Optimal Road Trip Planning with Hybrid Genetic Algorithm

V. Arun [1], Harsh Pandey [2], Himanshu Badlani [3], Rishi Pratap Singh Sisodiya [4], Anshum Sethi [5]
[1, 2, 3, 4, 5] Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India.

Abstract – Finding an accurate and optimal model for traffic forecasting has always been of utmost importance.Through this paper we will be focusing on achieving the above aim through the use of Hybrid Genetic Algorithm which combines the features of local search with the adaptability and features of genetic algorithm thus enhancing local search and global search capabilities to forecast the most optimal route for the trip. Given how dicy and diverse road trips can get especially in a varied terrain country such as India, it will be of great benefit if there is an optimal route response with respect to the constantly changing state. Through this paper, we propose an algorithm that shall yield the optimal route taken into consideration the maximum tourist points that can be visited or traversed between the initial and final points.

Index Terms – Genetic Algorithm, Hybrid Genetic Algorithm, optimal route planning.

1. INTRODUCTION

With the advent of technology and the internet we are in an age where we can gather information and access it all throughout the time with zero or no restraints. This knowledge is very helpful and can be utilized for more than one purposes such as optimizing satellite data and images, weather forecasting and also to analyze maps and topographic contours. Proper extraction of this knowledge and its optimal use is what this paper revolves around. Traffic surveys and analysis are an integral part of the society we today live in.

Based on this colossal data we are efficiently able to analyze these huge amounts of traffic for anomalies, national security purposes and to keep a keen view on the traffic flow so as to ensure security, map geographically manned and unmanned areas and allow real-time survey. We currently deploy more than one methods for traffic analysis such as heat signatures, IR signatures, etc. We also deploy open CV and image processing technology for the same purpose. Fast detection algorithms are also used such as Image Texture Analysis. Gray level occurrence matrix is also one of the most frequently used algorithms. This very real time traffic analysis is also used for Detection of Hot Spots through GPS data articles.

Fuzzy neural networks is also a concept which combine the complementary capabilities of both neural networks and fuzzy logic, thus giving a more than promising technique for short term traffic flow prediction and analysis. The above can also be utilized for road detection, collision detection, avoidance, sign detection and obstacle detection. Traffic data analysis is also utilized for to optimize signal timings in saturated networks by taking queuing and congestion into account to predict equilibrium flows.

Gradient descent algorithm is also developed and utilized to solve the traffic signal problem. Traffic analysis is also used to establish Inter-Vehicle communication and help observe traffic congestion or incidents which can be further developed for safety applications. Its simulations can be run on Ad-Hoc Networks and road traffic simulators. The future applications are more than we can name and with the current rate at which we are making advancements we can be sure that data is the new gold and this very data can be used for vehicle identification and tracking purposes and retrieval of stolen transport modes and modules.

Vehicle route planning is categorized as an NP-hard problems due to the combinatorial nature of this problem and topology complexity of operational network. Obtaining the optimal solutions for NP-hard problems is computationally challenging issue and difficult to solve in practice. Generally, proposed solutions for mission route planning approach can be categorized into three main groups: grid-based methods, graph based strategies, and artificial intelligence based techniques. The grid-based strategies are inefficient in cases where the workspace is very large or complex because the large numbers of cells render such solutions intractable. On the other hand, topology-based (graph-based) methods, which are very popular, usually look for the shortest route between two points in a network (graph). The vehicle routing problem with time windows is a hard combinatorial optimization problem that has received considerable attention in the last decades. This paper proposes a two-stage hybrid algorithm for this transportation problem. The algorithm first minimizes the number of vehicles, using simulated annealing. It then minimizes travel cost by using a large neighborhood search that may relocate a large number of customers.

The major drawback of these methods is that they are time consuming owing to redundant computations and makes them expensive in terms of time complexity. Some of the popular graph search algorithms like A* or Dijkstra operate based on

cell decomposition and determine the cell based route from the start to the destination point. Another category of methods used for mission route planning is the Artificial and computational intelligence (AI and CI) approaches. While various deterministic techniques have been developed over three last decades ,evolution-based ,heuristic and meta-heuristic methods still remain appropriate possibilities for real time applications with larger dimensionality.

2. RELATED WORK

Ant colony optimization (ACO) is based on meta-heurestics that are implemented on population to find an approximate solution to difficult optimization issues. In ant-colony optimization we use artificial ants which are software agents that search for good solution thus optimizing the problem. To better define and ease out the procedure of obtaining the result and apply the algorithm, we transform the problem into a weighted graph problem which searches for optimal path. The software agents or the artificial ants thereafter, incrementally build the solution by moving through and across the graph using a stochastic construction process that is biased by a pheromone model. The pheromone model is a set of parameters associated with the graph components with values that can be modified at run time.

