



Missouri University of Science and Technology
Scholars' Mine

Engineering Management and Systems
Engineering Faculty Research & Creative Works

Engineering Management and Systems
Engineering

01 Sep 2008

SR-2: A Hybrid Algorithm for the Capacitated Vehicle Routing Problem

Angel A. Juan

Javier Faulin

Josep Jorba

Barry Barrios

et. al. For a complete list of authors, see https://scholarsmine.mst.edu/engman_syseng_facwork/224

Follow this and additional works at: https://scholarsmine.mst.edu/engman_syseng_facwork

 Part of the [Systems Engineering Commons](#)

Recommended Citation

A. A. Juan et al., "SR-2: A Hybrid Algorithm for the Capacitated Vehicle Routing Problem," *Proceedings of the 8th International Conference on Hybrid Intelligent Systems, (HIS '08)*, Institute of Electrical and Electronics Engineers (IEEE), Sep 2008.

The definitive version is available at <https://doi.org/10.1109/HIS.2008.148>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Engineering Management and Systems Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

SR-2: A Hybrid Algorithm for the Capacitated Vehicle Routing Problem

Angel A. Juan¹, Javier Faulin², Josep Jorba³, Scott E. Grasman⁴ and Barry Barrios⁵
 {1,3} Open University of Catalonia, Barcelona, Spain { ajuana@uoc.edu, jjorba@uoc.edu }
 {2, 5} Public University of Navarre, Pamplona, Spain { javier.faulin@unavarra.es, bcubeb3@mit.edu }
 {4} Missouri University of Science and Technology, Rolla, USA { grasmans@mst.edu }

Abstract

During the last decades a lot of work has been devoted to develop algorithms that can provide near-optimal solutions for the Capacitated Vehicle Routing Problem (CVRP). Most of these algorithms are designed to minimize an objective function, subject to a set of constraints, which typically represents aprioristic costs. This approach provides adequate theoretical solutions, but they do not always fit real-life needs since there are some important costs and some routing constraints or desirable properties that cannot be easily modeled. In this paper, we present a new approach which combines the use of Monte Carlo Simulation and Parallel and Grid Computing techniques to provide a set of alternative solutions to the CVRP. This allows the decision-maker to consider multiple solution characteristics other than just aprioristic costs. Therefore, our methodology offers more flexibility during the routing selection process, which may help to improve the quality of service offered to clients.

1. Introduction

In the Capacitated Vehicle Routing Problem (CVRP), a fleet of vehicles supplies customers using resources available from a depot or central node. Each vehicle has the same capacity (homogeneous fleet) and each customer has a certain demand that must be satisfied. Additionally, there is a cost matrix that measures the costs associated with moving a vehicle from one node to another. These costs usually represent distances, traveling times, number of vehicles employed or a combination of these factors. Traditionally, the goal here is to find an optimal solution, i.e., a set of vehicle routes that minimize the total costs of satisfying each customer demand while not violating a set of constraints regarding the vehicle

maximum capacity and the maximum number of available vehicles, among others.

Different approaches to the CVRP have been explored during the last decades [1], [2]. These approaches range from the use of pure optimization methods, such as linear programming, for solving small-size problems with relatively simple constraints to the use of heuristics and metaheuristics that provide near-optimal solutions for medium and large-size problems with more complex constraints. As mentioned before, most of these methods focus on minimizing an aprioristic cost function subject to a set of well-defined constraints. However, real-life problems tend to be complex enough so that not all possible costs, e.g., environmental costs, work risks, etc., constraints and desirable solution properties, e.g., time or geographical restrictions, balanced work load among routes, solution attractiveness, etc., can be considered a priori during the mathematical modeling phase [3]. For that reason, there is a need for more flexible methods that provide a large set of alternative near-optimal solutions with different properties, so that decision-makers can choose among different alternative solutions according to their concrete necessities and preferences. Accordingly, the main purpose of this paper is to present a hybrid approach to the CVRP, based on the combined use of Monte Carlo Simulation (MCS) and Parallel and Grid Computing (PGC) techniques, which is designed to fulfill this lack of solution alternatives. The algorithm that sustains this approach is called SR-2.

