Scientific Research and Essays Vol. 5(22), pp. 3529-3537, 18 November, 2010
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2010 Academic Journals

Full Length Research Paper

Population pre-selection operators used for generating a non-random initial population to solve vehicle routing problem with time windows

Ocotlán Díaz-Parra, Jorge A. Ruiz-Vanoye* and José C. Zavala-Díaz

Universidad Autónoma del Estado de Morelos - FCAeI, Mexico.

Accepted 13 October, 2010

In this paper, we propose two population pre-selection operators to be improved by the k-means algorithm (clustering) and neighborhood techniques in the determination of the initial population (non-random population) used in the evolutionary and genetic algorithms to solve the vehicle routing problem with time windows (VRPTW). The results provide individual or group chromosomes in the initial population that is close to the optimal in VRPTW instances.

Key words: Pre-selection operator, k-means algorithm, neighborhood, logistics, vehicle routing, VRPTW.

INTRODUCTION

The vehicle routing problem (VRP) is a combinatorial optimization problem that is very complex. It is considered as a central problem in the areas of transportation, logistics and distribution. In some sectors of the industry, transportation means a high percentage of value added to products, therefore the uses of computational methods offer good results to transportation, and as such, it has great savings from 5 to 20% in the total costs (Toth and Vigo, 2001).

The vehicle routing problem with time windows (VRPTW), a variant of VRP, consists of minimal transportation costs that are fulfilling time restrictions of each route and vehicle capacity, considering the demand of each client (Toth and Vigo, 2001).

In the VRPTW, there exist different types of instances: *C-type* = clustered data, *R-type* = uniformly distributed data or random data, *RC-type* = random and clustered data. The value one is for small time window and small vehicle capacity, while the value two is for large time window and large vehicle capacity. The parameters of the VRPTW instances are: *VN* = vehicle number, *C* = capacity, *CN* = customer number, *XCO* = X coordinate, *YCO* = Y coordinate, *D* = demand, *RT* = ready time, *DT* = Due date and *ST* = service time. However, the Equations 1 to 11 represent the VRPTW model (Toth and Vigo, 2001):

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ijk} \tag{1}$$

$$\sum_{k \in K} \sum_{j \in \Delta^+(i)} x_{ijk} = 1; \qquad \forall i \in N \tag{2}$$

$$\sum_{j \in \Delta^+(0)} x_{0jk} = 1; \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in \Delta^-(j)} x_{ijk} - \sum_{i \in \Delta^+(j)} x_{ijk} = 0 \quad \forall k \in K, j \in N \tag{4}$$

$$\sum_{i \in \Delta^-(n+1)} x_{i,n+1,k} = 1 \quad \forall k \in K \tag{5}$$

$$w_{ik} + s_i + t_{ij} - w_{jk} \le (1 - x_{ijk}) M_{ij} \quad \forall k \in K, (i,j) \in A \tag{6}$$

$$a_i \sum_{j \in \Delta^+(i)} x_{ijk} \le w_{ik} \le b_i \sum_{j \in \Delta^+(i)} x_{ijk} \quad \forall k \in K, i \in N \tag{7}$$

$$E \le w_{ik} \le L \quad \forall k \in K, i \in (0, n+1) \tag{8}$$

$$\sum_{i \in N} d_i \sum_{j \in \Delta^+(i)} x_{ijk} \le C \quad \forall k \in K \tag{9}$$

*Corresponding author. E-mail: jorge@ruizvanoye.com.

$$x_{ijk} \geq 0 \quad \forall k \in K, (i, j) \in A \tag{10}$$

$$x_{ijk} \in \{0,1\} \quad \forall k \in K, (i, j) \in A \tag{11}$$

where: 0 = the deposit at the beginning of the route, A = the set formed by ordered pairs (i, j), a = the time window limit in a node, b = the time window limit in a node, C = capacity of vehicle, c = cost of service, d = demand, E = beginning time window in the deposit, i = origin node, j = destiny node, K = the fleet of vehicles, k = vehicle, L = the time window limit in the deposit, M = a very big positive number, N = the set of nodes, n+1 = the deposit when the route has already been realized, S = service time, t = the time of arrival to the following node, W = the beginning of the service time in the node, X = the execution of an operation, $i \in \Delta^-(j)$ = an origin i together with a destiny j with direction of j directly to i and $j \in \Delta^+(i)$ = a destiny j together with an origin i with direction of i to j.

The motivation of this paper is to apply a new concept called population pre-selection operators to improve the determination of the initial population of the vehicle routing problem with time windows.

