Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 34 (2007) 2734 – 2742
computers & operations research
www.elsevier.com/locate/cor

A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem

Feiyue Li[a], Bruce Golden[b,*], Edward Wasil[c]

[a] *Princeton Optimization Inc., 2 Research Way, Princeton, NJ 08540, USA*
[b] *R.H. Smith School of Business, University of Maryland, College Park, Maryland 20742, USA*
[c] *Kogod School of Business, American University, Washington, DC 20016, USA*

Available online 19 January 2006

Abstract

In the heterogeneous fleet vehicle routing problem (HVRP), several different types of vehicles can be used to service the customers. The types of vehicles differ with respect to capacity, fixed cost, and variable cost. We assume that the number of vehicles of each type is fixed and equal to a constant. We must decide how to make the best use of the fixed fleet of heterogeneous vehicles.

In this paper, we review methods for solving the HVRP, develop a variant of our record-to-record travel algorithm for the standard vehicle routing problem that takes a heterogeneous fleet into account, and report computational results on eight benchmark problems. Finally, we generate a new set of five test problems that have 200–360 customers and solve each new problem using our record-to-record travel algorithm.

Keywords: Vehicle routing problem; Heuristics; Record-to-record travel

1. Introduction

In the standard version of the vehicle routing problem (VRP), a fleet of *homogeneous* vehicles is based at a single depot. Each vehicle has the same capacity and must leave from and return to the depot. Each customer has a known demand and is serviced by exactly one visit of a single vehicle. A sequence of deliveries must be generated for each vehicle so that all customers are serviced and the total distance traveled by the fleet is minimized.

In the *heterogeneous* fleet vehicle routing problem (HVRP), there are several different vehicle types. For vehicle type t, the capacity is Q_t, the fixed cost is f_t, and the variable cost per unit distance is α_t. There are n_t vehicles of type t that are available for servicing the customers. Note that n_t might be very large or, essentially, unlimited. Some customers may have such large demands that they can be serviced only by large capacity vehicles.

In the *fixed* fleet version of the HVRP, the values of n_t are fixed. More specifically, the number of vehicles of type t is *limited* and the fleet composition is known in advance. We must decide how to make the best use of a fixed fleet of heterogeneous vehicles. This is the version of the HVRP studied in this paper.

* Corresponding author. Tel.: +1 301 405 2232; fax: +1 301 405 3364.
E-mail address: bgolden@rhsmith.umd.edu (B. Golden).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2005.10.015

When the number of vehicles of type t is *unlimited*, we must determine the best composition of the fleet. We are not studying this version of the HVRP in this paper and refer the reader to [1] for a literature review and computational results with a tabu search heuristic.

In practice, there are several applications of the fixed fleet HVRP according to [2]. In residential and commercial waste collection, it is common for fleets to have different capacity vehicles. In residential pickups, when the actual service at a location must be completed by a specific vehicle such as a side loader with mechanical arms that pick up the trash cans, there are site dependency issues rather than the vehicle capacity issues encountered in the HVRP (see [3]) for details on the VRP with site dependencies).

Levy [2] points out that FedEx Ground has different capacity vehicles in its fleet with vehicles oriented to bulk, ground contractors, and home delivery contractors. The capacity of the different vehicles is taken into account in a fleet mix decision and then in the allocation of deliveries (and pickups) to the fixed fleet after the fleet mix decision is made.

In some instances, single copy newspaper delivery has a varied fleet [2]. Large vehicles will make larger drops of newspapers and smaller vehicles will make smaller drops and even make bundle drops for individual carriers.

In Section 2, we review three algorithms that have been developed to solve the fixed fleet version of the HVRP. We develop a variant of our record-to-record travel algorithm (see [4]) to handle a heterogeneous fixed fleet and report computational results for all four procedures on eight test problems taken from the literature.

In Section 3, we develop five new test problems that have 200–360 customers. We apply our record-to-record travel algorithm to these five problems and report our results. In Section 4, we give our conclusions.

2. Solving the fixed fleet version of the HVRP

2.1. Algorithms for the HVRP

In the last 6 years or so, three algorithms have been developed to solve the fixed fleet version of the HVRP.