The rest of the paper is structured as follows: Section 2 gives a more formal description of the CVRP; Section 3 reviews some relevant CVRP literature; Section 4 discusses the use of MCS in CVRP; Section 5 introduces the main ideas behind our approach; Section 6 presents the SR-2 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. The Capacitated Vehicle Routing Problem

We assume a set Ω of $n+1$ nodes, each of them representing a vehicle destination (depot node) or a delivery point (demanding node) [4]. The nodes are numbered from 0 to n , node 0 being the depot and the remaining n nodes the delivery points. A demand $q_i > 0$ of some commodity has been assigned to each non-depot node i ($1 \leq i \leq n$). On the other hand, $E = \{(i, j) / i, j \in \Omega ; i < j\}$ represents the set of the $n \cdot (n+1) / 2$ existing links connecting the $n+1$ nodes. Each of these links has an associated aprioristic cost $c_{ij} > 0$, which represents the cost of sending a vehicle from node i to node j . These c_{ij} are assumed to be symmetric ($c_{ij} = c_{ji}, 0 \leq i, j \leq n$), and they are frequently expressed in terms of the Euclidean distance between the two nodes, d_{ij} . The delivery process is to be carried out by a fleet of NV vehicles ($NV \geq 1$) with equal capacity, $C \gg \max_{1 \leq i \leq n} \{q_i\}$. Some additional constraints associated to the CVRP are the following:

- (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

As stated in the introduction, traditionally the main goal of this problem is the construction of a feasible solution (set of feasible routes, one for each non-idle vehicle), which minimizes the sum of the total costs involved in the delivery process.

3. Literature review for the CVRP

Probably the most cited approach to the CVRP is the Clarke and Wright's Savings algorithm (CWS) [5], which presents several variations. Gaskell published a paper contrasting the difficulties to optimize some cases of CVRP by using the CWS algorithm [6]. The Gillett and Miller's sweep algorithm [7] is other well-known constructive method to obtain CVRP solutions in an easy way. After that, Mole and Jameson [8] generalized the definition of the savings function, introducing two parameters for controlling the savings behavior. Similarly, Holmes and Parker [9] 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 [10] adapted the CWS method to the optimization of inter-customer travel times. Correspondingly, Dror and Trudeau [11] developed a version of the CWS method for the Stochastic VRP. Some years later, Paessens [12] depicted the main characteristics of the CWS method and its performance in generic VRP.

Buxey [13] described a simulation-based method. As far as we know, this author applied Monte Carlo Simulation in CVRP for the first time. Later, this method was improved with the introduction of an entropy function 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 [14].

Other algorithms that have also been proposed to solve the VRP are the GRASP procedures [15], [16]. 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 [17], which introduced the Tabu Route algorithm, and [4], which includes a thorough discussion of classical and modern heuristics. Some years later, Tarantilis and Kiranoudis [18] presented the Boneroute for routing and fleet management, and Toth and Vigo [19] the Granular Tabu Search as a new method to solve the CVRP.

Other important references about metaheuristics that can be applied to CVRP are [20] and [21], who introduced some genetic algorithms in routing; [22], who make a good review of new routing algorithms; and [23], who developed a new evolutionary algorithm.

Finally, [24] applied MCS to solve the Rural Postman Problem (RPP), and the CVRP [25].

4. Our approach to the CVRP

Our goal is to develop a methodology that provides the decision-maker with a set of alternative near-optimal or "good" solutions for a given CVRP instance. We are not especially interested in obtaining the best solution from an aprioristic point of view—that is, the solution that minimizes the aprioristic costs as expressed in the objective function. As we have discussed before, in practical real situations there are important cost factors, constraints and desirable solution properties that usually can not be modeled or accounted for a priori. In order to generate this set of "good" solutions, we will make use of Monte Carlo Simulation to randomly select the next node in an open route according to an efficiency criterion.

