Transportation Research Part E 45 (2009) 434–445

Contents lists available at ScienceDirect

Transportation Research Part E

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

An effective genetic algorithm for the fleet size and mix vehicle routing problems

Shuguang Liu [a,*], Weilai Huang [b], Huiming Ma [b]

[a] School of Business, State University of New York at New Paltz, 1 Hawk Drive, New Paltz, NY 12561, United States
[b] School of Management, Huazhong University of Science and Technology, Wuhan, 430074, China

ARTICLE INFO

Article history:
Received 24 March 2008
Received in revised form 11 September 2008
Accepted 12 October 2008

Keywords:
Genetic algorithm
Single parent crossover
Local search
Fleet size and mix vehicle routing problem

ABSTRACT

This paper studies the fleet size and mix vehicle routing problem (FSMVRP), in which the fleet is heterogeneous and its composition to be determined. We design and implement a genetic algorithm (GA) based heuristic. On a set of twenty benchmark problems it reaches the best-known solution 14 times and finds one new best solution. It also provides a competitive performance in terms of average solution.

1. Introduction

The capacitated vehicle routing problem (CVRP) is a problem risen in the fields of transportation, distribution and logistics in which a fleet of delivery vehicles must service known customer demands for a single commodity from a common depot at minimum cost. The vehicles are homogeneous, having the same capacity and costs, while the size of the fleet is unlimited. There are many variants of the CVRP that relax one or both of these conditions.

One variant of the CVRP is the fleet size and mix vehicle routing problem (FSMVRP), where one has to decide how many vehicles of each type to use given a mix of vehicle types differing in capacity and costs. Here the fleet is heterogeneous, however, the available number of vehicles for each type remains unlimited. This problem is similar to the CVRP, but the difference is that FSMVRP has different types of vehicles to choose from, while CVRP has only one type. The decision is on both the fleet composition and the vehicle routing. There are two lines of inquiries on FSMVRP addressing different cost structures. The first cost structure involves vehicle types having a uniform variable cost (normally assumed to be 1) and different fixed costs. Golden et al. (1984) present the first mathematical formulation for this FSMVRP. The second cost structure introduced by Salhi et al. (1992) deals with vehicle types that have no fixed cost but different variable costs. In the following, we refer to these two cases as FSMVRP with fixed cost and FSMVRP without fixed cost, respectively.

Some researchers study both cases, while others focus on only the FSMVRP with fixed cost; no research studies only the FSMVRP without fixed cost. Among those that studied both cost structures are: Taillard (1999), Gendreau et al. (1999), Wassan and Osman (2002), Choi and Tcha (2007) and Brandao (2008). Among those discussing only the problems with fixed cost include: Desrochers and Verhoog (1991), Salhi and Rand (1993), Osman and Salhi (1996), Ochi et al. (1998), Liu and Shen (1999), Renaud and Boctor (2002), Lima et al. (2004) and Yaman (2006).

* Corresponding author.
E-mail addresses: liush@newpaltz.edu (S. Liu), huangwl@mail.hust.edu.cn (W. Huang), mahm@mail.hust.edu.cn (H. Ma).

1366-5545/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.tre.2008.10.003

Another related variant of CVRP is the heterogeneous vehicle routing problem (HVRP). Here not only the vehicles in the fleet have different capacity and costs, but also the number of available vehicles is fixed as a priori. The decision is how to best utilize the existing fleet to sever customer demands. This variant is first studied by Taillard (1999), and later by Tarantilis et al. (2003), Tarantilis et al. (2004), Gencer et al. (2006) and recently by Li et al. (2007). Even though the two cost structures still apply, the literature seems to deal exclusively with the case without fixed cost, with Gencer et al. (2006) being the exception.

A recent survey paper by Baldacci et al. (2008) summarizes these two and other variants of the CVRP involving heterogeneous fleet. Lower bounds and heuristic algorithms in the literature are reviewed. Different computational results from various papers on benchmark instances are compared.

In this paper we apply genetic algorithm to solve the FSMVRP where the fleet size is to be determined, and address both cost structures, with and without fixed cost. Before we introduce our algorithm, we first review the literature related to FSMVRP.

Lenstra and Rinnooy Kan (1981) proved that the FSMVRP with fixed cost is NP-Hard. It is easy to show that the FSMVRP without fixed cost is also NP-Hard. One only needs to reduce the heterogeneous fleet of FSMVRP to homogeneous and obtain a CVRP, which is NP-Hard.

Classical heuristics have been designed to tackle the FSMVRP. Golden et al. (1984) developed several heuristics by revising the saving algorithm of Clarke and Wright (1964), the sweep algorithm of Gillett and Miller (1974) and the generalized assignment of Fisher and Jaikumar (1981). Matching based saving algorithms were proposed by Desrochers and Verhoog (1991), Salhi and Rand (1993) and Osman and Salhi (1996).

These traditional heuristics are outperformed by a new generation of heuristics, including mathematical programming based heuristics and meta-heuristics, particularly tabu search (TS). Mathematical programming based methods have been proposed in several works; among them are Taillard (1999), Renaud and Boctor (2002), Yaman (2006) and Choi and Tcha (2007). Taillard (1999) uses heuristic column generation (HCG). First, tabu search is used to generate a set of good solutions. Then an integer linear program, where each column in the program is a route from the initial solution, is solved to obtain the final solution. Renaud and Boctor (2002) developed a sweep heuristic to find initial routes, and then solved a set partitioning problem to attain solutions to FSMVRP. Some tried to derive upper and lower bounds of the FSMVRP. Yaman (2006) gave several formulations of the FSMVRP with fixed cost, generalized subtour elimination and multistar inequalities. Based on derived valid inequalities, constraint lifting was used to improve the linear programming lower bounds. Choi and Tcha (2007) developed a set covering formulation and solved its linear relaxation by column generation to obtain the bounds. Comparing these methods on the Golden et al. (1984) benchmark instances, the best solutions have been obtained by Choi and Tcha (2007).

