

Instituto de Engenharia de Sistemas e Computadores de Coimbra
Institute of Systems Engineering and Computers
INESC – Coimbra

Luís Santos

João Coutinho-Rodrigues

John Current

**An ellipse based heuristic for the
capacitated arc routing problem**

No. 5

2008

ISSN: 1645-2631

Instituto de Engenharia de Sistemas e Computadores de Coimbra
INESC – Coimbra
Rua Antero de Quental, 199; 3000-033 Coimbra; Portugal
www.inescc.pt

An ellipse based heuristic for the capacitated arc routing problem

Luís Santos^{a,d}, João Coutinho-Rodrigues^{b,d}, John R. Current^c

^aPhD Student, Department of Civil Engineering, Faculty of Sciences and Technology,
Polo II, University of Coimbra, 3030-788 Coimbra, Portugal
e-mail: lsantos@dec.uc.pt

^bDepartment of Civil Engineering, Faculty of Sciences and Technology,
Polo II, University of Coimbra, 3030-788 Coimbra, Portugal
e-mail: coutinho@dec.uc.pt

^cDepartment of Management Sciences, The Fisher College of Business, The Ohio State University,
632 Fisher Hall, 2100 Neil Avenue, Columbus, OH 43210-1144, USA
e-mail: current.1@osu.edu

^dINESC-Coimbra, Rua Antero Quental 199, 3000-033 Coimbra, Portugal

Abstract

The Capacitated Arc Routing Problem (CARP) is an important and practical problem in the OR literature. In short, the problem is to identify routes to service (e. g., pickup or deliver) demand located along the edges of a network such that the total cost of the routes is minimized. In general, a single route cannot satisfy the entire demand due to capacity constraints on the vehicles. CARP belongs to the set of NP-Hard problems; consequently numerous heuristic and metaheuristic solution approaches have been developed to solve it. In this paper an “ellipse rule” based heuristic is proposed for the CARP. This approach is based on the path-scanning heuristic, one of the mostly used greedy-add heuristics for this problem. The innovation consists basically of selecting edges only inside ellipses when the vehicle is near the end of each route. This new approach was implemented and tested on three standard test networks and the solutions are compared against: i) the original path-scanning heuristic, ii) two other path-scanning heuristics and iii) the three best known metaheuristics. The results indicate that the “ellipse rule” approach lead to improvements over the three path-scanning heuristics, reducing the average distance to the lower bound in the test problems by about 43%.

Keywords: vehicle routing, capacitated arc routing problem, heuristics

1. Introduction

Vehicle routing is an important activity in both the public and private sectors. In the United States for example, transportation costs account for approximately 6% of the GDP (MacroSys Research and Technology, 2005). As a consequence, even small improvements in routing efficiency can result in large cost reductions. Improving routing efficiency will become even more important as a result of recent increases in fuel prices. A common vehicle routing problem involves the pickup or delivery of items located along the arcs of a road network (e. g., trash collection and parcel delivery). Other examples of real applications can be founded in Assad and Golden (1995), Dror (2000), and Santos et al. (2008). Many of these applications can be structured as a Capacitated Arc Routing Problem (CARP).

Golden and Wong (1981) introduced CARP. The problem may be stated as follows.

Let $G = (N, E)$ be an undirected graph, where $N = \{v_0, \dots, v_n\}$ is a node set and $E = \{[v_i, v_j] : v_i, v_j \in N, i < j\}$ is an edge set. Each edge $[v_i, v_j]$ of E has a nonnegative cost or length c_{ij} and nonnegative demand or weight q_{ij} . Node v_0 represents a depot at which λ identical vehicles of capacity w ($w \geq \max q_{ij}$) are based. The number of vehicles (λ) is a decision variable. The CARP consists of designing a set of vehicle routes, such that: (1) each positive-demand edge is serviced by exactly one vehicle; (2) each route starts and ends at the depot; (3) the total demand of all edges serviced by any vehicle does not exceed w ; (4) the total routing cost is minimized.

CARP can be formulated as an integer linear programming and solved optimally using a branch-and-bound algorithm (e. g., Hirabayashi et al., 1992; Longo et al., 2006). Given that CARP belongs to the class of problems known as NP-Hard (Golden and Wong, 1981) such approaches are limited to small problem instances. For example, Hirabayashi et al. (1992) solved problems with up to 30 edges and Longo et al. (2006) with up to 87 edges.

