow this guidance in setting a small population size and a low mutation probability, as a mutation rate that is too high may lead to premature convergence of the algorithm. Instead of using convergence criteria such as generations of population reached or generations of no improving solution found, we set a limit of computing time to terminate the algorithm.

For the computational results below, we run this population heuristic ten times for each test problem, set the mutation rate at $p_m = 0.1$, and the algorithm terminates after one thousand seconds. Note that we did not fine-tune these values to find the best possible combination of parameters (which might be problem/instance specific). Instead, we use the same set of values to all benchmark instances. We intend to illustrate the effectiveness of the algorithm without too much fine-tuning of the parameters, compared with other heuristics.

4. Computational results

The population heuristic just described was tested on the eight benchmark instances proposed by Taillard (1999), who adapted the CVRP instances developed by Golden et al. (1984). These test problems were first designed for the FSMVRP, where the travel costs (variable costs) are the same for all vehicle types and there is no limit on vehicles available. These problem were modified later by Taillard (1999) to include the variable cost and the number of vehicles available for each type. We use the numbering scheme (instance 13 to 20) given by Golden et al. (1984) for these instances. Problem 13 to 16 have 50 customers, Problem 17 and 18 have 75 customers, and Problem 19 and 20 have 100 customers. There are between three and six types of vehicles. The corresponding parameters of these instances can be found in Taillard (1999).

We run the population heuristic on each test problem ten times and terminate when the computation time of one thousand seconds is reached.

4.1. HVRPD

Minimizing total variable costs, from the literature six metaheuristics have been developed and tested on the instances 13–20: the tabu search based heuristic column generation algorithm of Taillard (1999), the threshold accepting algorithms of Tarantilis et al. (2003) and Tarantilis et al. (2004), the Record-to-record method of Li et al. (2007), the memetic algorithm of Prins (2009), and iterated local search based algorithms by Penna et al. (2011) and Subramanian et al. (2012). The comparison of best solutions are given in Table 2. Taillard (1999) reports their best results over five runs, while Tarantilis et al. (2003) and Tarantilis et al. (2004) tested different values of the search parameters and reported the best results and their corresponding parameters for each test instance. Since Tarantilis et al. (2004) gave better solutions than those from Tarantilis et al. (2003), only the results from the former are included in Table 2. Similar to Tarantilis et al., Prins (2009) provided the best solutions found by their memetic algorithms using various settings of parameters. It is worth noting that Li et al. (2007) obtained their solutions from only a single run of their record-to-record algorithm, which is deterministic in nature.

Table 3 gives the relative gap between the solution and the best-known solution. The gap is computed as $\frac{Solution-BestKnownSolution}{BestKnownSolution} \times 100\%$, where Solution represents the objective value of the best solution found by each algorithm, and BestKnownSolution denotes the objective value of the best solution found by all of these five methods.

Tables 2 and 3 show that our algorithm outperforms the heuristic column generation algorithm of Taillard (1999) on all instances, except instance 19, and that our results are better than those from Tarantilis et al. (2004) on all instances, except instances 15 and 16. Our results are on par with those from Penna et al. (2011), Subramanian et al. (2012), Li et al. (2007) and Prins (2009), while the first three have a gap of 0.03%, Prins (2009) 0.06%, and our Algorithm 0.08%. One could conclude that all these algorithms produce good quality solutions.

Tabu search, threshold accepting algorithm, and population based heuristics are probabilistic in the sense that different runs of same set of parameters might yield different solutions. Hence, another way to show their effectiveness is to measure average performance over multiple runs. In the literature, only Taillard (1999) reported their average performance, in terms of average total cost and average time used to achieve those average results. In Table 4 we present the comparison of our algorithm with that of Taillard (1999) on average performance.

Table 4 indicates that our population based algorithm outperforms the HCG of Taillard (1999) in term of average solutions. As for the comparison of computation time, we need to be careful since different computers are used in these studies. The related information about the computers used by authors cited here is shown in Table 5. We consult the Dongarra (2011) tables (from year 2007 to 2011) to estimate the relative performance of different computers used. Last column in the table is the CPU speed in millions of floating-point operations per second (Mflop/s). The computation time can be scaled by using the CPU speeds.