To be more specific, in the CVRP context, for the current active node i in an open route we consider the

random variable X_i = “node that follows node i in the current active route”. Notice that X_i can take any value x_i in $\Omega - \{i\}$ as far as this value corresponds to the depot or to a node that has not been served yet, that is, $x_i \in \Omega^* = \{0\} \cup \{j \in \Omega / q_j > 0\}$. Also, for each pair of distinct nodes, i and j , we define the concept of “efficiency value associated to moving from i to j ”, e_{ij} , as the quotient between the demand that will be satisfied with this moving, q_j , and the cost of the shipment, c_{ij} ($c_{ij} > 0$ since $i \neq j$), that is:

$$\forall i, j \in \Omega, i \neq j \quad e_{ij} = \frac{q_j}{c_{ij}} \quad (1)$$

Now, we will use this concept of efficiency to assign a discrete probability distribution to the random variable X_i . There are several possibilities to perform this distribution assignment. One interesting option could be to use a methodology similar to the one employed in [25]. In such case, thought, we would use our concept of efficiency instead of the simpler concept of “distance between two nodes” used by these authors. That is, we could define the following probability function for X_i :

$$\forall j \in \Omega^* \quad P(X_i = j) = \frac{e_{ij}^\alpha}{\sum_{k \in \Omega^*} e_{ik}^\alpha} \quad (2)$$

where $\alpha \geq 0$ is a weighting parameter that can be used to change (fine-tuning) the discrete probability function. Notice that for $0 \leq \alpha < 1$ all nodes have almost the same probabilities of being selected regardless their efficiency levels, while for $\alpha = 1$ the probability assigned to each node is directly proportional to its efficiency level. Also, notice that for $\alpha > 1$ the probability of being selected decreases exponentially as we move from higher to lower efficiency levels.

Even when this assignment method is quite interesting and should be explored in a future work, it also seemed to us that the optimal selection of α is not a trivial task, due to the wide range of possible values for ,the fine-tuning (or “learning”) process could be highly complex and time-consuming, which could represent a severe restriction in many practical situations. For that reason, we decided to use another option to construct the probability function, one inspired by the exponential smoothing method used in time series analysis [26]. Given a value β (smoothing constant), $0 < \beta < 1$, it follows that:

$$\sum_{k=0}^{+\infty} \beta \cdot (1-\beta)^k = 1 \quad (3)$$

This way, assuming that the current open route is located at node i ($0 \leq i \leq n$) and that $\Omega^* = \{j_1, j_2, \dots, j_l\}$, with $1 \leq l \leq n$ and $e_{ij_1} \geq e_{ij_2} \geq \dots \geq e_{ij_l}$, is the set of potential nodes that can be visited next, we construct the following probability function for X_i :

$$\forall s \in \{1, 2, \dots, l\}, P(X_i = j_s) = \beta \cdot (1-\beta)^{s-1} + \frac{\varepsilon}{l} \quad (4)$$

where

$$\varepsilon = \sum_{r=l}^{+\infty} \beta \cdot (1-\beta)^r = 1 - \sum_{r=0}^{l-1} \beta \cdot (1-\beta)^r \quad (5)$$

In other words, we are sorting out all remaining nodes in Ω^* according to their efficiency values and then to assign a probability of (approximately) β to the node with the highest efficiency level, the rest of the sorted nodes receiving their corresponding probabilities according to an (approximately) exponential diminishing pattern (Fig. 1).

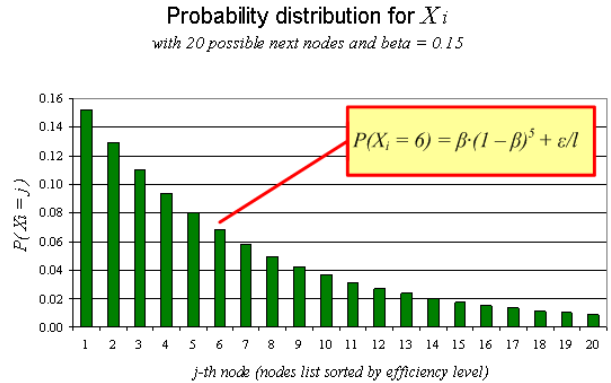


Fig. 1: Construction of the probability function for X_i

Since the smoothing factor β is restricted to the interval $(0,1)$, we expect that this parameter will be easier to fine-tune in most practical situations. Nevertheless, a lot of computations might be required to explore the solution space using efficiency-based random search, as described above, with different values of β . This is where Parallel and Grid Computing techniques (PGC) come into play [27], [28]. For instance, several tasks could be launched in parallel, each of them performing iterative random searches with different values of the smoothing factor (e.g., five parallel tasks with the k -th task

