Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

SR-1: A SIMULATION-BASED ALGORITHM FOR THE CAPACITATED VEHICLE ROUTING PROBLEM

Javier Faulin

Dep. of Statistics and Operations Research
Public University of Navarre
Campus Arrosadia
Pamplona, 31006, SPAIN

Miquel Gilibert
Angel A. Juan
Xavier Vilajosana

Dep. of Computer Science, Multimedia and Telecommunication
Open University of Catalonia
Rambla Poblenou, 156
Barcelona, 08018, SPAIN

Rubén Ruiz

Instituto Tecnológico de Informática
Dep. of Statistics, Applied Operations Research and Quality
Valencia University of Technology
Camino de Vera, s/n, Edificio 7A
Valencia, 46022, SPAIN

ABSTRACT

In this paper we present SR-1, a simulation-based heuristic algorithm for the Capacitated Vehicle Routing Problem (CVRP). Given a CVRP instance, SR-1 uses an initial "good solution", such as the one provided by the classical Clarke and Wright heuristic, in order to obtain observations for the variable "distance between two consecutive nodes in a route". These observations are then fitted by a statistical distribution, which characterizes the inter-node distances in good solutions. Then, the fitted distribution is employed to generate a large number of new random solutions with similar edge-size distribution. Thus, a random but oriented local search of the space of solutions is performed, and a list of "best solutions" is obtained. This list allows considering several properties per solution, not only aprioristic costs, which can be practically used when making multiple-criteria decisions. Several tests have been performed to discuss the effectiveness of this approach.

1 INTRODUCTION

The purpose of this paper is to introduce SR-1 (acronym for SimuRoute algorithm, first release), an algorithm that uses Monte Carlo Simulation (MCS) to generate routes and solutions for the Capacitated Vehicle Routing Problem (CVRP), which is described in the following.

We assume a set Ω of $n+1$ nodes, each of them representing a vehicle destination or a delivery point. The nodes are numbered from 0 to n, being node 0 the depot and the remaining n nodes the delivery points. In this context, d_{ij} represents the distance (cost) between the nodes i and j. These d_{ij} are assumed to be symmetrical distances ($d_{ij} = d_{ji}$, $0 \leq i, j \leq n$) verifying the triangular inequality. Additionally, a demand $q_i > 0$ of some commodity has been assigned to each non-depot node i. The delivery process is to be carried out by a fleet of NV vehicles with equal capacity, $c >> max\{q_i\}$. The main goal of this problem is the construction of several routes, one for each active (non-idle) vehicle, which minimizes the sum of the total distances (costs) involved in the delivery process.

Although the CVRP constraints are clearly well-known (Toth and Vigo 2002), we define them as follows:

(i) Each non-depot node is supplied by a single vehicle
(ii) All vehicles begin and end their routes at the depot (node 0)
(iii) A vehicle cannot stop twice at the same non-depot node
(iv) No vehicle can be loaded exceeding its maximum capacity

The characteristics of this CVRP are shown in Table 1. It is worth drawing attention to the following details in the CVRP formulation:

The objective function is defined by the sum of the (usually Euclidean) distances or costs between the nodes visited by each vehicle, taking all the vehicles into account.

Constraints (i), (ii) and (iii) are typical in any VRP definition while constraint (iv) is specific for CVRP.

Table 1: Characteristics of CVRP

Characteristics of VRP	Options for the CVRP
Size of fleet	Multiple vehicles with capacity constraints
Type of fleet	Homogeneous
Origin of vehicles	Single depot
Kind of demand	Known deterministic demand
Location of demand	At each node
Kind of network	Non-oriented. Symmetrical in distances
Maximum time on route	No constraint
Activities	Only deliveries
Constraints	(i), (ii), (iii) and (iv)
Objective	Minimize distances (costs)

The rest of the paper is structured as follows: Section 2 discusses the use of MCS in CVRP; Section 3 describes some constructive algorithms for the CVRP; Section 4 reviews some important CVRP literature; Section 5 introduces the main ideas behind our approach; Section 6 presents the SR-1 algorithm in detail; Section 7 explains how this algorithm has been implemented by using an object-oriented approach; Section 8 discusses some experimental results; finally, Section 9 highlights the originality and advantages of our approach over the existing ones.

2 USE OF MC SIMULATION IN VRP

Monte Carlo simulation can be defined as a set of techniques that make use of random number generation to solve certain stochastic or deterministic problems (Law 2007). MCS has proved to be extremely useful for obtaining numerical solutions to complex problems which cannot be efficiently solved by using analytical approaches. The Monte Carlo term was first applied to a class of mathematical algorithms by scientists working on the development of nuclear weapons in Los Alamos, during the period ranging from 1943 to 1946 (Gass and Assad 2005a). The effectiveness of numerical or simulated gambling, as a serious scientific pursuit, has been enormously enhanced by the power of modern computers and, therefore, MCS has found extensive use during the last twenty five years in Operations Research and Statistics (Fishman 1996, Gass and Assad 2005b). Other applications of MCS can be found in areas such as Physics, Biology, Medicine, Business and, specially, Engineering (Duby 2000).

