A Hybrid Algorithm combining Path Scanning and Biased Random Sampling for the Arc Routing Problem

Sergio González[1], Angel A. Juan[1], Daniel Riera[1], and José Cáceres[1]

Estudis d'informàtica, Multimèdia i Telecomunicació
Open University of Catalonia - IN3
Barcelona, Spain
{sgonzalezmarti,ajuanp,drierat,jcaceresc}@uoc.edu

Abstract. The Arc Routing Problem is a kind of NP-hard routing problems where the demand is located in some of the arcs connecting nodes and should be completely served fulfilling certain constraints. This paper presents a hybrid algorithm which combines a classical heuristic with biased random sampling, to solve the Capacitated Arc Routing Problem (CARP). This new algorithm is compared with the classical Path scanning heuristic, reaching results which outperform it. As discussed in the paper, the methodology presented is flexible, can be easily parallelised and it does not require any complex fine-tuning process. Some preliminary tests show the potential of the proposed approach as well as its limitations.

Keywords: Arc Routing Problem, Combinatorial Optimisation, Hybrid Algorithms, Metaheuristics, Simulation.

1 Introduction

The Arc Routing Problem (ARP) is the counterpart to the Vehicle Routing Problem (VRP). In the latter, the demand is placed in nodes (i.e. clients) whereas in the ARP, it is located in arcs. Existing literature on ARP is not so extensive as for the VRP, although there are approaches to the VRP which can be adapted to the ARP, obtaining fairly good results. The objective of this research is to adapt the general idea proposed in [15], which were proposed for the Capacitated Vehicle Problem (CVRP) and apply them to the Capacitated Arc Routing Problem (CARP).

The structure of this paper is as follows. First, the CARP problem is stated, establishing some assumptions and the basic notation. Section 3 visits the existing literature to revise the state of the art. Later, in section 4, the proposed

Proceedings of the 18th RCRA workshop on *Experimental Evaluation of Algorithms for Solving Problems with Combinatorial Explosion* (RCRA 2011).
In conjunction with IJCAI 2011, Barcelona, Spain, July 17-18, 2011.

algorithm is introduced, presenting the classic Path Scanning algorithm proposed by Golden et al. in [9] first. Section 5 displays and discusses some results, and finally, section 6 states some conclusions and future work.

2 The Capacitated Arc Routing Problem

The Arc Routing Problem is a set of NP-hard problems where the objective is to determine a routing plan on a graph to serve a given set of nodes and arcs (also known as tasks), in contrast with the Vehicle Routing Problem, where the customer's demands occur in nodes and the arcs are only used to model paths interconnecting nodes. The CARP is a particular case of the ARP in which every vehicle serving a route has a maximum capacity.

The CARP problem is subjected to the following constraints:

(a) All routes begin and end at the depot,
(b) every vehicle has a maximum load capacity, which is considered to be the same for all vehicles,
(c) every arc with positive demand must be satisfied,
(d) every arc is served by a single vehicle,
(e) every arc can be traversed as many times as required (though it can only be served by a single vehicle),
(f) and the total routing cost is minimised.

2.1 Basic Notation and Assumptions

The CARP is defined over a non-complete graph $G = (V, E)$, where $V = \{0, 1, 2, \ldots, n\}$ represents the set of n nodes, and $E \subseteq A, A = \{(i, j)|i, j \in V; i \neq j\}$ represents the set of m arcs connecting pairs of nodes, in which the demand is located. A fleet of identical vehicles of capacity W, based on depot node 0 must serve the set of customers denoted by E. Every edge $e = (i, j)$ has a traversal cost or length c_{ij}, a positive or zero demand d_{ij} and, as undirected arc, can be traversed in both directions. Thus, a subset R of required edges E (or tasks) must be served by the fleet, which are those with positive demands, $d_{ij} > 0$.

In this scenario, the classical CARP goal is to determine a minimum length set of vehicle trips covering all the tasks with the following constraints:

(a) Every trip leaves the depot, visits a subset of tasks whose total demand does not exceed W and returns to the depot,
(b) and every task must be served by one and only one vehicle, though edges can be traversed multiple times.

