Programación Matemática y Software (2014) 6 (2): 36-45. ISSN: 2007-3283

Solving the capacitated vehicle routing problem with stochastic demands applying the simulated annealing algorithm

Resolviendo el problema de ruteo de vehículos de capacidad con demandas estocásticas aplicando el algoritmo de recocido simulado

Ernesto Liñán-García [1]*, Linda Crystal Cruz-Villegas [1], David S. González-González[1]

[1] Facultad de Sistemas. Universidad Autónoma de Coahuila. Carretera 57, km. 13, Arteaga, Coahuila, México.
* Correo-e: ernesto_linan_garcia@uadec.edu.mx

KEYWORDS:

simulated annealing, savings algorithm, vehicle routing problem, metaheuristic

ABSTRACT

A new hybrid algorithm for solving the capacitated vehicle routing problem (CVRP) with stochastic demands is proposed. This new approach combines the simulated annealing (SA) with the savings algorithms and is implemented to obtain results of several Solomon's instances of CVRP. This approach is named simulated annealing with metropolis cycle based on savings algorithm (SAMCSA). SA is a bio-inspired metaheuristic, an emulation of the heating and cooling processes of solids to solve optimization problems. On the other hand, the savings algorithm is a deterministic heuristic for solving CVRP. In order to generate high quality solutions of CVRP, this approach applies the savings algorithm into metropolis cycle of simulated annealing. An initial solution of SA is also generated by the savings algorithm. This simple change has lead to increase the solution quality for the CVRP with respect to the classical simulated annealing and savings algorithms.

PALABRAS CLAVE:

algoritmo de recocido simulado, algoritmo de ahorro, problema de ruteo de vehículos, metaheurísticas

RESUMEN

Se propone un nuevo algoritmo híbrido para resolver el problema de ruteo de vehículos de capacidad (CVRP en inglés) con demanda estocástica. Este nuevo enfoque combina los algoritmos de recocido simulado (RS) y de ahorro y es implementado para obtener resultados de varias instancias Solomon del CVRP. El RS es una metaheúristica bioinspirada, una simulación del proceso de calentamiento y enfriamiento de los sólidos para resolver problemas de optimización. Por otro lado, el algoritmo de ahorro es una heurística determinística para resolver el CVRP. Con el objeto de generar soluciones de alta calidad del CVRP, este enfoque aplica el algoritmo de ahorro dentro del ciclo de metrópolis del recocido simulado; la solución inicial del algoritmo también es generada por el algoritmo de ahorro. Este simple cambio ha permitido incrementar la calidad de la solución del CVRP con respecto al recocido simulado clásico y al algoritmo de ahorro.

 http://dx.doi.org/10.30973/progmat/2014.6.2/5

1 INTRODUCTION

In order to improve the delivery services of goods to customers with different demands, distribution centers must arrange optimal vehicle routes to ensure the lowest transportation and distribution costs. Planning optimal routes for the distribution of goods to customers can generate important savings for companies; several benefits can be obtained if the vehicle routing problem (VRP) is solved. Consequently, VRP is an important task in many private and public corporations.

VRP is basic in distribution management and has become one of the most widely studied problems in the area of combinatorial optimization. This problem was introduced by Dantzig [1,2] and is applied to the design of optimal routes, which seek to service a number of customers with a fleet of vehicles with a set of constraints. The objective of this problem is to determine the optimal route to serve multiple clients, using a group of vehicles to minimize the overall transportation cost.

VRP and its variants are classified as NP-hard problems [3-5]. In order to find an optimal solution, there are many factors to be considered and many possibilities of permutation and combinations. VRP becomes more complex as constraints and number of customers increase.

There are variations of VRP, such as capacitated VRP (CVRP), multi-depot VRP (MDVRP), periodic VRP (PVRP), stochastic VRP (SVRP) [6], and VRP with time windows (VRPTW), among others.

Several computational algorithms, clustered in exact, heuristic and metaheuristic methods, have been applied to solve VRP, but it is best solved by heuristics and metaheuristics [7].

Some exact algorithms have been proposed for solving the vehicle routing problem [8,9]. The advantage of using exact algorithms is that they always obtain optimal solutions to a certain size of the problem, but they become inefficient for large instances in function of time. Due to this reason, several metaheuristic methods have been designed to obtain sub-optimal solutions of VRP variants, for example Ant colony optimization [10], simulated annealing [11,12], Genetic algorithm [13,14] and Tabu search [15,16]. The disadvantage of metaheuristics is that they do not guarantee optimal solutions, but can generate solutions very close to the optimal in a reasonable processing time.