To the best of our knowledge, MCS has been applied to routing only in a limited number of cases. Buxey (1979) employed MCS along with the Clarke and Wright's (1964) algorithm to build a procedure to solve the CVRP. Twenty years later, Fernández de Córdoba et al. (1998) applied it to solve the Rural Postman Problem (RPP), and the CVRP (Fernández de Córdoba et al. 2000). As compared with other Operations Research fields, where simulation techniques are widely used, MCS is still in an early stage of implementation in the VRP arena.

3 SOME CVRP CONSTRUCTIVE METHODS

This section describes some of the traditional algorithms for solving the Vehicle Routing Problem (VRP). In spite of the existence of exact methods as a general approximation to solve VRP, we will discuss in this paper only heuristic methods, from which we will obtain some ideas for the SR-1 algorithm. Among several methods that appear in the current literature for solving VRP, we would draw attention to:

A) **The nearest neighbor insertion (NNI) method** (Bodin et al. 1983). This method can be used to construct routes by the successive addition of the nearest point of the network *Ω* after the last one introduced in the current solution. This action is repeated until the vehicle is full. Notice that this procedure runs in time proportional to n^2.

B) **Clarke and Wright's savings (CWS) algorithm** (Clarke and Wright 1964). This algorithm assumes a baseline scenario in which each customer is supplied by a separate vehicle. In this situation, Clarke and Wright proposed that customers *i* and *j* should be supplied by the same vehicle as long as its capacity constraints were not violated. This new scenario, therefore, achieves a saving in the delivery distance which can be calculated in the following way:

$$S_{ij} = d_{0i} + d_{0j} - d_{ij}$$

where point *0* is the depot and points *i* and *j* are any pair of nodes in network *Ω*. Clearly $S_{ij} = S_{ji}$, $0 \leq i, j \leq n$ because of the distance symmetrical property. This algorithm can be easily applied: the node presenting the greatest saving is introduced into the route under construction, taking into account the relevant constraints. The CWS algorithm has a computational complexity proportional to $n^2 \log_2 n$.

C) **Fernandez, Garcia, Mayado and Sanchis (FGMS) method** (Fernández de Córdoba et al. 2000). These authors applied MCS to the NNI method to solve the CVRP. While most heuristic methods, including the aforementioned algorithms, build routes by the introduction of the best node according to a fixed criterion, the FGMS method designs a discrete probability function for selecting candidate nodes to be added to an existing route. Monte Carlo techniques are therefore used to model the random components of the node selection. Another crucial feature of this algorithm is that the probability function is based on the distances between nodes: big distances from a current node to a candidate node imply less likelihood to add the candidate node to the route under construction. Thus, these authors make use of the next formula to define a probabilistic way to choose nodes in a route:

$$p_{i,j} = \frac{\left(\frac{1}{d_{ij}}\right)^{\alpha}}{\sum_j \left(\frac{1}{d_{ij}}\right)^{\alpha}} \quad \text{with} \quad 0 \leq i, j \leq n$$

where the probability distribution $p_{i,j}$ ($0 \leq i, j \leq n$) defines a probability matrix $P = (p_{i,j})$ which supplies two indices, *i* and *j*, for each probabilistic iteration of the algorithm. The first of these two indices, *i*, indicates where to insert the next node (immediately before node *i*), while the second, *j*, determines which node should be inserted. That is, the probability distribution matrix *P* simultaneously stores the insertion position and the node number.

Consequently, each node's probability of being added to a route is inversely proportional to its distance from the last connected node. Moreover, the parameter $\alpha \geq 0$, also known as the weighting parameter, is used to change (fine-tuning) the discrete probability function. The FGMS method has a time complexity proportional to n^2.

4 LITERATURE REVIEW FOR THE SR-1

In this section we offer a review of the most recent literature on the CVRP algorithms which are directly related to SR-1. The first reference in importance and novelty is Toth and Virgo's (2002) book, which provides an excellent guide to new algorithms that can be compared against our method and it is indispensable in order to understand the state-of-the-art in VRP. Likewise, Ralphs et al. (2001) paper presents an excellent survey about the CVRP and develops a specific branch, cut and price framework. A good update of various heuristic methods for the CVRP appears in Van Breedam (2001).

Given the CVRP, the SR-1 algorithm uses a classical heuristic to obtain an initial solution. So, among other possibilities, this classical heuristic could be the CWS procedure, which presents several variations. Gaskell (1967) published a paper contrasting the difficulties to optimize some cases of VRP by using the CWS algorithm. The Gillet and Miller's (1974) sweep algorithm is other well-known constructive method to obtain CVRP solutions in an easy way. After that, Mole and Jameson (1976) generalized the definition of the savings function, introducing two parameters for controlling the savings behavior. Similarly, Holmes and Parker (1976) developed a procedure based upon the CWS algorithm, using the same savings function but introducing a solution perturbation scheme in order to avoid poor quality routes. Beasley (1981) adapted the CWS method to the optimization of inter-customer travel times. Correspondingly, Dror and Trudeau (1986) developed a version of the CWS method for the Stochastic VRP. More recently, Paessens (1988) depicted the main characteristics of the CWS method and its performance in generic VRP. Other traditional papers on heuristic algorithms are those of Golden et al. (1977), Bodin et al. (1983), Bodin and Berman (1979), and Gendreau et al. (1994). This list of papers would not be complete without Laporte and Nobert's (1987) research on exact algorithms for routing problems.