Meta-heuristics, such as TS algorithms, have been applied by various authors. Osman and Salhi (1996) developed a short-term memory tabu search using moves in 1-interchange neighborhood. Gendreau et al. (1999) presented a tabu search algorithm that is embedded in an adaptive memory procedure. Their tabu search uses GENIUS (Gendreau et al., 1992) to obtain solutions and a shortest path program to choose vehicle type for each route. Wassan and Osman (2002) developed new variants of a tabu search meta-heuristic. These variants use a mix of different components, including reactive concepts, variable neighborhoods, hashing functions and special data memory structures, similar to adaptive memory procedures. Brandao (2008) developed a deterministic tabu heuristic, which restricts the moves in a nearest neighborhood whose size is determined by the estimated number of customers in a route. GENIUS algorithm and giant tour are used to obtain initial solutions.

Evolutionary algorithms have also been attempted by Ochi et al. (1998) and Lima et al. (2004) on the FSMVRP with fixed cost, but with little success. Ochi et al. (1998) developed a parallel genetic algorithm (GA) hybrid with scatter search. A petal decomposition procedure is designed to build chromosomes. Each chromosome is a set of routes, multiple depots are used as route delimiters, and each route is optimized through GENIUS. A vehicle type is assigned to a customer by choosing the vehicle with the lowest product of its remaining capacity and its fixed cost. Using a crossover extended from the TSP edge recombination crossover (ERX), Ochi et al. (1998) evaluated their GA on the Golden et al. (1984) instances. However, no detailed results are given. A histogram showing the comparison with Taillard (1999) seems to contain some error. For example, their cost for instance 14 is only around 1500 while the best solution obtained by various authors, so far, is around 9119.

Lima et al. (2004) proposed a memetic algorithm for the FSMVRP with fixed cost. A chromosome coding and vehicle assignment similar to that of Ochi et al. (1998) is used. Edge recombination crossover is applied and two local search procedures, GENUIS and 1- and 2-interchange, are used for mutation. The same Golden et al. (1984) instances are solved. The best and average of 10 runs of their algorithm were reported. No comparison with results from other studies is shown.

Among these meta-heuristics designed for solving FSMVRP, better results are obtained by tabu search based algorithms, particularly those developed by Brandao (2008) and Wassan and Osman (2002). And the two GAs byOchi et al. (1998) and Lima et al. (2004) are not as effective. However, in related articles, Prins (2004) applies an evolutionary algorithm to the CVRP and Wang et al. (2008) design a GA to solve the generalized orienteering problem. Both research show the effectiveness of a well-designed GA. This paper will demonstrate that GA could also provide competitive performance in solving the FSMVRP. While the general structure of the genetic algorithm proposed here is similar to those of Ochi et al. (1998), Lima etal. (2004), Prins (2004) and Wang et al. (2008), this paper designs different initial solution procedures, a new chromosome evaluation procedure and some local search moves that are specific to the FSMVRP. Also proposed is a new single parent crossover operator.

The paper is organized as follows. A formal definition of the FSMVRP is given in Section 2. Section 3 presents the genetic algorithm proposed and its important elements, such as, how to represent and evaluate a chromosome, how to construct an initial population and how to reproduce and mutate the chromosomes. Section 4 reports computational results. The paper is concluded in Section 5.

2. The problem

We formally describe the FSMVRP as follows. Let 0 represent the depot, and there are n customer locations numbered $\{1, 2, 3, \ldots, n\}$. A fleet of T types of vehicles are available at the depot. The number of vehicles for each vehicle type is unlimited. How many vehicles of each type to use is one of the decision variables. Each vehicle type t has a capacity Q_t, a fixed cost f_t and a variable cost v_t. It is assumed that between two vehicle types u and v, we have $f_u < f_v$ if $Q_u < Q_v$. For the FSMVRP with fixed cost, we have:

$$v_t = 1, \quad F_t > 0 \ \forall t \tag{1}$$

And for the FSMVRP without fixed cost, we have:

$$v_t > 0, \quad F_t = 0 \ \forall t. \tag{2}$$

Each customer node $i > 0$ has a non-negative demand d_i. The traveling distance between locations i and j is a non-negative $\tau_{i,j}$. These distances are symmetric and satisfy the triangle inequality, i.e., $\tau_{i,j} = \tau_{j,i}$ and $\tau_{i,j} + \tau_{j,k} \geqslant \tau_{i,k}$. The variable cost of traveling from location i to location j is $v_t \tau_{i,j}$.

The FSMVRP consists of determining the composition of vehicles to be used; the route of each vehicle, so that the total cost of delivering to all customers is minimized while each route starts and ends at the depot; each customer is visited exactly once; customer demands are satisfied; and vehicle capacity is not violated.

3. The genetic algorithm based heuristic

In this section, we describe the genetic algorithm and its elements, such as chromosomal representation, initial population, GA operators (such as crossover and mutation) and control parameters for stopping the heuristic.

The general structure of the heuristic is shown in Algorithm 1. This heuristic is hybrid in the sense that local search is used as mutation in the genetic algorithm.