Taillard [5] used tabu search and linear programming to solve the HVRP. His method is known as heuristic column generation (HCG). For each vehicle type and an unlimited number of vehicles, a homogeneous VRP is solved. A tabu search algorithm is used to generate a set of good solutions. Single vehicle routes are extracted and then combined into a partial solution using tabu search. The process is repeated and routes are memorized as candidates for the final solution to the HVRP. Once the homogeneous VRPs are solved for each vehicle type, useless routes (these are routes servicing customers already served on other, larger routes) are deleted leaving a set R of routes. R contains only a small number of all the routes that can be generated for the homogeneous VRPs. The best solution to the HVRP is found by solving an integer linear program. Each column in the integer linear program corresponds to a route in R.

Threshold accepting is a deterministic variant of simulated annealing in which a threshold value T is specified as the upper bound on the amount of objective function increase allowed (uphill moves can be made). Tarantilis et al. [6,7] developed a list-based threshold accepting algorithm (denoted by LBTA) and a backtracking adaptive threshold accepting algorithm (denoted by BATA) for solving the HVRP. In the list-based algorithm, a list of values for T is used during the search. In the backtracking algorithm, T is allowed to increase during the search. LBTA and BATA use two-opt moves, 1–1 exchanges (swap two customers from either the same or different routes), and 1–0 exchanges (move a customer from its position on one route to another position on either the same route or a different route) when performing local search.

HCG, LBTA, and BATA use heuristic solution methods, so they are not guaranteed to produce an optimal solution. All three algorithms have been computationally tested on eight benchmark problems (more about this in Section 2.3).

We adapted our record-to-record travel algorithm for the VRP [4] to handle the HVRP and we denote our algorithm by HRTR. The details of HRTR are given in Table 1.

Dueck [8] developed record-to-record travel as a deterministic variant of simulated annealing. Let S be the current solution and S' be an alternative solution in the neighborhood of S. In record-to-record travel, Record is the total distance of the best solution observed so far and Deviation is r% $\times$ Record. If the objective function value of S' is less than Record + Deviation, then S' is selected as the new solution. Uphill moves are allowed in order to avoid becoming trapped at a poor local minimum.

In HRTR (see Table 1), most of the work is performed between lines 10 and 27 where we apply record-to-record travel with uphill moves (line 12) followed by strictly downhill moves (line 21). In line 37, we can accept a perturbed

Table 1
Record-to-record travel algorithm for the HVRP

```
1   Parameters are I, K, M, and NBListSize.
        I: number of iterations record-to-record travel with uphill moves runs on the current solution.
        K: maximum number of consecutive iterations allowed when the local record has not been updated.
        M: maximum number of consecutive iterations allowed when the global record has not been updated.
        NBListSize: size of the neighbor list for each node.
2   Set I = 30, K = 5, M = Max (Number of nodes/2, 30) and NBListSize = 20.
3   Set Global Record = ∞, Global Deviation = 0.01× Global Record.
4   Generate a feasible solution using the least cost insertion algorithm.
5   Set Record = objective function value of the current solution.
6   Set Deviation = 0.01 × Record.
7   Set itr = 0.
8   while itr ⩽ M do
9       Set count = 0.
10      while count ⩽ K do
11          for i = 1 to I do
12              One Point Move with record-to-record travel, Two Point Move
                with record-to-record travel between routes, and Two-opt Move
                with record-to-record travel. Feasibility must be maintained.
13              if no feasible record-to-record travel move is made then
14                  go to line 21.
15              end if
16              if a new record is produced then
17                  update Record and Deviation.
18                  Set count = 0.
19              end if
20          end for
21          For the current solution, apply One Point Move (within and between routes)
            Two Point Move (between routes), Two-opt Move (within and between routes),
            and OR-opt move (within and between routes). Only downhill moves are allowed.
22          if a new record is produced then
23              update Record and Deviation.
24              Set count = 0.
25          end if
26          count = count + 1.
27  end while
28  if the current solution is better than the global best solution then
29          update the Global Record and Global Deviation.
30          Set itr = 0.
31  end if
32  Perturb the solution and go through line 10 and line 27.
33  if the perturbed solution is better than the global best solution then
34          update the Global Record and Global Deviation.
35          Set itr = 0.
36  end if
37  Accept the perturbed solution if it is less than Global Record + Global Deviation.
38  itr = itr + 1.
39  end while
40  Output the solution corresponding to the Global Record.
```

solution that is slightly worse than the best solution we have generated so far (this is in contrast to our record-to-record travel algorithm for the VRP which only accepts a solution that is better than the best solution). We use this two-level approach with record-to-record travel in HRTR since the objective function landscape is rugged with many poor local minima and we need additional ways in our heuristic to escape from them.

