A Multirecombinative Algorithm for Capacitated Vehicle Routing Problem

Silvia Villagra, Andrea Villagra, Daniel Pandolfi

LabTEm - Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral
Caleta Olivia (9011) – Santa Cruz, Argentina
{svillagra,avillagra,dpandolfi}@uaco.unpa.edu.ar

ABSTRACT

The capacitated vehicle routing problem (CVRP) deals with the assignment of a set transport routes to a fleet of vehicles. Especially the problem consists of obtaining a sequence of route for every vehicle to minimize transport cost. Evolutionary Algorithms (EAs) with multirecombinative and multiple parent approaches have been successful to solve different types of scheduling or routing problems. In this work two approaches to solve the CVRP problem are compared. An EA with a multirecombinative approach known as MCMP-SRI (Stud and Random Immigrants) and Simulated Annealing algorithm have been used. Details of the algorithms and the results of the experiments show a promissory behavior of the MCMP-SRI for this problem.

Keywords: capacitated vehicle routing problem, multirecombinative genetic algorithms.

1. INTRODUCTION

The vehicle routing problem (VRP) is a combinatorial optimization problem of great importance in the field of the transport, distribution and logistics [1]. The VRP is a generic name given to great types of problems, where a set of routes for a fleet of vehicles based in one or more depots, must be determined for a number of customers or geographically dispersed cities [2]. The problem consists of assigning to each vehicle a customers’ route, so that the cost of transport is minimized. In Figure 1 we can graphically see a solution to the problem, where 4 different routes can be observed with the start and end at a central depot.

This problem, VRP, has been classified as NP-hard problem [3] due to the great quantity of computational resources used to find an optimal solution that exponentially grows with the size of the problem. This type of problem deals with algorithms that do not have the need to explore the whole space search associated to give an approximate solution. The metaheuristics algorithms are a family of algorithms which goal is precisely to give approximate solutions to general problems of NP type, without need to explore the whole space search.

Different variants of the vehicle routing problem exist [2], which include additional restrictions and the incorporation of multiple variables.

Some more important restrictions are:
- every vehicle has limited capacity (Capacitated VRP - CVRP)
- every customer has to be attended inside a certain window of time (VRPTW)
- the vendor uses many depots to supply the customers (Multiple Depot) VRP - MDVRP
- the customers may return some goods (VRPPD)
- the customers may be served by different vehicles (SDVRP)
- some values (like the number of customers, their demands, serve time or the travel time) are arbitrary (Stochastic VRP - SVRP)
- the deliveries may be made in some days (Periodic VRP - PVRP).

Figure 1 - Graphical Representation of a solution of the VRP

Different metaheuristics have been applied for different variants of the problem, such as Tabu Search [4], Simulated Annealing [5], Ant Colony System [6], and Genetic Algorithms [7], among others.

Local Search Algorithms (LSA), are very fast methods that often start from an initial random solution to which iterative replace with another best solution that results from the application of some variation mechanism in a predetermined neighborhood space. A neighborhood structure with a problem instance defines the search space, where the solutions can be visualized as a graph, where the nodes are the solutions and the edges represent the relation between two neighbors [8].

Simulated Annealing (SA), is the first algorithm that was developed based on a strategy to escape of local minimums [9]. This algorithm makes a partial stochastic search inside the allowed region for a set of optimization variables. In minimization problems, the perturbations that cause increases in the objective value of the function are accepted by a controlled probability using the Metropolis criterion [9]. These perturbations are made in several occasions and allow the algorithm to escape from local minimums.

Generally, SA can find the global optimum of the objective function or an approximation to this one, in reasonable computation time. Diverse algorithms have been developed for
the SA, which principally defer in the mechanisms to perturb the optimization variables and in the procedure to modify the parameters of the SA during the optimization sequence [11]. The SA initiates generally with a random solution and selects the temperature of beginning. While a condition of completion is not produced, the algorithm iterates in the process of local search. Whereas it finds a best solution, this best solution will replace the best found in the previous iteration in the neighborhood search. On the contrary, if the algorithm does not find a best solution the replacement depends on a probability adapted in every iteration by the Temperature variable. This Temperature is decremented along the iterations and is used to decrease decreases the probability of acceptance solutions that are not “good”. The structure of the algorithm is presented bellow:

Algorithm: Simulated Annealing

\[
\text{s} \leftarrow \text{generateInitialSolution}(); \\
\text{k} \leftarrow 0; \\
\text{T}_0 \leftarrow \text{defineInitialTemperature}(); \\
\text{While} \ (\text{not max_evaluaciones}) \ do \\
\text{s}' \leftarrow \text{generateNeighborhood} (\text{s}); \\
\text{if} \ (f(\text{s}') < f(\text{s})) \ then \\
\text{s} \leftarrow \text{s}' \\
\text{else} \\
\text{Accept s with probability} \ p(\text{s}')/T_k \\
\text{endif} \\
\text{adaptTemperature} (\text{T}_k) \\
\text{k} \leftarrow k + 1 \\
\text{EndWhile} \\
\text{Output:} \ \text{Best solution found}
\]