using $\beta_k = 0.4 + 0.1k$, $1 \leq k \leq 5$). After some thousand iterations, these tasks could interchange messages to compare the respective solutions that are being obtained at each task and, consequently, choose the more convenient β -value for future iterations. PGC techniques also allow introducing some interesting “risky/conservative” strategies for the routing selection. In effect, notice that high values of β are conservative in the sense that, each time a new node has to be added to an open route, they promote the selection of those nodes with associated high-efficiency values. On the contrary, low values of β do not promote the selection of nodes with associated high-efficiency values on the very short-run, but they give priority to the search of alternative routes that, on the long run, might result in equivalent or even better solutions. Again, PGC techniques can be used to test different “risky-to-conservative” strategies such as: (a) start solutions with a low β -value (e.g. $\beta = 0.4$) and progressively increase that value at each new route; or (b) start each route in a solution with a low β -value and progressively increase that value as new nodes are added to that route. At the end, the logic behind this strategy is that initial steps in a route or in a solution could be less conservative in order to explore more alternative routes or solutions and, as the route or solution evolves, these steps become more conservative in order to keep high efficiency levels. Notice that this approach contributes to avoid the local minimum problem.

5. The SR-2 algorithm

As it has been described before, our approach makes use of an iterative process to generate a huge number of random solutions based on the efficiency criterion. Each of these solutions is a set of roundtrip routes that, altogether, satisfy all nodes demand by visiting and serving them. The actual SR-2 algorithm, which defines how these random solutions are constructed in each iteration h ($h = 1, 2, \dots, m$, being m a user-defined parameter), is detailed next.

1. Initialize a new solution, $CS[h]$, at current iteration h .
2. Make Θ the set of all served nodes (other than the depot) by $CS[h]$. Initially, set $\Theta = \emptyset$.
3. Reset the counter u of routes in $CS[h]$.
4. While $\Theta \neq \Omega - \{0\}$ (i.e., while there are still nodes with unsatisfied demand) do the following.

- 4.1. Initialize a new route, $CR[h][u]$, inside $CS[h]$. This route will be served by a new vehicle. Set the vehicle current capacity, VC , equal to its initial capacity, $C \gg 0$.
- 4.2. Add the depot to $CR[h][u]$ and set the depot as the current node, CN , in $CR[h][u]$.
- 4.3. While there is still any node, j , with unsatisfied demand, q_j , such as $0 < q_j < VC$, do the following.
 - 4.3.1. According to the efficiency criterion and to equations (4) and (5), construct the probability function for all non-served nodes.
 - 4.3.2. Using Monte Carlo Simulation, determine the next node in $CR[h][u]$, NN (Fig. 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 $CR[h][u]$.
 - 4.3.5. Set NN as the current node: $CN \leftarrow NN$.
- 4.4. Close the route $CR[h][u]$ by adding a link between CN and $\{0\}$.
- 4.5. Increase the counter u of routes in $CS[h]$.
5. Add the current solution $CS[h]$, including all of its routes, to the array of completed solutions.

As can be seen, the SR-2 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-2 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. 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 [29], [30]. 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.