Algorithm 1. GA pseudo code

Input: FSMVRP, population size N, probability p_m, stopping criteria
Output: Solution to FSMVRP
Set population $P := \phi$
$P \leftarrow P \cup$ chromosomes from savings and sweep algorithm
while $|P| \neq N$ **do**
  $P \leftarrow P \cup$ chromosomes from randomized algorithm
end while
while $\neg$ stopping criteria **do**
  $p_1, p_2 \leftarrow$ tournament-selection(P)
  if $p_1 \neq p_2$ **then**
    $c \leftarrow$ Order-crossover(p_1,p_2)
  else
    $c \leftarrow$ Single-parent-crossover(p_1)
  end if
  With probability p_m, $c \leftarrow$ local-search(c)
  Place c into the population
  if The placement is successful **then**
    generation ++
    if bestcost(P) $\leqslant$ cost(c) **then**
      convergence ++
    else
      convergence = 0
    end if
  end if
end while
return the best solution(P)

3.1. Coding and decoding of chromosomes

To code a solution to the FSMVRP in a chromosome, one way is to adopt the representation used in Ochi et al. (1998) and Lima et al. (2004), where depots are used as trip delimiters. A more straightforward way is to use a sequence of customers without trip delimiters, as has been done by Prins (2004) for the CVRP. The latter approach is implemented in this study.

To decode (evaluate) a sequence of customers (a chromosome) we need to decide the routes of customers and the vehicles used for the routes. When the vehicles are homogeneous, Prins (2004) developed a polynomial time procedure for delivering a single product, while separately Liu et al. (2008) proposed a shortest path method for delivering multiple products with a fixed route. However in both cases no decision on usage of heterogeneous vehicles is considered. In the following, we revised the shortest path method to consider the heterogeneous fleet used in the FSMVRP.

The evaluation of a given chromosome $X = \{1, 2, 3, \ldots, n\}$ is concerned with partitioning customer orders along X into groups so that the total load of each group does not exceed vehicle capacity, while the total cost required to deliver all customer orders is minimized. We 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 $G(X)$, where $(i,j) \in E(G)$ iff $\sum_{m=i+1}^{j} d_m \leqslant Q_t$, where t is a chosen vehicle type. Each arc (i,j) represents a feasible trip, where the vehicle for trip (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 trip (i,j) is equal to $\sum_{m=i+1}^{j} d_m$. A vehicle type t is chosen as the cheapest vehicle type with a capacity Q_t not smaller than the load. In the case of FSMVRP with fixed cost, the cost of the arc/trip (i,j) is equal to the variable cost plus the fixed cost, that is $c_{i,j} = \tau_{0,i+1} + \sum_{h=i+1}^{j-1} \tau_{h,h+1} + \tau_{j,0} + f_t$. For FSMVRP without fixed cost the arc cost of (i,j) is $c_{i,j} = (\tau_{0,i+1} + \sum_{h=i+1}^{j-1} \tau_{h,h+1} + \tau_{j,0}) v_t$.

The shortest path on $G(X)$ defines the optimal partition of the sequence. An optimal decoding can be found in $O(n^2 \log(n))$ time. Faster algorithms can be designed for solving our shortest path problem in $O(n^2)$ or even $O(n)$ time. Interested readers are directed to Prins (2004).

Fig. 1a shows an example for a FSMVRP with fixed cost. There are $n = 6$ customers and two vehicle types. The number on arc (i,j) denotes $\tau_{i,j}$, and the numbers on node j denote d_i respectively. The chromosome to be evaluated is $X = 1, 2, 3, 4, 5, 6$. The corresponding acyclic graph $G(X)$ is constructed in Fig. 1b. Arc (1,3) represents trip (0,2,3,0) with a total cost of 280 + 168 + 120 + 500 = 1068 using vehicle type 2. Fig. 1c gives the resulting solution with trip information and the vehicle used for each trip.

Along with each chromosome is also stored its solution (including trips and vehicles used) for later manipulation in local search. 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 trips into a single sequence.

3.2. Initialization of population

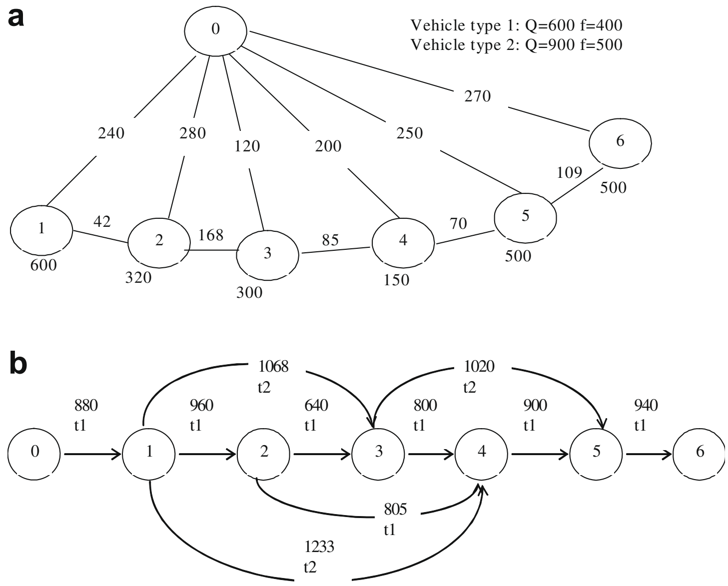
GA is a population based optimization strategy. To populate the initial population with chromosomes, some of the chromosomes are generated as random sequences, and some by heuristics. The savings algorithm and the sweep algorithm are adapted. These two heuristics are originally proposed to solve CVRP, but it is not difficult to revise them to consider heterogeneous fleet.

The savings algorithm from Clarke and Wright (1964) is developed for homogeneous vehicles. Here we apply the algorithm to the FSMVRP with different types of vehicles, one type at a time. When there are T types of vehicles, the savings algorithm is run for T times, each time with a different vehicle capacity Q_t. In total, T solutions could be generated. We implemented both sequential and parallel versions of the Clark and Wright method, and our computational experiments show that the parallel version works better.