Correspondingly, Buxey (1979) described a simulation-based method. This author applied MCS in CVRP for the first time, defining the following probability function for node selection:

$$P(I) = \frac{S_I^M}{\sum_{K=1}^{J} S_k^M}$$

where I represents a point-pair *(i, j)*, M is a weighting parameter, S_I is the Clarke and Wright's savings function for the set $I = (i, j),$ and J is the number of point-pairs contained in a selection list of "good behavior" pairs. Buxey assumed the set *{2, 3, 4,..., 20}* as the potential values for index J, i.e. this method only considers up to twenty values of the best savings. According to this author, $M \in [0, 5]$. Later, this method was improved with the introduction of the Entropy function (Shannon and Weaver 1963) to control the random selection of nodes using the probability functions defined in the former equation. This new approach using the Entropy function was named as ALGACEA-1 algorithm (Faulin and Juan 2008).

Other algorithms that need to be taken into account in the construction of our method are the GRASP procedures (Feo and Resende 1995, Feo and Resende 1989), which are iterative randomized sampling techniques that provide a solution to the problem within each iteration. Each GRASP iteration usually consists of two phases: the first is to construct an initial solution via an adaptive randomized greedy function; the second is to develop a local search procedure for the current solution in hope of finding an improvement.

Likewise, the use of metaheuristics in VRP became popular during the nineties. Two of the most important papers on the use of heuristics and metaheuristics in that moment were Gendreau, Hertz and Laporte (1994), which introduced the Tabu Route algorithm, and Laporte, Gendreau, Potvin and Semet (2000), which includes a thorough discussion of classical and modern heuristics. Some years later, Tarantilis and Kiranoudis (2002) presented the Boneroute for routing and fleet management, and Toth and Vigo (2003) the Granular Tabu Search as a new method to solve the CVRP.

Other important references about metaheuristics which can be applied to CVRP are the following ones: Alba and Dorronsoro (2004) and Berger and Barkaoui (2003), who introduced some genetic algorithms in routing; Cordeau et al. (2004), who make a good review of new routing algorithms; Prins (2004), who developed a new evolutionary algorithm; and Doerner et al. (2002) and Reimann et al. (2004), who developed some savings ants procedures to solve the CVRP.

5 OUR APPROACH TO THE CVRP

We employ the Laporte et al. (2000) graph notation for the sake of a more straightforward explanation of the method. Therefore, an instance of the CVRP can be denoted as a graph $G = (\Omega, E)$, where:

a) Ω is a defined set of $n+1$ nodes, being node 0 the depot, and where node i ($i = 1, 2,\ldots, n$) has an assigned demand, $q_i > 0$, that must be satisfied (it is assumed that the depot has no demand)

b) $E = \{(i, j) / i, j \in \Omega, i < j\}$ represents the set of the $k = n(n+1)/2$ existing links connecting the $n+1$ nodes, each of them with an associate cost c_{ij}. Frequently, this aprioristic cost is expressed in terms of the Euclidean distance between the two nodes, d_{ij}, that is: $c_{ij} = d_{ij}= d_{ji} = c_{ji}$

Usually, it is assumed that the depot has as many vehicles as it will be necessary to satisfy the total demand of the nodes. The capacity of each vehicle is a constant $c >> 0$, being every transporter able to serve several clients before exhausting its full capacity, i.e.: $c >> max(q_i)$.

The general idea of our approach is explained next: given a starting "good solution" –one provided by a classical heuristic such as, for instance, the CWS algorithm–, our first goal is to improve it by performing an iterative local random search until a given stopping criterion is met. At each iteration, an oriented random search is carried out among the set of all possible solutions that present similar characteristics to those of the initial solution. Of course, the definition of this similarity is neither naïve nor simple as it represents one of the key elements of the method. In this sense, we say that two given solutions for the CVRP are *d*-similar, or similar in distances distribution, if their respective inter-node distances follow similar statistical distributions, i.e., one solution S_1 will be *d*-similar to other S_2 if both share approximately the same statistical distribution for the variable X = *"distance between two consecutive nodes in the solution routes"*. The fundamental hypothesis that guides our approach is that good solutions of a given CVRP instance will have similar properties. In particular, the size of the edges connecting two consecutive nodes in any solution route will follow approximately the same statistical distribution (i.e., good solutions for a CVRP instance will likely have edges of a certain size, neither "too big" or "too small"). So we will restrict the random search of new solutions to those that are somewhat similar to a good initial solution.