We also benchmark our results with those of another population based memetic algorithm from Prins (2009). They run their memetic algorithms many times using various settings of parameters. For each instance, the average computation time of the runs is reported. However, no average results (in terms of total cost) on the problems is reported. Hence, we compare the best solutions in light of average computation time in Table 6.

Results in Table 6 reveals that both population based algorithms provide good solutions in short computation time.

4.2. HVRPFD

4.2.1. Comparison with known results

Here we compare our results with those obtained by the PPA of Gencer et al. (2006), the mathematical programming of Baldacci and Mingozzi (2009), the multistart adaptive memory programming metaheuristic (MAMP) of Li et al. (2010), as well as the iterated local search algorithms by Penna et al. (2011) and Subramanian et al. (2012). Gencer et al. (2006) did not present their machine information or how they conducted their computational experiment (e.g., the number of runs

Table 2
Comparison of best solutions of different algorithms HVRPD.

Problems	n	Best known	Our best	Taillard 99	Tarantilis et al. 04	Li et al. 07	Prins 09	Penna et al. 11	Subramanian et al. 12
13	50	1517.84	1517.8366	1518.05	1519.96	1517.84	1517.84	1517.84	1517.84
14	50	607.53	607.5290	615.64	611.39	607.53	607.53	607.53	607.53
15	50	1015.29	1015.8338	1016.86	1015.29	1015.29	1015.29	1015.29	1015.29
16	50	1144.94	1148.5673	1154.05	1145.52	1144.94	1144.94	1144.94	1144.94
17	75	1061.96	1061.9570	1071.79	1071.01	1061.96	1064.07	1061.96	1061.96
18	75	1823.58	1823.5801	1870.16	1846.35	1823.58	1823.58	1823.58	1823.58
19	100	1117.51	1120.3438	1117.51	1123.83	1120.34	1120.34	1120.34	1120.34
20	100	1534.17	1534.1666	1559.77	1556.35	1534.17	1534.17	1534.17	1534.17
Average		1227.85	1229.0406	1240.48	1236.21	1228.21	1228.47	1228.21	1228.21
Count		8	6	1	1	7	6	7	7

Table 3
The percentage deviations of different algorithms HVRPD.

Problems	n	Best known	Our best (%)	Taillard 99 (%)	Tarantilis et al. 04 (%)	Li et al. 07 (%)	Prins 09 (%)	Penna et al. 11 (%)	Subramanian et al. 12 (%)
13	50	1517.84	0.00	0.01	0.14	0.00	0.00	0.00	0.00
14	50	607.53	0.00	1.33	0.64	0.00	0.00	0.00	0.00
15	50	1015.29	0.05	0.15	0.00	0.00	0.00	0.00	0.00
16	50	1144.94	0.32	0.80	0.05	0.00	0.00	0.00	0.00
17	75	1061.96	0.00	0.93	0.85	0.00	0.20	0.00	0.00
18	75	1823.58	0.00	2.55	1.25	0.00	0.00	0.00	0.00
19	100	1117.51	0.25	0.00	0.57	0.25	0.25	0.25	0.25
20	100	1534.17	0.00	1.67	1.45	0.00	0.00	0.00	0.00
Average			0.08	0.93	0.62	0.03	0.06	0.03	0.03

of the algorithm it took to obtain their results). Baldacci and Mingozzi (2009) used an AMD Athlon 64 X2 Dual Core 4200 + processor at 2.6 GHz and with 3 GB of RAM, while Li et al. (2010) used a PC with an Intel CPU at 2.2 GHz.

We report in Table 7 the best results found by these four algorithms. Our best results are the best over ten runs, while MAMP results are from the “best configuration” (Li et al., 2010). Table 8 gives the relative gaps between the best solutions and the best-known solutions. The gap is computed similarly as those in Table 3.