3 Related work on the Capacitated Arc Routing Problem

The ARP has been widely studied, though existing publications on this field are considerably lower in number than those for the VRP. Some authors have published reviews of the advances performed on the ARP field. Particularly good results are [25], [14] and [5].

The study of the ARP was introduced on 1735, when Leonhard Euler presented his solution for the Königsberg bridge problem [12]. This problem, also known as the Euler Tour Problem, proposes, given a connected graph $G = (V, E)$, to find a closed tour visiting every edge in E exactly once, or to determine that no such tour exists.

The next ARP historically proposed was the Chinese Postman Problem (CPP), suggested on [21]. This is stated as given a connected graph $G = (V, E, C)$ where C is a distance matrix. The aim is to find a tour which crosses every edge at least once and does this in the shortest possible way. When G is completely directed or completely symmetric, it can be resolved in polynomial time [4].

Following the CPP, in [22], Orloff suggested the Rural Postman Problem (RPP), which is formally stated as follows. Given an undirected graph $G = (V, E, C)$ where C is the cost matrix for the edges, find a minimum cost tour which passes through every edge in a subset R of E at least once. It can be proved that the RPP is NP-hard [16] and its hardness comes from determining how the tour should connect the various components of R. After that, the Min-Max k-Chinese Postman Problem (MM k-CPP) can be found in [7]. In this case, given a connected undirected graph $G = (V, E, C)$ where C is a cost matrix with a special depot node, the aim is to find k tours, starting and ending in the depot node, such that every edge is covered by at least one tour and the length of the longest tour is minimised. It should be noted that for this problem the objective is to minimise the makespan, whereas most other problems with multiple postmen (or vehicles) seek to minimise the total distance travelled.

The CARP was introduced in [8], and was originally stated as follows. Given a connected undirected graph $G = (V, E, C, Q)$ where C is a cost matrix and Q is a demand matrix, and given a number of identical vehicles with capacity W, find the necessary tours such that:

(a) Every arc with positive demand is served by exactly one vehicle,
(b) the sum of demands on those arcs served by every vehicle does not exceed W,
(c) and the total cost of the tours is minimised.

This problem is considered as the classical CARP. It can be proved that the Capacitated Vehicle Problem (CVRP) can be transformed into a CARP [8], and the CARP can be transformed in CVRP [1], which make the two classes of problems equivalent, so algorithms used to solve one class can easily be adapted to solve the other, as we intend to do with our algorithm. For the transformation of CARP into CVRP, the resulting CVRP requires either fixing the variables or using edges with infinite cost. Furthermore, the resulting CVRP is a complete

graph of larger size than that of the original CARP. There also exists a problem in half way between CARP and CVRP, which is the Stringed CVRP [20], in which customers are to be served as in the CVRP but some of these customers are located along the streets.

During the eighties, problem specific heuristics were the most widely used methods for solving the CARP. They include the Construct-Strike algorithm, the Path-Scanning and the Augment-Merge algorithms [9]. The performance of those classical methods is generally 10% to 40% above the optimal solution. Pearn [23] proposed modified versions for those heuristics by adding several types of randomness, outperforming original heuristics. There exist several benchmarks to test the performance of the algorithms against the classical CARP, which can be downloaded from [2], and which will be used in this paper to test the proposed algorithm.

More recently, other problem specific heuristics have been proposed. Some of them are the Double Outer Scan heuristic [24], which combines the Augment-Merge and the Path-Scanning methods, and the Node Duplication heuristic [24], which uses similar ideas to those proposed in the Node Duplication Lower Bound [11].

Recently, most advances in development of heuristics for the classical CARP regard metaheuristics. Tabu Search algorithms have been constructed for solving the CARP. The first, called CARPET, was proposed in [13]. In it, unfeasible solutions are allowed but are also penalised. This algorithm outperformed the existing ones and is still one of the best performing algorithms for CARP. Also a combination of Tabu Search and Scatter Search to construct Tabu Scatter Search was proposed by Greistorfer [10]. Lacomme presented both a Genetic Algorithm [17] and a Memetic Algorithm [18]. In both algorithms crossover is performed on a giant tour, and fitness of a chromosome is based on the partitioning of the tour into vehicle tours. Currently these algorithms are among the very best performing solutions for the CARP.