A hybrid algorithm is proposed in this work, which combines the simulated annealing algorithm with the savings algorithm. An initial solution is requested for simulated annealing algorithm, which is generated by savings algorithm as well.

In the Metropolis cycle of SA, new solutions of CVRP are generated by the savings algorithm, each of which can be accepted when it is better than the previous one, or by Boltzmann probability, or otherwise rejected. In order to obtain new CVRP solutions, the customers' demands are generated for a stochastic way.

This paper is organized as follows: in section II, the capacitated vehicle routing problem is formally described; the classical simulated annealing algorithm is detailed in section III; in section IV savings algorithm is described; in section V the algorithm proposed is explained in detail; experimentation and results are shown in section VI; finally, in section VII the conclusions are discussed.

2 CAPACITATED VEHICLE ROUTING PROBLEM

The capacitated VRP is formally defined as follows. Let $G(V,A)$ be an undirected graph where V denotes a set of $n+1$ vertices. A denotes an arc set, which is defined as $A=\{(v_i,v_j):v_i,v_j \epsilon V, i \neq j\}$. V defines the set $V=\{0,1,2,...,n\}$ where n defines the number of customers. The vertex number $\{0\}$ represents a depot while the remaining $\{1,...,n\}$ corresponds to n customers. Each (v_i,v_j) of set A is associated to non-negative cost C_{ij}. Each customer of the set {1,...,n} has a non-negative demand q_i, which defines the units that are supplied from depot 0 to customer i. The demand of depot q_0 is equal to zero $(q_0=0)$. A set of m identical vehicles of capacity Q must be used to supply goods to the n customers. The m vehicles must start and finish at depot. A route is defined as a least cost simple cycle of graph G passing through depot 0 and such that the total demand of the vertices visited does not exceed the vehicle capacity. The objective of this problem is to minimize the total travelled distance, time or cost by the m vehicles, subject to: (1) each route starts and ends at the depot, (2) each customer is visited exactly once by exactly one vehicle, and (3) the total demand of each route does not exceed Q. The CVRP is formulated as follows:

$$min \sum c_{ij} x_{ij} \qquad (1)$$

where $(i,j) \epsilon V$, $V=\{0,1,2,\ldots,n\}$, $X_{ij} \epsilon \{0,1\}$

subject to

$\sum x_{0j}=m$ (1.1)

$\sum x_{i0}=m$ (1.2)

$\sum_{j=1} x_{ij}=1\ (i=1,\ldots,n)$ (1.3)

$\sum_{i=1} x_{ij}=1\ (j=1,\ldots,n)$ (1.4)

$m \geq 1$ (1.5)

The objective function (1) is the total cost of the solution. Constraints 1.1 and 1.2 indicate that m is the number of vehicles used in the solution and that all vehicles that leave the deposit should be returned. Constraints 1.3 and 1.4 ensure that every client is an intermediate node of a route.

3 SIMULATED ANNEALING ALGORITHM

The simulated annealing algorithm (SA) is applied to complex problems which have a large solution space to obtain results close to the global optimum value in a short period of time. The SA concept was independently presented by Kirkpatrick [17] and Cerny [18]. Simulated annealing is based on the solids annealing physical process and Boltzmann distribution; it is a Montecarlo method generalization, which was developed by Nicholas Metropolis [19], who introduced a simple algorithm called Metropolis algorithm (MA) to emulate the evolution of a solid in a heat bath until thermal equilibrium, at a given temperature, is reached.

Classical simulated annealing emulates the equilibrium state of a collection of atoms at any given temperature; it begins with an initial solution at a very high temperature, which is slowly decreased, allowing MA to generate good configurations of the solution until the final temperature is reached, simulating the gradual metal cooling for crystallization [17].

This SA algorithm starts at a high temperature value, and then this parameter is decreased until a final temperature is reached. The final temperature is typically very close to zero [17,18]. The temperature value is decreased by a cooling function.

There are several cooling functions that have been used in the simulated annealing algorithm [20-23], the most common of which is defined by $T_{t+1} = \alpha * T_t$, where T_t is the temperature value at t time, T_{t+1} is the temperature value at $t+1$ time, and α is the decrement factor of temperature.

