IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

A Survey on Genetic Algorithm for Vehicle Routing Problem

Ms. S. Geetha Msc.[1], Mrs. N. Vijayalakshmi MCA., M. Phil., [2]

M.Phil Scholar, Department of Computer Science, Kovai kalaimagal College of Arts and Science, Coimbatore, India[1]

Assistant Professor, Dept of Computer Science, Kovai kalaimagal College of Arts and Science, Coimbatore, India[2]

Abstract: A Vehicle routing problem (VRP) attracts much attention due to the increased interest in various geographical solutions and technologies as well as their usage in logistics and transportation. Many researches on different heuristic approaches can be found for the solution of the vehicle routing problem, where specific situations and constraints are analyzed. The common genetic algorithm approaches involve additional repair and improvement methods that are designed for a specific constraint to keep the generated solutions in the feasible search space. The usage of the repair and improvement methods designed for specific constraints or genetic operators specially designed for a specific problem can produce an inadequate result when they are applied to different problems. In this research we investigate genetic algorithm approaches for solving vehicle routing problem with different constraints. Due to stochastic characteristics, genetic algorithms generate solutions in the whole search space including the infeasible space. We propose a genetic algorithm based on a random insertion heuristics for the vehicle routing problem with constraints. The random insertion heuristic is used to construct initial solutions and to reconstruct the existing ones. The process of random insertion preserves stochastic characteristics of the genetic algorithm and preserves feasibility of generated individuals.

Keywords: Vehicle routing problem (VRP), genetic algorithm, VRPTW, VRPPD, MDVRP.

I. INTRODUCTION

The vehicle routing problem (VRP) is a well known combinatorial problem that attracts researchers to investigate it by applying the existing and newly created optimization algorithms. The VRP is defined as a routing problem with a single depot, a set of customers, multiple vehicles and the objective to minimize the total cost while servicing every customer. A set of constraints can be defined as follows: VRP with capacity limitations (CVRP), where vehicles are limited by the carrying capacity; VRP with time windows (VRPTW), where a customer can be serviced within a defined time frame or time frames; VRP with multiple depots (MDVRP), where goods can be delivered to a customer from a set of depots; VRP with pick-up and delivery (VRPPD), where rules are defined to visit pick-up places and later to deliver goods to the drop-off location. The defined genetic algorithm crossover and mutation operators incorporate random insertion heuristics, analyze individuals and select which parts should be preserved and which should be reconstructed. The second population increases the probability that the solution, obtained in the mutation process, will survive in the first population, thus increasing diversity in the population and the probability to find the global optimum. The important part in reduction of transportation costs is a better organization of routes by solving a vehicle routing problem. For example, a better organization of fleet routes in various distribution areas, delivery of post, supply delivery to markets etc.

I.I. Objectives of the research

The objective of the thesis is to design a new genetic algorithm for vehicle routing problem that handles constraints in genetic operators and that can be efficiently applied for solving rich vehicle routing problem. In order to achieve the objective, the following tasks are stated:

- To study existing genetic algorithms for solving vehicle routing problems.
- To analyze approaches in genetic algorithms for dealing with constraints in vehicle routing problems and investigate search intensification approaches in genetic algorithm operators.
- To analyze the existing formulations of rich vehicle routing problem and detail them. To investigate Dijkstra's shortest path algorithm speed up techniques in order to efficiently apply the proposed genetic algorithm to the real vehicle routing problems taking into account the road network.
- To evaluate the proposed genetic algorithm by applying it on public available benchmark instances and compares it with other known genetic algorithms.

The genetic algorithm is proposed that involves insertion heuristic, feasibility preservation, a search of common parts in the crossover operators and the second population used in the mutation operator. Solutions obtained in the second population remain competitive in the main population: they have a higher probability to be selected for reproduction and involve the diversification in the population. The proposed algorithm produces solution in short time and solutions are better or equal to results obtained by other genetic algorithms. The advantage of a new developed genetic algorithm is that it can be applied to rich vehicle routing problem and the formulation. To proposes a new genetic algorithm for rich vehicle routing problem, where genetic operators handle constraints in solutions in each iteration.