We have lots of experience in applying our record-to-record travel algorithm to the standard vehicle routing problem, very large-scale vehicle routing problems [4], and the open vehicle routing problem [9]. Our choices for values of I, K, M, and *NBListSize* in HRTR are very similar to values that produced good solutions to very large-scale problems

Table 2
Specifications for eight benchmark problems with at most six types of vehicles

Problem	n	Q_A	f_A	α_A	n_A	Q_B	f_B	α_B	n_B	Q_C	f_C	α_C	n_C	Q_D	f_D	α_D	n_D	Q_E	f_E	α_E	n_E	Q_F	f_F	α_F	n_F	%
13	50	20	20	1.0	4	30	35	1.1	2	40	50	1.2	4	70	120	1.7	4	120	225	2.5	2	200	400	3.2	1	95.39
14	50	120	100	1.0	4	160	1500	1.1	2	300	3500	1.4	1													88.45
15	50	50	100	1.0	4	100	250	1.6	3	160	450	2.0	2													94.76
16	50	40	100	1.0	2	80	200	1.6	4	140	400	2.1	3													94.76
17	75	50	25	1.0	4	120	80	1.2	4	200	150	1.5	2	350	320	1.8	1									95.38
18	75	20	10	1.0	4	50	35	1.3	4	100	100	1.9	2	150	180	2.4	2	250	400	2.9	1	400	800	3.2	1	95.38
19	100	100	500	1.0	4	200	1200	1.4	3	300	2100	1.7	3													76.74
20	100	60	100	1.0	6	140	300	1.7	4	200	500	2.0	3													95.92

n: number of customers; Q_t: capacity of vehicle type t (t = A, B, C, D, E, F); f_t: fixed cost of vehicle type t; α_t: variable cost per unit distance of vehicle type t; n_t: number of vehicles of type t available; %: 100 × (total demand/total capacity).

and open VRPs. Our sense is that small changes to the values of the four parameters will not have much of an effect on the solution quality. We point out that, if the value of I is set too low, then we may not be able to escape from a poor local minimum. Increasing the values of K and M will lead to more iterations which, in turn, may lead to different solutions. Although a fixed-length neighbor list with 20 nearest neighbors works well, we have used a variable-length neighbor list in solving very large-scale problems with record-to-record travel [4].

2.2. Test problems

Golden et al. [10] developed eight test problems for the vehicle fleet size and mix routing problem which can be viewed as a special case of the HVRP where the travel costs are the same for all vehicle types and the number of vehicles of each type is unlimited. Taillard [5] adapted the Golden et al. problems to the HVRP by specifying the variable cost per unit distance for each type of vehicle and the number of vehicles of each type available. The specifications for the HVRP problem set are given in Table 2. We use the numbering scheme (problem 13, …, problem 20) given by Golden et al. [10].

The problems range in size from $n = 50$ to $n = 100$ customers. There are no route-length restrictions and no customer service times. The eight problems have between three and six types of vehicles available. In Table 2, in the right-most column labeled %, we compute the ratio of total customer demand to total capacity of all available vehicles. This provides an indication of how much potential flexibility we have in using the available vehicles. For six problems, we see that the percentages are greater than 94% and expect that nearly all of the available vehicles will be used in a solution.

2.3. Computational results

In Table 3, we present the solution values and the computation times for HCG, LBTA, BATA, and HRTR. The results for HCG, LBTA, and BATA are taken from the literature and were generated using a single set of parameter values. We see that HRTR generated six new best-known solutions (problems 13, 14, 16, 17, 18, 20) and produced the same best-known solution as BATA on problem 15. HCG generated the best-known solution to problem 19. LBTA did not generate any best-known solutions. Over all eight problems on different computing platforms, LBTA was the fastest procedure (1781 s) followed by HRTR (2286 s). Converting the running times of the slow Pentium III processor used by LBTA to the fast Athlon 1 GHz processor used by HRTR would yield much smaller times for LBTA in Table 3.

In Table 4, we give the percentage deviation of each algorithm's solution from the best-known solution for each problem. All four algorithms are very accurate with an average deviation that is less than 1% for the eight test problems.

In Fig. 1, we show the best-known solutions produced by HRTR to problems 17 and 18. The routes generated by HRTR for all eight problems are given in Ref. [11].