As a result of CARP's many practical applications and its computational complexity, considerable research has been devoted to developing heuristic procedures to solve it. Path-scanning (Golden et al., 1983) is the most commonly used greedy-add solution approach to CARP. At each iteration, these heuristics "scan" the remaining unserved edges to determine which one should be visited next on the current vehicle route. This research introduces a new path-scanning heuristic that includes an "ellipse rule" to solve the problem. In essence, the ellipse rule only considers edges inside an ellipse when the vehicle is near the end of a route (i. e., when the vehicle load is nearing its capacity). This new heuristic was implemented and tested on three standard test networks. The results are compared to prior path-scanning heuristics and the three best known metaheuristics for the problem. When compared to the existing path-scanning heuristics, the experimental tests demonstrate that the new heuristic generates higher quality results in comparable CPU time. When compared to the metaheuristics, the experimental tests demonstrate that the new heuristic generates lower quality results, as expected, in considerably less

CPU time. Other reasons for the development of improved heuristics for CARP are presented in Section 2.1.

The remainder of this paper is organized as follows. Existing heuristic solutions for CARP are presented in the next section. The new ellipse-based heuristic is introduced in the third section. Computational results and comparisons are given in the fourth section followed by a summary and conclusions in the last section.

2. Existing heuristic solution procedures for CARP

2.1 General overview

Over the years, various heuristics (e. g., Golden et al., 1983; Golden and Wong, 1981; Pearn, 1989) and lower bound generating techniques (e. g., Belenguer and Benavent, 2003; Benavent et al., 1992; Golden and Wong, 1981; Longo et al. 2006) have been proposed for CARP. These heuristics (usually referred to as “greedy-add heuristics”) generally provide good approximate solutions in acceptable CPU time, given the computational complexity of the problem. Descriptions and computational performance comparisons of these heuristics are presented in Coutinho et al. (1993) which concluded that the CARP solution heuristic first introduced by Golden et al. (1983) is one of the best in terms of the quality of the solutions obtained and CPU time required to obtain the solutions.

More recently, metaheuristics have been proposed to solve CARP. These include tabu search (Hertz et al., 2000; Greistorfer, 2003), genetic algorithms (Lacomme et al., 2004a), and ant colony optimization (Lacomme et al., 2004b). In general, these metaheuristics identify better solutions than do the path-scanning heuristic. However, this improvement comes with an increase in solution time.

The development of improved greedy-add heuristics for CARP is still an important area of research even though the application of metaheuristics to CARP has led to improved solutions. Several reasons exist for this in addition to reduced solution times required. First, greedy-add heuristics are more intuitive as they mimic the way that people approach many problems. Consequently, they are more likely to be accepted by managers. Second, they are easier to implement (program and encode) and do not require the determination and fine tuning of various metaheuristic parameters such as mutation rate, aspiration levels, tabu list length, number of ants, evaporation coefficient, etc. Third, their simplicity makes them more flexible as they can be more easily modified to accommodate changes in the underlying problem (e. g., prohibited turns and maximum route duration). Finally, they form the starting point for various metaheuristics. For example, the genetic algorithm of Lacomme et al. (2004a) and Belenguer et al. (2006), and the algorithm based on ant colony optimization of Lacomme et al. (2004b) use the original and a modified path-scanning heuristic to generate one of its “good” starting solutions and the tabu search approach of Greistorfer (2003) starts with a greedy-add heuristic solution presented in Greistorfer (1995).

2.2 Previous path-scanning heuristics for CARP

In the original path-scanning heuristic (Golden et al., 1983) each solution is constructed by adding one edge at a time. In determining the edge to add, the heuristic considers variously edge length, edge demand, distance to the depot node, v_0 , distance to the next unserved edge, and unused capacity of the vehicle. Specifically, Golden et al. (1983) proposed five criteria to determine which edge to service next on a route. These criteria may be stated as follows:

1. minimize the ratio c_{ij} / q_{ij} ;
2. maximize the ratio c_{ij} / q_{ij} ;
3. minimize the cost from node v_j back to the node v_0 ;
4. maximize the cost from node v_j back to the node v_0 ;
5. if the vehicle is less than half-full use criterion 4, otherwise use criterion 3;

where v_i is the last node visited by the route to date and $[v_i, v_j]$ is a feasible (i. e., $q_{ij} \leq$ remaining vehicle capacity) unserved edge.