Tables 7 and 8 demonstrate that the best performance is obtained by the set partition based iterated local search algorithm of Subramanian et al. (2012), which found eight best solutions. The second best results are obtained by Penna et al. (2011). The mathematical program of Baldacci and Mingozzi (2009), our population heuristic and the MAMP of Li et al. (2010) are closely behind. Both produced five best solutions and have average gaps around tenth of a percentage. For each instance Li et al. (2010) reported detail solution information (e.g., customer sequence in each route), while Baldacci and Mingozzi (2009) did not. For instance 16, we found a better solution than that by Li et al. (2010). The presented detailed solution for instance 20 in Li et al. (2010) is not complete and omits a route. Thus, we report our HVRPFD solutions to instances 16 and 20 in the Appendix.

For these eight instances, the exact method of Baldacci and Mingozzi (2009) found their solution within computation times ranging from 5.9 s to 6446.8 s, with an average of 259.9 s. Our population heuristic and the MAMP of Li et al.

Table 4
Average results for HVRPD.

Problems	Best known	Our average		Taillard 99	
		Cost	Time	Cost	Time
13	1517.84	1534.04	57.42	1536.55	473
14	607.53	609.17	86.98	623.05	575
15	1015.29	1025.14	4.85	1022.05	335
16	1144.94	1153.14	13.51	1159.14	350
17	1061.96	1083.55	115.88	1095.01	2245
18	1823.58	1843.16	97.98	1894.73	2876
19	1117.51	1160.98	77.21	1156.93	5833
20	1534.17	1560.62	115.49	1592.16	3402
Average	1227.85	1246.22	71.17	1259.95	2011.13
Deviation (%)		1.50		2.61	

Table 5
Performance of the computers used.

	Model	CPU	Mflop/s
Taillard 99	Sun Sparc	50M	27
Tarantilis et al. 04	Pentium II	400M	262
Gencer et al. 06	N/A	N/A	N/A
Li et al. 07	Athlon	1.0G	1168
Prins 09	Intel Pentium 4M	1.8G	1564
Baldacci et al. 09	Athlon Due Core	2.6G	6818
Li et al. 10	Intel	2.2G	1350
Penna et al. 11	Intel Core i7	2.93G	5839
Subramanian et al. 12	Intel Core i7	2.93G	5839
Ours	Intel Pentium 4	3G	1573

Table 6
Comparison between two population algorithms for HVRPD.

Problems	Best known	Our		Prins 09	
		Best cost	Avg. time	Best cost	Avg. time
13	1517.84	1517.8366	57.42	1517.84	33.2
14	607.53	607.5290	86.98	607.53	37.6
15	1015.29	1015.8338	4.85	1015.29	6.6
16	1144.94	1148.5673	13.51	1144.94	7.5
17	1061.96	1061.9570	115.88	1064.07	81.5
18	1823.58	1823.5801	97.98	1823.58	190.6
19	1117.51	1120.3438	77.21	1120.34	177.8
20	1534.17	1534.1666	115.49	1534.17	223.3
Average	1227.85	1229.04	71.17	1228.47	94.76
Deviation (%)		0.08		0.05	

Table 7
Comparison of best solutions of different algorithms HVRPFD.

Problems	n	Best known	Our best	Gencer et al. 06	Baldacci et al. 09	Li et al. 10	Penna et al. 11	Subramanian et al. 12
13	50	3185.09	3185.0887	3242.86	3185.09	3185.09	3185.09	3185.09
14	50	10107.53	10107.5290	10193.83[a]	10107.53	10107.53	10107.53	10107.53
15	50	3065.29	3065.8338	3119.99	3065.29	3065.29	3065.29	3065.29
16	50	3265.41	3268.7012	3392.13	3265.41	3278.96	3265.41	3265.41
17	75	2076.96	2076.9570	2166.41	2076.96	2076.96	2076.96	2076.96
18	75	3743.58	3743.5801	3912.48	3743.58	3743.58	3743.58	3743.58
19	100	10420.34	10420.3438	11480.75	10423.34	10420.34	10420.34	10420.34
20	100	4761.26	4834.1666	5091.05	4806.69	4834.17	4788.49	4761.26
Average		5078.18	5087.7750	5324.94	5084.24	5088.99	5081.59	5078.18
Count		8	5	0	6	6	7	8

[a] Gencer et al. reported 7493.83 due to an erroneous fixed cost used for vehicle type A.

Table 8
Percentage deviations of different algorithms HVRPFD.

