

## A MULTI OBJECTIVE GENETIC ALGORITHM FOR SOLVING VEHICLE ROUTING PROBLEM

Padmabati Chand<sup>1</sup>, Bhabani Sankar Prasad Mishra<sup>2</sup> & Satchidananda Dehuri<sup>3</sup>

---

VRP is a NP-Complete and a multi-objective problem. The problem involves optimizing a fleet of vehicles that are to serve a number of customers from a central depot. Each vehicle has limited capacity and each customer has a certain demand. Genetic Algorithms maintain a population of solutions by means of a crossover and mutation operators. For crossover and mutation best cost route crossover techniques and swap mutation procedure is used respectively. In this paper, we focus on two objectives of VRP i.e. number of vehicles and total cost (distance). The proposed MOGA finds optimum solutions effectively.

Keywords: Vehicle Routing Problem (VRP), Genetic Algorithm, Multi-objective Optimization

---

### 1. INTRODUCTION

The Vehicle Routing is a complex combinatorial optimization problem which was first introduced by Dantzig and Ramser in 1959[15]. Fisher [16] describes the problem as the efficient use of a fleet of vehicles, which must make a number of stops to pick up and deliver passengers or products. The term customer is used to denote the stops to pick up and deliver the product. Every customer has to be assigned to exactly one vehicle in a specific order, which is done with respect to the capacity in order to minimize the total cost. The problem can be considered as a combination of the two well-know optimization problems i.e. the Bin Packing Problem (BPP) and the Travelling Salesman Problem (TSP). Relating this to the VRP, customers can be assigned to vehicles by solving BPP and the order in which they are visited can be found by solving TSP. The rest of the paper is organized as follows section 2 gives a background study of the VRP, section 3 gives the multi objective genetic search of VRP, section 4 describes on experimental results.

### 2. BACKGROUND

The VRP is defined on a set  $V = \{v_0, v_1, \dots, v_N\}$  of vertices, where vertex  $v_0$  is a depot which is based on  $m$  identical vehicles of capacity  $C$ , while the remaining  $N$  vertices

represent customers, also called requests or demands. Each customer has a demand  $d_i$ . The VRP consists of designing a set of  $m$  vehicle routes of the least total cost, each starting and ending at the depot, such that each customer is visited exactly once by a vehicle, the total demand of any route does not exceed. Each vertex  $v_i$  has a location in the plane, where the travel cost is given by the Euclidean distance  $d(v_i, v_j)$  for each edge  $(v_i, v_j) \in V \times V$ . Then, the main objective of the problem is to minimize the total number of vehicles used to service the customers and minimize the distance traveled by the vehicles. There are two constraints associated with the vehicle routing problem they are vehicle capacity constraint and each customer should be serviced exactly once.

### 3. MULTI-OBJECTIVE GENETIC SEARCH FOR THE VRP

Multi-objective optimization, also known as multi-criteria optimization, is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. If a multi-objective problem is well formed, there should not be a single solution that simultaneously minimizes each objective to its fullest. In each case we are looking for a solution for which each objective has been optimized to the extent that, if we try to optimize it any further, then the other objective(s) will suffer as a result. Finding such a solution, and quantifying how much better this solution is compared to other such solutions is the goal when setting up and solving a multiobjective optimization problem.

In GA, each chromosome in the population pool is transformed into a cluster of routes. The chromosomes are then subjected to an iterative evolutionary process until a minimum possible number of route clusters is attained or the termination condition is met. The evolutionary part is

---

<sup>1</sup>Department of Computer Science, Trident Academy of Technology, Orissa, INDIA

<sup>2</sup>School of Computer Engineering, KIIT University, Orissa, INDIA

<sup>3</sup>Department of ICT, F.M.University, Balasore, Orissa, INDIA  
Email: padma024@gmail.com, mishra.bsp@gmail.com, satchi.lapa@gmail.com

carried out as in ordinary GA using selection, crossover, and mutation operations on chromosomes. Tournament selection is used to perform fitness-based selection of individuals for reproduction. A crossover operator that ensures solutions generated through genetic evolution are all feasible is also proposed. Hence, both checking of the constraints and repair mechanism can be avoided, thus resulting in increased efficiency.