A problem instance is solved five times, using a different selection criterion each time. The best of the 5 solutions generated is the final solution. Golden et al. (1983) did not state which edge should be selected if all edges $[v_i, v_j]$ incident to the last node added (v_i) have no unserved demand. The generally accepted interpretation (e. g., Lacomme et al., 2004a; Belenguer et al., 2006; Evans and Minieka, 1992) in such situations is that the feasible unserved edge nearest to node v_i ($[v_p, v_j]$) is added, where nearest is measured by shortest path distance between node v_i and node v_p . The appropriate selection criterion is used whenever a tie for the nearest edge occurs (in this case, v_i is replaced by v_p in the criteria described above). This interpretation is used in this research.

Pearn (1989) modified this approach by selecting one of the 5 criteria at random, with equal probability, whenever a tie for the nearest edge (incident or not to the last node added to the route) occurs. Each problem was solved k times (results for $k = 30$ were published) and the best solution was selected at the end. One of the main advantages of this approach vis-à-vis the Golden et al. (1983) heuristic, is that it generally generates more solutions; consequently, the probability of identifying a better solution increases.

Recently, Belenguer et al. (2006) proposed another path-scanning heuristic. At each iteration, this heuristic adds the feasible unserved edge that is nearest to the last node added (v_i), as in the other path-scanning heuristics described above. Each problem is solved k times and the best solution is selected at the end. At any iteration, if there exist more than one feasible unserved edge that is the minimum distance from node v_i , these edges are referred to as “tied” edges. In such situations, Belenguer et al.

(2006) randomly select one of the tied edges to be the next edge added to the route. This differs from the Golden et al. (1983) and Pearn (1989) heuristics in that they would select the next edge to be added in such situations based upon the particular selection criterion in effect at the time.

Belenguer et al. (2006) compared their heuristic with the random selection of criteria (Pearn, 1989) using three standard test networks (referred to as: *gdb*, *val* and *egl*, and described in Section 4 below). Belenguer et al. (2006) presented results for $k=20$ and $k=50$. These results suggested that the random selection of tied edges is slightly worse for the *gdb* and *val* problems, and is slightly better for the *egl* problems. However, in all three problem sets, the differences are very small. In order to confirm these results, we solved the problems with $k=1000$, $k=10000$ and $k=20000$. Our results, presented in Table 1, confirm the results obtained by Belenguer et al. (2006).

3. A new “ellipse rule” based path-scanning heuristic

The new heuristic is similar to the one proposed by Belenguer et al. (2006) in that it randomly selects tied edges and solves each problem k times. It constructs a route by adding a new edge at each iteration. The edge added is the nearest edge to the last one added. In cases where there are ties for the nearest edge, the heuristic randomly selects one of the tied edges to add. The new heuristic differs from the Belenguer et al. (2006) one in that it utilizes an “ellipse rule” when the vehicle load is near capacity. Intuition suggests that as a vehicle’s load approaches its capacity, its route should stay closer to the depot to reduce its cost to return to the depot when full. The “ellipse rule” is designed to implement this intuition. When the vehicle is near the end of a route, this rule forces the vehicle to service only edges near the shortest path between the last serviced edge and the depot (v_0). This rule is similar to the one proposed by Norback and Love (1977) for solving the travelling salesman problem.

The “ellipse rule” is defined as follows. Let ned be the number of edges with positive demand in the network, td the total demand to be collected, tc the total cost assigned to edges with positive demand and $[v_h, v_i]$ the last serviced edge on the route. If the remaining capacity of the vehicle is less than or equal to td/ned , then the next edge to be serviced $[v_p, v_j]$ must be the nearest edge to $[v_h, v_i]$ ($v_i = v_p$, if the edges are adjacent) satisfying the condition:

$$SP(v_i, v_p) + c_{pj} + SP(v_j, v_0) \leq tc/ned + SP(v_i, v_0), \quad (1)$$

where, for example, $SP(v_i, v_p)$ is defined as the shortest path cost between the nodes v_i and v_p , and v_i and v_0 are the foci of the ellipse. If no feasible edge (i. e., an edge with demand less than or equal to the remaining capacity of the vehicle) satisfies (1) then the vehicle returns directly to the depot.