To improve the determination of the initial population of the VRPTW, Blanton and Wainwright (1993), Thangiah (1998), Affenzeller (2002), Rahoual et al. (2001) and Cruz-Chávez et al. (2008) used a random method in generating the initial population. Thangiah et al. (1995) use the cheapest insertion heuristic to route customers within each group. Potvin and Bengio (1996), Zhu (2000) and Tan et al. (2001) propose to create the initial population with Solomon's insertion heuristic, while Homberger and Gehring (1999) and Gehring and Homberger (2001) use a saving heuristic to generate an initial population. Berger et al. (1998) generated the initial population using Solomon's nearest neighbor heuristic, whereas Bräysy (1999a) proposed the modified heuristics for the initial population and randomly created routes. Later, Bräysy (1999b) proposed the creation of the initial population using the nearest neighbor heuristic, after which Berger et al. (2001) generated an initial population by insertion procedure. Bräysy et al. (2001) used a modification of Bräysy's genetic algorithm to create an initial population for an evolutionary algorithm, whereas Rajmohan et al. (2008) generated the initial population using 'greedy randomized adaptive search procedure'.

The paper is organized as follows. Firstly, it describes the operators of pre-selection and afterwards, presents the results, discussion and conclusions.

K-means algorithm and OR neighborhood technique

The K-means algorithm was introduced by MacQueen (1967), and it is one of the group algorithms that is well-known and which classifies the groups of data with similar characteristics. However, these groups know themselves like clusters (Perez et al., 2007). K-means algorithm is used (in spite of its disadvantages) as a traditional method of clustering, besides being the method used in diverse commercial applications such as SPSS (2010) and Weka (Witten and Frank, 2005). The k-means algorithm has four phases:

1. Initialize phase: It obtains the training set $x = [x_1, x_2, ..., x_n]$ and initializes the values of m with 0, where m is the number of iterations. Also, it selects the starting points (k) within the space represented by the objects that are desired to group the well-known training set of the random or systematic form. As such, these points will represent the initial centroids $c_k(0)$ of the groups or initial clusters $c = \{c_1, c_2, ..., c_k\}$ with $c_k(0), 1 \leq k \leq m$.
2. Classification phase: It assigns, classifies or distributes to the set of training (x) within clusters $c_k(m)$, using the rule of nearer neighbor or nearer centroid:

a. It calculates the centroid (M) that is given by the average M_k of each group c_k, (equation 12) where: M_k = the average of each group, x = the training set and n_k = the number of elements:

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} \tag{12}$$

b. It calculates the similarity measurement using the error to square and the total square error with the purpose of verifying the variation within the cluster (Equations 13 and 14), where: $e_k^{\ 2}$ = the square error, M_k = the average of each group, x = training set and E_k^2 = the total square error:

$$e_k^{\ 2} = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2 \tag{13}$$

$$E_k^2 = \sum_{k=1}^{k} e_k^2 \tag{14}$$

c. It calculates the measurement of the Euclidian distance (or the measurement of Manhattan distance) (Equation 15), where: $d(M_k, x_n)$ = Euclidean distance, M_k = the average of each group and x = the training set:

$$d(M_k, x_n) = \sqrt{|M_{k1} - x_{n1}|^2 + |M_{k2} - x_{n2}|^2 + ... + |M_{ki} - x_{ni}|^2} \tag{15}$$

d. Cluster the training set x_n into the clusters c_k (m) using the minimum distance to the nearer centroid (Equation 16).

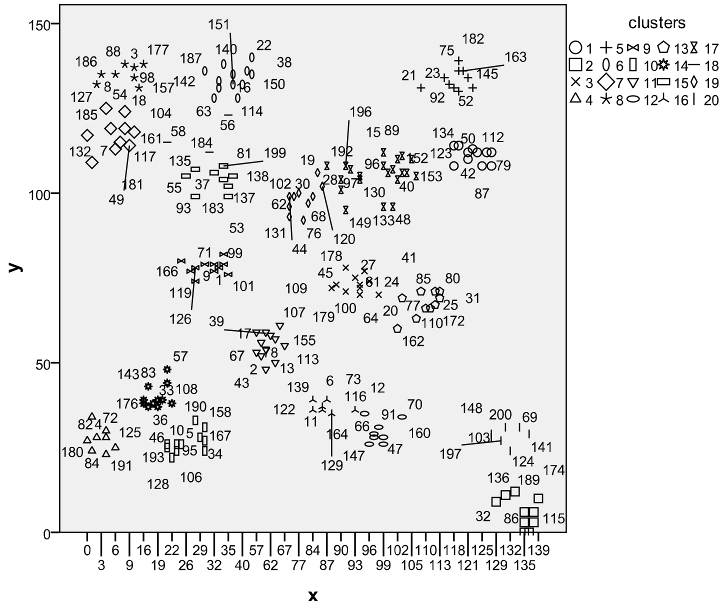

Figure 1. Clustering.

$$x \in C_k(m),\ \text{if}\ d(M_i, x_n) \leq d(M_j, x_n),\ j \neq i \tag{16}$$

3. Update phase: It updates the value of m increasing the value in 1 and updates each to cluster the recalculation of centroids M_k (m) of the training set.
4. Stop criterion phase of the k-means algorithm: It repeats steps two and three until the centroids do not move (Equations 17):

$$M_k(m) = M_k(m-1) \tag{17}$$

Figure 1 shows the initial population (c1_2_1 instance of VRPTW) obtained by the k-means algorithm (the figure is obtained courtesy, SPSS software).