The Evolutionary Algorithms (EAs) are metaheuristic that use computational models of the evolutionary process. A great variety of EAs exist, the principal ones are: Genetic Algorithms [12], Evolutionary Programming [13], Evolutionary Strategies [14], and Genetic Programming [15]. All these algorithms share a common base concept that is to simulate the evaluation of the individuals that form the population using a set of predefined operators. Commonly two types of operators are used: selection and search. The search operators more used are mutation and recombination. Current trends in EAs use approaches with multirecombination [16] and multiple parents [17]. In scheduling or routing problems these approaches have been successful strategies, particularly in scheduling problems introducing to the multirecombinative approach a new variant known as MCMP-SRI (Multiple Crossover Multiples Parents - Stud and Random Immigrants) [18].

From the old population an individual, designated as the stud, is selected by means of proportional selection. The number of \( n_2 \) parents in the mating pool is completed with randomly created individuals (random immigrants). The stud mates every other parent, the couples undergo PMX (Partial Mapped Crossover) crossover and \( 2*n_2 \) offspring are created. The best of these \( 2*n_2 \) offspring is stored in a temporary children pool. The crossover operation is repeated \( n_1 \) times, for different cut points each time, until the children pool is completed. Finally, the best offspring created from \( n_2 \) parents and \( n_1 \) crossover is inserted in the new population. The algorithm is described below:

Algorithm: EA-MCMP-SRI

\[
t=0; \ \text{[Initial Generation]} \\
\text{initialize}(\text{Stud}()); \\
\text{evaluate}(\text{Stud}()); \\
\text{while} \ (\text{not max_evaluaciones}) \ do \\
\text{mating_pool=} \ random \ generated \ Immigrants \ \text{[Select} (\text{Stud}()); \\
\text{while} \ (\text{not max_parent}) \ do \\
\text{while} \ (\text{not max_recombination}) \ do \\
\text{evaluate(mating_pool}); \\
\text{end while} \\
\text{end while} \\
\text{end while} \\
\text{Stud(t+1) = select new population from mating pool.} \\
t = t+1;
\]

The MCMP-SRI method was applied in combinatorial optimization problems and more specifically in sequentially problems, for static and dynamic cases, as for example Earliness and Tardiness [19], Weighted Tardiness [20], Average Tardiness [21] and Weighted Number of Tardy Jobs [22], were satisfactory results were obtained.

2. DESCRIPTION OF THE PROBLEM

The Capacitated Vehicle Routing Problem (CVRP) is one of the variants of the VRP that is considered to be emblematic in the field of the logistics and traffic, overcoming its treatment since 1960s [23]. This problem includes \( n \) customers, an only depot, and in addition the distances between each customer and the depot are known, as well as the distances between the customers. The vehicles that are in use have identical capacity. The CVRP can be depicted as an unidirectional graph \( G = (V, E) \) that consists in \( n+1 \) nodes (V), where the node 0 represents the depot and the nodes \( 1,..., n \) the customers, and a set of edges (E). Each edge \( (i, j) \) represents the way to go from node \( i \) towards node \( j \) (distance \( d_{ij} \)). Each customer \( i \) has associated a demand \( q_i \) and the vehicles have an identical capacity \( Q \). Each route must start and end at the depot, and each customer must be served by exactly once by one vehicle and the demand of each customer cannot be divided, that is to say the demand of a customer is transported completely by a vehicle.

The problem is to minimize the total travel distance of a routing plan such that the total demand of any route does not exceed a vehicle capacity \( Q \). The CVRP has been dealt with different metaheuristics approaches such as Genetic Algorithms [24, 25].

3. METAHEURISTICS APPLIED TO THE CVRP

In this section details and results of the methods applied in this work to solve the problem CVRP, with the objective to find a minimal total travel distance of a routing plan that satisfies the constrain of capacity vehicle are presented. First an EA with a multirecombinative approach (MCMP-SRI) is used. To codify the visits to the customers, who represent a possible solution, a permutation of integer numbers was used. Where each permutation \( p = (p_1, p_2, ..., p_n) \) is a chromosome in which \( p_i \) represents the customer \( i \) that must be visited and \( n \) represents the quantity of customers to visit. The chromosome defines the order of the sequence to follow to visit each customer. The function objective is to minimize the total distance of the general route plan in order to satisfy the demands of all
customers, keeping in mind the capacity constrain of the vehicle, Q.

The population size is set to 15 individuals. The initial population was randomly generated. The recombination operator was applied with a probability of 0.05 and mutation operator was set with a probability of 0.65. The number n1 (number of operations of recombination) and n2 (number of parents) was set to 16 and 18 respectively; these parameters were not chosen at random, but rather by an examination of values previously used with success [26]. Table 1 presents the parameters used in the EA MCMP-SRI.