Sweep algorithm from Gillett and Miller (1974) is also used to find initial solutions. First, a ray shooting out of the depot sweeps all the customers starting from the x-axis. Customers that are hit by the ray are added sequentially to a single tour, which after a complete sweep is a giant tour of all customers. Then, the shortest path procedure proposed in Section 3.1 is applied to find a optimal solution given the fixed sequence of the tour. In the solution, both the routes and vehicle types used have been determined. Next, each route (trip) in the solution is improved by using a simple TSP nearest insertion heuristic. Additional solutions can be found by repeating the above procedure with other customers as a start node. In fact, we generate only the giant tours starting with the customers from the first trip created. We note that the sweep algorithm could only be applied to data set with x–y coordinates. The benchmark problems we study later provide x–y coordinates of all locations.

An initial solution produced by the heuristics is then converted into its corresponding chromosome by concatenating its routes. For the benchmark instances we tested, there are five to eight initial chromosomes generated by these heuristics.

The rest of the population are consisted of chromosomes that are randomly generated. A randomized procedure is developed to produce random permutation of the customer locations. In generating the random sequences, no demand and vehicle information is considered; the customers are checked not to enter a chromosome more than once.



c

Trip	*Vehicle type*
0-1-0	1
0-2-3-0	2
0-4-5-0	2
0-6-0	1
Total cost	3908

Fig. 1. Decoding of a chromosome. (a) Example data, (b) graph and (c) solution.

Before a chromosome is admitted into the initial population its cost (fitness) is compared against the costs of the existing chromosomes to make sure that no two solutions of the same evaluated cost are in the initial population.

3.3. Selection and crossover

In genetic algorithms, parents are selected to generate offsprings. Among many methods of selection, we implement the tournament selection. Tournament selection runs a ''tournament'' among a few individuals chosen at random from the population and selects the one with the best fitness. Individual chromosomes are ranked by their total costs. A chromosome with smaller total cost has a better fitness. Selection pressure can be adjusted by changing the tournament size. In this study, tournament sizes of two and five are tested. If the tournament size is larger, weak individuals have a smaller chance to be selected. However, to ensure diversity a smaller size may be a better choice. In fact, our computational experiments show that a tournament size of two produces better results.

Reproduction operators are designed in GA to generate children from selected parent(s). One of the reproduction operators we used is a 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.) Note that OX is operated on two parent chromosomes not the solutions (trips).

Sometimes two parents selected by the selection procedure have the same fitness (or total cost). This happens more frequently when the population converges. Hence, a single parent crossover (SPO) operator is developed to allow more diversity. Algorithm 2 shows the pseudo code of this crossover.

Algorithm 2. Single parent crossover

Input: parent solution S
Output: child chromosome X
for each trip in S **do**
  with probability 0.5, place the trip into child solution S_1.
end for
for all customers not yet assigned to S_1 **do**
  Pick a customer randomly.
  Insert it in S_1 at the best position among
  (a) all positions in existing trips, or
  (b) a new empty trip.
end for
Convert S_1 into chromosome X.

The mechanism of the SPO is similar to that of binary fission in asexual reproduction, where one organism splits into two equal-sized offsprings. SPO is performed on the solution instead of the chromosome of a parent. Each trip (a sequence of customers) in the parent solution has a 50% chance of being duplicated in the offspring solution. All other customers not being duplicated will be inserted into the offspring solution one by one in random order. Among all possible positions of inserting a customer, the customer is inserted in the position resulting in the smallest increase in the total cost. When the offspring solution is completed, it is converted into a chromosome.

3.4. Local search

With some mutation probability p_m, the offspring reproduced in the crossover step will be subject to mutation. Instead of using only simple mutation operators, such as insertion and swap, we revise and implement the edge operations for the CVRP as our local improvement strategy for the FSMVRP.

There are a large number of edge exchange schemes developed by several authors, for example Stewart and Golden (1984), Dror and Levy (1986), Salhi and Rand (1987), Fahrion and Wrede (1990), Potvin et al. (1992) and Taillard (1993). Van Breedam (1994) further classifies the multi-route edge exchanges for the CVRP into three categories.

- String relocation: a string of $k = 1$ or 2 vertices is moved from one place to another. Both $k = 1$ and $k = 2$ are implemented. This is basically an insertion move. Remove a vertex or an edge and insert it to another place. It could be a place in the same route (thus a single route move), or in a different route (thus a multi-route insertion).
- String cross: two strings of k vertices are exchanged by crossing two edges. We implement the $k = 2$ cross, i.e., between two edges (strings of two vertices). If the two edges belong to the same trip, it is also called 2-opt. However, two edges being crossed could be from two different trips.
- String exchange: two strings of at most k vertices are exchanged. This, in essence, is a swap move. Only $k = 1$ and $k = 2$ exchanges are implemented, i.e., swaps between two vertices, between a vertex and an edge, and between two edges. As in a string cross and a string allocation, a swap could be a multi-route move.

For each chromosome that is subject to mutation, these local search steps are performed on each pair of customers (vertices) (i,j), as shown in algorithm 3. The local search is restarted whenever an improving move is found and applied. The local search on a chromosome is repeated until no improving move can be found for any pair (i,j).