To be more precise, our approach is based on a series of simple steps, being one of them the actual application of the SR-1 algorithm (Figure 1):

1. Given a CVRP instance, construct the corresponding data model and select a classical heuristic –e.g. the parallel version of the CWS algorithm– to solve it.
2. Apply the selected classical heuristic to the data model and obtain the corresponding initial solution, which will have an associated aprioristic cost.
3. From the routes in the initial solution, determine the statistical distribution of the X variable: first, try to fit the observed values by some theoretical distribution; if no easy fitting is found, non-parametric methods, like bootstrapping techniques or experimental distributions, can be used instead.
4. Use the statistical distribution obtained in the previous step as an input for the SR-1 algorithm.
5. Also, use the initial CVRP instance to construct a data model that can serve as an input for the SR-1 algorithm.
6. Apply the SR-1 algorithm (explained in the next section) to generate a new random solution which, by construction, will likely be *d*-similar to the initial one; compare this new solution with the existing ones and rank the set of solutions according to their respective aprioristic costs. If the new solution is better than the initial one, repeat steps 3 and 4 considering the new solution as the initial one.
7. Repeat (iterate) step 6 until a stopping criterion (maximum number of iterations, elapsed computation time, etc.) is reached.

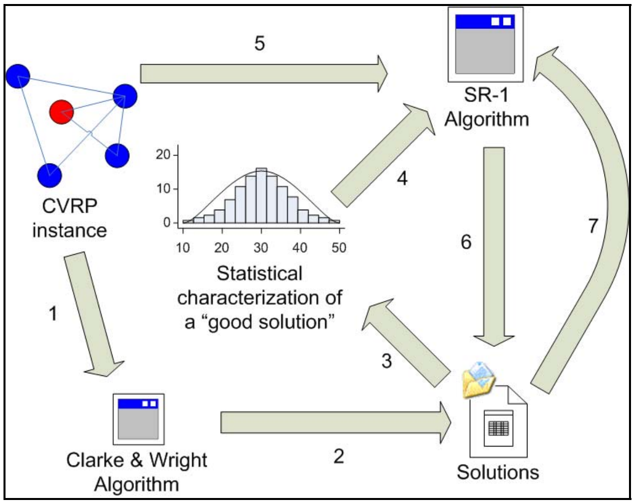
5
SR-1 Algorithm
CVRP instance
20
10
0
10
20
30
40
50
Statistical characterization of a "good solution"
4
6
7
1
3
Clarke & Wright Algorithm
2
Solutions

Figure 1: Scheme of our approach for the CVRP

Notice the following important details regarding the approach just introduced:

- The third step is no more than an attempt at giving a statistical characterization of a good solution. This characterization can then be sampled to obtain new –and hopefully improved– solutions to the given CVRP instance.
- Even when we are currently using just one random variable, *X*, to perform this statistical characterization of a "good solution", a richer characterization could be attained by using a multivariable approach –that is, by including other random variables such as the number of nodes per route, the served demand per route, etc. Therefore, other similarity relationships could be defined.
- Likewise, a richer statistical characterization of a "good solution" could be obtained if, instead of

using just an initial solution, multiple initial solutions from different classical heuristics are used.

- In order to identify the statistical distribution of X, outliers should be removed from observed data. In particular, depending on the concrete topology of the CVRP instance, distances from the depot to the first node in each route and distances from the last node to the depot in each route should not be considered, since they are likely to be larger than regular distances between two internal (non-depot) nodes in a route. An alternative approach to overcome the "outliers problem" would consist in using the trimmed mean and standard deviation of the initial observations as the parameters of a normal distribution which models X. When we use this approach, only positive sampled values from the normal distribution will be considered.
- Step 6 introduces a learning mechanism in the approach, so that the statistical characterization is always performed using data from the best solution so far (or from a set of best solutions so far). Other learning mechanisms are possible (e.g.: fine-tuning of distribution parameters, etc.)
- Even when one important target in our approach is to find out solutions with a lower aprioristic cost than the initial one, this is not our main target: as Poot et al. (2002) point out, in most practical situations there are additional restrictions and costs than cannot be easily computed a priori –such as environmental costs, work load, time or geographical restrictions. In this sense, our approach provides the decision-maker a set of alternative "good solutions", from which s/he can choose the best one according to multiple criteria.
- This approach can be generalized to a wide range of combinatorial problems, other than the CVRP, as far as an initial solution could be attained by using some classical heuristic. In fact, even in the case when no classical heuristic is available, a random solution could be employed as initial solution, which would be iteratively improved during the MCS process.

6 THE SR-1 ALGORITHM

As it has been described before, our approach makes use of an iterative process to generate a huge number of random solutions. Each of these solutions is a set of roundtrip routes that, altogether, satisfy all nodes demand by visiting and serving them. Furthermore, each of these solutions will be constructed such in a way that it will mimic the properties of the initial good solution. The actual SR-1 algorithm, which defines how these random solutions are constructed in each iteration k ($k = 1, 2,\ldots, m$, being m a user defined parameter), is detailed next:

1. Initialize a new solution, CS[k], at current iteration k
2. Make Θ the set of all served nodes (other than the depot) by CS[k]. Initially, set $\Theta = \varnothing$
3. Set the counter h of routes in CS[k] to 0
4. While $\Theta <> \Omega \setminus \{0\}$ (i.e., while there are still nodes with unsatisfied demand) do the following:
    - 4.1. Initialize a new route, CR[k][h], inside CS[k]. This route will be served by a new vehicle. Set the vehicle current capacity, VC, equal to its initial capacity, $c >> 0$
    - 4.2. Add the depot to CR[k][h] and set the depot as the current node, CN, in CR[k][h]
    - 4.3. While there is still any node, i, with unsatisfied demand, d_i, such as $d_i < VC$, do the following:
        - 4.3.1. Using the theoretical distribution of X (or, alternatively, using bootstrapping or an empirical distribution), sample a random value, x, for the distance between two consecutive nodes
        - 4.3.2. Determine the next node in CR[k][h], NN, as the unsatisfied node which distance to CN is closer to x (Figure 2)
        - 4.3.3. Update Θ by adding NN to it (in other words, set $\Theta \leftarrow \Theta \cup \{NN\}$)
        - 4.3.4. Add the link between CN and NN to route CS[k][h]
        - 4.3.5. Set NN as the current node: CN ← NN
    - 4.4. Close the route CS[k][h] by adding a link between CN and $\{0\}$ to it
    - 4.5. Increase the counter h of routes in CS[k]
5. Add the current solution CS[k], including all of its routes, to the array of completed solutions

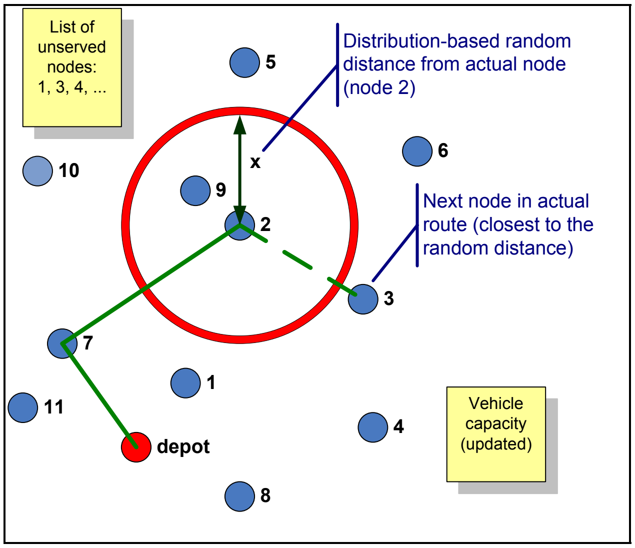
List of unserved nodes: 1, 3, 4, ...
Distribution-based random distance from actual node (node 2)
5
6
10
9
x
2
Next node in actual route (closest to the random distance)
3
7
1
11
4
depot
8
Vehicle capacity (updated)

Figure 2: Selecting the next node with SR-1

Notice that an interesting variant of step 4.3.2 would be to restrict the node selection to those nodes which are inside the circle with center CN and radius x. If no unsatisfied node matches those criteria, then the depot would be the next node and the route would be closed. This variant considerably reduces the possibility of selecting extremely remote nodes, which depending on the problem topology (e.g.: existence of clusters) should be avoided. In particular, selecting an extremely remote node could be a bad policy if the current available vehicle capacity after serving this remote node is close to zero (i.e.: if the current vehicle will not be able to serve other nodes in the neighborhood of the remote one).

As can be seen, the SR-1 algorithm has many desirable characteristics. First of all, it is a simple method which requires little instantiation. With little effort, similar algorithms based on the same key basic idea could be easily developed for other routing problems and, in general, for other combinatorial optimization problems. Second, SR-1 returns not only one solution or set of routes for the CVRP problem, like most existing algorithms, but rather a large set of solutions. Such behavior is highly desirable, as it allows for multiple criteria decision making as the set of solutions can be ranked according to different objectives. Notice that this is somewhat different to Genetic Algorithms (GAs), see Goldberg (1989) or Michalewicz (1996). While GAs maintains a population of solutions as SR-1 does, the size of this population is usually limited to 100 or even less, which is several orders of magnitude less from what SR-1 can provide. Moreover, some of the most efficient heuristics and metaheuristics are not used in practice because of the difficulties they present when dealing with real-life problems and restrictions (Kant et al. 2008). On the contrary, simulation-based heuristics, like the one presented here, tend to be more flexible and, therefore, they seem more appropriate to deal with real restrictions and dynamic work conditions.

7 SOFTWARE IMPLEMENTATION

We have used an object-oriented approach to implement the described methodology as a computer program. In order to do this, we have employed the Java programming language. The implementation process is not a trivial task, since there are some details which deserve special attention, in particular: (i) the use of a good random number generator (L'Ecuyer 2001), (ii) a correct design of the different classes, so that convinient cohesion and coupling levels will be reached (Horstmann 2006), and (iii) the code levels of accuracy and effectiveness –as an example of this, to perform mathematical operations the use of the package `java.lang.StrictMath` is usually preferred over the use of the more classical package `java.lang.Math`.

Regarding the generation of random number and variates, we have employed the SSJ library (L'Ecuyer 2002). In particular, we have used the subclass `GenF2W32`, which implements a generator with a period value equal to 2^{800}-1.