This rule is called the “ellipse rule” for the following reason. If “cost” is defined as “distance” and one measures the shortest path between node v_p and node v_0 via a straight line, then condition (1) forces the vehicle to collect edges inside the ellipse represented in Figure 1, where $c_2 + c_3 = 2c_1$.

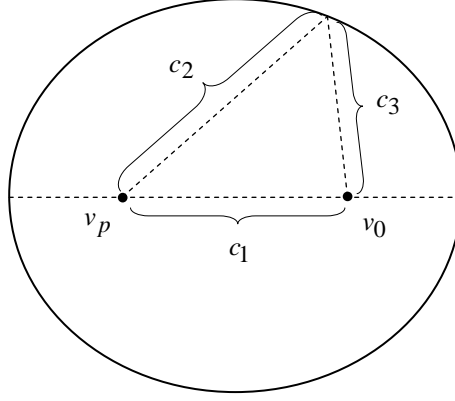


Figure 1 – Ellipse defined by the condition (1).

Let $nmit$ be the maximum number of iterations, S_{cur} the solution being constructed in the current iteration, S_{best} the best solution obtained so far, CS_{cur} and CS_{best} the total cost of solutions S_{cur} and S_{best} respectively, rvc the remaining vehicle capacity, w the vehicle capacity, LB the lower bound for the problem, and F the set of feasible unserved edges that are the minimum distance from the last node added to S_{cur} .

Given these definitions, the new path-scanning heuristic may be stated as follows.

```

 $CS_{best} \leftarrow +\infty$ 
 $iter \leftarrow 1$ 
while ( $(iter \leq nmit)$  and  $(CS_{best} > LB)$ ) do //if both the maximum nr. of iterations and the LB are not attained
     $S_{cur} \leftarrow v_0$ 
     $rvc \leftarrow w$ 
    for  $i = 1$  to  $ned$  do //for all the edges with positive demand
        if ( $rvc > td/ned$ ) then
            Determine  $F$ 
        else
            Determine  $F$  satisfying equation (1)
        end if
        if ( $F = \emptyset$ ) then
             $S_{cur} \leftarrow S_{cur} \cup \{v_0\}$  //the vehicle returns to the depot
             $rvc \leftarrow w$ 
        else
            Select  $[v_p, v_j]$  randomly from  $F$  //an additional edge is serviced in the solution being constructed
             $S_{cur} \leftarrow S_{cur} \cup \{[v_p, v_j]\} \cup \{v_j\}$ 
             $rvc \leftarrow rvc - q_{pj}$ 
        end if
    end for
    if ( $CS_{cur} < CS_{best}$ ) then //if the current solution is better than the best so far
         $S_{best} \leftarrow S_{cur}$ 
    end if
     $iter \leftarrow iter + 1$ 
end while.

```

4. Computational results

The new “ellipse rule” based heuristic, and the path-scanning heuristics of Golden et al. (1983), Pearn (1989), and Belenguer et al. (2006) were implemented and tested on the 23 problems proposed by DeArmon (1981) (*gdb* files), the 34 problems proposed by Benavent et al. (1992) (*val* files) and the 24 problems proposed by Belenguer and Benavent (2003) (*egl* files), based on real road networks used by Li and Eglese (1996). The first set of problems ranges from 7 to 27 nodes and from 11 to 55 edges. In keeping with general practice, problems 8 and 9 were removed from the 25 originally proposed because they contain inconsistencies. The second set of problems ranges from 24 to 50 nodes and from 34 to 97 edges. The third set of problems ranges from 77 to 140 nodes and from 98 to 190 edges. These data sets were used as they are standard test problems for CARP (data can be downloaded at <http://www.uv.es/~belengue/carp.html>). These problems were also solved with three of the best known metaheuristics. These include the Tabu search procedure of Hertz et al. (2000), the genetic algorithm-based procedure of Lacomme et al. (2004a), and the ant colony approach of Lacomme et al. (2004b). The solutions generated by the various heuristics and metaheuristics were compared to each other and to the lower bounds (LB) identified by Belenguer and Benavent (2003). The heuristics were run on a 1 GHz Pentium III with 368 K RAM (similar to the computers used by the three metaheuristics in order to facilitate comparisons).