Problems	Best known	Our best (%)	Gencer et al. 06 (%)	Baldacci et al. 09 (%)	Li et al. 10 (%)	Penna et al. 11 (%)	Subramanian et al. 12 (%)
13	3185.09	0.00	1.81	0.00	0.00	0.00	0.00
14	10107.53	0.00	0.85	0.00	0.00	0.00	0.00
15	3065.29	0.02	1.78	0.00	0.00	0.00	0.00
16	3265.41	0.10	3.88	0.00	0.41	0.00	0.00
17	2076.96	0.00	4.31	0.00	0.00	0.00	0.00
18	3743.58	0.00	4.51	0.00	0.00	0.00	0.00
19	10420.34	0.00	10.18	0.03	0.00	0.00	0.00
20	4761.26	1.53	6.93	0.95	1.53	0.57	0.00
Average		0.19	4.86	0.12	0.21	0.07	0.00

Table 9
Average results for HVRPFD.

Problems	Best known	Our average		Li et al. (2010)[a]	
		Average cost	Average time	Average cost	Average time
13	3185.09	3196.1229	129.88	3190.13	92
14	10107.53	10109.1673	51.78	10107.78	41
15	3065.29	3073.1183	87.29	3066.01	57
16	3265.41	3351.6000	73.85	3285.33	83
17	2076.96	2098.7298	128.65	2077.03	151
18	3743.58	3761.6809	115.31	3743.89	126
19	10420.34	10786.5480	238.67	10420.34	295
20	4806.69	4864.4005	190.96	4835.36	209
Average	5083.86	5155.1710	127.05	5090.73	131.75
Deviation (%)		1.40		0.14	

[a] Li et al. (2010) reported results from four versions of MAMP algorithm, and we use their best results here.

Table 10
Difference between HVRPFD solutions and HVRPD solutions.

Problems	HVRPFD			HVRPD		
	Variable	Fixed	Total	Variable	Fixed	Total
13	**1525.0887**	**1660**	**3185.0887**	**1517.8366**	**1680**	**3197.8366**
14	607.5290	9500	10107.5290	607.5290	9500	10107.5290
15	1015.8338	2050	3065.8338	1015.8338	2050	3065.8338
16	**1168.7012**	**2100**	**3268.7012**	**1148.5673**	**2200**	**3348.5673**
17	1061.9570	1015	2076.9570	1061.9570	1015	2076.9570
18	1823.5801	1920	3743.5801	1823.5801	1920	3743.5801
19	1120.3438	9300	10420.3438	1120.3438	9300	10420.3438
20	1536.6773	3300	4836.6773	1536.6773	3300	4836.6773
Average			5088.0889			5099.6656

Table 11
Problem 16.

Route	Vehicle	Customer sequence
Solution to problem 16–50 customers		
1	2	42 19 40 41 13
2	3	11 2 20 35 36 3 28 22
3	1	7 43 24
4	2	25 14 23 6
5	3	38 49 9 30 34 50 21 29 16 32
6	2	1 31 26 8 48 27
7	3	5 10 39 33 45 15 44 37 17 12
8	2	18 4 47 46
Total fixed cost: 2100		
Total variable cost: 1168.7012		
Total cost: 3268.7012		

(2010) are probabilistic in nature. Thus computing time and solutions arrived are not the same during each run. In Table 9, we compare the average solutions achieved by our population heuristic and those by MAMP. Li et al. (2010) run four different versions of MAMP with ten trials and report the average values of the best solutions and average CPU time required to find the corresponding solutions. The best results of the four versions are used here to compare with our heuristic. As we run our heuristic ten times, our average solution (time) is the average over these ten runs.

Table 9 shows that the performances of both algorithms are good. Over all of test problems the average percentage deviation of our average solution from the best known solutions is 1.4% in an average computation time of around 127 s. The MAMP performed slightly better with a smaller gap and a little bit longer time. Having a method that provides good solutions within a short time is beneficial for transportation companies who generate day-to-day dispatching schedules.

4.2.2. Solution difference between HVRPD and HVRPFD

It is apparent that the HVRPD routes, which minimize variable cost, do not, in general, utilize the smallest possible number of vehicles. When the objective is to minimize the total variable and fixed cost, sometimes the HVRPD solution may coin-

Table 12
Problem 20.