Table 3
Computational results for HVRP algorithms on eight test problems

Problem	Taillard HCG	Time (s)	Tarantilis et al. LBTA	Time (s)	Tarantilis et al. BATA	Time (s)	Li et al. HRTR	Time (s)
13	1518.05	473	1519.96	110	1519.96	843	**1517.84**	358
14	615.64	575	612.51	51	611.39	387	**607.53**	141
15	1016.86	335	1017.94	94	**1015.29**	368	**1015.29**	166
16	1154.05	350	1148.19	11	1145.52	341	**1144.94**	188
17	1071.79	2245	1071.67	221	1071.01	363	**1061.96**	216
18	1870.16	2876	1852.13	310	1846.35	971	**1823.58**	366
19	**1117.51**	5833	1125.64	309	1123.83	428	1120.34	404
20	1559.77	3402	1558.56	675	1556.35	1156	**1534.17**	447
Total		16089		1781		4857		2286

Bold: Best-known solution; HCG: Heuristic column generation solution from Taillard [5], Sun Sparc workstation, 50 MHz; LBTA: List-based threshold accepting solution from Tarantilis et al. [6], Pentium III, 550 MHz, 128 MB RAM; BATA: Backtracking adaptive threshold accepting solution from Tarantilis et al. [7], Pentium II, 400 MHz, 128 MB RAM; HRTR: Record-to-record travel solution, Athlon, 1 GHz, 256 MB RAM.

Table 4
Percent deviation results for HVRP algorithms on eight test problems

Problem	Best-known solution	Taillard HCG	Tarantilis et al. LBTA	Tarantilis et al. BATA	Li et al. HRTR
13	1517.84	0.0138	0.1396	0.1396	0
14	607.53	1.3349	0.8197	0.6353	0
15	1015.29	0.1546	0.2610	0	0
16	1144.94	0.7956	0.2838	0.0506	0
17	1061.96	0.9256	0.9143	0.8521	0
18	1823.58	2.5543	1.5656	1.2486	0
19	1117.51	0	0.7275	0.5655	0.2532
20	1534.17	1.6686	1.5897	1.4457	0
Average		0.9309	0.7876	0.6171	0.0316

3. New test problems and computational results

We selected five large-scale vehicle routing problems with 200–360 customers from Golden et al. [12] and adapted them to the HVRP. Each problem has a geometric symmetry with customers located in concentric circles around the depot. The problem generator is given in the Appendix. The specifications for the five new problems are given in Table 5.

Table 5
Specifications for five new problems with at most six types of vehicles

Problem	n	Q_A	f_A	α_A	n_A	Q_B	f_B	α_B	n_B	Q_C	f_C	α_C	n_C	Q_D	f_D	α_D	n_D	Q_E	f_E	α_E	n_E	Q_F	f_F	α_F	n_F	%
H1	200	50	20	1.0	8	100	35	1.1	6	200	50	1.2	4	500	120	1.7	3	1000	225	2.5	1					93.02
H2	240	50	100	1.0	10	100	1500	1.1	5	200	3500	1.2	5	500	120	1.7	4									96.00
H3	280	50	100	1.0	10	100	250	1.1	5	200	50	1.2	5	500	120	1.7	4	1000	225	2.5	2					94.76
H4	320	50	100	1.0	10	100	200	1.1	8	200	400	1.2	5	500	120	1.7	2	1000	225	2.5	2	1500	250	3	1	94.12
H5	360	50	25	1.0	10	100	80	1.2	8	200	150	1.5	5	500	320	1.8	1	1500	225	2.5	2	2000	250	3	1	92.31

n: number of customers; Q_t: capacity of vehicle type t (t = A, B, C, D, E, F); f_t: fixed cost of vehicle type t; α_t: variable cost per unit distance of vehicle type t; n_t: number of vehicles of type t available; %: 100 × (total demand/total capacity).

