Multi Objective Genetic Approach for Solving Vehicle Routing Problem

Padmabati Chand and J. R. Mohanty

Abstract—**Vehicle Routing Problem (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 Algorithm (GA) maintains 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 Multi Objective Genetic Algorithm (MOGA) finds optimum solutions effectively.**

Index Terms—**Vehicle routing problem, genetic algorithm, multi-objective optimization, pareto ranking procedure, best-cost route crossover (BCRC).**

I. Introduction

The Vehicle Routing problem (VRP) is a complex combinatorial optimization problem which was first introduced by Dantzig and Ramser in 1959. Fisher [1] 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 II gives a back ground study of the VRP, section III gives the multi objective genetic search of VRP, section IV describes on experimental results.

II. Background

The VRP is defined on a set $V = \{v_0, v_1 . . . v_N\}$ 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). 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 [2]. There are two constraints associated with the vehicle routing problem: vehicle capacity constraint and each customer should be serviced exactly once.

Manuscript received February 5, 2013; revised April 5, 2013.
The authors are with School of Computer Application, KIIT University, Bhubaneswar, India (e-mail: padma024@gmail.com, jnyana1@gmail.com).

III. 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 (compared to other such solutions) is the goal when setting up and solving a multi-objective optimization problem [3].

In genetic algorithm (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 carried out as in the GA using selection, crossover, and mutation operations on chromosomes as per the following algorithm [4]. The time complexity of the following algorithm is O (MN^3).

Genetic Algorithm

Start

Step 1: Read problem instance data
Step 2: Set GA parameters
Step 3: Generate randomly an initial population
Step 4: For Generation =1 to MaximumGeneration
Step 5: Evaluate fitness of the individuals of population
Step 6: Apply pareto rank methods and select new population
Step 7: Apply GA operators (crossover (BCRC) and mutation (Swap))

End

Tournament selection is used to perform fitness-based selection of individuals for reproduction. A crossover operator that ensures solutions generated through genetic evolution is proposed which is feasible. Hence, both checking of the constraints and repair mechanism can be avoided, thus resulting in increased efficiency.

A. 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. In Fig. 1 *N* is 4. 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 [5]. Thus, the chromosome consists of integers, where new customers are directly represented on a chromosome with their corresponding index number and each committed customer is indirectly represented within one of the groups (shown by a * mark) representing a given deployed vehicle [6], [7].

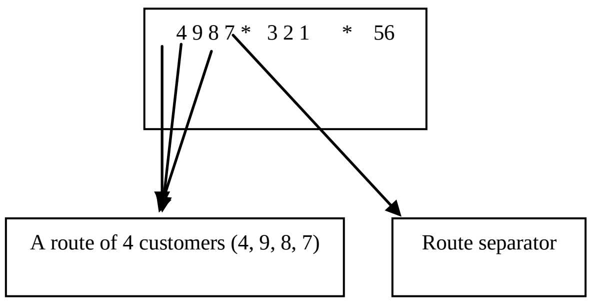

Fig. 1. Chromosome representation.

B. Fitness Evaluation

In this approach we used dominance-information of the individuals of the population (pareto ranking procedure) by calculating for each individual, the number of alternatives from which this individual is dominated [8], [9]. Individuals that are not being dominated by others should receive a higher fitness value than individuals that are being dominated. In Pareto ranking scheme fitness of chromosomes represented by pareto ranks. Solutions assigned rank 1 are non-dominated and those of rank $i+1$ are dominated by all solutions of rank1 through *i*. First the set of non-dominated vectors in the population are assigned rank 1. These solutions are removed, and the remaining non–dominated solutions are assigned rank 2. This is repeated until the entire population is ranked. In pareto ranks we will not get a single solution, we get a set of solutions. Every generation in a run have a rank 1 set [2]. In order to determine whether an actual solution has been found or not we have applied diversity method.

Diversity Method: In pareto approximation diversity is important because all the solutions are different. Density information gives us good metric to increase this diversity. This means probability to select solution decreases the greater density of solutions in its neighborhood. So for density information we applied nearest-neighbor Fig. 2 method in which, distance between a given point and its' *i*-th nearest neighbor is to estimate density in its neighborhood.

C. 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 [10]. An example of the procedure utilized by the proposed crossover (Best-Cost Route Crossover, BCRC) is given in Fig. 3. According to Fig. 3, 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 [2]. This means only integers which represent uncommitted customers are reinserted into the current chromosome [11], [12].

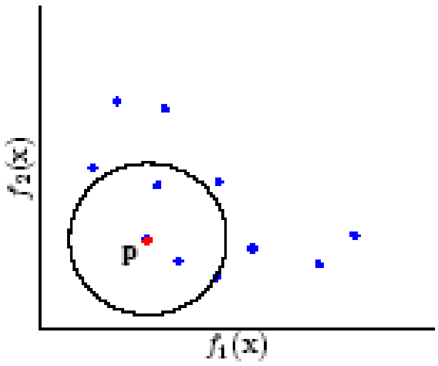

Fig. 2. Nearest neighbour.