Route	Vehicle	Customer sequence
Solution to problem 20–100 customers		
1	C	13 87 42 14 44 86 16 61 5 89
2	A	53
3	C	40 21 72 75 56 23 67 39 25 55 4 26
4	C	94 95 59 98 37 100 85 93 99 96 6
5	B	28 76 77 3 79 68 80 12
6	B	33 81 9 35 71 65 66 20 51
7	B	31 10 63 90 32 30 70 1 69 27
8	B	52 7 82 48 47 19 62 88
9	A	54 24 29 34 78 50
10	A	60 84 17 45 8 83 18
11	A	11 64 49 36 46
12	A	58 41 22 74 73
13	A	2 57 15 43 38 91 92 97
Total fixed cost: 3300		
Total variable cost: 1534.1666		
Total cost: 4534.1666		

cide with the HVRPFD solution, and at other times, the HVRPFD improves HVRPD solutions. Table 10 indicates that the improvement happened for two instances: instances 13 and 16. For both instances, the HVRPFD solutions use fewer number of vehicles and thus incur smaller fixed cost, while the variable costs are larger than the HVRPD counterparts.

Another observation is the difference in computation time. Tables 4 and 9 show that the solution time increases for all instances when the fixed cost is considered in the objective, with an average of 71 s for HVRPD and 127 s for HVRPFD.

5. Conclusion

In this paper we designed and implemented a hybrid population heuristic for the HVRPD and HVRPFD problems. The local search moves are applied to a neighborhood with varying size.

The contributions of the paper to the literature that surveyed by Baldacci et al. (2008) are three fold. First, on the Taillard (1999) benchmark problems, our heuristic improved the solutions to the HVRPFD obtained by Gencer et al. (2006) and Li et al. (2010), while provided a competitive performance to that of Penna et al. (2011), Subramanian et al. (2012) and Baldacci and Mingozzi (2009) in a shorter computing time. We report one new solution to instance 16 over that of Li et al. (2010) and give a complete description of solution to instance 20. Second, we exhibit that the population heuristic is as competitive as other approaches currently employed in the literature on HVRPD, such as the mathematical programming based heuristics of Taillard (1999), the threshold accepting algorithms of Tarantilis et al. (2003), Tarantilis et al. (2004), and the record-to-record algorithm of Li et al. (2007). Third, compared with another population based memetic algorithms of Prins (2009), our heuristic is as effective, yet we provide a simpler implementation in terms of the evaluation of chromosomes in polynomial time.

In the future we plan to explore new variants of the HVRP problems, for example, HVRP with time windows and/or with pickup and delivery. Another direction is to design more effective local search moves specific to HVRPFD and HVRPD. Currently our local search moves are applied to both problems. Finally, we plan to further test the effectiveness of our approach by applying the heuristic to more difficult set of test instances with a much larger number of customers.

Acknowledgement

The author thank the Editor-in-Chief and two anonymous reviewers for their valuable comments.

Appendix A

We present the best solutions for HVRPFD to the Taillard (1999) benchmark Problems 16 and 20. In each solution, routes and vehicles used are listed. A sequence of customer locations is given for each route (see Tables 11 and 12).

References

Altinkemer, K., Gavish, B., 1987. Heuristics for unequal weight delivery problems with a fixed error guarantee. Operations Research Letters 6 (4), 149–158.
Baldacci, R., Battarra, M., Vigo, D., 2008. Routing a heterogeneous fleet of vehicles. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, pp. 3–27.
Baldacci, R., Mingozzi, A., 2009. A unified exact method for solving different classes of vehicle routing problems. Mathematical Programming 120 (2), 347–380.
Beasley, J., 1983. Route-first cluster-second methods for vehicle routing. Omega 11 (4), 403–408.