In the 'Or' (1976) Neighborhood technique, the neighborhood of a solution is in equation 18 and it is the set of all those solutions that can be attainable from a *s′* solution by a σ movement (Martínez-Morales, 2006). As such, a movement can be interchanged among elements that conform to the *s* solution:

$$N(s) = \{s' \in S : s \xrightarrow{\sigma} s'\} \tag{18}$$

where N(s) represents the neighborhood with respect to s, s represents a solution that is taken from the S solutions' total space and s' represent a neighbor of s generated by σ movements.

A movement can be an insertion, elimination or interchange of components in a solution. For this case in particular, a movement will be an interchange of two genes in an individual. In Figure 2, it is the graphical representation of the neighborhood generated from an s solution.

Given an example, the N(s) neighborhood generated by an s solution is represented in Figure 1. However, N(s) is shown with a circle and the neighbors by $S = \{s_1, s_2, s_3, \ldots, s_i\}$. The structure of neighborhood by a search allows a feasible solution to be found in the landscape. The neighborhood is based on the Or´s proposition for the traveling salesman problem or TSP (1976). The Or method of interchange is a variant of the well-known r-optimal interchanges developed by Lin and Kernighan for the symmetric TSP (Lin, 1965; Lin and Kernighan, 1973). A r-optimal interchange consists of moving chains of r elements. For example, it is a 0-1-2-3-4-0-5-6-7-0-8-9-10-0 valid route for a problem of routes. However, a 2-optimal interchange consists of interchanging chains of two numbers; then for the given initial chain, chains 3-4 and chains 8-9 of such form will be interchanged and the following feasible route will be defined by: 0-1-2-8-9-0-5-6-7-0-3-4-10-0. For this work, the use of Or technique is proposed only with 1-optimal taking movements forward (Pacheco and Delgado, 2000).

POPULATION PRE-SELECTION OPERATORS

We propose two population pre-selection operators to improve the use of k-means algorithm (clustering) and neighborhood techniques in determining the initial population (non-random population) used for solution of the vehicle routing problem with time windows. The population pre-selection operators proposed are based on:

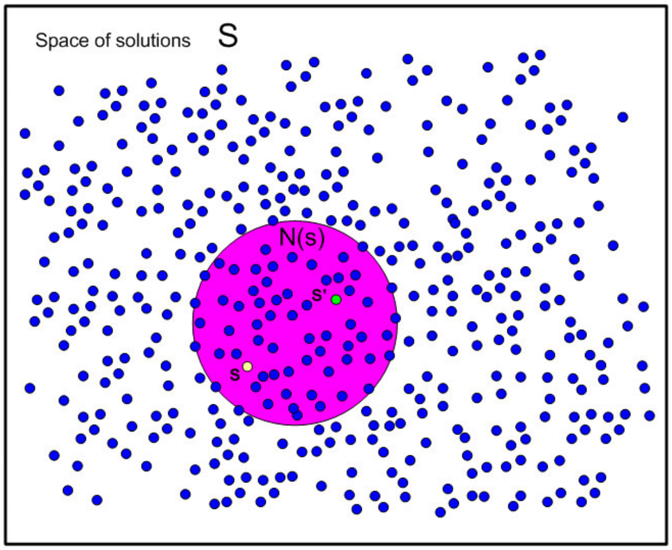

Figure 2. Neighborhood generated from individual solution.

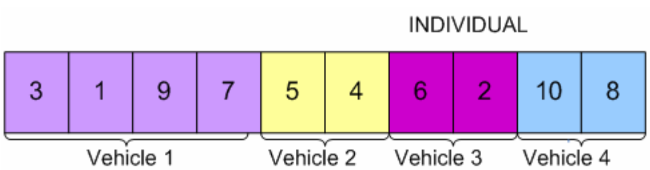

Figure 3. One individual obtained by 'clusterization' and 'cluster and neighborhood' phases.

Clusterization

The k-means algorithm for clustering (data-mining techniques), introduced by MacQueen in 1967, is one of the clustering algorithms that is well-known and which classifies the groups of data with similar characteristics. These groups know themselves like clusters and the pre-selection operator groups have similar characteristics of VRPTW instances. The parameters provided to the k-means algorithm are: XCO = X coordinate and YCO = Y coordinate; and the number of clusters (the number of vehicles' average reported in Solomon (1987) benchmark for type C, RC, or R instances). With the use of k-means algorithm, one individual that represents the initial population is obtained (Figure 3). The initialization of the k-means algorithms was obtained by the characteristic instance called vehicles or K (Equation 19), where: K is the number of vehicles or clusters for the initial population, b = the time window limit in a node, C = Capacity of vehicle, d = demand, N = the set of nodes or customers:

$$K = Random[(\sum_{i=1}^{N} d_i / C) + Frequency(b_i)];\ i = 1, 2, ..., N; Random \in Z^{+} \quad (19)$$