Furthermore, we needed a software implementation of the CWS heuristic in order to be able to test the efficiency of our approach against the CWS approach. Since we did not find any available implementation for the CWS algorithm –either on the Internet or in any book or journal–, we have developed our own object-oriented implementation of this algorithm. As a matter of fact, there are several variants of the CWS heuristic, so we decided to base our implementation in the one described in the following webpage from the Massachusetts Institute of Technology: <http://web.mit.edu/urban_or_book/www/book/chapter6/6.4.12.html>

8 EXPERIMENTAL RESULTS

In order to test the efficiency of our approach, we started by using some CVRP instances referenced in the book of Toth and Vigo (2002). We also used other CVRP instances from the TSPLIB, a library of instances freely available from the University of Heidelberg at: <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>. At first, we were somewhat concerned with the results of our CWS implementation, since they did not seem to perfectly match the expected results according to the Toth and Vigo's (2002) book. After discussing this matter with Professor Vigo, we understood that the expected CWS results should be considered with caution, since some of them were obtained by using CWS implementations that were not making use of real numbers, but only integer numbers. Moreover, sometimes these integers were obtained by rounding, while sometimes they were obtained by direct truncation. These details may have a significant impact over the CWS results, even for the same instance. Because of this, we decided to construct and use our own CVRP instances, which always use real numbers of high precision (all numbers and calculations in our code use the `double` Java type).

8.1 Case 1: A Small Size CVRP

As a first CVRP instance to test our algorithm, we generated a random set of *20* nodes (nodes *1* to *20*) uniformly distributed inside the square defined by the corner points *(-100, -100)*, *(-100, 100)*, *(100, 100)* and *(100, -100)*. The depot (node *0*, with no demand), was placed at the square center. The demand for each node was randomly generated (with an average demand of *83* and a maximum individual demand of *144*). Finally, a value of *345* was assigned as the vehicle total capacity. Figure 3 shows the scatterplot for this instance. Each point represents a node, and the corresponding number represents its associated demand. In this example, the traveling cost from one node to other was calculated as the Euclidean distance between the two nodes.

On one hand, this instance was solved by using the CWS heuristic, which provided a solution with a total cost of *1,208*. On the other hand, we solved this instance by employing our SR-1 algorithm with the bootstrapping variant: using a standard PC (Pentium *4* CPU, *2.8* GHz and *2* GB RAM), it took less than a minute to perform *50,000* iterations (i.e., to generate *50,000* random solutions); after those iterations SR-1 provided nine alternative solutions with a lower cost than the one given by the CWS heuristic (costs for these nine solutions were in the range between *1,185* and *1,205*).

A second set of calculations was performed by using the normal distribution variant, that is: (a) determine the trimmed mean, $\bar{x}^*$, and standard deviation, s, associated to the observed values of *X*, and (b) use a normal distribution with parameters $\mu = \bar{x}^*$ and $\sigma = s$ to generate the random values for *x*. Results using this variant were very similar to the ones obtained by using bootstrapping. Therefore, it seems reasonable to conclude that in small-size scenarios, SR-1 can easily offer a set of alternative solutions that improve the solution provided by the CWS heuristic.

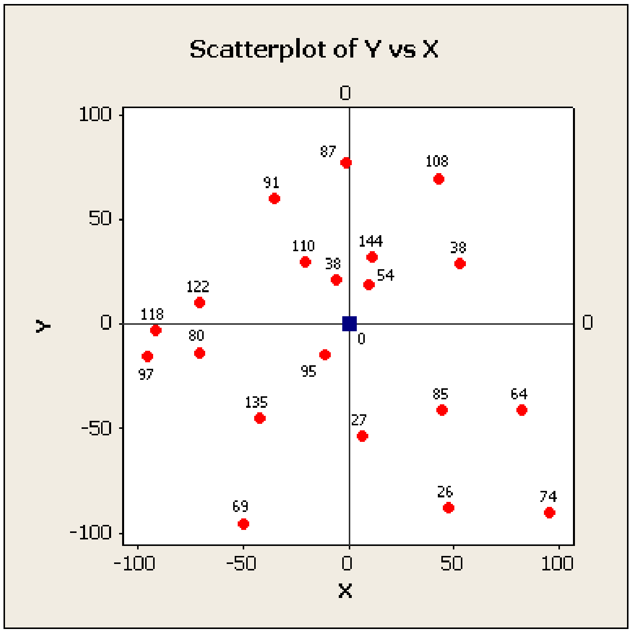

Figure 3: Random instance of nodes and demands

8.2 Case 2: A Medium Size CVRP

As a second CVRP instance, we generated a random set of *50* nodes (nodes *1* to *50*) uniformly distributed inside the square defined by the points *(-1000, -1000)*, *(-1000, 1000)*, *(1000, 1000)* and *(1000, -1000)*. As before, the depot was node *0* and it had zero demand. Demands for nodes other than the depot were randomly generated, with an average demand of *162* and a maximum individual demand of *290*. Finally, a value of *1,266* was assigned as the vehicle total capacity. Again, we solved this instance by using the CWS heuristic and by using the SR-1 algorithm. The solution given by the CWS heuristic has a cost of *17,356.33*. After approximately two minutes of computing time, the SR-1 algorithm provided three alternative solutions with an individual cost inferior to the one obtained before (costs for these three solutions were in the range between *17,271.15* and *17,326.03*). Results for both instances are summarized in Table 2.