Algorithm 3. Local search on a vertex pair (i,j)

Input: a vertex pair (i,j), a chromosome and its solution,
Output: the improved chromosome
string relocation (i,j)
string exchange (i,j)
if $\exists m$ such that $i-m$ is an edge **then**
  string relocation $(i-m,j)$
  string exchange $(i-m,j)$
  if $\exists n$ such that $j-n$ is an edge **then**
    string cross $(i-m,j-n)$
    string exchange $(i-m,j-n)$
  end if
end if

When calculating the total cost, the vehicle types used must be considered, since in the FSMVRP different trips might use different vehicles. There is no need to check for the vehicle change for the single route moves, but possible change in the vehicles used has to be accounted for when a move is multi-route. For example, when considering a multi-route insertion move for the FSMVRP with fixed cost, in addition to the change in the variable traveling cost because of modification in routes, there might be a change in the fixed cost f_t. It is possible that a vehicle with smaller capacity Q_t, thus smaller f_t, now is used for the route where a string has been moved out, while a larger vehicle might be used for the route the string is to be inserted. Similarly in the FSMVRP without fixed cost, the variable cost of vehicle type used v_t may change due to the change in vehicle type t.

3.5. Control variables

The proposed GA has four parameters including population size, mutation probability, and convergence limits. Prins (2004) provided some guidance of setting these parameter values when solving the vehicle routing problem. We follow this guidance in setting a small population size, and a low mutation probability. 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 GA without too much fine-tuning the parameters, comparing with other heuristics.

Our genetic algorithm uses a population size of 30; the population is sorted in ascending order in terms of solution cost. To place a offspring into the population (line 16 in Algorithm 1) and maintain population diversity, the population management technique from Sorensen and Sevaux (2006) is adapted.

Sorensen and Sevaux (2006) suggest a distance measure in solution space to control the replacement of some existing solution by the offspring. In our implementation, first a random solution in the worse half of the population is chosen to be replaced by the offspring. Then, instead of using a distance measure, we simple 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. If a offspring is placed into the population, then it is counted as a successful generation. The heuristic is stopped after 900,000 generations.

If a successful generation does not provide a new best cost, a convergence counter is increased by one. The convergence counter is reset if a new best solution is found. The heuristic is stopped after 300,000 consecutive generations without an improvement. These values were chosen based on preliminary experiments.

4. Experimental results

In this section, we report our computational results and compare them with those from the existing literature. Our GA has been programmed in GNU C and executed on an Intel Pentium 4 machine running GNU/LINUX at 3 GHz.

4.1. Test problems

Our computational experiment is based on the 12 benchmark instances developed by Golden et al. (1984). These test problems are designed for the FSMVRP with fixed cost, where the travel costs (variable costs) are the same for all vehicle types. We use the numbering scheme (instance 3, 4, 5, 6, and 13–20) given by Golden et al. (1984) for these instances. The corresponding cost specifications are given in Table 1.

In Table 1, Q_t stands for the capacity of vehicle type t, ($t = A, B, C, D, E, F$); f_t represents the fixed cost of vehicle type t; v_t represents the variable cost of vehicle type t. Problem 3 to problem 6 have 20 customers, problem 13–16 have 50 customers, problem 17 and 18 have 75 customers, and problem 19 and 20 have 100 customers. These problems are for the FSMVRP with fixed cost. We can see from Table 1 that there are between three and six types of vehicles.

Table 1
Specifications for the twenty benchmark problems.

Problem	Q_A	f_A	v_A	Q_B	f_B	v_B	Q_C	f_C	v_C	Q_D	f_D	v_D	Q_E	f_E	v_E	Q_F	f_F	v_F
3	20	20	1.0	30	35	1.0	40	50	1.0	70	120	1.0	120	225	1.0			
4	60	1000	1.0	80	1500	1.0	150	3000	1.0									
5	20	20	1.0	30	35	1.0	40	50	1.0	70	120	1.0	120	225	1.0			
6	60	1000	1.0	80	1500	1.0	150	3000	1.0									
13 (23)	20	20	1.0	30	35	1.1	40	50	1.2	70	120	1.7	120	225	2.5	200	400	3.2
14 (24)	120	100	1.0	160	1500	1.1	300	3500	1.4									
15 (25)	50	100	1.0	100	250	1.6	160	450	2.0									
16 (26)	40	100	1.0	80	200	1.6	140	400	2.1									
17 (27)	50	25	1.0	120	80	1.2	200	150	1.5	350	320	1.8						
18 (28)	20	10	1.0	50	35	1.3	100	100	1.9	150	180	2.4	250	400	2.9	400	800	3.2
19 (29)	100	500	1.0	200	1200	1.4	300	2100	1.7									
20 (30)	60	100	1.0	140	300	1.7	200	500	2.0									

Taillard (1999) adapted the test problems from 13 to 20 to the FSMVRP without fixed cost, where the variable cost per unit distance for each type of vehicle is specified. We number these instances as instance 23–30, as has been done by Brandao (2008).

In total, there are 12 problems for the FSMVRP with fixed cost, and eight for the FSMVRP without fixed cost.

We compare our results to those obtained by the mathematical programming based heuristics ofTaillard (1999) and Choi and Tcha (2007); by the tabu search heuristics of Gendreau et al. (1999), Wassan and Osman (2002) and Brandao (2008); and by the evolutionary algorithm of Lima et al. (2004). No classical heuristic is included since they are dominated by the above mathematical related methods and the tabu search heuristics.

Different computers are used in these studies. The related information about the computers used is shown in Table 2. We use the Dongarra (2007) table to estimate the relative performance of different computers used in the study.

Last column in the table is the CPU speed in millions of floating-point operations per second (Mflop/s).

4.2. Computational results

Here we present our best solutions and average results on the twenty benchmark test problems. Our GA uses a population size of 30, the outer iteration of 900,000 and the inner iteration for convergence of 300,000. We run our algorithm under two settings with different mutation probabilities of 0.05 and 0.1. For each setting, the algorithm has been run five times. The best solutions, average solutions and average times are computed over these 10 runs.