The following notation will be used in presenting the results of these tests.

- PSG: original path-scanning heuristic with 5 criteria (Golden et al., 1983) (as implemented by Lacomme et al., 2004; Belenguer et al., 2006; Evans and Minieka, 1992);
- PSP (k): path-scanning heuristic modification of Pearn (1989) with 5 random selection criteria in each iteration (uniform distribution used) and k runs for each problem (as implemented by Lacomme et al., 2004; Belenguer et al., 2006; Evans and Minieka, 1992);
- RSE (k): path-scanning heuristic with random selection of tied edges and k runs for each problem as proposed and implemented by Belenguer et al. (2006);
- RSE_ER (k): “ellipse rule” based path-scanning heuristic, with k runs for each problem, introduced in this article;
- CARPET: metaheuristic proposed by Hertz et al. (2000) based on tabu search (CPU times were scaled to 1 GHz Pentium III PC by Lacomme et al., 2004a);
- MA: metaheuristic proposed by Lacomme et al. (2004a) based on genetic algorithms (CPU times were obtained by 1 GHz Pentium III PC);
- BACO: metaheuristic proposed by Lacomme et al. (2004b) based on ant colony optimization (CPU times were obtained by 800 MHz Pentium III PC).

The deviation percentage between the obtained solution cost (SC) and the LB, for each problem, are used to evaluate the solution quality (i. e., deviation percentage is equal to $((SC - LB) / LB) * 100\%$).

Table 1 presents the results obtained by two of the existing path-scanning heuristics, PSP and RSE, and the new approach, RSE_ER, for $k=1000$, $k=10000$ and $k=20000$. The PSG results are presented in Table 2. We conclude that the global results (i. e., overall averages for the problem sets) for the 81 problems obtained by RSE are similar to those obtained by PSP. The RSE results are slightly worse than those for PSP for the *gdb* and *val* files and are slightly better for *egl* files (for all three values of k). These results demonstrate that selecting among tied edges randomly results in solutions comparable to those generated by using an optimization criterion. The introduction of the “ellipse rule” (RSE_ER) significantly improves the quality of the solutions (the average distance to the LB was globally reduced by about 43% relatively to the PSP and RSE heuristics).

Table 1 – Average deviation to the LB (%) for the PSP (k), RSE (k) and RSE_ER (k).

	k	PSP (k)	RSE (k)	RSE_ER (k)
<i>gdb</i> files	1000	3.73	3.94	1.76
	10000	2.75	2.79	1.20
	20000	2.61	2.53	1.09
<i>val</i> files	1000	8.83	9.21	6.48
	10000	6.90	6.55	4.77
	20000	5.83	6.19	4.22
<i>egl</i> files	1000	18.53	18.30	9.91
	10000	16.91	16.63	8.70
	20000	16.64	16.35	8.61
Totals		82.73	82.49	46.74

Table 2 presents the results obtained by the three path-scanning heuristics, PSG, PSP and RSE, by the new approach, RSE_ER, and also by the best known metaheuristics (CARPET, MA and BACO). RSE and RSE_ER were tested for 5 iterations in order to attain results comparable with PSG (which is executed 5 times, one for each criterion). Conclusions are similar to those obtained from the results presented in Table 1. For example, comparing the ratios of average deviation to the LB one finds that the ratio of PSG to RSE_ER(5) solutions ranged from 1.04 to 1.64 and the ratio of RSE(5) to RSE_ER(5) ranged from 1.17 to 1.61, for the three problem sets. Similar ratios for the PSP(10000) to RSE_ER(10000) ranged from 1.45 to 2.29 and RSE(10000) to RSE_ER(10000) ranged from 1.37 to 2.33. On average, the new heuristic generated better solutions than did the existing path-scanning heuristics in all three problem sets. The times reported in Table 2 indicate that these improved results were attained with little or no additional CPU time.