Cluster and neighbourhood

The proposed pre-selection operator is based first on clusterization (k-means algorithm) of the initial population and Or (1976) neighborhood technique. The population pre-selection operators provide an initial population that is neither non-feasible nor optimum. The non-feasibilities will become feasible by means of verifying the demand, time and vehicle restrictions. In VRPTW, each movement in the total route infers *n*-searches for not violating the own restrictions of the problem. It is advisable to use movements with one chain, to this new adjustment we called Or neighborhood technique. The Or neighborhood then takes two nodes with 1-optimal interchange of chains and the movement is realized forward and the search in the neighborhood is a classic local search. As such, the size of the neighborhood is based on the size of the individual and the number of genes to be interchanged (Equation 20):

$$_{N(s)} = \frac{SizeIndiv(SizeIndiv - 1)}{numGenMuta} \quad (20)$$

where: $T_{N(s)}$ represents the size of the neighborhood generated from *s*. For example, for an individual formed by ten genes with an interchange of two genes, *SizeIndiv* = 10 and *numGenMuta* = 2, therefore $T_{N(s)}$ = 45, which infers that only 45 neighbors will be generated. The criterion of unemployment of the neighborhood proposes twice the size of the neighborhood in Equation 21:

$$Stop_{N(s)} = 2T_{N(s)} \quad (21)$$

where: $Stop_{N(s)}$ represents the stop criterion of the search in the $N(s)$ neighborhood.

The pre-selection operators are used only for generating the initial population (an individual with *n* chromosomes, set of genes or vehicles) (Figure 3) of an intelligent way (the non-random way is likely used normally by other algorithms), and not for obtaining the solution of the instance (but we obtain a feasible solution close to the optimum). Later, the individual generated in a non-random way can serve in the future like a parameter to the next phase of the genetic algorithm (to the selection, crossover and mutation operator).

For example, the proposed pre-selection operators could be used as a mechanism of pre-selection in the initial population within the genetic algorithm, and as such, the main difference is that the generation of the initial population is realized randomly. Therefore, the study's proposal states that the generation of the initial population was done by a pre-selection operator (clusterization or neighborhood). The modified genetic algorithm consists of the following phases (Figure 4):

1. Pre-selection: Generate a non-random initial

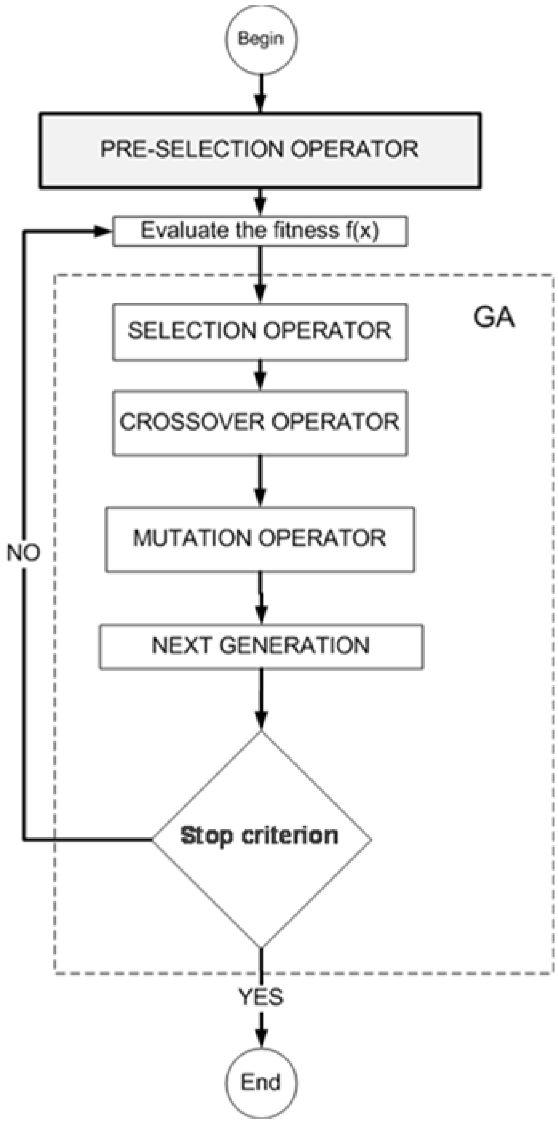

Figure 4. GA with pre-selection operators.

population (one individual) of *n* chromosomes suitable for solution of the problem and the phase proposed in this paper.
2. New population: Create a new population by repeating the following steps until the new population is complete:

a. Generate a random population of *n* chromosomes based on the non-random initial population or the population of the new generation that is interchanged randomly by a pair of genes of the individuals.
b. Fitness: Evaluate the fitness *f*(*x*) of each chromosome *x* in the population.
c. Selection operator: Select two parent chromosomes from a population according to their fitness (the better the fitness, the bigger its chance of being selected).
d. Crossover operator: With a crossover probability, it crosses over the parents to form a new offspring (children). If no crossover was performed, the offspring is an exact copy of the parents.
e. Mutation operator: With a mutation probability, it mutates new offspring at each locus (position in chromosome).