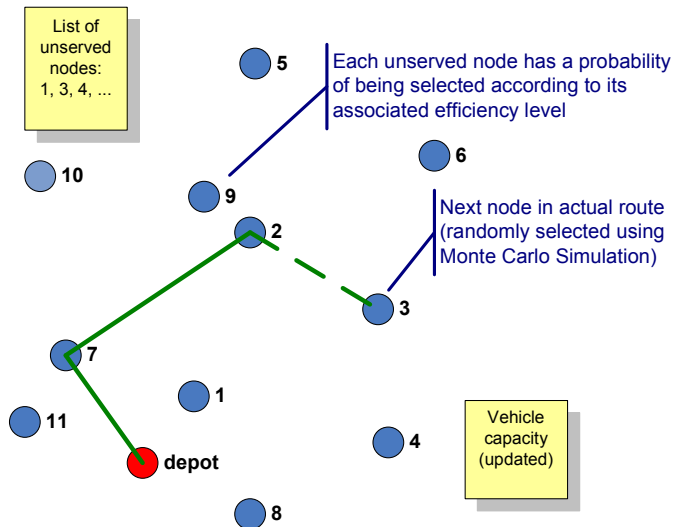


Fig. 2: Selecting the next node with SR-2

6. 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, and (ii) the code levels of accuracy and effectiveness.

Regarding the generation of random number and variates, we have employed the SSJ library [31]. 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 on the one described in the following webpage from the Massachusetts Institute of Technology:

web.mit.edu/urban_or_book/www/book/chapter6/6.4.12.htm.

7. A preliminary test

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. 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-2 algorithm: using a standard PC (Pentium 4 CPU, 2.8 GHz and 2 GB RAM), it took only some seconds to perform 50,000 iterations (i.e., to generate 50,000 random solutions); after those iterations SR-2 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).

Therefore, it seems reasonable to conclude that in small-size scenarios, SR-2 can easily offer a set of alternative solutions that improve the solution provided by the CWS heuristic.

8. Conclusions

In this paper, we have presented a general methodology, based on the combined use of Monte Carlo Simulation and Parallel and Grid Computing, to solve the Capacitated Vehicle Routing Problem. This methodology makes use of the concept of efficiency to randomly generate a set of alternative solutions for a CVRP instance. Although more tests and benchmarks are needed before establishing definitive conclusions, the SR-2 algorithm has proven to be effective in some preliminary tests. 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 that 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. The main disadvantage of using simulation-based algorithms is that they use to be computationally intensive. This is where Parallel and Grid Computing techniques can play an important role in order to make these algorithms more efficient.

9. References

- [1] P. Toth and D. Vigo. 2002. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. SIAM.