Brandão, J., 2009. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. European Journal of Operational Research 195 (3), 716–728.
Choi, E., Tcha, D.-W., 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. Computers & Operations Research 34, 2080–2095.
Desrochers, M., Verhoog, T.W., 1991. A new heuristic for the fleet size and mix vehicle routing problem. Computers & Operations Research 18, 263–274.
Dongarra, J., 2011. Performance of various computers using standard linear equations software. Tech. Rep. Report CS-89-85, University of Tennessee.
Gencer, C., Top, I., Aydogan, E.K., 2006. A new intuitional algorithm for solving heterogeneous fixed fleet routing problems: passenger pickup algorithm. Applied Mathematics and Computation 181, 1552–1567.
Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D., 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research 26, 1153–1173.
Goldberg, D.E., 1989. Genetic Algorithms in Search Optimisation and Machine Learning. Addison Wesley.
Golden, B., Assad, A., Levy, L., Gheysens, F., 1984. The fleet size and mix vehicle routing problem. Computers & Operations Research 11, 49–66.
Lenstra, J.K., Rinnooy Kan, A.H.G., 1981. Complexity of vehicle routing and scheduling problems. Networks 11, 221–227.
Li, C., Simchi-Levi, D., 1990. Worst-case analysis of heuristics for multidepot capacitated vehicle routing problems. INFORMS Journal on Computing 2 (1), 64.
Li, F., Golden, B., Wasil, E., 2007. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers & Operations Research 34, 2734–2742.
Li, X., Tian, P., Aneja, Y., 2010. An adaptive memory programming metaheuristic for the heteroigeneous fixed fleet vehicle routing problem. Transportation Research Part E: Logistics and Transportation Review 46 (6), 1111–1127.
Lima, C.M.R.R., Goldbarg, M.C., Goldbarg, E.F.G., 2004. A memetic algorithm for the heterogeneous fleet vehicle routing problem. Electronic Notes in Discrete Mathematics 18, 171–176.
Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. Transportation Research Part E: Logistics and Transportation Review 45 (3), 434–445.
Ochi, L.S., Vianna, D.S., Drummond, L.M., Victor, A.O., 1998. A parallel evolutionary algorihtm for the vehicle routing problem with heterogeneous fleet. Future Generation Computer System 14, 285–292.
Osman, I., Salhi, S., 1996. Local search strategies for the vehicle fleet mix problem. In: Rayward-Smith, V.J., Osman, I.H., Reeves, C.R., Smith, G.D. (Eds.), Modern Heuristic Search Methods. Wiley, New York, pp. 131–153.
Penna, P., Subramanian, A., Ochi, L., 2011. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. Journal of Heuristics, 1–32.
Prins, C., 2002. Efficient heuristic for the heterogeneous fleet multitrip vrp with application to a large-scale real case. Journal of Mathematical Modelling and Algorithms 1, 135–150.
Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 1985–2002.
Prins, C., 2009. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Engineering Applications of Artificial Intelligence 22 (6), 916–928.
Renaud, J., Boctor, F.F., 2002. A sweep-based algorithm for the fleet size and mix vehicle routing problem. European Journal of Operational Research 140, 618–628.
Salhi, S., Rand, G.K., 1993. Incorporating vehicle routing into the vehicle fleet composition problem. European Journal of Operational Research 66, 313–330.
Sorensen, K., Sevaux, M., 2006. Mapm: memetic algorithms with population management. Computers & Operations Research 33, 1214–1225.
Subramanian, A., Vaz Penna, P., Uchoa, E., Ochi, L., 2012. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research, 285–295.
Taillard, E.D., 1999. A heuristic column generation method for the heterogeneous fleet vrp. RAIRO 33, 1–34.
Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S., 2003. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Journal of the Operational Research Society 54, 65–71.
Tarantilis, C.D., Kiranoudis, C.T., Vassiliadis, V.S., 2004. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research 152, 148–158.
Wang, X., Gloden, B., Wasil, E., 2008. Using a genetic algorithm to solve the generalized orienteering problem. In: Golden, B., Raghavan, S., Wasil, E. (Eds.), The Vehicle Routing Problem: Latest Advances and New Challenges. Springer, pp. 263–274.
Wassan, N.A., Osman, I.H., 2002. Tabu search variants for the mix fleet vehicle routing problem. Journal of the Operational Research Society 53, 768–782.
Yaman, H., 2006. Formulations and valid inequalities for the heterogeneous vehicle routing problem. Mathematical Programming 106, 365–390.