II .VEHICLE ROUTING PROBLEM

Vehicle routing problem (VRP) is a general name given for a class of problems, in which a set of vehicles service a set of customers. This statement was first defined by Dantzig and Ramser (1959). VRP is a generalization of a traveling salesman problem (TSP), where only one traveler is taken into account. The TSP is defined as a set of cities, where a single traveler needs to visit all of them and return to the starting city.

The objective of the TSP is to find the shortest route. The vehicle routing problem typically is described as a graph G = (N, E) and a set of homogeneous vehicles V = {v1, …, vt}, where t is the number of vehicles. The graph G consists of the nodes N = {n0, n1, ..., nk}, where n0 is a depot and N\{n0} are k customers that need to be serviced, and edges E = {eij}, where i j, 0 i k, 0 j k, eij = (ni, nj). Each vehicle that services customers starts the travel from the depot and finishes it in the depot as well. The objective of the typical VRP is to find the solution, at first, minimizing the total vehicle number required, and secondly, minimizing the length of the total traveled path. Usually the VRP is treated as symmetric, where dij = dji. In the real world problem, the cost matrix is asymmetric and needs to be calculated from geographic data by using the shortest path algorithms. Real situations can give another type of constraints where goods need not only to be brought from a depot to a customer, but also to be picked up from a number of customers and brought back to depot or to any other customer.

III. HEURISTICS FOR VRP AND GENETIC ALGORITHM

Branch and bound (B&B). Branch and bound is an optimization technique which search of all possible solutions while discarding (pruning) a large number of non-promising solutions by estimating upper and lower bounds of the quantity to be optimized. Constructive heuristics are methods that start from the empty solution and iteratively extend it until the full solution is constructed. Construction heuristics that are typically used for solving VRP are as follows Savings algorithm, Route-first cluster-second, Cluster-first route-second, Insertion heuristics. Savings algorithm search and merge two routes by maximizing the saving cost, where cost typically is a distance. Merge is possible, if merged route remains feasible. Route-first cluster, the construction starts from the initial route that visits all the nodes. The route is then split into several routes starting from the depot. Cluster-first route-second, the nodes are firstly added to clusters and then routes are optimized in each cluster. Algorithm inserts new nodes to route. Nodes are inserted at the end of the route, if insertion is feasible and if no insertion found, new route is started. Afterwards each route is optimized. Insertion heuristics: The main principle of insertion heuristics is to start from a single node that is usually called a seed node and that forms the initial route from the depot. Other nodes are inserted one by one evaluating certain functions to select a node and the place in the route for insertion. Local-improvement heuristic local search is an iterative process that takes the initial solution x and, in each iteration, searches for the improved solution x' in the neighborhood of x. The search stops at solution x" when the improved solution is not found in neighborhood Nh(x"). Such a search approach finds a local optimum and is called Hill Climbing (HC) is a popular method used in other algorithms for improvement of solutions. Metaheuristic is another approach for solving a complex problem that may be too difficult or time-consuming by traditional techniques. Genetic algorithms are based on ideas of evolution theory .The main principle here is that only the fittest entities survive (Reid, 2000; Jung and Moon, 2002; Lukasiewycz et al., 2008a). A genetic algorithm can be divided into several sub-parts that are used in this algorithm. representation, fitness function evaluation, initialization, selection, recombination (crossover and mutation), and termination. The whole process of genetic algorithm is described in various steps.

- The initial population is created, where each individual is expressed via defined representation;
- The fitness function is evaluated for the initial population;
- The subset of the population (so-called parents) is selected that will be used in recombination operators to generate offspring;
- The crossover operator is applied to parents to create new offspring;
- The mutation operator is applied with a certain probability;
- The fitness function is evaluated and the individuals with the worst fitness value are removed;
- If the stopping criterion is not met, go to Step 3.

The proposed algorithm can be applied to any problem that can be defined as a graph and which solution depends on the sequence of the elements. When the second population is created in mutation operation the generated solutions have better than average fitness value, so these solutions have higher probability to "survive" and to be selected for crossover operator. This can increase the diversification in the genetic algorithm.