Another even younger generation of metaheuristics is that of the Ant Colony Systems. Lacomme [18] proposes an algorithm where two types of ants are used, one that makes the solution converge towards a minimum cost solution and another which ensures diversification to avoid getting trapped in a local minimum. A Guided Local Search algorithm is proposed in [3], suggesting that the distance of each edge is penalised according to some function which is adjusted throughout the algorithm. Computational experiments shows that this approach is promising.

4 Proposed algorithm

In this section the proposed algorithm is detailed. To do that, first of all the Path Scanning algorithm is reviewed to present later our approach on the Randomised Path Scanning.

4.1 Path Scanning algorithm

The Path Scanning algorithm [9] is a simple and efficient algorithm which aims to get competitive results for CARP in low computational times. Its main idea is to construct five complete solutions, every one of them following an optimisation criterion. The final solution of the algorithm is the best —in terms of cost— of the five obtained.

The way every route is constructed is not clearly defined in the original Golden paper, so it allows different interpretations when trying to implement it. The approach followed in this paper is to extend the current route by selecting only adjacent arcs with unserved demand, selecting that which best accomplishes the given criterion. Those five criterion consider that the vehicle is at node i and the route through the selected arc e to the node j:

(1) Minimise the cost per unit demand ($min\{c_{ij}/d_{ij}\}$)
(2) Maximise the cost per unit demand ($max\{c_{ij}/d_{ij}\}$)
(3) Minimise the distance from node j back to depot.
(4) Maximise the distance from node j back to depot.
(5) If vehicle is less than half-full, minimise distance from node j back to depot, otherwise maximise this distance.

In the case of non adjacent arcs with existing unserved demand, the closest arc —in terms of the shortest path distance— is selected. If there exists more than one arc at the same minimum distance, then the best arc accomplishing the current optimisation criterion is selected.

Finally, once the vehicle capacity is exhausted, the current route is closed by returning the vehicle to the depot through the shortest path. The original algorithm does not state how the shortest path is computed. In our approach an implementation of Dijkstra's algorithm is used.

4.2 Randomised Path Scanning

Recent advances in development of high-quality pseudo-random number generators have opened perspectives regarding the use of Monte Carlo Simulation (MCS) in combinatorial problems. As stated previously in this paper, the idea behind our algorithm is based on that from Juan et al. [15] for the CVRP. In that paper, a classical heuristic for the CVRP, as the Clarke and Wright Savings (CWS) heuristic, was chosen and combined with the MCS methodology. Thus, some random behaviour was introduced to the CWS heuristic in order to start an efficient search process inside the space of feasible solutions, which allows to improve original CWS results.

Notice that this general approach has similarities with the Greedy Randomised Adaptive Search Procedure (GRASP) [6]. But our approach does not contain an expensive local search phase and includes a more detailed randomised construction step.

In the studied case, the Path Scanning heuristic for CARP has been chosen and is combined with MCS to add randomness allowing it to reach better results.

With that, the Randomised Path Scanning (RPS) is obtained. In the proposed algorithm, two random processes are introduced into the original algorithm:

(1) When constructing every solution, the optimisation criterion to select the next arc is not known beforehand. A criterion is randomly selected, with uniform probability distribution ([23] states that it gets better results than other probability distributions).
(2) When selecting the next arc, the arc which better accomplishes the selected criterion is not chosen by default. All the candidate arcs are sorted following the selected criterion and a weight is given to every one of them, following a geometrical distribution. Thus, the next arc is selected randomly with a geometric probability distribution.

With this randomisation, many valid solutions can be generated. An efficient search process inside the feasible solutions is started where each of these feasible solutions consists on a set of round-trip routes from the depot that, altogether, satisfy the demand of the arcs.

