

# An Improved Clarke and Wright Algorithm to Solve the Capacitated Vehicle Routing Problem

Louis Caccetta

Dpt of Mathematics and Statistics  
Curtin University of Technology  
Australia  
l.caccetta@curtin.edu.au

Mamoon Alameen

Dpt of Engineering  
The Australian College of Kuwait  
Kuwait  
m.radiy@ack.edu.kw

Mohammed Abdul-Niby

Dpt of Engineering  
The Australian College of Kuwait  
Kuwait  
m.nibi@ack.edu.kw

**Abstract**— This paper proposes an effective hybrid approach that combines domain reduction with the Clarke and Wright algorithm to solve the capacitated vehicle routing problem. The hybrid approach is applied to solve 10 benchmark capacitated vehicle routing problem instances. The dimension of the instances was between 21 to 200 customers. The results show that domain reduction can improve the classical Clarke and Wright algorithm by about 18%. The hybrid approach improves the large instances significantly in comparison with the smaller size instances. This paper will not show the time taken to solve each instance, as the Clarke and Wright algorithm and the hybrid approach took almost the same CPU time.

**Keywords**- Clarke and Wright; capacitated vehicle routing problem; domain reduction

## I. INTRODUCTION

The Vehicle Routing Problem (VRP) is an important problem in the distribution network and has a significant role in cost reduction and service improvement. The problem is one of visiting a set of customers using a fleet of vehicles, respecting constraints on the vehicles, customers, drivers etc [1]. The goal is to produce a minimum cost routing plan specified for each vehicle. The problem of vehicle scheduling was first formulated in 1959 [2] and may be stated as a set of customers, each with a known location and a known requirement for some commodity, that is to be supplied from a single depot by delivery vehicles, subject to the following conditions and constraints:

- The demands of all customers must be met.
- Each customer is served by only one vehicle.
- The capacity of the vehicles may not be violated (for each route the total demands must not exceed the capacity).

The objective of a solution may be stated, in general terms, as that of minimizing the total cost of delivery, namely the costs associated with the fleet size and the cost of completing the delivery routes [3]. The problem frequently arises in many diverse physical distribution situations. For example bus routing, preventive maintenance inspection tours, salesmen routing and the delivery of any commodity such as mail, food

or newspapers [4]. The vehicle routing problem is an integer programming problem that falls into the category of NP-Hard problems. As the problems become larger, there will be no guarantee that optimal tours will be found within reasonable computing time [5].

## II. PROBLEM FORMULATION

The Capacitated Vehicle Routing Problem (CVRP) is to satisfy the demand of a set of customers using a fleet of vehicles with minimum cost. The problem is described as follows [4]:

Let:

- $C = \{1, 2, \dots, n\}$ : the set of customer location.
- 0: depot location.
- $G = (N, E)$ : the graph representing the vehicle routing network with  $N = \{0, 1, \dots, n\}$  and  $E = \{(i, j) : i, j \in N, i < j\}$ .
- $q_j$ : demand of customer  $j$ .
- $Q$ : common vehicle capacity.
- $m$ : number of delivery vehicles.
- $C_{ij}$ : distance or associated cost between locations  $i$  and  $j$ .
- $L$ : maximum distance a vehicle can travel.
- $P_j$ : a lower bound on the cost of travelling from the depot to customer  $j$ .
- $\ell(S)$ : lower bound on the number of vehicles required to visit all locations of  $S$  in an optimal solution. Note that  $S \subseteq C$  and  $\ell(S) \geq 1$ .
- $\bar{S}$ : the complement of  $S$  in  $C$
- $x_{ij}$ : 1, 2, or 0

The problem is to:

$$\bullet \text{ minimize } Z = \sum_{i \in N} \sum_{i < j} c_{ij} x_{ij} \quad i \in N, i < j \quad (1)$$

subject to:

$$\bullet \sum_{i \in C} x_{0i} = 2m, i \in C \quad (2)$$

$$\bullet \sum_{j < i} x_{ij} + \sum_{i < j} x_{ji} = 2, i \in C \quad (3)$$

$$\bullet \sum x_{ij} \leq |S| - \ell(S), i, j \in S, S \subseteq C, 3 \leq |S| \leq n - 2 \quad (4)$$

$$\bullet x_{ij} = 1, 2, \text{ or } 0 \quad (5)$$

III. THE CLARKE AND WRIGHT ALGORITHM

The Clarke and Wright algorithm [6] is the most popular heuristic algorithm for the VRP. The algorithm calculates all the savings  $S_{ij}$  between customers  $i$  and  $j$ . Assuming that  $c_{i0}$  is the cost of travelling from the depot to customer  $i$  and  $c_{ij}$  is the cost of travelling from customer  $i$  to  $j$ . The following is a description of the Clarke and Wright algorithm to solve the CVRP:

- **Step 1:** Compute the savings  $S_{ij} = C_{i0} + C_{0j} - C_{ij}$  for  $i, j = 1, \dots, n$  and  $i \neq j$ . Rank the savings  $S_{ij}$  and list them in descending order.
- **Step 2:** Create the "savings list." Process the savings list beginning with the topmost entry in the list (the largest  $S_{ij}$ ). For the savings under consideration ( $S_{ij}$ ), include link  $(i, j)$  in a route if no route constraints will be violated through the inclusion of  $(i, j)$ . The following three cases need to be considered.
  - **Case 1:** If neither  $i$  nor  $j$  have already been assigned to a route, then a new route is initiated including both  $i$  and  $j$ .
  - **Case 2:** If exactly one of the two points ( $i$  or  $j$ ) has already been included in an existing route and that point is not interior to that route (a point is interior to a route if it is not adjacent to the depot in the order of traversal of points), then the link  $(i, j)$  is added to that same route. If the point is interior and not violating the capacity then add  $(i, j)$  to the same route. If it's violating the capacity make a new route with the point (customer)  $i$ .
  - **Case 3:** If both  $i$  and  $j$  have already been included in two different existing routes and neither point is interior to its route, then the two routes are merged by connecting  $i$  and  $j$ . If they are interior then the merge cannot be done.
- **Step 3:** If the savings list  $S_{ij}$  has not been exhausted, return to Step 2, processing the next entry in the list; otherwise, stop.

IV. DOMAIN REDUCTION

Observing the work of [7-11], one can see that some significant work has been done to improve the accuracy of the classical algorithms. However, one of the weaknesses in classical algorithms lays in accuracy, especially when dealing with large instances. This paper shows that domain reduction can solve the problem of the accuracy for large instances.

Domain reduction is a method that deletes some unneeded values from the domain using a logical based approach [12]. In this paper the domain reduction will be applied by adding a new constraint that deletes some large numbers from the distance matrix and thus forbids the use of certain links. The new restriction is:

$$c_{ij} \leq R \quad i, j = 1, 2, \dots, n$$

where  $c_{ij}$  represent the cost between  $i$  and  $j$ , and  $R$  is a threshold that depends on the maximum number in the distance matrix. The domain reduction procedure starts by dividing the distance matrix into four parts based on the distance. In the other hand the demand matrix will be divided into four parts starting from the lowest to the greatest.

Customers with lowest demands and high (or very high) costs will be highlighted as critical customers. In the other hand, customers with high (or very high) demands and with lowest (or average) costs will be referred to as leading customers. The domain reduction will delete all the high and very high costs from the cost matrix. For the critical customers we delete the highest 50% of the edges. Taking the obtained solution in consideration, the same procedure should be repeated by applying the domain reduction on the remaining values.

The domain reduction procedure will delete some unneeded values from the distance matrix. This will improve the solution especially when combined with the Clarke and Wright algorithm. Table I illustrates how the domain reduction procedure divides the cost and demand matrices.

TABLE I. DOMAIN REDUCTION LOGIC

<b>Very high cost path</b>	<b>Very high demands</b>
<b>High cost path</b>	<b>High demands</b>
<b>Average cost path</b>	<b>Average demands</b>
<b>Lowest cost path</b>	<b>Lowest demands</b>

V. COMPUTATIONS

This section combines the domain reduction with the Clarke and Wright algorithm. The algorithm is employed to solve 10 benchmark VRP instances [13-16]. A look up table for the instances used can be seen in Table II. The details of each instance can also be found in [17]. Table III shows the obtained results.

TABLE II. LOOK-UP TABLE FOR THE 10 BENCHAMRK VRP INSTANCES

<b>Instance</b>	<b>Reference</b>
1-3	13
4	14
5-6	15
7-10	16

TABLE III. THE C&amp;W SAVING ALGORITHM AND DOMAIN REDUCTION

Customer number	Classical C&W	Modified C&W	Domain reduced	Improvement percentage	Simulating Annealing
21	3754	3658	62%	2.7%	3314
24	1659	1404	70%	15.3%	1473
26	1891	1888	50%	0.15	1779
31	1336	1278	5%	4.3	1269
42	2391	1747	55%	26.9	1528
48	19342	19181	65%	0.83	17888
101	803	672	25%	16%	409.918
121	934	574	30%	38%	336.485
151	958	480	45%	49.8	368.996
200	1291	697	45%	46%	652.158

Table III shows that domain reduction improved the classical Clarke and Wright rapidly. It's clear that as the instances become larger, the solution improves better and get closer to simulating annealing results. Surveying the results for the last five instances one can see that the improved Clarke and Wright provides better results on large instances. The 48 customers problem improved by about 1% but other large instances show better improvements. The 101 customers problem improved by 16%. The 121 customers problem improved by 38%. The problem 151 customers problem improved by 49.8%. For the 200 customers problem, the improvement was 46%.