This cooling function decreases the temperature value by an α factor, which comprises a range of $0.70<\alpha<1.0$. A gradual cooling is applied when α is very close to 1, and a fast cooling is applied when α is very close to 0.70.

The classical simulated annealing has two cycles; the first one is named Temperature cycle. Into this cycle, the temperature value is decreased by a cooling function. The second cycle is named Metropolis cycle (MC), and it is applied to generate, accept or reject solutions for optimizing the problem.

Figure 1 shows the SA algorithm, which can be described in four sections, the first one is the setting section, where all initial parameters and initial values are calculated; the second section corresponds to the Temperature cycle; the third one is the Metropolis cycle, an internal procedure of Temperature cycle; and the algorithm results are shown at the last section.

Several parameters are determined at the setting section. An initial solution ($S_{current}$) is calculated as well as its associated energy value ($E_{current}$). The best solution (S_{best}) and best energy (E_{best}) are variables assigned to store the best values of all the algorithm execution; they are set at the beginning with their respective current values. An important algorithm parameter is temperature, which determines the initial and stop criteria. Initial temperature (T) is tuned in this section, as well as final temperature (T_{final}). Finally, the cooling factor (α) has the task of decreasing temperature, each time that Metropolis cycle is completed, in order to reach the final stage, so α is tuned before the Temperature cycle begins.

The second SA section corresponds to the Temperature cycle, also called external cycle, which contains the Metropolis cycle and the temperature decreasing function; once MC is completed, the function sets a new temperature value and Metropolis cycle starts again until final temperature is reached.

Metropolis cycle or SA internal cycle is started into temperature one. The first step is to make a little change to $S_{current}$ and assign the perturbed solution to the new solution (S_{new}). The new energy solution (E_{new}) is calculated as the difference between E_{new} and $E_{current}$; if the difference is less than zero (at minimization case), the $S_{current}$ and $E_{current}$ are replaced by S_{new} and E_{new} values respectively, this means that

```
Scurrent = Initial solution
E(Scurrent) = Energy of initial solution
Sbest = Scurrent, E(Sbest) = E(Scurrent)
T = Initial Temperature value
Tfinal = Final temperature value
α= Alpha value
While (T > Tfinal) do // Temperature cycle
    While (stop criteria) // Metropolis cycle
      Snew = perturbation (Scurrent)
       Obtain difference between Snew and Scurrent
      If (difference <= 0) then
        Accept Snew
       If E(Scurrent) < E(Sbest) then
         Sbest = Scurrent
         Ebest = Ecurrent
       End if
      End if
     else
       Boltzmann probability = exp(-difference/T)
          If (Boltzmann probability > random(0,1) then
            Accept Snew
        end if
     end if
    end while
    Decrease T by a cooling function
end while
```

Figure 1. Pseudo code of classical SA

the new solution is better than the current one, and a new difference is calculated, but now between $S_{current}$ and S_{best}; if this difference is less than zero, the current solution is better than S_{best} and its value is replaced by $S_{current}$ value. The same happens with energy values, $E_{current}$ is assigned to E_{best}; in this case, the MC starts again if stop criteria is not reached yet. On the other hand, when the current solution is better than the new one, Boltzmann probability is calculated and if this value is greater than a random number between zero and one, the new solution (a bad solution) is accepted and replaces $S_{current}$ value and E_{new} replaces $E_{current}$ value; if not, the thermal equilibrium (stop criteria) is evaluated and Metropolis Cycle is repeated until thermal equilibrium is reached.

Finally the best solution obtained by the simulated annealing algorithm is stored at S_{best} and its evaluation at E_{best}.

```
Read customer coordinates
Read customer demands
Calculate distances between pair of customers
Calculate the savings matrix
Sort the savings matrix
While (there are savings without to process) do
 customers i and j of the maximum savings not processed
 if customer i or j has not been processed then
   Obtain Ri or Rj (routes of i or j respectively)
  Modify Route Rj or Ri
 else // merge two routes
  Merge two routes
 end if
End while
```

Figure 2. Pseudo code of classical SA

4 SAVINGS ALGORITHM

One of the most widely used algorithms for the VRP is the savings algorithm [24]. In order to maximize the saving, two different routes *(0,...,i,0)* and *(0,j,...,0)* can be combined to form a new one. In the new route, the arcs *(i,0)* and *(0,j)* will not be used and the arc *(i,j)* will be added.