IV. GENETIC ALGORITHMS AND VRP

VRP is a generalization of the TSP problem. Genetic algorithm approaches to solve the VRP can be categorized according to the following feature like Representation: Solution in GA can be encoded as a chromosome (expressed as a literal string), or uuencoded, where encoding of the solution within chromosome is not addressed. Feasibility handling in Genetic algorithm operators can be designed to preserve the feasibility of individuals within a population or allow the generation of infeasible individuals.

An example of VRP solution, where 3 routes are used to service customers expressed as a chromosome is as follows (Berger et al., 1998), where "ne belongs to one route, "nf belongs to the second route and "ng belongs to the third route:| ne1 ne2 … | nf1 nf2 … | ng1 ng2 …| The standard genetic operators can be applied to such a chromosome, however, such a representation does not

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 2, February 2016

hold any problem specific information and, depending on the encoding approach, the selected genetic algorithm can be ineffective. Different approaches for encoding the VRP solution can be found in the literature.

A chromosome representation based on the angles of vectors starting from a depot node is proposed, where the VRP is treated as a planar graph problem. Researches can be found that compare cross over operators designed to work with the chromosome representation. When dealing with constraints, a stochastic approach to find optimal solutions can compute very long, until an acceptable solution has been found (Reid, 2000). For a constrained problem, there exist feasible and infeasible search spaces SF (x SF does not violate any of the defined constraints) and SU (x SU does violate at least one defined constraint). Let us define the whole search space S, then SF S, SU S, SU SF = S, SU SF =. The solution x belongs to the feasible search space SF, if Fc(x) = 0. Highly constrained problems are those, where the feasible search space is very small. The following approaches are used to deal with the infeasibility in genetic algorithms. Genetic algorithm approaches that deal with infeasible individuals require additional approaches to intensify a search to a feasible search space. Insertion heuristics are popular because they are easy to implement and they show good characteristics in creating feasible solutions. In this approach of the genetic algorithm is defined to find the best values of coefficients the coefficient values in the range [0,1] are mapped to values [0, 127] and encoded in 7 symbol substrings as a binary expression and a single point crossover operator is used. The authors argue that the results of insertion heuristic can be greatly improved by a careful search for coefficients. In a push-forward insertion heuristic (PFIH**)** is used to create an initial solution and also as part of the crossover operator. PFIH originally was defined for the VRPTW by Solomon PFIH starts by selecting the first node and forming the initial route from a depot. The algorithm inserts all the other nodes into the constructed route by minimizing the insertion cost function for each node. Insertion heuristic usage in genetic algorithms for the VRP we can see that usually the insertion heuristic is used in the initialization step of GA to create the initial set of solutions

V. GENETIC ALGORITHM FOR VEHICLE ROUTING PROBLEM

The definition "genetic algorithm" can describe either a general approach or a set of the specific genetic operators. In this thesis the proposed version of genetic algorithm for VRP with constraints will be called "new genetic algorithm" Genetic algorithms and insertion heuristics combine together their best characteristics to search for the optimal solution. It is generally accepted that any genetic algorithm for solving a problem should have basic components, such as a genetic representation of solutions, the way to create the initial solution, the evaluation function for ranking solutions, genetic operators, values of the parameters (i.e. population size, probabilities for applying genetic operators, etc.). Genetic algorithm VRP the insertion heuristic is used in the initialization step.