### 3.1. Chromosome Representation and Initial Population Creation

In our approach, a chromosome representing route of length  $N$ , where  $N$  is the number of customers in a particular problem instance. A gene in a given chromosome indicates the original node number assigned to a customer, while the sequence of genes in the chromosome indicates the order of visitation of customers, \* indicates a node representing a group of clustered customers that have already been committed to a given vehicle. Thus, the chromosome consists of integers, where new customers are directly represented on a chromosome with their corresponding positive index number and each committed customer is indirectly represented within one of the groups (shown by a \* mark) representing a given deployed vehicle.

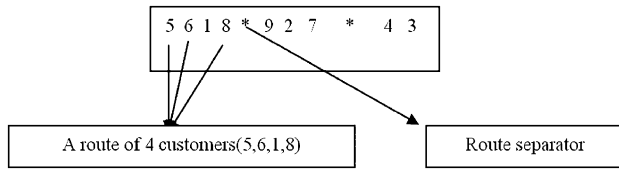


Fig. 1: Chromosome Representation

### 3.2. Fitness Evaluation

The fitness of a chromosome is determined after each chromosome has been transformed directly into a route network topology.

$$f(x) = \sum_{(i,j) \in R, r_i \in R} \alpha d_{ij} + \beta(|V| - V_{\min}) \quad (1)$$

where  $\alpha$  and  $\beta$  are weight parameters associated with the number of vehicles and the total distance traveled by vehicles respectively. The weight values of the parameters used in this function were established empirically and set at  $\alpha = 0.01$  and  $\beta = 100$ . In the above expression  $d_{ij}$  is the distance from node  $i$  to node  $j$ ,  $r_i$  is the sub route of the route  $R$ ,  $V$  is the total no. of vehicle of the route.  $V_{\min}$  is the minimum no. of vehicle per route. It is calculated as

$$V_{\min} = \frac{\sum_{i=1}^n \text{demand}_i}{\text{Capacity of vehicle}} \text{ where, } i \text{ is the customer from } 1 \text{ to } n \quad (2)$$

### 3.2. Cross Over

Initial experiments using standard crossover operators such as Partially-Mapped-Crossover (PMX) and uniform order crossover (UOC) yielded non-competitive solutions. Hence, we utilized a problem-specific crossover operator that generates feasible route schedules. An example of the procedure utilized by the proposed crossover (Best-Cost Route Crossover, BCRC) is given in Fig. 2. According to Fig. 2, two parents A and B are selected from the population. A route from each parent chromosome is randomly selected and the customer orders present in each route are removed from the other parent. Since \* marks represent existing vehicles, their customers are left untouched. This means only integers which represent uncommitted customers are reinserted into the current chromosome. This is shown in the Fig. 2. Then the customers that have been removed are reinserted at the location which minimizes the overall cost of the entire tour. This requires computing the cost of inserting each of the remaining customers at each location in the chromosome without constraint violation. If no insertion location for a particular customer is found, a new route is created. In the figure 2 A and B are two parents. In step 1 for parent A there is three routes ( $r_1:2\ 4\ 3\ 6\ r_2:1\ 9\ r_3:5\ 7\ 8$ ) similarly in B ( $r_1:8\ 7\ 9\ r_2:3\ 1\ 6\ r_3:2\ 5\ 4$ ). In step 2 select randomly a route from step 1 of parent A ( $r_2:1\ 9$ ) and in B ( $r_3:2\ 5\ 4$ ). In step 3 what route we selected from step 2 of parent A (1 9) that we remove from parent B of the given routes in step 1. Similarly in what route we selected from step 2 of parent B (2 5 4) that we remove from parent A of the given routes in step 1. In step 4 what route we are deleted from parent A(2 5 4) and B(1 9) that we have to again insert. From the route (2 5 4) we have to randomly select a route suppose 5 that we have to insert again to the route (what route we got in step 3) by satisfying all the constraints i.e vehicle capacity and also after inserting there should by optimum solution means distance should be minimum and also minimum no. of vehicles.