Table 1-Parameters of the MCMP-SRI Algorithm

<table>
<thead>
<tr>
<th>Population Size</th>
<th>Chromosome Size</th>
<th>Number of generations</th>
<th>Mutation PMX</th>
<th>Recombination Probability</th>
<th>Mutation Probability</th>
<th>Nº of operations of recombination (n1)</th>
<th>Nº of parents (n2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>75</td>
<td>9000</td>
<td>SW</td>
<td>0.65</td>
<td>0.05</td>
<td>16</td>
<td>18</td>
</tr>
</tbody>
</table>

4. EXPERIMENTS AND RESULTS

For the experiments we used C2 instance of Christofides et al. [2], which has 75 customers, vehicle capacity is 140 and the best known solution was 835.26.

MCMP-SRI algorithm was compared with a version of SA algorithm, for which the following performance variables were analyzed, Best (best found solution), E_best (Percentage error among the best known solution (835.26.) and the median value of the best-found solution found in 30 generations), Avg (Number of thousands of evaluations made by the algorithm to achieve the best found solution). For each algorithm, 30 independent runs were executed.

From the obtained results, see Table 2, it can be observed that the MCMP-SRI algorithm finds the best solution for the CVRP, (899.51), with an error of 7.69 % compared with the best known solution (835.36). Also, achieves a best maximum solution (1.056,17) compared with the best solution obtained by the SA (1.077,86). Nevertheless the computational effort of the MCMP-SRI is higher than the SA.

The convergence of the MCMP-SRI algorithm, (30) independent runs for different maximum of generations (500...9000) were executed. The following performance values were obtained: Best (best-found solution), G_Best (generation where the best-found solution was found), Avg (average value of the best-found solutions found in 30 generations), Median (median value of the best-found solutions found in 30 generations), E_Best (percentage error as regards the best known solution (835.26.); the best-found solution / best known solution * 100), E_Avg (percentage error as regards the best known solution (835.26.); the average of the best-found solutions of the 30 runs / best known solution * 100) and E_Median (percentage error as regards the best known solution (835.26.) and the median value of the best-found solution in the 30 runs / best known solution* 100).

Table 2 – Comparison of algorithms SA and MCMP-SRI

<table>
<thead>
<tr>
<th>Simulated Annealing</th>
<th>MCMP-SRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best</td>
<td>E_Best</td>
</tr>
<tr>
<td>Evals</td>
<td>Best</td>
</tr>
<tr>
<td>Avg</td>
<td>Evals</td>
</tr>
</tbody>
</table>

Table 3 – Results obtained by MCMP-SRI algorithm

<table>
<thead>
<tr>
<th>Nro Gen</th>
<th>Best</th>
<th>G_Best</th>
<th>Avg</th>
<th>Median</th>
<th>E_Best</th>
<th>E_Avg</th>
<th>E_Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>899.38</td>
<td>617</td>
<td>223</td>
<td>899.38</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>2</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>3</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>4</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>5</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>6</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>7</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>8</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>9</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
<tr>
<td>10</td>
<td>899.20</td>
<td>982</td>
<td>223</td>
<td>899.20</td>
<td>1.0776</td>
<td>18.02</td>
<td>16.15</td>
</tr>
</tbody>
</table>

Table 3 shows the comparative results obtained by the MCMP-SRI algorithm, each row shows the solutions obtained for a set of 30 independent runs according to the maximum number of generations Nro Gen (maximum number of generation). The best-found solution (899.50) is found with the maximum number of generations 9000, specifically in the 8286 generation. As the maximum number of generations is increased the algorithm converges adequately, this convergence is showed in Figure 2, where the dotted line is the median value and the full line is the average for the instances showed in Table 2.
5. CONCLUSION

Evolutionary Algorithms are robust search algorithms in the sense that provide "good" solutions to a wide class of the problems that otherwise would be computationally untreatable. To improve EAs behavior, multirecombiantive approaches allow multiple exchanges of genetic material between multiple parents and with it to improve the convergence speed.

To improve the search process, by means of a best balance between the exploration and the exploitation, the concept of stud and random immigrants was inserted in MCMP-SRI. The presence of the stud assures the keeping back of good features of previous solutions and the random immigrants, as a constant source of genetic diversity, avoids premature convergence. Capacitated Routing Vehicle Problem (CVRP) is one of the variants of the VRP that is considered to be emblematic in the field of the distribution, logistics and traffic. The promissory results obtained by MCMP-SRI improve SA for the principal analyzed variables (Best, Ebest) for CVRP. With regard to the speed of convergence, SA needs fewer evaluations respect of the MCMP-SRI, though the last shows its capacity achieving best solutions while increases the search (iterations).

Future works will include the hybridization of the algorithm with other methods of local search, the application of other methods of recombination and the study of the behavior for other variants of VRP.

6. ACKNOWLEDGMENTS

We acknowledge the cooperation of the project group of LabTEm for providing new ideas and constructive criticisms. Also to the Universidad Nacional de la Patagonia Austral from which we receive continuous support.

7. REFERENCES