Tables 3 and 4 present the best solutions obtained for the FSMVRP with fixed cost and without fixed cost, respectively. All the authors provide their best solutions. Taillard (1999) does not indicate if the best solutions obtained are during the five runs the average results reported or over a more extensive experiment. The best solutions from Gendreau et al. (1999) are obtained within their 10 runs, for which the average solutions are reported later in the section. The best solutions obtained by Lima et al. (2004) are over 10 runs. Wassan and Osman (2002) run their algorithm five times for the first set of problems and twice for the second, and then present the best solutions. Choi and Tcha (2007) provide their best solutions over five

Table 2
Performance of the computers used.

	Model	CPU	Mflop per Second
Taillard (1999)	Sun Sparc	50M	10
Gendreau et al. (1999)	Sun Sparc	50M	10
Wassan and Osman (2002)	Sun Sparc	50M	10
Lima et al. (2004)	Intel Pentium 4	2G	1033
Choi and Tcha (2007)	Intel Pentium 4	2.6G	1190
Brandao (2008)	Intel Pentium M	1.4G	350
Ours	Intel Pentium 4	3G	1573

Table 3
Best computational results for FSMVRP with fixed cost on 12 test problems.

Problem	Best known	Fist found by	Taillard (1999)	Gendreau et al. (1999)	Wassan and Osman (2002)	Lima et al. (2004)	Choi and Tcha (2007)	Brandao (2008)	Ours (2008)
3	961.03	Taillard (1999)	**961.03**	**961.03**	**961.03**	**961.03**	**961.03**	**961.03**	**961.0257**
4	6437.33	Taillard (1999)	**6437.33**	**6437.33**	**6437.33**	**6437.33**	**6437.33**	**6437.33**	**6437.3312**
5	1007.05	Gendreau et al. (1999)	1008.59	**1007.05**	**1007.05**	**1007.05**	**1007.05**	**1007.05**	**1007.0509**
6	6516.47	Taillard (1999)	**6516.47**	**6516.47**	**6516.47**	**6516.47**	**6516.47**	**6516.47**	**6516.4684**
13	2406.36	Choi and Tcha (2007)	2413.78	2408.41	2422.10	2408.60	**2406.36**	**2406.36**	**2406.3607**
14	9119.03	Gendreau et al. (1999)	**9119.03**	**9119.03**	9119.86	**9119.03**	**9119.03**	**9119.03**	**9119.0300**
15	2586.37	Gendreau et al. (1999)	**2586.37**	**2586.37**	**2586.37**	2586.88	**2586.37**	**2586.37**	**2586.3695**
16	2720.43	Choi and Tcha (2007)	2741.50	2741.50	2730.08	**2721.76**	**2720.43**	2728.14	2724.2228
17	1734.53	Brandao (2008)	1747.24	1749.50	1755.10	1758.53	1758.53	**1734.53**	**1734.5314**
18	2369.65	Brandao (2008)	2373.63	2381.43	2385.52	2396.47	2371.49	**2369.65**	**2369.6464**
19	8659.74	Wassan and Osman (2002)	8661.81	8675.16	**8659.74**	8691.00	8664.29	8661.81	8662.9452
20	4039.49	Choi and Tcha (2007)	4047.55	4086.76	4061.64	4093.29	4039.49	4042.59	[a]**4038.4556**
Deviation			0.20%	0.30%	0.29%	0.29%	0.06%	0.03%	0.01%
# of best sol.	11		5	6	6	6	8	9	10

[a] New best solution.

Table 4
Best computational results for FSMVRP without fixed cost on eight test problems.

Problem	Best known	First found by	Taillard (1999)	Gendreau et al. (1999)	Wassan and Osman (2002)	Choi and Tcha (2007)	Brandao (2008)	Ours (2008)
23	1491.86	Gendreau et al. (1999)	1494.58	**1491.86**	1499.69	**1491.86**	**1491.86**	**1491.8599**
24	603.21	Taillard (1999)	**603.21**	**603.21**	608.57	**603.21**	**603.21**	**603.2129**
25	999.82	Gendreau et al. (1999)	1007.35	**999.82**	**999.82**	**999.82**	**999.82**	**999.8236**
26	1131.00	Wassan and Osman (2002)	1144.39	1136.63	**1131.00**	**1131.00**	**1131.00**	**1131.0008**
27	1031.00	Gendreau et al. (1999)	1044.93	**1031.00**	1047.74	1038.60	1038.60	1038.5974
28	1800.80	Brandao (2008)	1831.24	1801.40	1814.11	1801.40	**1800.80**	1801.3977
29	1100.56	Wassan and Osman (2002)	1110.96	1105.44	**1100.56**	1105.44	1105.44	1105.4392
30	1530.16	Wassan and Osman (2002)	1550.36	1541.18	**1530.16**	1530.43	1530.43	1534.3672
Deviation			0.93%	0.21%	0.47%	0.15%	0.15%	0.19%
# of best sol.	8		1	4	4	4	5	4

runs, and each run has different parameters. Brandao (2008) presents their best solutions during all experiments with different values of parameters. Also provided are the best solutions from two different versions of the tabu search heuristic with different initial solutions. Here we use the best solutions from Brandao (2008) during all its computation runs.

No computation time is presented in Tables 3 and 4. First, most of the authors did not report the computation time for the best solution. Second, the average computation time for all runs conducted might be a better number to compare, which we will report in Tables 5 and 6. In Tables 3 and 4 boldface figures indicate the best-known solutions.