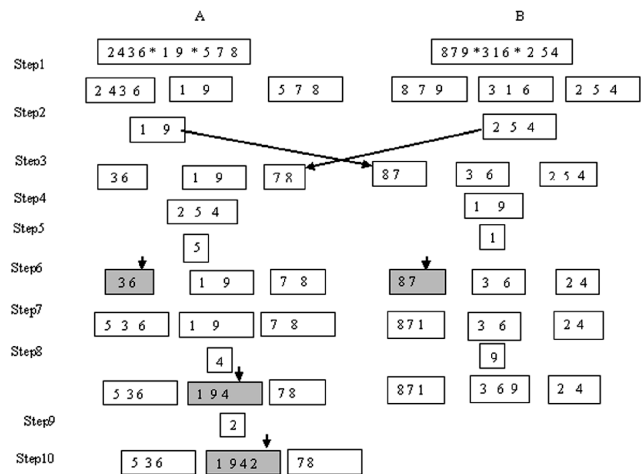


Fig. 2: Best Cost Route Cross Over

The suitable location is grey in figure and also there is an arrow mark in step 6. After inserting the new route is (r<sub>1</sub>:5 3 6 r<sub>2</sub>:1 9 r<sub>3</sub>:7 8). Similarly insert 2 and 4. If any removed node is not satisfying constraints then make a new route. Similarly in parent B from 1 and 9 we have randomly select a customer and that we have to insert by satisfying all the constraints. If not satisfying make a new route. So finally the optimum solution we got from parent A (r<sub>1</sub>:5 3 6 r<sub>2</sub>:1 9 4 2 r<sub>3</sub>:7 8) and B (r<sub>1</sub>:8 7 1 r<sub>2</sub>:3 6 r<sub>3</sub>:2 4 r<sub>4</sub>:9).

Mutation is done by doing swap mutation. Select any two customer from any two route randomly and exchange their position if satisfying all the constraints. After swap, insertion is done in which we select randomly a customer from a route and try to insert rest of any one route if it satisfies all the constraints.

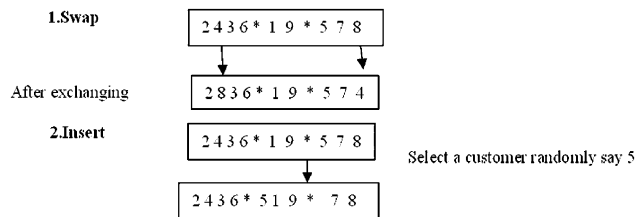


Fig. 3: Mutation

4. EXPERIMENTAL RESULTS

This section describes computational experiments carried out to investigate the performance of the proposed GA. By our given fitness function, it minimizes both number of vehicles and travel costs without bias. The algorithm was coded in mat lab and run on an Intel Pentium IV 1.6 MHz PC with 512 MB memory and it gives optimum result. We have applied our MOGA to 18 of the instances from Solomon’s benchmark set. The summary of the results are shown in Table 2.

Table 1  
GA Parameters

Parameter type	Values
Number of runs	20
Population size	100
Generation span	700
Crossover rate	0.90
Mutation rate	0.10
Selection type Tournament	2
Crossover type BCRC	Probability rate(0.90)
Mutation type swap	Probability rate(0.15)

Table 2  
Test Result

Instance name	No. routes	Traveled distance	Result
C101_25	3	191.81	101.9181
C201_25	2	237.15	2.3715
R101_25	8	623.52	606.2352
R201_25	2	597.22	5.9722
RC101_25	4	473.50	204.735
RC201_25	2	526.86	5.2686
C101_50	6	429.63	404.2963
C201_50	2	495.45	4.9545
R101_50	11	1223.11	912.2311
R201_50	2	1069.32	10.6932
RC101_50	8	951.79	609.5179
RC201_50	3	1135.48	111.3548
C101_100	12	1467.47	1014.6747
C201_100	3	773.27	107.7327
R101_100	19	1997.87	1719.9787
R201_100	5	1700.59	317.0059
RC101_100	15	1986.58	1319.8658
RC201_100	5	2027.62	320.2762

5. CONCLUSION

This paper presented a GA based approach for the static VRP. The approach was tested using problem instances reported in the literature, derived from publicly available Solomon’s benchmark data for VRP. The experimental results showed that the GA approach was able to find high quality solutions. Future goal is to generate larger problem instances, and further evaluate the GA’s performance on these problems by considering other objectives like time window and speed of the vehicle.