Pseudo-code The algorithm is implemented as described next. First, the problem instance is loaded from the data files. Next, the arcs are extracted from the problem instance and stored in a static structure. After that, all the shortest paths between all pairs of nodes are computed using a Dijkstra implementation for the shortest path algorithm. A loop constructing complete solutions is started and finally the best solution among all the generated solutions, is selected.

```
procedure RandPS;
begin
  arp = getInstanceInputs();
  arcs = getArcs(arp);
  paths = constructShortestPathsMatrix(arcs);
  while stopping criterion not satisfied
    sol = buildRandomizedSolution(arcs, paths);
    if sol.cost < bestSolution.cost
      bestSolution = sol;
    end if
  end while
  return bestSolution;
end
```

5 Results

The implementation of the RPS algorithm has been done as a Java application, using some state-of-the-art pseudo-random number generator. In particular, some classes from the SSJ library [19] were implemented, among them the

LFSR113 with a period of approximately 2^{113}. To test the new algorithm, instances from [2] have been used, which are based on those of [9] and will allow the result comparison with the original Path Scanning algorithm.

Table 1 shows results obtained with the *gdb* instances. The Path scanning solutions were obtained from Golden's original article [9]. RPS results and times are obtained from the Java implementation described in previous section, generating 10000 solutions on the loop which selects the best one.

Problem	Nodes	Arcs	LB	BKS	PS	PS Time	RandPS	RandPS Time	Gap (%)
gdb1	12	22	316	316	**316**	0,005	**316**	0,61	0,00
gdb2	12	26	339	339	367	0,006	**339**	0,71	7,63
gdb3	12	22	275	275	289	0,003	**275**	0,63	4,84
gdb4	11	19	287	287	320	0,002	**287**	0,51	10,31
gdb5	13	26	377	377	417	0,002	383	0,76	8,15
gdb6	12	22	298	298	316	0,001	**298**	0,56	5,70
gdb7	12	22	325	325	357	0,003	**325**	0,57	8,96
gdb8	27	46	344	348	416	0,015	358	2,45	13,94
gdb9	27	51	303	303	355	0,017	324	2,74	8,73
gdb10	12	25	275	275	302	0,003	**275**	0,62	8,94
gdb11	22	45	395	395	424	0,003	**395**	1,90	6,84
gdb12	13	23	458	458	560	0,001	**490**	0,58	12,50
gdb13	10	28	536	536	592	0,002	**536**	0,68	9,46
gdb14	7	21	100	100	102	0,001	**100**	0,46	1,96
gdb15	7	21	58	58	**58**	0,001	**58**	0,42	0,00
gdb16	8	28	127	127	131	0,002	**127**	0,68	3,05
gdb17	8	28	91	91	93	0,002	**91**	0,66	2,15
gdb18	9	36	164	164	168	0,003	**164**	0,92	2,38
gdb19	11	11	55	55	57	0,001	**55**	0,23	3,51
gdb20	11	22	121	121	125	0,002	**121**	0,53	3,20
gdb21	11	33	156	156	168	0,002	**156**	0,89	7,14
gdb22	11	44	200	200	207	0,003	**200**	1,33	3,38
gdb23	11	55	233	233	241	0,005	235	1,80	2,49

Table 1. Results obtained with gdb instances. LB=Lower Bound; BKS=Best Known Solution obtained from [18]. Times expressed in seconds. Bold indicates that it achieves the BKS and underline that it outperforms PS solution.

6 Conclusions and future work

From the obtained results it can be seen that the classical Path Scanning is outperformed by the new RPS. Furthermore, competitive results are obtained in small-medium sized instances, so the new algorithm accomplishes the main objective of this research, which is to prove if the ideas from [15] for the CVRP are valid for the CARP.

Future work in this research will be to add splitting and cache techniques to the algorithm, trying to improve results and optimize the algorithm. Due to the independence of all the generated solutions, the algorithm could easily be parallelised in order to improve its performance when attempting to solve larger instances of CARP problems.

An additional future objective of the research is to apply the algorithm to different variants inside the ARP, specially the Arc Routing Problem with Stochastic Demand (ARPSD) since having the proposed algorithm randomisation, we think that the RPS algorithm will be well suited for this problem with random behaviour on the arcs' demand.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation (TRA2010-21644-C03), and has been developed in the context of the CYTED-IN3-HAROSA network (http://dpcs.uoc.edu).