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 above Fig. 3 A and B are two parents. In step 1 for parent A there are 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 we selected randomly a route from step 1 of parent A (r_2:1, 9) and in B (r_3:2 5 4). In step 3 the selected route from step 2 of parent A i.e. (1 9), removed from parent B of the given routes in step 1. Similarly the selected route e from step 2 of parent B i.e. (2 5 4), removed from parent A of the given routes in step 1. In step 4 the deleted routes from parent A (2 5 4) and B (1 9) again inserted. From the route (2 5 4) we have randomly selected a route (suppose 5) again inserted to the route of 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. The suitable location is darkened in figure and there is also an arrow mark in step 6. After inserting the new route is created (r_1:5 3 6 r_2:1 9 r_3:7 8). Similarly we inserted 4 and 2. If any removed node is not satisfying constraints then we make a new route. Similarly in parent B from 1 and 9 we have randomly selected a customer and that we have inserted by satisfying all the constraints. If not satisfied make a new route. So finally the optimum solution we got from parent A (r_1:5 3 6 r_2:1 9 4 2 r_3:7 8) and B (r_1:8 7 1 r_2:3 6 9 r_3:2 5 4) [2].

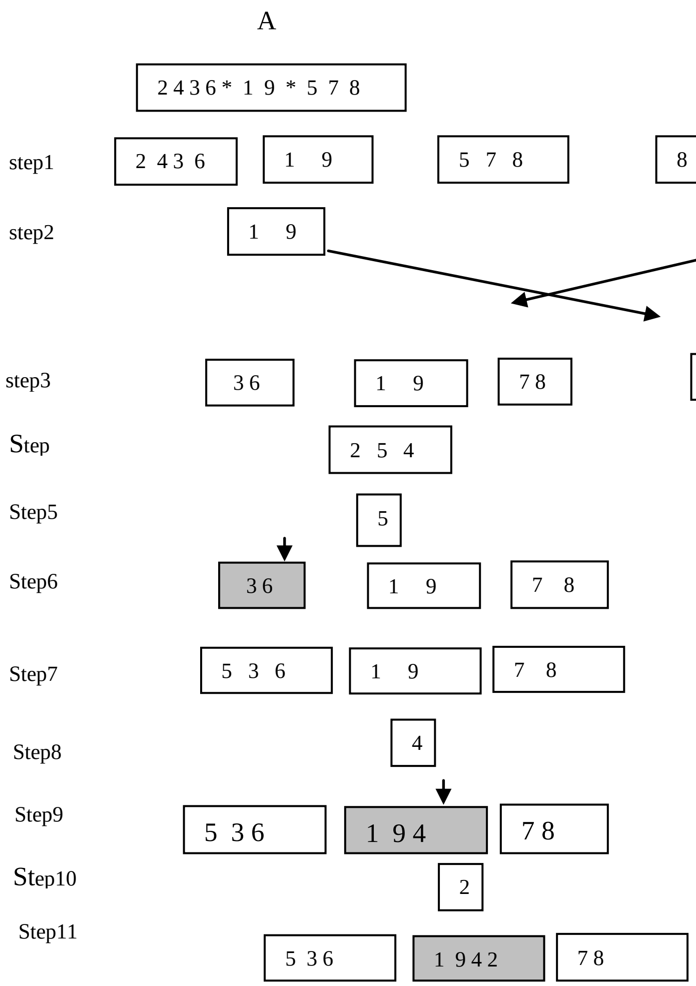

Fig. 3. Bes-cost route cross over.

D. Mutation

Mutation is done by doing swap mutation in Fig. 4. We selected any two customers from any two routes randomly and exchange their position after satisfying all the constraints.

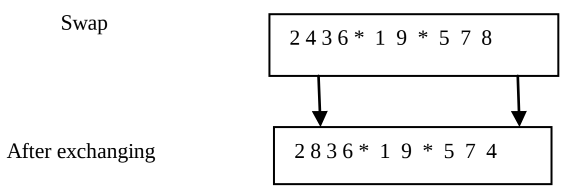

Fig. 4. Mutation.

IV. 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 MOGA to some instances from Solomon's benchmark set, it gives better result. The following figures illustrate the progress of the genetic algorithm. Table I represents genetic algorithm (GA) parameters, Table II represents obtained results. Fig. 5 represents initial population, Fig. 6 represents Pareto optimal front. Fig. 7 result for the instance C101_50 (number of vehicle=5, distance=378.27). Figure 8 result for the instance C201_25 (number of vehicle=3, distance=220.91).

TABLE I: GA Parameters

Parameter type	Values
Number of runs	20
Crossover rate	90
Mutation rate	0.10
Selection type Tournament	2
Crossover type	BCRC
Mutation type	Swap

TABLE II: Obtained Results

Solomon data set	Net Value	Distance
C101_25	3	191.98
C201_25	3	220.91
R101_25	8	768.34
R201_25	2	470.36
RC101_25	4	473.89
C101_50	5	378.27
C201_50	2	520.90

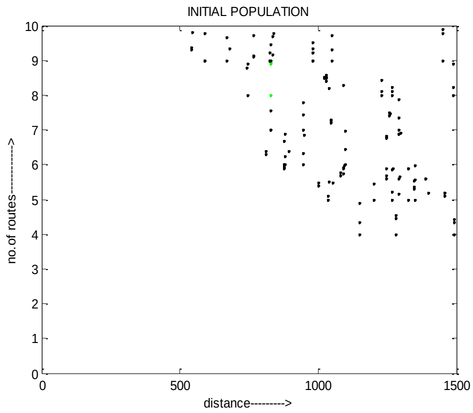

Fig. 5. Initial population.