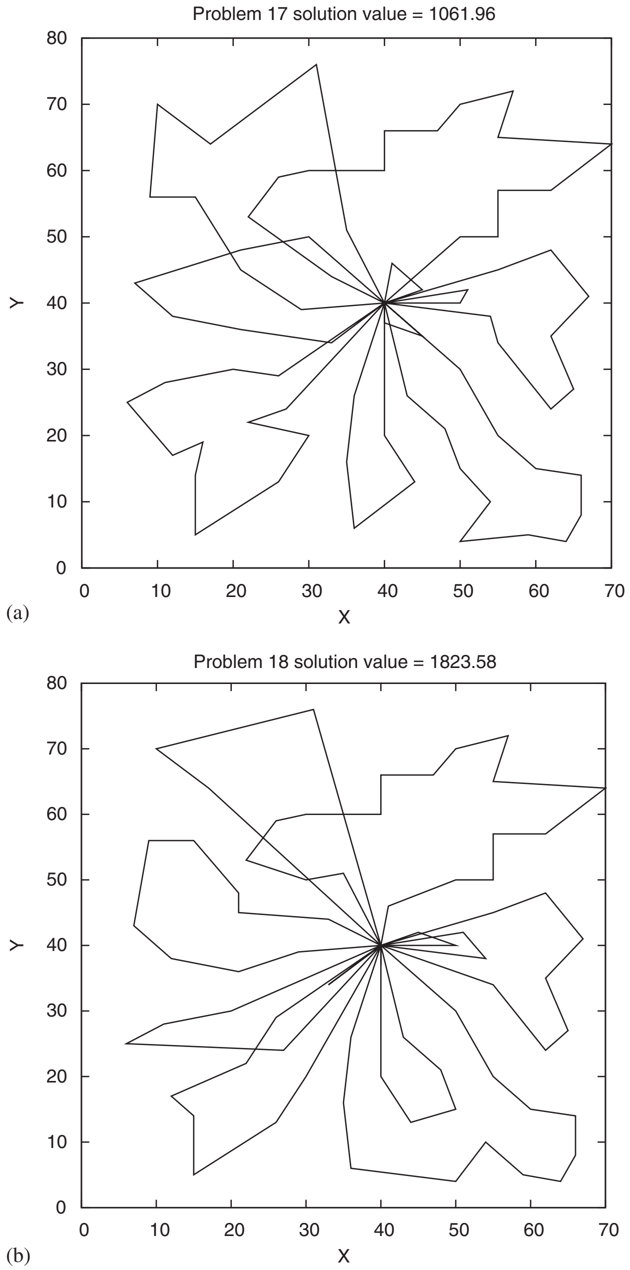
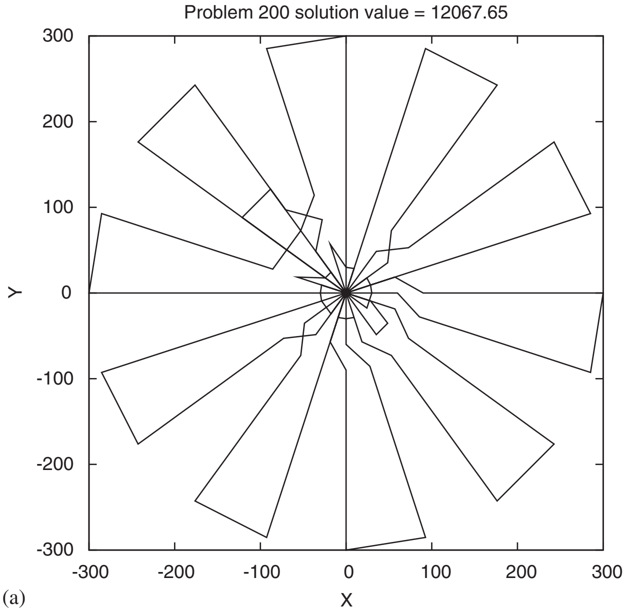


Fig. 1. Best-known solutions to two problems produced by HRTR. (a) Problem 17 and (b) Problem 18.

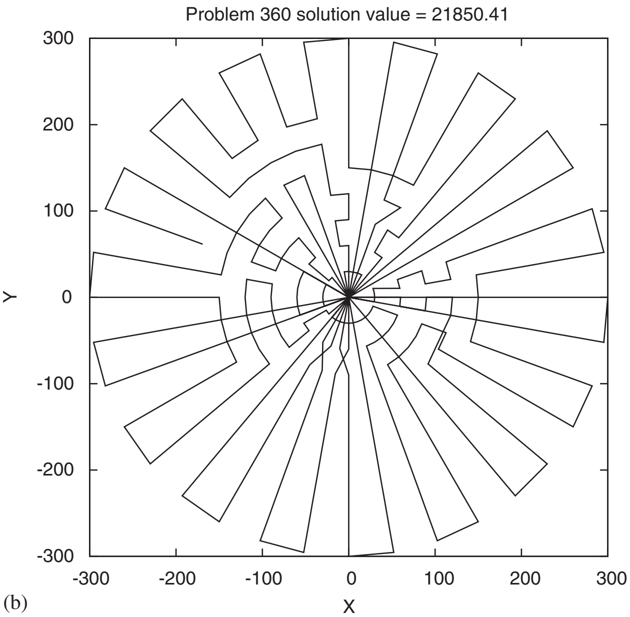
Table 6
Computational results for HRTR on five new test problems

		Li et al.	
Problem	n	HRTR	Time (s)
H1	200	12067.65	687.82
H2	240	10234.40	995.27
H3	280	16231.80	1437.56
H4	320	17576.10	2256.35
H5	360	21850.41	3276.91

HRTR: Record-to-record travel solution, Athlon, 1 GHz, 256 MB RAM.