References

[1] A.A. Assad, B.L. Golden and W.L. Pearn. Transforming arc routing into node routing problems. *Computers and Operations Research, 14(4):285-288, 1987.*

[2] J.M. Bleneguer. *http://www.uv.es/belengue/carp.html*

[3] P. Beullens, D. Cattrysse, L. Muyldermans and D. Van Oudheusden. A guided local search heuristic for the capacitated arc routing problem. *European Journal of Operational Research, 147:629-643, 2003.*

[4] N. Christofides. The optimum traversal of a graph. *OMEGA, The International Journal of Management Science, 1(6):719732, 1973.*

[5] A. Corberán and C. Prins. Recent results on arc routing problems: an annotated bibliography. *Networks, 56(1):50-69, 2010.*

[6] T.A. Feo and M.G. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization, 6:109-133, 1995.*

[7] G.N. Frederickson, M.S. Hecht and C.E. Kim. Approximation algorithms for some routing problems. *SIAM Journal of Computing, 7(2):178-193, 1978.*

[8] B.L. Golden and R.T. Wong. Capacitated arc routing problems. *Networks, 11:305-315, 1981.*

[9] B.L. Golden, J.S. DeArmon and E.K. Baker. Computational experiments with algorithms for a class of routing problems. *Computers and Operations Research 10:47-59, 1983.*

[10] P. Greistorfer. A Tabu Scatter Search Metaheuristic for the Arc Routing Problem. *Computers & Industrial Engineering, 44:249-266, 2003.*

[11] R. Hirabayashi, N. Nishida and Y. Saruwatari. Node duplication lower bounds for the capacitated arc routing problems. *Journal of the Operations Research Society of Japan, 35(2):119-133, 1992.*

[12] H. Sachs, M. Stiebitz and R.J. Wilson. An historical note: Euler-s Königberg letters. *Journal of Graph Theory, 12(1):133-139, 1988.*

[13] A. Hertz, G. Laporte and M. Mittaz. A tabu search heuristic for the capacitated arc routing problem. *Operations Research, 48(1):129-135, 2000.*

[14] A. Hertz. Recent Trends in Arc Routing. in *Graph theory, Combinatorics and algorithms: Operations research/computer science interfaces series, M.C. Golumbic and I.B.A Hartman. 2005.*

[15] A.A. Juan, J. Faulin, R. Ruiz, B. Barrios and S. Caballé. The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem. *Applied Soft Computing, 10:215-224, 2010.*

[16] A.H.G. Rinnooy Kan and J.K. Lenstra. On general routing problems. *Networks, 6:273-280, 1976.*

[17] P. Lacomme, C. Prins and W. Ramdana-Chérif. Competitive genetic algorithms for the capacitated arc routing problem and its extensions. *Lecture Notes in Computer Science, 2037:473-483, 2001.*

[18] P. Lacomme, C. Prins and W. Ramdane-Chérif. Competitive memetic algorithms for arc routing problems. *Annals of Operations Research, 131:159-185, 2004.*

[19] P. L'Ecuyer. SSJ: A framework for stochastic simulation in Java. *Proceedings of the Winter Simulation Conference, 2002, 234-242.*

[20] A. Løkketangen and J. Oppen. Arc routing in a node routing environment. *Computers and Operations Research, 33(4):1033-1055, 2006.*

[21] K. Mei-Ko. Graphic programming using odd or even points. *Chinese Mathematics, 1:237-277, 1962.*

[22] C.S. Orloff. A fundamental problem in vehicle routing. *Networks, 4:35-64, 1974.*

[23] W.L. Pearn. Approximate solutions for the capacitated arc routing problem. *Computers & Operations Research, 16(6):589-600, 1989.*

[24] S. Wøhlk. Contributions to arc routing. *PhD thesis, University of Southern Denmark, 2005.*

[25] S. Wøhlk. A decade of the capacitated arc routing problem. *The Vehicle Routing Problem: Latest Advances and New Challenges. Springer 2010.*