## VI. CONCLUSION

In general, the saving approach used by the Clarke and Wright algorithm can provide good solutions for small size instances. However, for large instances calculating the savings may consider large values which affect the solution. Hence, combining domain reduction with the Clarke and Wright algorithm is far better approach than using the Clarke and Wright algorithm alone to solve large instances. In addition, classical heuristics are easy to understand and implement compared to meta-heuristics. However, meta-heuristics can provide better results in most of the cases. This paper showed that classical algorithms can also provide accurate results.

## REFERENCES

- [1] C. Groër, Parallel and serial algorithms for vehicle routing problems, PhD Thesis, University of Maryland, ProQuest, 2008
- [2] G. B. Dantzig, J. H. Ramser, "The truck dispatching problem", *Management Science*, Vol. 6, No. 1, pp. 80-91, 1959
- [3] N. Christofides, S. Eilon, "An algorithm for the vehicle dispatching problem", *Operational Research*, Vol. 20, No. 3, pp. 309-318, 1969
- [4] N. R. Achuthan, L. Caccetta, S. P. Hill, "A new subtour elimination constraint for the vehicle routing problem", *European Journal of Operational Research*, Vol. 91, No. 3, pp. 573-586, 1996
- [5] N. R. Achuthan, L. Caccetta, "Integer linear programming formulation for a vehicle routing problem", *European Journal of Operational Research*, Vol. 52, No. 1, pp. 86-89, 1991
- [6] G. Clarke, J. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research*, Vol. 12, No. 4, pp. 568-581, 1964
- [7] A. A. Juan, J. Faulin, R. Ruiz, B. Barrios, S. Caballé, "The SR-GCWS hybrid algorithm for solving the capacitated vehicle routing problem", *Applied Soft Computing*, Vol. 10, No. 1, pp. 215-224, 2010
- [8] A. A. Juan, J. Faulin, J. Jorba, D. Riera, D. Masip, B. Barrios, "On the use of Monte Carlo simulation, cache and splitting techniques to improve the Clarke and Wright savings heuristics", *Journal of the Operational Research Society*, Vol. 62, No. 6, pp. 1085-1097, 2011
- [9] T. Pichpibul, R. Kawtummachai, "New enhancement for Clarke-Wright savings algorithm to optimize the capacitated vehicle routing problem", *European Journal of Scientific Research*, Vol. 78, No. 1, pp. 119-134, 2012
- [10] K. Kanthavel, P. Prasad, "Optimization of capacitated vehicle routing problem by nested particle swarm optimization", *American Journal of Applied Sciences*, Vol. 8, No. 2, pp. 107-112, 2011
- [11] L. Xiaodong, "A summarization of algorithms for vehicle routing problem with time windows", *European Journal of Scientific Research*, Vol. 73, No. 3, pp. 296-309, 2012
- [12] A. Caprara, M. Locatelli, "Global optimization problems and domain reduction strategies", *Mathematical Programming*, Vol. 125, No. 1, pp. 123-137, 2010
- [13] M. Groetschel, C. L. Monma, M. Stoer, "Computational results with a cutting plane algorithm for designing communication networks with low connectivity constraints", *Operations Research*, Vol. 40, No. 2, pp. 309-330, 1992
- [14] S. Eilon, C. D. T. Watson-Gandy, N. Christofides, *Distribution management: mathematical modelling and practical analysis*, Griffin, London, 1971
- [15] M. Held, R. M. Karp, "The traveling salesman problem and minimum spanning trees", *Operations Research*, Vol. 18, No. 6, pp. 1138-1162, 1970
- [16] N. Christofides, A. Mingozzi, P. Toth, "The vehicle routing problem", in *Combinatorial Optimization*. Wiley, Chichester, pp. 315-338, 1979
- [17] Branch Cut and Price Resource Web, Vehicle Routing Data Sets USA, <http://www.coin-or.org/SYMPHONY/branchandcut/VRP/data>