(a)



(b)

Fig. 2. Solutions to two new problems produced by HRTR. (a) Problem H1 and (b) Problem H5.

For all five new problems, there are no route-length restrictions and no customer service times. One problem has four types of vehicles available, two problems have five types of vehicles available, and two problems have six types of vehicles available. For each of the five problems, we see that the ratio of total customer demand to the total capacity of all available vehicles is greater than 92% and we expect nearly all of the available vehicles will be used in a solution.

We applied HRTR with a single set of parameter values to the five new problems. In Table 6, we present the solution values and computation times. In Fig. 2, we show the solutions produced by HRTR to the smallest and largest problems (H1 and H5). These routes are visually appealing. The computation times for all five problems ranged from 11 min for the smallest problem ($n = 200$) to 54 min for the largest problem ($n = 360$).

4. Conclusions

Over the last 6 years, three algorithms (HCG, LBTA, BATA) have been proposed in the vehicle routing literature to solve the HVRP with a fixed fleet. We developed a solution procedure (HRTR) based on the record-to-record travel algorithm and compared the results of HRTR to the results of HCG, LBTA, and BATA on eight benchmark problems. All four algorithms generated accurate results that were within 1% of the best-known solutions on average. HRTR produced six new best-known solutions and was reasonably fast (averaging 285 s to solve a problem). Finally, we developed five new HVRPs and solved them with HRTR.

Appendix: HVRP Generator

$(x(i), y(i))$ is the coordinate of customer i, where $i = 0$ is the depot, $q(i)$ is the demand of customer i, A and B are parameters that determine the number of customers n, where $n = \mathrm{A} \times \mathrm{B}$. All data recorded to four decimal places.

```
begin
    ω = 0
    x(ω) = 0, y(ω) = 0, q(ω) = 0
    for k := 1 to B do
        begin
            γ = 30k
            for i := 1 to A do
                begin
                    ω = ω + 1
                    x(ω) = γ cos [2(i − 1)π/A]
                    y(ω) = γ sin [2(i − 1)π/A]
                    if mod(i, 4) = 2 or 3
                        then q(ω) = 30
                        else q(ω) = 10
                end
        end
end
```

Problem	A	B	n
H1	20	10	200
H2	40	6	240
H3	28	10	280
H4	40	8	320
H5	36	10	360

References

[1] Gendreau M, Laporte G, Musaraganyi C, Taillard E. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research 1999;26:1153–73.

[2] Levy L. Private communication. RouteSmart Technologies, Inc.; 2005.

[3] Chao I-M, Golden B, Wasil E. A computational study of a new heuristic for the site-dependent vehicle routing problem. INFOR: Information Systems and Operational Research 1999;37(3):319–36.

[4] Li F, Golden B, Wasil E. Very large-scale vehicle routing: new test problems, algorithms, and results. Computers & Operations Research 2005;32:1165–79.

[5] Taillard E. A heuristic column generation method for the heterogeneous fleet VRP. RAIRO 1999;33(1):1–14.

[6] Tarantilis C, Kiranoudis C, Vassiliadis V. A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. Journal of the Operational Research Society 2003;54:65–71.

[7] Tarantilis C, Kiranoudis C, Vassiliadis V. A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. European Journal of Operational Research 2004;152:148–58.

[8] Dueck G. New optimization heuristics: the great deluge algorithm and the record-to-record travel. Journal of Computational Physics 1993;104: 86–92.
[9] Li F, Golden B, Wasil E. The open vehicle routing problem: algorithms, large-scale test problems, and computational results. Computers & Operations Research 2006; to appear.
[10] Golden B, Assad A, Levy L, Gheysens F. The fleet size and mix vehicle routing problem. Computers & Operations Research 1984;11:49–66.
[11] Li F. Solving variants of the vehicle routing problem: algorithms, test problems, and computational results. PhD dissertation, Department of Mathematics, University of Maryland, College Park, Maryland; 2005.
[12] Golden B, Wasil E, Kelly J, Chao I-M. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results. In: Crainic T, Laporte G, editors. Fleet management and logistics. Boston, MA: Kluwer; 1998. p. 33–56.