Table 3 presents the best computational results for FSMVRP with fixed cost on 12 test problems. We can see that our GA produces a new best solution for test problem 20. The route and vehicle information for this best solution is provided in the appendix. Table 3 provides a count of best-known solutions reached among the 12 test problems. Our algorithm is the best to attain 10 out of 12 best solutions, compared with nine from Brandao (2008), eight from Choi and Tcha (2007) and six from Lima et al., 2004. Also provided in Table 3 is the percentage deviation of each algorithm's solution from the best-known

Table 5
Average results for FSMVRP with fixed cost on 12 test problems.

Problem	Taillard (1999)		Gendreau et al. (1999)		Lima et al. (2004)		Choi and Tcha (2007)		Ours (2008)	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
3	**961.03**	N/A	**961.03**	164	963.40	89	**961.03**	0	**961.03**	0
4	**6437.33**	N/A	6441.01	253	6437.33	85	6441.01	1	**6437.33**	0
5	1008.59	N/A	1008.72	164	1009.90	85	**1007.05**	1	1007.20	2
6	**6516.47**	N/A	6517.98	309	**6516.47**	85	6516.84	0	**6516.47**	0
13	2436.78	470	2424.88	724	**2409.10**	559	2409.77	10	2414.50	91
14	9123.60	570	9121.98	1033	9121.62	669	**9119.13**	51	9119.48	42
15	2593.61	334	2590.68	901	2590.20	554	2588.92	10	**2586.37**	48
16	2744.25	349	2743.96	815	**2729.60**	507	2731.08	11	2735.54	107
17	1753.74	2072	1752.29	1022	1770.30	1517	1746.26	207	**1745.54**	109
18	2382.80	2744	2392.57	691	2401.00	1613	**2375.78**	70	2377.92	197
19	8665.40	12528	8682.50	1687	8695.30	2900	**8665.08**	1179	8666.04	778
20	4063.18	2117	4100.20	1421	4142.30	2383	4057.33	264	**4052.81**	1004
Average	4057.23	2648	4061.48	765	4065.54	920	4051.60	150	4051.69	198
Deviation	0.27%		0.37%		0.47%		0.13%		0.13%	

Table 6
Average results for FSMVRP without fixed cost on eight test problems.

Problem	Taillard (1999)		Gendreau et al. (1999)		Choi and Tcha (2007)		Ours (2008)	
	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)	Cost	Time (s)
23	1494.58	473	1494.21	626	**1493.36**	4	1510.23	117
24	**603.21**	575	603.33	669	**603.21**	37	603.28	26
25	1007.35	335	1001.80	736	**1001.33**	6	1004.21	37
26	1144.39	350	1137.01	852	1135.62	6	**1134.10**	54
27	1044.93	2245	1046.36	1453	**1038.60**	103	1047.12	153
28	1831.24	2876	1812.00	1487	**1801.40**	81	1812.86	394
29	1110.96	5833	1117.09	1681	**1106.62**	299	1109.44	479
30	1550.36	3402	1553.72	1706	**1536.10**	112	1540.76	826
Average	1223.38	2011	1220.69	1151	1214.53	81	1220.25	261
Deviation	1.02%		0.80%		0.29%		0.76%	

solution. In this measurement, our algorithm again is the best to have an average deviation of 0.01%. However, we have to note that all these algorithms are very effective to produce less than 1% deviation.

Table 4 presents the best computational results for FSMVRP without fixed cost on eight test problems. Table 4 provides the same count of best-known solutions reached as in Table 3. The best heuristic on this account is the tabu search by Brandao, 2008 with five best solutions attained. Our algorithm reaches four best solutions out of eight possible solutions, so are the algorithms by Gendreau et al. (1999), Wassan and Osman (2002) and Choi and Tcha (2007). Taillard (1999) generates one best solution for test problem 14. In terms of the percentage deviation of each algorithm's solution from the best-known solution, our algorithm has a deviation of 0.19%, performing better than those from Taillard (1999), Gendreau et al. (1999), Wassan and Osman (2002), and slightly worse than the 0.15% deviation produced by both Choi and Tcha (2007) and Brandao (2008). Again, we note that all these algorithms are very effective to produce less than 1% deviation.

We also present the comparison of these algorithms over the quality of average solution in Tables 5 and 6. Tables 5 and 6 provide the average solutions obtained by selected algorithms for the FSMVRP with fixed cost and without fixed cost, respectively. There is no average solution reported by Wassan and Osman (2002), and Brandao (2008) reports only the best solutions found from their numerous runs with different values of the parameters. Thus Wassan and Osman (2002) and Brandao (2008) are excluded from Tables 5 and 6. The results of Taillard (1999) are the average results of five runs. The results obtained by Gendreau et al. (1999) are the average results over 10 runs. The average solutions from Lima et al., 2004 are over ten runs. Choi and Tcha (2007) also provide the average solution and average computation time for each instance over five runs.

Table 5 presents average computational results for FSMVRP with fixed cost on 12 test problems. Both solution cost and computation time for each instance are included for each algorithm. In Table 5, we also give the percentage deviation of each algorithm's solution from the best-known solution for each problem. All five algorithms are very accurate with an average deviation that is less than 1% for the 12 test problems. We see that, in average, our genetic algorithm produces better results than those of Lima etal. (2004), Taillard (1999), and Gendreau et al. (1999), and results that are as good as those of Choi and Tcha (2007). Table 5 gives also the computation times reported by these authors.

Table 6 presents the average computational results for FSMVRP without fixed cost on eight test problems. Lima et al. (2004) did not report their results for the FSMVRP without fixed cost, thus are not included in Table 6. We can see that the best average results for this case are produced by the mathematical programming from Choi and Tcha (2007), while ours are better than those of the tabu search heuristics from both Taillard (1999) and Gendreau et al. (1999). All four algorithms demonstrate remarkable accuracy that the average results are less than or around 1% deviation from the best-known solutions.