- [2] B. Golden, S. Raghavan and E. Edward Wasil (eds.). 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer.
- [3] A. Poot, G. Kant and A. Wagelmans. 2002. "A Savings Based Method for Real-Life Vehicle Routing Problems". *Journal of the Operational Research Society*, vol. 53, pp. 57-68.
- [4] G. Laporte, M. Gendreau, J. Potvin and F. Semet. 2000. "Classical and Modern Heuristics for the Vehicle Routing Problem". *International Transactions in Operational Research*, vol. 7, pp. 285-300.
- [5] G. Clarke, and J. Wright. 1964. "Scheduling of Vehicles from a central Depot to a Number of Delivering Points". *Operations Research*, vol. 12, pp. 568-581.
- [6] T. Gaskell. 1967. "Bases for the Vehicle Fleet Scheduling". *Operational Research Quarterly*, vol. 18, pp. 281-295.
- [7] B. Gillet, and L. Miller. 1974. "A Heuristic Algorithm for the Vehicle Dispatch Problem". *Operations Research*, vol. 22, pp. 340-349.
- [8] R. Mole and S. Jameson. 1976. "A Sequential Route-building Algorithm Employing a Generalised Savings Criterion". *Operational Research Quarterly*, vol. 27, pp. 503-511.
- [9] R. Holmes, and R. Parker. 1976. "A Vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme". *Operational Research Quarterly*, vol. 27, pp. 83-92.
- [10] J. Beasley. 1981. "Adapting the Savings Algorithm for Varying Inter-Customer Travel Times". *Omega*, vol. 9, pp. 658-659.
- [11] M. Dror and P. Trudeau. 1986. "Stochastic Vehicle Routing with Modified Savings Algorithm". *European Journal of Operational Research*, vol. 23, pp. 228-235.
- [12] H. Paessens. 1988. "The Savings Algorithm for the Vehicle Routing Problem". *European Journal of Operational Research*, vol. 34, pp. 336-344.
- [13] G. Buxey. 1979. "The Vehicle Scheduling Problem and Monte Carlo Simulation". *Journal of Operational Research Society*, vol. 30, pp. 563-573.
- [14] J. Faulin and A. Juan. 2008. "The ALGACEA-1 Method for the Capacitated Vehicle Routing Problem". *International Transactions in Operational Research*, (in press).
- [15] T. Feo and M. Resende. 1989. "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem". *Operations Research Letters*, vol. 8, pp. 67-71.
- [16] T. Feo and M. Resende. 1995. "Greedy Randomized Adaptive Search Procedures". *Journal of Global Optimization*, vol. 6, pp. 109-133.
- [17] M. Gendreau, A. Hertz and G. Laporte. 1994. "A Tabu Search Heuristic for the Vehicle Routing Problem". *Management Science*, vol. 40, pp. 1276-1290.
- [18] D. Tarantilis and C. Kiranoudis. 2002. "Bonerooute: an Adaptive Memory-Based Method for Effective Fleet Management". *Annals of Operations Research*, vol. 115, pp. 227-241.
- [19] P. Toth and D. Vigo. 2003. "The Granular Tabu Search and its Application to the Vehicle Routing Problem". *INFORMS Journal on Computing*, vol. 15, pp. 333-346.
- [20] E. Alba, and B. Dorronsoro. 2004. "Solving the Vehicle Routing Problem by using Cellular Genetic Algorithms". In Jens Gottlieb and Günter R. Raidl, editors, *Evolutionary Computation in Combinatorial Optimization – EvoCOP 2004*, LNCS 3004, pp. 11-20, Coimbra, Portugal, 5-7 April, 2004. Springer-Verlag.
- [21] J. Berger and M. Barkaoui. 2003. "A Hybrid Genetic Algorithm for the Capacitated Vehicle Routing Problem". In E. Cantó-Paz, editor. *Proceedings of the International Genetic and Evolutionary Computation Conference – GECCO03*, LNCS 2723, pp. 646-656, Illinois. Chicago, USA. Springer-Verlag.
- [22] J. Cordeau, M. Gendreau, A. Hertz, G. Laporte, G. and J. Sormany. 2004. "New Heuristics for the Vehicle Routing Problem". In A. Langevin and D. Riopel, editors, *Logistics Systems: Design and Optimization*. Kluwer Academic Publishers
- [23] C. Prins. 2004. "A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem". *Computers and Operations Research*, vol. 31, pp. 1985-2002.
- [24] P. Fernández de Córdoba, L. García Raffi, A. Mayado and J. Sanchis. 2000. "A Real Delivery Problem Dealt with Monte Carlo Techniques." *TOP*, vol. 8, pp. 57-71.
- [25] P. Fernández de Córdoba, L. García Raffi, and J. Sanchis. 1998. "A Heuristic Algorithm Based on Monte Carlo Methods for the Rural Postman Problem". *Computers and Operations Research*, vol. 25, pp. 1097-1106.
- [26] J. Hanke and D. Wichern. 2008. *Business Forecasting*. Prentice Hall. Upper Saddle River, NJ, USA.
- [27] J. Mohcine, S. Contassot-Vivier And R. Couturier. 2007. *Parallel Iterative Algorithms: From Sequential to Grid Computing*. Chapman & Hall/CRC.
- [28] S. Rajasekaran and J. Reif (eds.). 2007. *Handbook of Parallel Computing: Models, Algorithms and Applications*. Chapman & Hall/CRC.
- [29] G. Kant, M. Jacks, and C. Aantjes. 2008. "Coca-Cola Enterprises Optimizes Vehicle Routes for Efficient Product Delivery". *Interfaces*, vol. 38, pp. 40-50.
- [30] A. Goel and V. Gruhn. 2008. "A General Vehicle Routing Problem" *European Journal of Operational Research*, vol. 191, pp. 650-660.
- [31] P. L'Ecuyer, P. 2002. "SSJ: A Framework for Stochastic Simulation in Java". In *Proceedings of the 2002 Winter Simulation Conference*, pp. 234-242.