Table 2 – Comparison between heuristics and metaheuristics approaches.

		Av. deviation to the LB (%)			Av. CPU time (sec.)		
		<i>gdb</i>	<i>val</i>	<i>egl</i>	<i>gdb</i>	<i>val</i>	<i>egl</i>
Heuristics	PSG	10.62	16.34	26.05	0.00	0.01	0.02
	RSE (5)	10.59	18.51	25.54	0.00	0.01	0.02
	RSE_ER (5)	7.86	15.77	15.84	0.01	0.02	0.03
	PSP (10000)	2.75	6.90	16.91	2.59	8.51	22.15
	RSE (10000)	2.79	6.55	16.63	2.41	8.45	21.85
	RSE_ER (10000)	1.20	4.77	8.70	3.02	6.62	22.96
Metaheuristics	CARPET	0.48	1.96	4.74	9.03	63.86	*
	MA	0.15	0.61	2.47	5.29	38.35	526.99
	BACO	0.28	0.90	4.11	24.77	276.32	2341.3

* Unknown CPU time

As expected, the metaheuristics generated better solutions, on average, than did the path-scanning heuristics. For example, the average ratios of average deviation to the LB for all of the problems of RSE_ER(10000) to CARPET, MA, and BACO were 2.26, 6.45, and 3.90 respectively. These improvements came at a large cost in increased solution times. For example, the ratios of average solution times for the problem sets for the metaheuristics to RSE_ER(10000) varied from 2.99 to 9.65 for CARPET, 1.75 to 22.95 for MA, and 8.20 to 101.97 for BACO.

5. Conclusions

Due to its practicality and computational complexity, the capacitated arc routing problem (CARP) has received considerable attention in the Operations Research literature. Several heuristics and metaheuristics have been developed to solve it. This article introduces a new path-scanning heuristic for the CARP. The heuristic constructs a route by adding feasible edges (i. e., have demand less than or equal to the remaining vehicle capacity) one at a time based on a nearest neighbor strategy. If there is a tie for the nearest neighbor, one of the tied edges is selected at random. As the vehicle nears the end of a route, an ellipse rule is invoked which constrains the options for the next edge added to those near the shortest path between the last serviced edged and the depot. The heuristic was tested on three well-known test networks and the solutions were compared to the lower bounds for the problems obtained by Belenguer and Benavent (2003) and to the solutions generated by the three best path-spanning heuristics for CARP (Golden et al., 1983; Pearn, 1989; and Belenguer et al., 2006) and the three best metaheuristics for the problem (Hertz et al., 2000; and Lacomme et al., 2004a,b).

The results of these tests indicate that a “nearest” edge selection rule and random selection of tied edges performs as well as the multi-criterion selection procedures proposed by Golden et al. (1983) and Pearn (1989). This supports the findings of Belenguer et al. (2006) and suggests that greedy-add heuristics should be compared to random-add heuristics during their testing and evaluation.

The results also indicate that the new path-scanning heuristic (RSE_ER) generates better quality solutions than do the existing path-scanning heuristics over all three standard problem sets. This improvement (about a 43% reduction in overall average distance to the lower bounds) comes with little or no increase in solution time. As expected, the new heuristic did not generate as good of solutions as do the metaheuristics. However, solution times were significantly longer for the metaheuristics. For example, on average they increased solution times by factors ranging from 1.75 for the smallest networks to 101.97 for the larger networks.

In general, metaheuristics generate better solutions for the CARP. However, the development of extremely fast heuristics is still an important area of research. This is true for several reasons. First, as Table 2 indicates, solution time for metaheuristics can become quite long for even medium sized problems like those in the test set. Real world problems may be much larger. Second, decisions makers often find greedy-add heuristics intuitive; consequently, they may be more willing to implement their results. Third, they do not need the determination of various parameters associated with metaheuristics. Fourth, their simplicity makes them easier to encode and to modify to accommodate various real-world modifications to the underlying problem. Finally, they are often used to identify “starting” solutions for metaheuristics.

Our research indicates that the new path-scanning heuristic presented in this article is the best path-scanning based heuristic to solve CARP. It can be used to either solve instances of CARP directly or to generate starting solutions for various metaheuristics.