3. Accepting: Place new offspring in a new population.
4. Replace: Use a new generated population for a further run of algorithm.
5. Test: If the end condition is satisfied, stop and return the best solution in the current population.
6. Loop: Go to step 2.

RESULTS

The experimentation was carried on an Acer Travelmate 2330LC computer with an Intel Celeron processor at 1.5GHz, 512 MB and 80 GB hard disk. We used the k-means algorithm for clustering (clusterization operator) similar characteristics and the geographical localization for clustering the individual and verifying the demand, time and vehicle restrictions. For the cluster and neighborhood operator, we use the k-means algorithm of the initial population and the Or neighborhood technique.

The instances of problem VRPTW (Table 1) were obtained from Solomon (1987) benchmark (it contains a set of VRPTW instances of different sizes, or number of customers: 25, 50, 100, 200, 400, 600, 800 and 1000 nodes, while the sets of problem instances of 200, 400, 600, 800 and 1000 customers contain 60 instances in each set). Experiments are repeated 30 times for each instance, where: VN = vehicle number, Ca = capacity, CN = customer number, XCO = X coord., YCO = Y coord., D = demand, RT = ready time, DT = due date and ST = service time. In the VRPTW, additional parameters existed, such as: C = clustered data, R = random data, RC = random and clustered data, the value 1 is for small time window and small vehicle capacity and 2 is the big time window and big vehicle capacity. Here, we only mention some results obtained from the experimentation, while in Table 2, the results of the pre-selection operators are obtained.

In Table 3, some results obtained for the initial populations from the pre-selection operators were compared with the results obtained for the initial populations from a validation tool (Heuristic Lab). Heuristic lab is a framework for heuristic and evolutionary algorithms. As such, we use GGA, SSGA and SXGA (GGA: general genetic algorithm, SSGA: steady state genetic algorithm and SXGA: sexual genetic algorithm), contained in the heuristics lab (Wagner and Affenzeller, 2004), for obtaining the random initial population. In GGA algorithm, the input values were used: Generations = 1000, population size = 100, selection operator = roulette, crossover operator = OPX, crossover rate = 1 and mutation rate = 0.05. The data for the SSGA algorithm

Table 1. VRPTW instances.

VN	*Ca*					
CN	XCO	YCO	D	RT	DT	ST
0	x_0	Y_0	d_0	RT_0	DT_0	ST_0
…	…	…	…	…	…	…
100	x_{100}	Y_{100}	d_{100}	RT_{100}	DT_{100}	ST_{100}

Table 2. Some results obtained for the pre-selection operators (30 times for each instance).

Instances	Nodes	Bestknown	Clusterization	Cluster and neighborhood
c101	100	828.94/10	1085.46/10	849.49/10
c102	100	828.94/10	1126.79/10	841.76/10
c103	100	828.06/10	1054.31/10	969.54/10
c104	100	824.78/10	1122.44/10	875.99/10
rc101	100	1696.94/14	2400.12/14	1900.00/14
rc102	100	1554.75/12	2010.14/12	1774.53/12
r101	100	1645.79/19	1869.72/19	1653.20/19
r102	100	1486.12/17	1934.78/17	1502.98/17
c1_2_1	200	2704.57/20	2879.56/20	2715.40/20
c1_2_2	200	2917.89/18	9112.25/18	5250.06/18
rc1_2_1	200	3602.80/18	10395.54/18	5188.42/18
rc1_2_2	200	3249.50/18	11016.71/18	6014.61/18
r1_2_1	200	5024.65/19	10221.45/19	5065.06/19
r1_2_2	200	4040.60/18	9567.26/18	4200.33/18
c1_4_1	400	7152.02/40	18977.05/40	9987.43/40
c1_6_1	600	14095.64/60	24725.12/60	16656.88/60
c1_8_1	800	25030.36/80	50289.29/80	30004.23/80
c1_10_1	1000	42478.95/100	120902.40/100	60409.55/100

were used: selection operator = roulette, crossover operator = OPX, mutation operator = RandomSwap, mutation rate = 0.05, population size = 100, replacement operator = worst and tournament group size = 2. The data for the SXGA algorithm were used: male selection operator = roulette, female selection operator = roulette, crossover operator = OPX, mutation operator = RandomSwap, mutation rate = 0.05, generations = 1000, population size = 100 and tournament group size = 2.

In Figure 5, we show the evolution of the genetic algorithms with the pre-selection operators versus other algorithms, such as: GGA: General Genetic Algorithm, SSGA: Steady State Genetic Algorithm, SXGA: SeXual Genetic Algorithm, GA-Cluster: Genetic algorithm with clusterization pre-selection operator and GA-Cluster and OR: Genetic algorithm with cluster and neighborhood pre-selection operator.