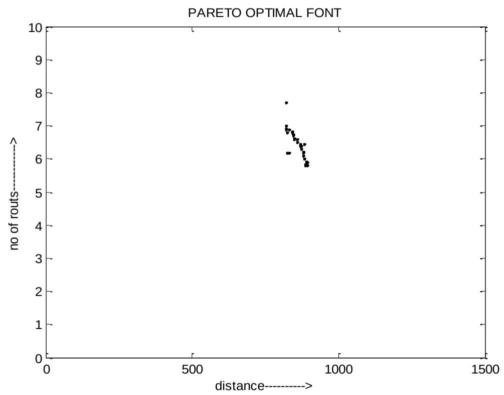

Fig. 6. Pareto optimal front.

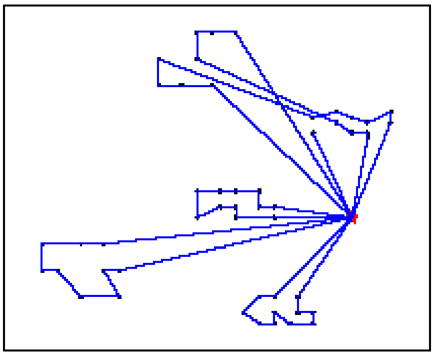

Fig. 7. Result for instance C101_50.

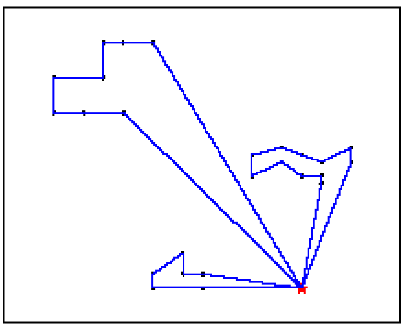

Fig. 8. Result for instance C201_25.

V. CONCLUSION

In this paper we 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] M. Fisher, *Vehicle routing. Handbooks of Operations Research and Management Science*, Elsevier, 1995, vol. 8, pp. 1-33.

[2] B. Ombuki, B. J. Ross, and F. Hanshar "Multi-Objective Genetic Algorithms for Vehicle Routing Problem with Time Windows," *Springer Science Trans. on Applied Intelligence*, vol. 24, pp. 17-30, February 2006.

[3] K. C. Tan and Y. H. Chew, "A Hybrid Multiobjective Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows," *Springer Science Trans. on Computational Optimization and Applications*, vol. 34, pp. 115–151, May 2006.

[4] S. Salhi and R. J. Petch, "A GA Based Heuristic for the Vehicle Routing Problem with Multiple Trips," *Springer Science Trans. on Journal of Mathematical Modelling and Algorithms*, vol. 6, pp. 591–613, July 2007.

[5] G. Perboli, F. Pezzella, and R. Tadei, "EVE-OPT: a hybrid algorithm for the capacitated vehicle routing problem," *Springer Science Trans. on Mathematical Methods of Operations Research*, vol. 68, issue 2, pp. 361–382, July 2008.

[6] M. Polacek, K. F. Doerner, R. F. Hartl, and V. Maniezzo, *A variable Neighborhood Search for the Capacitated Arc Routing Problem, with Intermediate Facilities Heuristics*, 2008, vol. 14, pp. 405–423.

[7] M. Laporte, J. Y. Gendreau, and F. S. Potvin, "Classical and modern heuristics for the vehicle routing problem," *International Trans. in Operational Research*, vol. 7, issue 4-5, pp. 285–300, September 2006.

[8] P. Reed, B. Minsker, and D. E. Goldberg, "Designing a competent simple genetic algorithm for search and optimization," *Water Resour. Res*, Wiley, 2000, vol. 36, issue 12, pp. 3757–3761.

[9] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithm," in G. J. E. Rawlins, (ed.), *Foundations of genetic algorithms*, CA: Morgan Kaufmann, 1991, pp. 69-93.

[10] T. Potter and T. Bossomaier, "Solving vehicle routing problems with genetic algorithms," in *Proc. Evolutionary Computation*, IEEE Press, December 1995, vol. 2, pp. 788-793.

[11] R. L. Haup and H. S. Haupt, *Practical Genetic Algorithms* (2nd Ed.), A John Wiley & Sons Inc., 2004.

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithm*, John Wiley & Sons Ltd., 2001.

Padmabati Chand is now a Ph.D student of School of Computer Application, KIIT University, whose areas of expertise are genetic algorithm and particle swarm optimization.

J. R. Mohanty is an associate professor of School of Computer Application, KIIT University, whose areas of expertise are genetic algorithm, particle swarm optimization, and data mining.