In this paper, a modified savings is described, for which the pseudo code is shown in figure 2. The customer coordinates are read from a file of instances. The customer demands are read too. Euclidean distance is calculated between pair of customers.

The cost matrix is used in order to calculate the savings by the equation $s_{ij} = c_{i0} + c_{0j} - c_{ij}$ [24]. The values of savings are sorted. There is a cycle which is done while there are savings processing. In this cycle, the savings are processed in order to add customers to the route or to merge two routes. Customer i and j are from the savings matrix. If customer i or j have been processed, the other one can be added to the customer processed route.

5 HYBRID SIMULATED ANNEALING

In order to solve the capacitated vehicle routing problem, a new hybridized algorithm has been designed and implemented. This new approach combines the classical simulated annealing with the

```
// Setting section
Setting initial temperature (Tinitial)
Setting final temperatures (Tfinal)
Setting cooling factor (alpha)
Generate stochastic demands of all customer
Create a factible initial solution of CVRP
Calculate the cost of this initial solution of CVRP
Scurrent is initialized with the initial solution of CVRP
Sbest is initialized with initial solution of CVRP
T is equal to Tinitial
Lcm_max is initialized
// Simulated annealig section
While (T > Tfinal) do // Temperature cycle
lcm = 1
While (lcm < Lcm_max) // Metropolis cycle
 Generate stochastic demand of a customer in random way
 Create new solution (Snew)of CVRP using Savings Algorithm or a random way
 Obtain difference of costs between Snew and Scurrent
  If (difference <= 0) then
   The new solution of CVRP is accepted
   If the new solution is best than the best solution then
    Sbest is equal to Snew
   End if
  Else
    Boltzmann probability = exp(-difference/T)
    If (Boltzmann probability > random(0,1) then
     The new solution of CVRP is accepted
   else
    The new solution of CVRP is rejected
    End if
  End if
  lcm = lcm + 1
End while
Decrease T by cooling function T = alpha * T
End while
Show Sbest, of CVRP
```

Figure 3. Pseudo code of SAMCSA

savings algorithm. The last algorithm is applied to generate new CVRP solutions with customer stochastic demands. Simulated annealing with Metropolis cycle based on savings algorithm (SAMCSA) is applied to generate both an initial and a new solution into Metropolis cycle. Simulated annealing is applied to reduce the temperature value and to accept or reject solutions. If a new CVRP solution is better than the previous one, the new solution is accepted; on the other hand, the new solution can be accepted or rejected by applying Boltzmann probability.

The pseudo code of SAMCSA algorithm is shown in figure 3. This algorithm is divided in two sections, the first one is named setting section, where several settings are specified. The second one is named simulated annealing section, in which the simulated annealing algorithm is specified. The initial and final temperatures ($T_{initial}$ and T_{final}) are set. The cooling factor alpha (α) is set. Demands of all customers are generated in a stochastic way. A feasible initial solution ($S_{initial}$) of CVRP is created, either by applying savings algorithm or in a random way. The CVRP initial solution's cost (CSinitial) is calculated. The CVRP current solution (Scurrent) is set to initial solution (see line 7). The best CVRP solution is set to initial solution. The variable T is set to initial temperature (Sinitial). The maximal length of Metropolis cycle is set.

In the simulated annealing section, the two cycles of SA are described. The first one is named Temperature cycle, which contains the Metropolis cycle and reduces the temperature values. While the variable T is greater than final temperature, this cycle is executed. The Metropolis cycle length is started at 1; then it is increased until the maximal length is reached. The Metropolis is executed while the length is less than the maximal length of MC. In MC, the stochastic demand of a customer is generated in a random way; this change of demand is then used to generate a new CVRP solution (S_{new}) by applying alternatively the savings algorithm or in a random way. The cost difference between S_{new} and $S_{current}$ is calculated. In order to solve a minimization problem, a new CVRP solution is accepted if the difference is less than or equal to zero. If the new CVRP solution is better than the best CVRP solution then the best solution is replaced by S_{new}. If the difference between S_{new} and $S_{current}$ is greater than zero, Boltzmann probability is calculated by e-Difference/T. If this probability is greater than a random number within range (0,1) then the new CVRP solution is accepted, else this solution is rejected.