DISCUSSION

In Table 2, the results of the pre-selection operators (clusterization, cluster and neighborhood) are close to the optimal solution in VRPTW instances. In other words, the initial population generated using the operators, is near to the optimal solution of the VRPTW instances.

In Table 3, the validation / comparison of the initial population of some algorithms versus the initial population obtained from the pre-selection operators are seen. We can observe that the pre-selection operator called clusterization gives the initial population with an acceptable solution, but it is even better with the other proposed pre-selection operator called ‘cluster and neighborhood’, compared with the results obtained from the generation of the random ‘initial population’ of the GGA, SSGA and SXGA after 30 execution times for each instance.

It is necessary to mention the differences of two concepts: Initial population and initial solution. In a genetic algorithm and evolutionary algorithm, a population of strings (chromosomes or the genotype) encodes candidate solutions (called individuals, creatures, or phenotypes) to a combinatorial optimization problem. Traditionally, solutions are represented in binary as strings of 0s and 1s, and other encodings. The initial population encodes initial candidate solutions represented

Table 3. Some results obtained for the initial populations (30 times for each instance).

I	N	IPGGA	IPSSGA	IPSXGA	IPCO	IPCNO	Best known
c101	100	4397.09/21	3418.41/22	4372.26/22	1085.46/10	849.49/10	828.94/10
c102	100	4400.28/23	4470.82/21	4230.78/22	1126.79/10	841.76/10	828.94/10
c104	100	4177.11/20	2742.74/13	4577.78/23	1122.44/10	875.99/10	824.78/10
c201	100	4464.10/21	3746.50/24	4561.21/22	892.71/4	714.54/4	589.1/3
c202	100	4379.70/22	1692.74/7	4715.70/22	981.44/5	766.29/4	589.1/3
rc101	100	4711.44/23	2561.74/23	4809.91/24	2400.12/14	1900.00/14	1696.94/14
rc102	100	4536.95/22	4024.91/20	4967.44/23	2010.14/12	1774.53/12	1554.75/12
rc201	100	4931.45/22	2905.42/12	4682.18/22	2970.63/9	1521.19/9	1261.8/9
rc202	100	5049.35/23	2611.92/9	4672.07/22	2541.30/9	1539.90/9	1092.3/8
r101	100	3717.61/23	3551.40/29	3715.51/21	1869.72/19	1653.20/19	1645.79/19
r102	100	3908.64/22	2960.11/25	3835.24/23	1934.78/17	1502.98/17	1486.12/17

I: Instances; N: Nodes; IPGGA: Initial population obtained from the general genetic algorithm; IPSSGA: Initial population obtained from the steady state genetic algorithm; IPSXGA: Initial population obtained from the sexual genetic algorithm; IPCO: Initial population obtained from the clusterization pre-selection operator; IPCNO: Initial population obtained from the cluster and neighborhood pre-selection operator.

Table 4. Related works.

Research	Methods used in generating the initial population
Blanton and Wainwright (1993)	Random method
Thangiah (1995)	Cheapest insertion heuristic
Potvin and Bengio (1996)	Solomon´s insertion heuristic
Thangiah (1998)	Random method
Berger (1998)	Solomon´s nearest neighbor heuristic
Homberger and Gehring (1999)	Saving heuristic
Homberger and Gehring (1999)	Threshold accepting meta-heuristic
Bräysy (1999a)	Random method
Bräysy (1999b)	Nearest neighbor heuristic
Gehring (1999)	Stochastic saving heuristic
Zhu (2000)	Solomon´s insertion heuristic
Tan (2001)	Solomon´s insertion heuristic
Rahoual (2001)	Random method
Gehring and Homberger (2001)	Saving heuristic
Berger (2001)	Insertion procedure or random insertion heuristic
Bräysy (2001)	A genetic algorithm
Affenzeller (2002)	Random method
Rajmohan et al. (2008)	Greedy randomized adaptive search procedure
Cruz-Chávez (2008)	Random method
This work (2010)	k-means algorithm and OR neighborhood technique

as strings and other encodings, whereas the initial solution is empty or the complete initial solutions are not represented as chromosomes or individuals. The approximate algorithms comprised: The construction algorithms (that is, greedy construction heuristic) which start with the empty initial solution and the local search algorithms (that is, iterative improvement) which start with the complete initial solution. The motivation of this paper is to apply a new concept called population pre-selection operators to improve the determination of the initial population of the vehicle routing problem with time windows and to improve the genetic and evolutionary algorithms (initial population). The improvement does not apply to other algorithms (initial solutions) such as ant colony optimization, tabu search and others.