Table 2: Summary of results for Case 1 and Case 2

Case (# nodes)	CWS Solution	Best SR-1 Sol.	# of SR-1 sol. improving the CWS sol.	Time (s)
Case 1 (21)	1,208	1,185	9	53
Case 2 (51)	17,356	17,271	3	118

9 CONCLUSIONS

We have presented here a general methodology, based on Monte Carlo simulation, to solve the Capacitated Vehicle Routing Problem. This methodology makes use of an initial solution and the SR-1 algorithm to randomly generate alternative solutions for a CVRP instance. Although more tests and benchmarks need to be performed before establishing definitive conclusions, the SR-1 algorithm has proven to be effective at least in some small size and medium size instances. One major advantage of simulation-based algorithms is the fact that they provide not only a good solution to the decision maker, but a set of alternative good solutions than can be ranked according to different criteria. Another major advantage of our approach is the flexibility of simulation-based algorithms, which allows them to deal with realistic situations defined by complex restrictions and dynamic working conditions. In our opinion, simulation techniques offer a new way to explore some traditional combinatorial problems, such as the CVRP and many others of similar characteristics.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Education and Science under grants TRA2006-10639 and 816902257-2257-4-8. We would also like to gratefully acknowledge the suggestions received from Professor Daniele Vigo.

REFERENCES

Alba, E., and B. Dorronsoro. 2004. Solving the Vehicle Routing Problem by using Cellular Genetic Algorithms. *Proceedings of Evolutionary Computation in Combinatorial Optimization – EvoCOP 2004*, ed. J. Gottlieb and G.R. Raidl. LNCS 3004, 11-20, Coimbra, Portugal, 5-7 April, 2004. Springer-Verlag.

Beasley, J. 1981. Adapting the Savings Algorithm for Varying Inter-Customer Travel Times. *Omega* 9: 658-659.

Berger, J., and M. Barkaoui. 2003. A Hybrid Genetic Algorithm for the Capacitated Vehicle Routing Problem. *Proceedings of the International Genetic and Evolutionary Computation Conference – GECCO03*, ed. E. Cantó-Paz. LNCS 2723, 646-656, Illinois. Chicago, USA. Springer-Verlag.

Bodin, L., and L. Berman. 1979. Routing and Scheduling of School Buses by Computer. *Transportation Science* 13: 113-129.

Bodin, L., B. Golden, A. Assad, and M. Ball. 1983. Routing and Scheduling of Vehicles and Crews. The State of the Art. *Computers and Operations Research* 10: 63-212.

Buxey, G.M. 1979. The Vehicle Scheduling Problem and Monte Carlo Simulation. *Journal of Operational Research Society* 30: 563-573.

Clarke, G., and J. Wright. 1964. Scheduling of Vehicles from a central Depot to a Number of Delivering Points. *Operations Research* 12: 568-581.

Cordeau, J.F., M. Gendreau, A. Hertz, G. Laporte, and J.S. Sormany. 2004. New Heuristics for the Vehicle Routing Problem. In *Logistics Systems: Design and Optimization*, ed. A. Langevin, and D. Riopel. Kluwer Academic Publishers.

Doerner, K., M. Gronalt, R.F. Hartl, M. Reimman, C. Strauss, and M. Stummer. 2002. Savings Ants for the Vehicle Routing Problem. In *Evo Workshops02*, ed. S. Cagnoni. LNCS 2279, 11-20. Springer-Verlag.

Dror, M., and P. Trudeau. 1986. Stochastic Vehicle Routing with Modified Savings Algorithm. *European Journal of Operational Research* 23: 228-235.

Duby, A. 2000. *Monte Carlo Applications in Systems Engineering*. Wiley.

Faulin, J., and A. Juan. 2008. The ALGACEA-1 Method for the Capacitated Vehicle Routing Problem. *International Transactions in Operational Research* 15: 1-23.

Feo, T.A., and M.G.C. Resende. 1989. A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem. *Operations Research Letters* 8: 67-71.

Feo, T.A., and M.G.C. Resende. 1995. Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization* 6: 109-133.

Fernández de Córdoba, P., L.M. García Raffi, A. Mayado, and J.M. Sanchis. 2000. A Real Delivery Problem Dealt with Monte Carlo Techniques. *TOP* 8: 57-71.

Fernández de Córdoba, P., L.M. García Raffi, and J.M. Sanchis. 1998. A Heuristic Algorithm Based on Monte Carlo Methods for the Rural Postman Problem. *Computers and Operations Research* 25: 1097-1106.

Fishman, G.S. 1996. *Monte Carlo. Concepts, Algorithms, and Applications*. Springer. New York.

Gaskell, T.J. 1967. Bases for the Vehicle Fleet Scheduling. *Operational Research Quarterly* 18: 281-295.

Gass, S.I., and A.A. Assad. 2005a. Model World: Tales from the Time Line. The Definition of OR and the Origins of Monte Carlo Simulation. *Interfaces* 35: 429-435.

Gass, S.I., and A.A. Assad. 2005b. *An Annotated Timeline of Operations Research*, Springer. New York.

Gendreau, M., A. Hertz, and G. Laporte. 1994. A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science* 40: 1276-1290.

Gillet, B.E., and L.R. Miller. 1974. A Heuristic Algorithm for the Vehicle Dispatch Problem. *Operations Research* 22: 340-349.

Goldberg, D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading.