Table 1. Solomon's instances

Instance number	Solomon´s instance	Customers
1	A-n32-k5	32
2	A-n33-k5	33
3	A-n33-k6	33
4	A-n34-k5	34
5	A-n36-k5	36
6	A-n37-k5	37
7	A-n37-k6	37
8	A-n38-k5	38
9	A-n39-k5	39
10	A-n39-k6	39
11	A-n44-k6	44
12	A-n45-k6	45
13	A-n45-k7	45
14	A-n46-k7	46
15	A-n48-k7	48
16	A-n53-k7	53
17	A-n54-k7	54
18	A-n55-k9	55
19	A-n61-k9	61
20	A-n65-k9	65
21	B-n31-k5	31
22	B-n34-k5	34
23	B-n35-k5	35
24	B-n38-k6	38
25	B-n39-k5	39
26	B-n41-k6	41
27	B-n43-k6	43
28	B-n44-k7	44
29	B-n45-k5	45
30	B-n45-k6	45
31	B-n50-k7	50
32	B-n51-k7	51
33	B-n52-k7	52
34	B-n56-k7	56
35	B-n57-k7	57
36	B-n57-k9	57
37	B-n63-k10	63
38	B-n64-k9	64
39	B-n67-k10	67

Table 2. Comparison of results

Instance number	Solomon´s instance cost	Solomon´s instance routes
1	784	5
2	661	5
3	742	6
4	778	5
5	799	5
6	669	5
7	949	6
8	730	5
9	822	5
10	831	6
11	937	6
12	944	6
13	1146	7
14	914	7
15	1073	7
16	1010	7
17	1167	7
18	1073	9
19	1034	9
20	1174	9
21	672	5
22	788	5
23	955	5
24	805	6
25	549	5
26	829	6
27	742	6
28	909	7
29	751	5
30	678	6
31	741	7
32	1032	7
33	747	7
34	707	7
35	1153	7
36	1598	9
37	1496	10
38	861	9
39	1032	10

Table 3. Comparison of results

Instance number	Minimum SAMCSA's cost	Routes obtained	SAMCCA's average cost
1	771.47	4	812.03
2	647.48	5	662.97
3	733.43	6	739.94
4	775.95	6	783.82
5	781.35	5	800.07
6	673.57	5	694.92
7	905.98	6	929.15
8	716.15	5	726.76
9	824.44	5	838.53
10	827.23	6	835.15
11	928.61	6	937.64
12	917.14	6	929.37
13	1148.84	7	1168.68
14	892.51	7	913.42
15	1064.61	7	1087.96
16	1014.15	7	1025.87
17	1162.11	7	1181.58
18	1076.55	9	1083.63
19	1033.58	9	1054.49
20	1182.21	9	1210.00
21	616.77	4	631.92
22	772.28	5	777.24
23	887.65	5	965.37
24	691.79	5	739.64
25	539.56	5	548.61
26	798.42	6	812.12
27	721.61	6	733.32
28	848.60	7	865.96
29	707.08	5	727.57
30	666.38	5	685.99
31	685.60	7	719.34
32	1007.75	7	1027.85
33	694.89	7	705.70
34	635.08	7	653.12
35	1141.53	7	1152.54
36	1524.67	8	1588.03
37	1511.30	10	1529.92
38	839.17	9	862.70
39	1032.37	10	1067.77

Table 4. Comparison of results between savings and SAMCSA algorithms

Instance number	Savings' average cost	Minimum SAMCSA's cost	SAMCSA's average cost
1	857.66	771.47	812.03
2	706.57	647.48	662.97
3	781.96	733.43	739.94
4	824.77	775.95	783.82
5	870.92	781.35	800.07
6	714.23	673.57	694.92
7	1000.79	905.98	929.15
8	772.74	716.15	726.76
9	895.33	824.44	838.53
10	872.58	827.23	835.15
11	987.56	928.61	937.64
12	1005.34	917.14	929.37
13	1224,48	1148.84	1168.68
14	974.99	892.51	913.42
15	1153.90	1064.61	1087.96
16	1093.51	1014.15	1025.87
17	1246.41	1162.11	1181.58
18	1134.50	1076.55	1083.63
19	1119.53	1033.58	1054.49
20	1297.23	1182.21	1210.00
21	684.27	616.77	631.92
22	845.82	772.28	777.24
23	1019.67	887.65	965.37
24	837.12	691.79	739.64
25	580.39	539.56	548.61
26	908.40	798.42	812.12
27	793.43	721.61	733.32
28	944.41	848.60	865.96
29	773.36	707.08	727.57
30	734.65	666.38	685.99
31	774.40	685.60	719.34
32	1140.61	1007.75	1027.85
33	793.64	694.89	705.70
34	741.77	635.08	653.12
35	1287.59	1141.53	1152.54
36	1671.24	1524.67	1588.03
37	1640.85	1511.30	1529.92
38	943.04	839.17	862.70
39	1115.96	1032.37	1067.77