REFERENCES

- [1] Franklin T. Hanshar, Beatrice M. “Ombuki-Berman Dynamic Vehicle Routing using Genetic Algorithms”, Appl Intell, 27, pp.89–99, 2007.
- [2] Beatrice Ombuki, Brian J. Ross and Franklin Hanshar Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows.
- [3] Jaber Jemai-Khaled Mellouli, “A Neural-Tabu Search Heuristic for the Real Time Vehicle Routing Problem J Math Model Algor”, pp.161–176, 2008.
- [4] K. C. Tan Y. H. Chew, “A Hybrid Multiobjective Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows Computational Optimization and Applications”, 34, Pp.115–151, 2006.
- [5] Olli Braˆ Ysy Evolutionary Algorithms for the Vehicle Routing Problem with TimeWindows Journal of Heuristics, 10 pp. 587–611, 2004.

- [6] S. Salhi · R. J. Petch A GA Based Heuristic for the Vehicle Routing Problem with Multiple Trips *J Math Model Algor*, 6, pp.591–613, 2007.
- [7] Guido Perboli · Ferdinando Pezzella·Roberto Tadei EVE-OPT: a Hybrid Algorithm for the Capacitated Vehicle Routing Problem *Math Meth Oper Res*, 68, pp.361–382, 2008.
- [8] Michael Polacek·Karl F. Doerner·Richard F. Hartl·Vittorio Maniezzo, “A Variable Neighborhood Search for the Capacitated Arc Routing Problem with Intermediate Facilities Heuristics”, 14, pp. 405–423, 2008.
- [9] Jean-Charles Créput·Abderrafiaâ Koukam The Memetic Self-organizing Map Approach to the Vehicle Routing Problem *Soft Comput*, 12, pp.1125–1141, 2008.
- [10] Nicolas Jozefowicz·Frédéric Semet·El-Ghazali Talbi Target Aiming Pareto Search and its Application to the Vehicle Routing Problem with Route Balancing *J Heuristics*, 13, pp. 455–469, 2008.
- [11] Minea Skok Davor Skrlec Slavko Krajcar, The Non-Fixed Destination Multiple Depot Capacitated Vehicle Routing Problem and Genetic Algorithms University of Zagreb Faculty of Electrical Engineering and Computing Department of Power Systems Unska 3, 10000 Zagreb, Croatia.
- [12] Keiichi Uchimura and Hideki Sakaguchi Vehicle Routing Problem using Genetic Algorithmsbased on Adjacency Relations Keiichi Uchimura and Hideki Sakaguch Dept. of Electrical Engineering and Computer Science, Kumamoto University Kurokami 2-39-1, Kumamoto 860, Japan.
- [13] Abel Garcia Najera A First Multi-objective Genetic Algorithm Approach to Solving the Vehicle Routing Problem with Time Windows School of Computer Science University of Birmingham February 11, 2008.
- [14] G. Laporte, M. Gendreau, J-Y. Potvin, and F. Semet. “Classical and Modern Heuristics for the Vehicle Routing Problem”. *International Transactions in Operational Research*, 7, pp. 285\_300, 2000.
- [15] M. Fisher. Vehicle routing. *Handbooks of Operations Research and Managemen Science*, Chapter 1, pp.1-31, 1995.
- [16] E. Falkenauer. “A Hybrid Grouping Genetic Algorithm for Bin Packing”, *Journal of Heuristics*, 2, pp. 5-30, 1996.