6. References

- Assad, A.A., Golden, B.L., 1995. Arc routing methods and applications. In: Ball, M.O. et al. (Eds.), *Handbook in Operations Research and Management Science* 8, Elsevier, pp. 375-483.
- Belenguer, J.M., Benavent, E., 2003. A cutting plane algorithm for the capacitated arc Routing problem. *Computers and Operations Research* 30 (5), 705-728.
- Belenguer, J.M., Benavent, E., Lacomme, P., Prins, C., 2006. Lower and upper bounds for the mixed capacitated arc routing problem. *Computers and Operations Research* 33, 3363-3383.
- Benavent, E., Campos, V., Corberán, A., Mota, E., 1992. The capacitated arc routing problem: Lower bounds. *Networks* 22, 669-690.
- Coutinho-Rodrigues, J., Rodrigues, N., Clímaco, J., 1993. Solving an urban routing problem using heuristics: A successful case study. *Belgian Journal of Operations Research, Statistics and Computer Science*, special issue on “Combinatorial Optimization in Applications” 33, 19-32.
- DeArmon, J.S., 1981. A comparison of heuristics for the capacitated chinese postman problem. Master’s thesis, University of Maryland at College Park.

- Dror, M., 2000. Arc routing: Theory, solutions and applications. Kluwer Academic Publishers.
- Evans, J., Minieka, E., 1992. Optimization algorithms for networks and graphs. 2nd ed., Marcel Dekker, USA.
- Golden, B.L., Wong, R.T., 1981. Capacitated arc routing problems. *Networks* 11, 305-315.
- Golden, B.L., DeArmon, J., Baker, E.K., 1983. Computational experiments with algorithms for a class of routing problems. *Computers and Operations Research* 10, 47-59.
- Greistorfer, P., 2003. A tabu scatter search metaheuristic for the capacitated arc routing problem. *Computers and Industrial Engineering* 44 (2), 249-266.
- Greistorfer, P., 1995. Computational experiments with heuristics for a capacitated arc routing problem. In: Derigs, U., Bachem, A., Drexl, A. (Eds.), *Operations Research Proceedings 1994*, Springer-Verlag, Berlin et al., pp. 185-190.
- Hertz, A., Laporte, G., Mittaz, M., 2000. A tabu search heuristic for the capacitated arc routing Problem. *Operations Research* 48, 129-135.
- Hirabayashi, R., Saruwatari, Y., Nishida, N., 1992. Tour construction algorithm for the capacitated arc routing problems. *Asia-Pacific Journal Operations Research* 9, 155-175.
- Lacomme, P., Prins, C., Ramdane-Chérif, W., 2004a. Competitive memetic algorithms for arc routing problems. *Annals of Operations Research* 131, 159-185.
- Lacomme, P., Prins, C., Tanguy, A., 2004b. First competitive ant colony scheme for the CARP. Research Report LIMOS/RR-04-21. Available from: < <http://www.isima.fr/limos/publi/RR-04-21.pdf>>
- Li, L.Y.O., Eglese, R.W., 1996. An interactive algorithm for vehicle routeing for winter-gritting. *Journal of the Operational Research Society* 47, 217-228.
- Longo, H., Aragão, M.P., Uchoa, E., 2006, Solving capacitated arc routing problems using a transformation to the CVRP. *Computers and Operations Research* 33, 1823-1837.
- MacroSys Research and Technology, 2005. Logistics costs and U.S. gross domestic product. Federal Highway Administration Department of Transportation, August 25.
- Available at: http://ops.fhwa.dot.gov/freight/freight_analysis/econ_methods/lcdp_rep/index.htm.
- Norback, J.P., Love, R.F., 1977. Geometric approaches to solving the traveling salesman problem. *Management Science* 23 (11), 1208-1223.
- Pearn, W.L., 1989. Approximate solutions for the capacitated arc routing problem. *Computers and Operations Research* 16, 589-600.

Santos, L., Coutinho-Rodrigues, J., Current, J.R., 2008. Implementing a multi-vehicle multi-route spatial decision support system for efficient trash collection in Portugal. *Transportation Research Part A: Policy and Practice* 42 (6), 922-934.