6 EXPERIMENTATION AND RESULTS

SAMCSA is implemented using Powerbuilder software tool. The parameters of simulated annealing are established in an experimental way. Initial temperature is set at 1 000, and final temperature is set at 1. The alpha factor is set to 0.85.

In this section, the results obtained are shown and analyzed. SAMCSA is tested with 39 CVRP instances (table I), which were defined in Solomon's instances. The information of each instance includes its name instance and the number of customers. The capacity of all instances is equal to 100.

The comparison of results between optimal solutions of Solomon's instances and SAMCSA are shown on tables II and III. In these tables are shown the Solomon's instance costs, number of routes of each one, as well as the SAMCSA's solution minimum and average costs and the number of routes obtained by this approach.

These tables show that the new proposed approach obtained better results than the optimal reported at Solomon's instances. At 30 of 39 (76.92%) instances, SAMCSA improved the minimal cost for a CVRP solution.

The algorithm was executed 30 times by each Solomon instance. An average cost was calculated. The proposed algorithm improved average cost of some instances.

On table IV, the comparison between savings algorithm and SAMCSA algorithm is shown. The new proposed approach obtained better results than the optimal ones reported by savings algorithm.

The results obtained by SAMCSA were compared with other algorithms as well, for example the hybrid algorithm based on Ant colony optimization (ACO) and Particle swarm optimization (PSO) proposed by Yucheng Kao and colleagues [25], an improved Clarke and Wright savings algorithm proposed by Pichpibul and Kawtummachai [26], and a multi-space sampling heuristic proposed by Mendoza and Villegas [27].The quality solutions obtained by SAMCSA are better than the quality solutions of these algorithms.

7 CONCLUSIONS

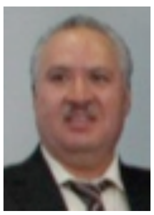
The new proposed approach to solve the capacitated vehicle routing problem with stochastic demands generates better high quality solutions at most of Solomon's instances, specifically in 76.92% of 39 instances. The average cost of 30 runs was better than the optimum reported at Solomon's instances. The average cost improved in 19 instances of 39 (48.71%). In this case, the instances improved were 3, 7, 8, 12, 14, 21, 22, 24-29, 31-36. SAMCSA algorithm reduced the number of routes and the cost of the solution at six Solomon's instances. In order to generate high quality solutions for CVRP, simulated annealing and savings algorithms work better combined than each by itself. As a future work, this new approach can be modified in order to generate high quality solutions for CVRP. This algorithm can be hybridized with other heuristics or metaheuristics.