Let us assume that we have a set of nodes N={n0, ..., nk}, where N\{n0} are the nodes that should be visited by a single vehicle and n0 is the depot. The constructed partial solution is x0=({n0}, r0= , Nr0=N\{n0}), where r0 is the empty set of arcs, Nr0 is a set of unvisited nodes. So the solution contains only the depot n0. In the first iteration the randomly selected node nr1 from Nr0 is inserted into a partial solution x0. The new constructed partial solution is x1=({n0, nr1}, r1={(n0, nr1), (nr1, n0)}, Nr1=Nr0\{nr1}= N\{n0, nr1}). Two new arcs (n0, nr1), (nr1, n0) have been created in the solution. Assume that the route is feasible and it can be agreed that it would be the shortest route for a single customer problem {n0, nr1}. In the second iteration a random node nr2 is selected from Nr1. For the newly selected node there exist two possible places for insertion in the solution x1: either in the arc (n0, nr1) or in the arc (nr1, n0). Assume that both insertions are feasible and the arc (nr1, n0) has a lower insertion cost than the arc (n0, nr1). So the newly constructed partial solution is x2=({n0, nr1, nr2}, r2={(n0, nr1), (nr1, nr2), (nr2, n0)}, Nr2=N\{n0, nr1, nr2}). The newly constructed partial solution is feasible and optimal. In the third iteration another random node nr3 is selected from Nr2 and a new optimal solution x3 is created. In each next iteration k a random node nrk is selected from Nr(k–1). If there exists such an arc in rk–1, where the inserted node does not violate any constraints and produces a new feasible partial solution xk, the added node nrk removes the existing arc (ni, nj) and adds two new arcs (ni, nrk) and (nrk, nj). If we find the optimal partial solution in the iteration k–1, the solution created in iteration k is not necessarily optimal, because two new arcs (ni, nrk) and (nrk, nj) are created and there exist a shorter path to some nodes in the route rk.

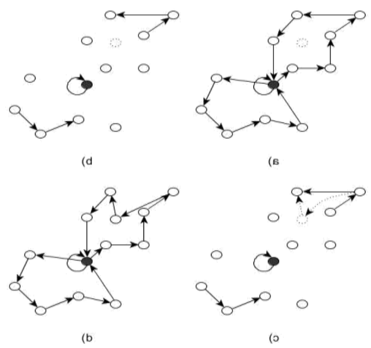

Fig.1. Node insertion process.

In the proposed genetic algorithm crossover and mutation operators are defined in the "remove and reinsert" approach. The approach is similar to a single point relocation method, where the node is extracted and inserted into a different place. However, reinsertion of a single node in a different place can be unsuccessful, because the constructed routes have reached constraint limits and cannot be extended by an additional node. If a single node has been chosen for reinsertion, there is a large probability that the node will be inserted in the same place from which it has been removed. In order to enable the node reinsertion, multiple nodes have to be extracted. In the proposed algorithm the mutation operator is applied with probability MP = 0.1 and the crossover operator is applied to all individuals selected for mating.

In the crossover operation new off springs are generated from two parent solutions that are selected from population by using the ranking method. The new off springs are added to the population and the worst individuals are removed from the population to keep the same population size in each iteration. The defined mutation operators are based on a random insertion and can produce individuals that will not survive. The dashed line presents the average fitness value of the population and the dots present the average fitness value obtained in the mutation operator in each iteration. The fitness value obtained in the mutation operator is not presented as a line because the mutation is applied with certain probability and in some iteration it is not applied at all and in some iteration it can be applied couple times where the averaged value is presented in this case.

VI. GENETIC OPERATORS FOR RICH VEHICLE ROUTING PROBLEM

The typical VRP can be extended by adding additional constraints and other parameters to the problem. The MDVRP includes additional depot nodes and CVRP includes load capacity limitation for a vehicle. VRP with time windows (VRPTW) is an extension, where time window constraints are added. The time window constraint defines a time frame in which a customer can be serviced, i.e. loading or unloading of a vehicle. A vehicle may arrive earlier, but it must wait until the start of the service is possible. The VRP can be extended with some additional constraints, like driver working hours, time, required for a driver to take a rest, etc. Similarly, depending on additional parameters, other variants of VRP are defined. Particular mathematical formulations can be found for each VRP, VRPTW, VRPPD, CVRP problem, where each formulation is based on a customer set, represented as nodes in a graph. The aim of this research is to create the algorithm for the general VRP: rich vehicle routing problem.