Tables 5 and 6 also show the computation times associated with these studies. Performance comparison only in terms of CPU time is not recommended, owing to other influencing factors such as cache, main memory and compiler. Still, the reported computation times support the good performance of our heuristic.

5. Conclusion

This paper presents a genetic algorithm based heuristic for the FSMVRP both with and without fixed cost. On the Golden et al. (1984) benchmark problems it reaches the best-known solution 14 times and find a new best-known solution. It also provides a very competitive performance in terms of average solution cost.

The contributions of the paper to the literature that surveyed by Baldacci et al. (2008) are three fold. First, it dramatically improved the performance of earlier attempts of genetic algorithm implemented by Ochi et al. (1998) and Lima et al. (2004). New initial solution procedure, chromosome evaluation algorithm and local search moves are developed. Second, it demonstrates that genetic algorithm based solution approach is as competitive as other approaches currently employed in the literature, such as the mathematical programming based heuristics of Taillard (1999) and Choi and Tcha (2007); the tabu search heuristics of Gendreau etal. (1999), Wassan and Osman (2002) and Brandao (2008). Third, a new best solution is found for instance #20 in the Golden et al. (1984) benchmark problem set.

In the future one possible research direction is to design more effective local search moves to speed up the computation. For example, moves could be limited to the locations in the nearest neighborhood - not all the locations currently being implemented. We are also considering apply the GA approach to more difficult set of test instances. Another direction is to tackle the similar but more difficult variant, the HVRP, where a resource constrained shortest path problem has to be solved in order to evaluate the solution. An efficient algorithm needs to be designed for this problem.

Appendix A

The new best solution for test problem 20. Customer locations but not depot are included in the routes and they are numbered from 0 to 99.

Route	Vehicle type used
5 95 4 88	A
32 80 78	A

(continued on next page)

Appendix A (*continued*)

Route	Vehicle type used
17 82 7 44 16 83 59	A
51 6 47 18 10 61 87 30 68	B
52 57 12	A
27 75 49 26	A
58 92 98	A
3 55 22 66 38 24 54 23 53	B
0 50 8 34 70 64 65 19 29 69	B
91 99 84	A
67 11	A
60 15 85 37 13 43 90 97 36	B
1 56 14 42 41 86	A
39 72 71 25	A
9 31 89 62 63 48 35 46 45 81	B
20 73 74 21 40	A
93 94 96	A
79 28 33 77 2 76	A

References

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.

Brandao, J., 2008. A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. European Journal of Operational Research. doi:10.1016/j.ejor.2007.05.059.

Choi, E., Tcha, D.-W., 2007. A column generation approach to the heterogeneous fleet vehicle routing problem. Computers and Operations Research 34, 2080–2095.

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

Desrochers, M., Verhoog, T.W., 1991. A new heuristic for the fleet size and mix vehicle routing problem. Computers and Operations Research 18, 263–274.

Dongarra, J., 2007. Performance of various computers using standard linear equations software. Technical Report CS-89-85. University of Tennessee.

Dror, M., Levy, L., 1986. A vehicle routing improvement algorithm. Computers and Operations Research 13, 33–45.

Fahrion, R., Wrede, W., 1990. On a principle of chain-exchange for vehicle routing problem. Journal of the Operational Research Society 41, 821–827.

Fisher, M., Jaikumar, R., 1981. A generalized assignment heuristic for vehicle routing. Networks 11, 109–124.

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., Hertz, A., Laporte, G., 1992. New insertion and post-optimization procedures for the traveling salesman problem. Operations Research 40, 1086–1094.

Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, E.D., 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers and Operations Research 26, 1153–1173.

Gillett, B., Miller, L., 1974. A heuristic for the vehicle dispatching problem. Operations Research 22, 340–349.

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 and 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, F., Golden, B., Wasil, E., 2007. A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. Computers and Operations Research 34, 2734–2742.

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, F.-H., Shen, S.-Y., 1999. Fleet size and mix vehicle routing problem with time windows. Journal of the Operational Research Society 50, 721–732.

Liu, S., Lei, L., Park, S., 2008. On the multi-product packing-delivery problem with a fixed route. Transportation Research Part E 44, 350–360.

Ochi, L.S., Vianna, D.S., Drummond, L.M., Victor, A.O., 1998. A parallel evolutionary algorithm 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.

Potvin, J.-Y., Kervahut, T., Garcia, B.L., Rousseau, J.M., 1992. The vehicle routing problem with time windows – part i: tabu search. INFORMS Journal on Computing 8, 158–164.

Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers and Operations Research 31, 1985–2002.

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., 1987. Improvements to vehicle routing heuristics. Journal of the Operational Research Society 38, 293–295.

Salhi, S., Rand, G.K., 1993. Incorporating vehicle routing into the vehicle fleet composition problem. European Journal of Operational Research 66, 313–330.

Salhi, S., Sari, M., Saidi, D., Touati, N.A.C., 1992. Adaptation of some vehicle fleet mix heuristics. OMEGA International Journal of Management Science 20, 653–660.

Sorensen, K., Sevaux, M., 2006. Mapm: memetic algorithms with population management. Computers and Operations Research 33, 1214–1225.

Stewart, W.R., Golden, B.L., 1984. A Lagrangean relaxation heuristic for vehicle routing. European Journal of Operational Research 15, 84–88.

Taillard, E.D., 1993. Parallel iterative search methods for vehicle routing problems. Networks 23, 661–673.

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.

Van Breedam, A., 1994. An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related customer-related and time-related constraints. Ph.D. Disseration, University of Antwerp.
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.