REFERENCIAS

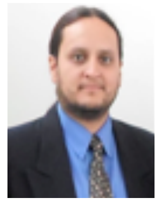
1. Dantzig, G., Fulkerson, R., Johnson, S. Solution of a large-scale traveling-salesman problems. *Operations Research*. 1954, 2 (4), 393-410.
2. Dantzig, G., Ramser, J. The truck dispatching problem. *Management Science*. 1959, 6 (1), 80-91.
3. Chiang, W. Ch., Russell, R. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research*. 1996, 63 (1), 3-27.
4. Bräysy, O., Dullaert, W., Gendreau, M. Evolutionary algorithms for the vehicle routing problem with time windows. *Journal of Heuristics*. 2004, 10 (6), 587-611.
5. Nagy, G., Salhi, S. Location-routing: Issues, models and methods. *European Journal of Operational Research*. 2007, 177 (2), 649-672.
6. Flatberg, T. *Dynamic and stochastic aspects in vehicle routing: A literature survey*. SINTEF report. Oslo: SINTEF ICT, 2005. ISBN 8214028430.
7. Bräysy, O., Gendreau, M. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*. 2005, 39 (1), 104-118.
8. Laporte, G., Nobert, Y. Exact algorithms for the vehicle routing problem. *Surveys in Combinatorial Optimization*. 1987, 31, 147-184.
9. Laporte, G. The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*. 1992, 59 (3), 345-358.
10. Bell, J. E., McMullen, P. R. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*. 2004, 18 (1), 41-48.
11. Osman, IH. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research*. 1993, 41 (1-4), 421-451.
12. Czech, Z. J., Czarnas, P. Parallel simulated annealing for the vehicle routing problem with time windows. In: *10th Euromicro Workshop on Parallel, Distributed and Network-based Processing*. Canary Islands-Spain, 2002, 376-383.
13. Thangiah, S. R. Vehicle routing with time windows using genetic algorithms. In: Chambers, L. (Ed.). *The Practical Handbook of Genetic Algorithms: New Frontiers, Volume II*. Florida: CRC Press, 1995, 253-277.
14. Homberger, J. H. G. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research*. 2005, 162 (1), 220-238.
15. Garcia, B. L., Potvin, J. Y., Rousseau, J. M. A parallel implementation of the tabu search heuristic for vehicle routing problems with time window constraints. *Computers and Operations Research*. 1994, 21 (9), 1025-1033.
16. Tas, D., Dellaert, N., Van Woensel, T., De Kok, T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers and Operations Research*. 2013, 40 (1), 214-224.
17. Kirkpatrick, S., Gelatt, C., Vecchi, M. Optimization by simulated annealing. *Science*. 1983, 4598, 671-680.
18. Cerny, V. Thermodynamical approach to the traveling salesman problem: An eficient simulation algorithm. *Journal of Optimization Theory and Applications*. 1985, 45, 41-51.
19. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*. 1953, 21 (6), 1087-1092.
20. Aarts, E., Korst, J. *Simulated annealing and Boltzmann machines: A stochastic approach to combinatorial optimization and neural computing*. New York: John Wiley and Sons, 1989.
21. Ingber, L. Simulated annealing: Practice versus theory. *Mathematical and Computer Modelling*. 1993, 18 (11), 29-57.
22. Kjaerulff, U. Optimal decomposition of probabilistic networks by simulated annealing. *Statistics and Computing*. 1992, 2, 7-17.
23. Van Laarhoven, P. J., Aarts, E. H. L. Simulated annealing: Theory and applications. Dordrecht: Kluwer Academic Publishers, 1987.
24. Clarke, G., Wright, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*. 1964, 12 (4), 568-581.
25. Kao, Y., Chen, M. H., Huang, Y. T. A hybrid algorithm based on ACO and PSO for capacitated vehicle routing problems. *Mathematical Problems in Engineering*. 2012. Available from: doi:10.1155/2012/726564
26. Pichpibul, T., Kawtummachai, R. An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem. *Science Asia*. 2012, 38, 307-318.
27. Mendoza, J. E., Villegas, J. G. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters*. 2013, 7, 1503-1516.

About the authors

Linda Crystal Cruz Villegas was born in Monclova, Coahuila, Mexico. She has a Computational Systems Engineer degree from the Faculty of Mechanical and Electrical Engineering of Autonomous University of Coahuila (2007). Currently, she is studying a Master on Applied Engineering with accent on Integral Manufacturing at the Faculty of Systems of the Autonomous University of Coahuila in Saltillo, Coahuila, Mexico. Her interest areas are optimization, metaheuristics, logistics and supply chain.



Ernesto Liñán García has a PhD in Computer Science, MSC in Computer Systems, and Electronic Systems Enginner degree from Instituto Tecnológico y de Estudios Superiores de Monterrey. He is a full Research-Professor in Computer Science at University of Coahuila, Mexico. His research area is metaheuristics applied to NP-hard problems, protein folding problems, DNA alignment sequences, and vehicle routing problems.



David Salvador González González has a PhD in Science and technology in Industrial and Manufacturing Engineering, a Master in Science and Technology in Industrial Engineering and is an Industrial and systems engineer. He specializes in the development of statistical tools for industrial applications and troubleshooting. Research development in reliability, experimental designs, probabilistic models and application of hybrid models and soft-computing for the solution of industrial problems. Interested in modeling and optimization of industrial processes. Fulltime research professor at University of Coahuila, Mexico.