Table 4 shows the related works of this research. The strengths of this research versus others are: the results that were provided (initial population) for the pre-selection

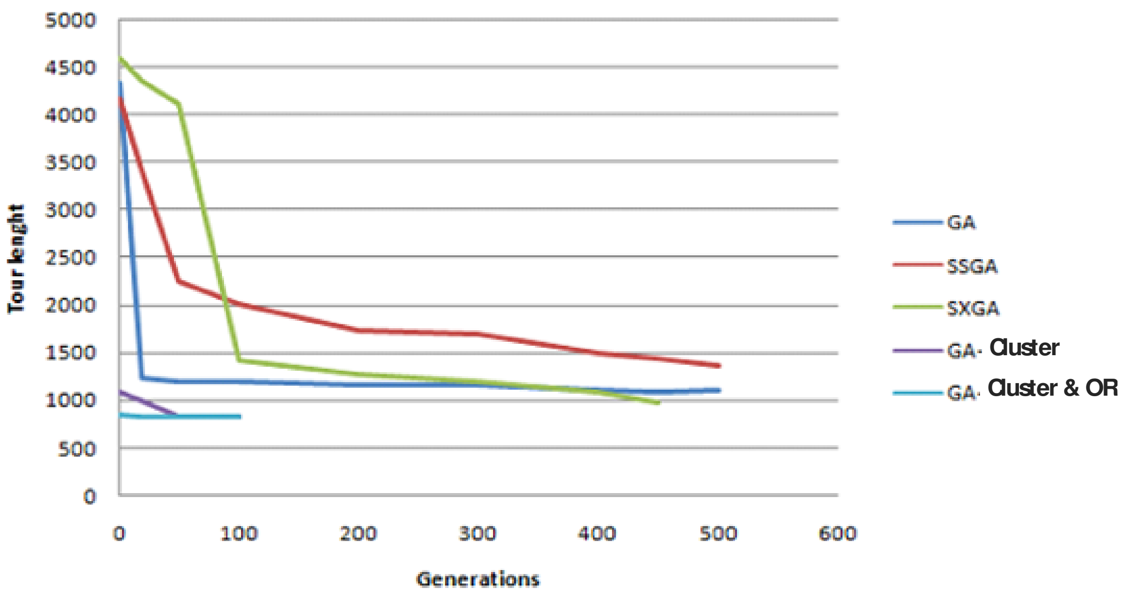

Figure 5. GA with pre-selection operators versus other algorithms.

operators were closer to the optimal solution in the VRPTW instances. As such, the study proposed new pre-selection operators to be used in algorithms or genetic algorithms to solve VRPTW (we have not found at the time of the investigation, related works about pre-selection operators for algorithms or pre-selection operators for genetic algorithms to VRPTW), and finally, some related works (perhaps weaknesses) generated the initial population randomly. However, in this research, the generation of the initial population was done in a non-random way.

Figure 5 shows the evolution of the genetic algorithms with the pre-selection operators versus General Genetic Algorithm, Steady State Genetic Algorithm and SeXual Genetic Algorithm. We can observe that the 'genetic algorithm' with clusterization pre-selection operator and the 'genetic algorithm' with 'cluster and neighborhood' pre-selection operator obtain results closer to the optimal in fewer generations than the other algorithms.

Conclusions

The main contribution of this work is the proposal of population pre-selection operators:

(a) Clusterization: Clustering the initial population by k-means algorithm (data-mining technique).
(b) Cluster and neighborhood: Clustering the initial population by k-means algorithm and neighborhood technique.

The population pre-selection operators improve (in a non-random way) the initial population for the solution of the VRPTW. From the experimental results, the solutions obtained are acceptable and closer to the optimal solution, taking into account that which is not based on heuristics. As such, the operators try to generate the initial population in a non-random way and created it with the purpose of improving the genetic and evolutionary algorithms.

ACKNOWLEDGEMENT

This work was supported by SEP-PROMEP (Mexico) through grant PROMEP/103.5/10/4453.

REFERENCES

Affenzeller M (2002). A Generic Evolutionary Computation Approach Based Upon Genetic Algorithms and Evolution Strategies, J. Syst. Sci., 28(2): 59-72.

Berger J, Salois M, Begin R (1998). A hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. In Proceedings of the 12th Biannual Conference of the Canadian Society for Computational Studies of Intelligence, Springer-Verlag, Berlin, pp. 14-127.

Berger J, Barkaoui M, Bräysy O (2001). A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. Working Paper, Defence Research Establishment Valcartier, Canada.

Blanton JL, Wainwright RL (1993). Multiple Vehicle Routing with Time and Capacity Constraints Using Genetic Algorithms. In Proceedings of the 5th international Conference on Genetic Algorithms S. Forrest, Ed. Morgan Kaufmann Publishers, San Francisco, CA, pp. 452-459.