Golden, B., T. Magnanti, and H. Nguyen. 1977. Implementing Vehicle Routing Algorithms. *Networks* 7: 113-148.

Holmes, R.A., and R.G. Parker. 1976. A Vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme. *Operational Research Quarterly* 27: 83-92.

Horstmann, C. 2006. *Big Java*. John Wiley & Sons.

Kant, G.; M. Jacks, and C. Aantjes. 2008. Coca-Cola Enterprises Optimizes Vehicle Routes for Efficient Product Delivery. *Interfaces* 38: 40-50.

Laporte, G., M. Gendreau, J.Y. Potvin, and F. Semet. 2000. Classical and Modern Heuristics for the Vehicle Routing Problem. *International Transactions in Operational Research* 7: 285-300.

Laporte, G., and Y. Nobert. 1987. Exact Algorithms for the Vehicle Routing Problem. *Annals of Discrete Mathematics* 31: 147-184.

Law, A. 2007. *Simulation Modeling & Analysis*. McGraw-Hill.

L'Ecuyer, P. 2001. Software for Uniform Random Number Generation: Distinguishing the Good and the Bad. In *Proceedings of the 2001 Winter Simulation Conference*, 95 – 105.

L'Ecuyer, P. 2002. SSJ: A Framework for Stochastic Simulation in Java. In *Proceedings of the 2002 Winter Simulation Conference*, 234 – 242.

Michalewicz, Z. 1996. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin.

Mole, R.H., and S.R. Jameson. 1976. A Sequential Route-building Algorithm Employing a Generalised Savings Criterion. *Operational Research Quarterly* 27: 503-511.

Paessens, H. 1988. The Savings Algorithm for the Vehicle Routing Problem. *European Journal of Operational Research* 34: 336-344.

Poot, A., G. Kant, and A Wagelmans. 2002. A savings based method for real-life vehicle routing problems. *Journal of the Operational Research Society* 53: 57-68.

Prins, C. 2004. A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem. *Computers and Operations Research* 31: 1985-2002.

Ralphs, T.K., L. Kopman, W.R. Pullesyblank, and L.E. Trotter. 2001. On the Capacitated Vehicle Routing Problem. <http://neo.lcc.uma.es/radi-aeb/WebVRP/data/articles/RKPT01.pdf> [accessed April 2, 2008].

Reimann, M., K. Doerner, and R.F. Hartl. 2004. D-ants: Savings Based Ants Divide and Conquer the Vehicle Routing Problem. *Computers and Operations Research* 31: 563-591.

Shannon, C. E., and W. Weaver. 1963. *Mathematical Theory of Communication.* Urbana, IL: University of Illinois Press.

Tarantilis, C.D., and C.T. Kiranoudis. 2002. Boneroute: an Adaptative Memory-Based Method for Effective Fleet Management. *Annals of Operations Research* 115: 227-241.

Toth, P., and D. Vigo. 2002. *The Vehicle Routing Problem.* SIAM Monographs on Discrete Mathematics and Applications. SIAM .

Toth, P., and D. Vigo. 2003. The Granular Tabu Search and its Application to the Vehicle Routing Problem. *INFORMS Journal on Computing* 15: 333-346.

Van Breedam, A. 2001. Comparing Descent Heuristics and Metaheuristics for the Vehicle Routing Problem. *Computers and Operations Research* 28: 289-315.

AUTHOR BIOGRAPHIES

JAVIER FAULIN <javier.faulin@unavarra.es> is an associate professor of Operations Research and Statistics at the Public University of Navarra (Pamplona, Spain). He holds a PhD in Economics and an MS in Applied Mathematics His research interests include logistics, vehicle routing problems and simulation modeling and analysis, especially techniques to improve simulation analysis in practical applications.

MIQUEL GILIBERT <mgilibert@uoc.edu> is a principal support engineer at CA, an enterprise software company with headquarters in Islandia (NY, USA). He holds a PhD in Physical Chemistry, an MBA and an MS in Computer Sciences. His research interests include the modeling and simulation of physical systems using statistical and approximate methods.

ANGEL A. JUAN <ajuanp@uoc.edu> is an associate professor of Computer Simulation and Data Analysis at the Open University of Catalonia (Barcelona, Spain). He holds a PhD in Industrial Engineering, an MS in Information Technology and an MS in Applied Mathematics. His research interests include discrete-event simulation and computer supported collaborative learning.

XAVIER VILAJOSANA <xvilajosana@uoc.edu> is an associate professor of Distributed Systems and Computer Networks at the Open University of Catalonia (Barcelona, Spain). He holds an MS in Computer Science and is currently finishing his PhD in Distributed Systems and Artificial Intelligence. His research interests include complex algorithms in distributed systems and economic models applied to computational environments.

RUBEN RUIZ <rruiz@eio.upv.es> is an associate professor in the Department of Applied Statistics, Operations Research and Quality at the Valencia University of Technology. He is also head of the Applied Optimization Systems (SOA) group at the Instituto Tecnológico de Informática (ITI). He received his BS and MS on Computer Science and a PhD on Statistics and OR from the Valencia University of Technology. His research interests include scheduling, metaheuristics and statistical analysis of algorithms' performance.