The first attempt to define rich vehicle routing problem can be found in. The paper refers to this problem as industrial vehicle problem. In it is called the general vehicle routing problem. Rich vehicle routing problem is a description of different information and constraints reflecting real world situation. VRP, it can be divided into the following components: data, used in the problem tasks, defined to be accomplished constraints that should be satisfied objective of the problem. Data definition includes the graph G = (N, E), which consists of the nodes N and edges E. The data definition also includes a set of vehicles V = {v1, …, vt}. Similarly as in define a set of targets to be achieved. Let us define a set M = {m1, …, mq} as a set of q requests and T = {t1, …, tk} as a set of k tasks to complete requests. Each request mi can be expressed via a set of tasks mi = {ti1, ti2,...}, where tij T, |mi| > 0, mi M, m1 … mq = and m1 … mq = T. The main difference between the request and task is that the task can be processed one at a time by a single vehicle and the requests may be processed in parallel.

VII. CONCLUSION

In contrast to crossover operators, where solutions are constructed from parts of the parent solutions, the proposed crossover operators, that search and preserve parts of the solution that are common to both parents, find the results that in most of the cases are more accurate than the ones found by other crossover operators. Some solutions are equal to the best known solutions even in the cases. As results of VRPTW instances show, the proposed algorithm, based on feasible reinsertion approach in genetic algorithm operators, on crossovers preserving common parts, and on the second population in mutation operator, finds better solutions for 4 out of 6 problem instance groups in comparison with other genetic algorithm approaches. By repeatedly applying random insertion heuristic, the diversification is enabled in the population and, by dealing only with feasible solutions, thus avoiding unnecessary computation and increasing overall computation speed. The solutions are found on average in 38.97 seconds. The proposed genetic algorithm performs ~2 times less floating point operations to find the results comparing to the best value of other algorithms.

REFERENCES

1. Alvarenga, G. B., de A. Silva, R. M., Sampaio, R. M. (2005). A Hybrid Algorithm for the Vehicle Routing Problem with Time Window, INFOCOMP Journal of Computer Science.
2. Anastopoulos, N., Nikas, K., Goumas, G., Koziris, N. (2009). Early Experiences on Accelerating Dijkstra's Algorithm Using Transactional Memory. Parallel & Distributed Processing, IPDPS 2009, IEEE, 1–8.
3. Campbell, A. M., Savelsbergh, M. W. P. (2004). Efficient Insertion Heuristics for Vehicle Routing and Scheduling Problems. Transportation Science
4. Dijkstra, E. (1959). A note on two problems in connexion with graphs. Numerische Mathematik
5. Edmonds, N., Breuer, A., Gregor, D., Lumsdaine, A. (2006). Single-Source Shortest Paths with the Parallel Boost Graph Library. In 9th DIMACS Implementation Challenge – Shortest Paths.
6. Felinskas, G. (2007). Investigation of heuristic methods and application to optimization of resource constrained project schedules. Doctoral dissertation, Vytautas Magnus University.
7. Garcia-Najera, A., Bullinaria, J. A. (2011). An improved multi-objective evolutionary algorithm for the vehicle routing problem with time windows. Computers & OR
8. Gendreau, M., Laporte, G., Musaraganyi, C., Taillard, É. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Computers & Operations Research.
9. Golberg, D. E., Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. Foundations of Genetic Algorithms, San Mateo, CA, Morgan Kaufmann.
10. Hartl, R., Hasle, G., Janssens, G. (2006). Special Issue on Rich Vehicle Routing Problems. Central European Journal of Operations Research.
11. Holland, J. (1975). Adaptation in Natural and Artificial Systems: An Introductory Analysis with applications to biology, Control and Artificial Intelligence. The University of Michigan Press.
12. Li, H., Lim, A. (2003). A Metaheuristic for the Pickup and Delivery Problem with Time Windows. International Journal on Artificial Intelligence Tools.
13. Madduri, K., Bader, D. A., Berry, J. W., Crobak, J. R. (2007). An Experimental Study of A Parallel Shortest Path Algorithm for Solving Large-Scale Graph Instances. In Proceedings of the Workshop on Algorithm Engineering and Experiments, ALENEX 2007, New Orleans, Louisiana, USA.
14. Redondo, J. L., Ortigosa, P. M., Žilinskas, J. (2012). Multimodal Evolutionary Algorithm for Multidimensional Scaling with City-Block Distances. Informatica.
15. Solomon, M.M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research.