Bräysy O (1999a). A Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows, Unpublished, Vaasa University Library, Vaasa, Finland.
Bräysy O (1999b). A new Algorithm for the Vehicle Routing Problem with Time Windows based on the Hybridization of a Genetic Algorithm and Route Construction Heuristics. Proceedings of the University of Vaasa, Research papers 227, Vaasa, Finland.
Bräysy O (2001). Local Search and Variable Neighborhood Search Algorithms for the Vehicle Routing Problem with Time Windows", Doctoral Dissertation, University of Vaasa, Finland.
Cruz-Chávez, MA, Díaz-Parra O, Juárez-Romero D, Martínez-Rangel MG (2008). Memetic Algorithm base on a Constraint Satisfaction Technique for VRPTW. L. Rutkowski et al. (Eds.). Artificial Intelligence and Soft Computing – ICAISC 2008. Springer-Verlag. Lecture Notes in Artificial Intelligence, 5097: 376-387.
Gehring H, Homberger J (1999). A Parallel Hybrid Evolutionary Metaheuristic for the Vehicle Routing Problem with Time Windows", in Proceedings of EUROGEN99 - Short Course on Evolutionary Algorithms in Engineering and Computer Science, Reports of the Department of Mathematical Information Technology, Series A. Collections, No. A 2/1999, K. Miettinen, M. Mäkelä and J. Toivanen (eds), pp. 57-64.
Gehring H, Homberger J (2001). Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows. Asia-Pacific J. Oper. Res., 18: 35-47.
Homberger J, Gehring H (1999). Two Evolutionary Meta-heuristics for the Vehicle Routing Problem with Time Windows. INFOR, 37: 297-318.
Lin S (1965). Computer Solutions to the Traveling Salesman Problem. Bell Syst. Tech. J., 44: 2245-2269.
Lin S, Kernighan BW (1973). An Effective Heuristic Algorithm for the Traveling Salesman Problem. Oper. Res., 21 (2): 498-516.
MacQueen JB (1967). Some methods for classification and analysis of multivariate observations. In Proceedings of the Fifth Symposium on Math, Statistics, and Probability, Berkeley, CA, University of California Press, 1: 281-297.
Martínez-Morales AA (2006). Algorithm Based on Tabu Search for Calculation of the Transmission Index of a Graph. J. Faraute of Sci. Technol., 1(1): 31-39.
Or I (1976). Traveling Salesman Type Combinatorial Problems y their Relations to the Logistics of Blood Banking. Ph.D. Thesis. Dpt. of Industrial Engineering y Management Sciences, Northwestern Univ., pp. 14-20.
Pacheco J, Delgado C (2000). Results of Different Experiences with Local Search Applied to Problems of Routes. Electronic Journal of Communications and Works of ASEPUMA: Rect@. ASEPUMA, 2(1): 54-81.
Perez J, Pazos R, Cruz L, Reyes G, Basave R, Fraire H (2007). Improving the Efficiency and Efficacy of the k-means Clustering Algorithm Through a New Convergence Condition. International Conference on Computational Science and Its Applications – ICCSA 2007, LNCS 4707, Springer Verlag, pp. 674-682.
Potvin JY, Bengio S (1996). The Vehicle Routing Problem with Time Windows part II: Genetic Search. J. Comput., 8(2): 165-172.
Rahoual M, Kitoun B, Mabed M-H, Bachelet V, Benameur F (2001). Multicriteria Genetic Algorithms for the Vehicle Routing Problem with Time Windows. Proceedings of the 4th Metaheuristics International Conference MIC'2001, Porto, Portugal.
Rajmohan M, Shahabudeen P (2008). Genetic Algorithm based approach for Vehicle Routing Problem with Time Windows, India Int. J. Logistics Syst. Manage., 4(3): 338-365.
Solomon MM (1987). Algorithms for Vehicle Routing and Scheduling problems with Time Window Constraints. Oper. Res., 35(2):254-265.
SPSS (2010). Inc. Headquarters, Chicago, Illinois. http://www.spss.com/es/
Thangiah S (1995). Vehicle Routing with Time Windows Using Genetic Algorithms. In Application Handbook of Genetic Algorithms: New Frontiers, Volume II, L. Chambers (ed). CRC Press, Boca Raton, pp. 253-277,
Thangiah SR (1998). Genetic Algorithms, Tabu Search and Simulated Annealing Methods for Vehicle Routing Problems with Time Windows. In Practical Handbook of Genetic Algorithms: Complex Structures. L. Chambers Ed. CRC Press, pp. 347–381.
Tan K C, Lee L H, Ou K (2001). Hybrid Genetic Algorithms in Solving Vehicle Routing Problems with Time Window Constraints. Asia-Pacific J. Oper. Res., 18: 121-130.
Toth P, Vigo D (2001). The Vehicle Routing Problem. Monographs on Discrete Mathematics and Applications. SIAM. Philadelphia.
Wagner S, Affenzeller M (2004). The HeuristicLab Optimization Environment. Technical Report. Institute of Formal Models and Verification. Johannes Kepler University Linz. Austria.
Witten IH, Frank E (2005). Data Mining Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers, Elsevier, pp. 366-367.
Zhu K Q (2000). A New Genetic Algorithm for VRPTW. Proceedings of the International Conference on Artificial Intelligence IC-AI 2000, Las Vegas, U.S.A.