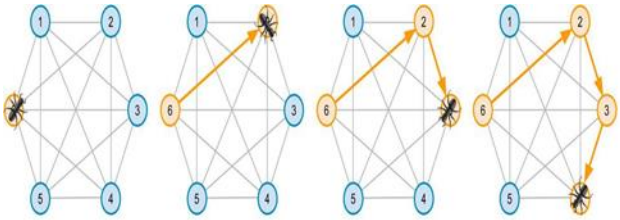

Fig. 1. Node Representation of ACO

For ease of understanding, we consider the travelling salesman problem(TSP) In TSP we have to find out a method to traverse maximum nodes with lowest weights, cost and distance. The same can be done using ACO by considering the cities as a set of vertices on the construction graph.Since in the TSP it is possible to move from any given city to any other city, the construction graph is fully connected and the number of vertices is equal to the number of cities. We set the lengths of the edges between the vertices proportional to the distances between the cities represented by these vertices and we associate pheromone values and heuristic values with the edges of the graph.

Pheromone values are modified at runtime and represent the cumulated experience of the ant colony, while heuristic values are problem dependent values which are set as inverse of the lengths of edges. The ants construct the solutions as follows. Each ant starts from a randomly selected city (vertex of the construction graph). Then, at each construction step it moves along the edges of the graph. Each ant keeps a memory of its path, and in subsequent steps it chooses among the edges that do not lead to vertices that it has already visited. An ant has constructed a solution once it has visited all the vertices of the graph. A teach construction step, an ant probabilistically chooses the edge to follow among those that lead to yet unvisited vertices. The probabilistic rule is biased by pheromone values and heuristic information: the higher the pheromone and the heuristic value associated to an edge, the higher the probability an ant will choose that particular edge. Once all the ants have completed their tour, the pheromone on the edges is updated. Each of the pheromone values is initially decreased by a certain percentage. Each edge then receives an amount of additional pheromone proportional to the quality of the solutions to which it belongs (there is one solution per ant).This procedure is repeatedly applied until a termination criterion is satisfied.

3. EXISTING SYSTEM

Brute force is an approach to the optimal solution by observing multiple patterns and giving the optimal solution among it.It is used to solve a problem based on the problem statement and definitions.It is considered as a easy approach to apply and it is useful for small size problem. The bruteforce methods generate all possible tours and compute their distances. The shortest tour is thus the optimal tour. To solve Travel Plan using Brute-force method we can use the following:

- Calculate the total number of Hamiltonian path.
- Draw and list all the possible Hamiltonian path.
- Calculate the distance of each path.
- Choose the shortest path; this will be the optimal solution.

This approach is feasible for small size problems. When dealing with problems with bigger sampling size or datasets, we have to check various conditions to get the optimal solution. Brute force works better for smaller datasets and data points. With increase in size, brute force becomes more complex, time-consuming and difficult to achieve result with the desired efficiency. Brute Force Algorithm have various disadvantages of Data Optimization and Data Efficiency. It is inefficient and hence useless when dealing with homogeneous problems of higher complexity.

4. PROPOSED MODELLING

Genetic algorithms (GAs) perform well as a global search technique, but many a times, they may take a relatively long time to converge to a global optimum.. Local search(LS) techniques have been incorporated into GAs to improve their performance through what could be termed as learning. Such Hybrid Genetic Algorithms, often known as memetic algorithms (MAs), and are viewed as a form of population based genetic algorithms hybridized with an individual learning procedure capable of fine tuning the global search.

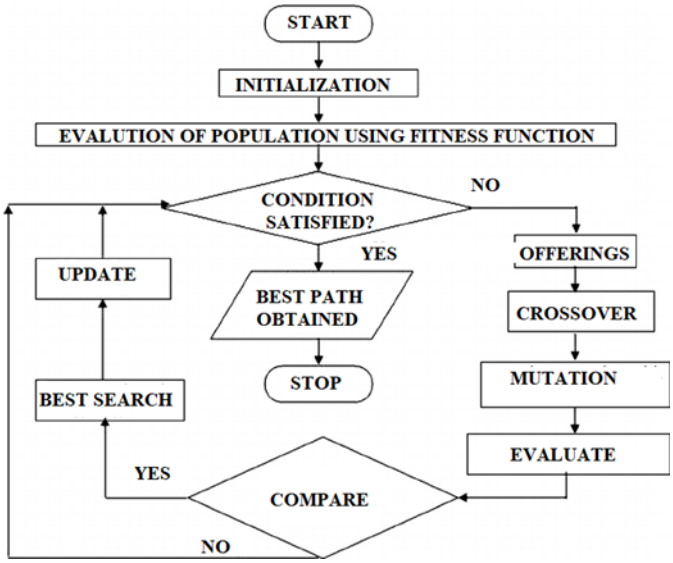

Figure 2 Steps for Hybrid genetic Algorithm

Unlike other search and optimization techniques, a genetic algorithm promises convergence but not optimality, not even that it will find local maxima. This implies that the choice of when to stop the genetic algorithm is not well-defined. We stop the genetic algorithm process when 50 generations have gone by with no better chromosome identified. Since there is no guarantee of optimality, successive runs of the GA will provide different chromosomes with varying fitness measures.

This is one of the drawbacks of using a genetic algorithm for optimization - since there is no guarantee of optimality, there is always the chance that there is a better chromosome lurking somewhere in the search space. Although there is no guarantee of optimality, we are assured of exponential convergence. If we run the GA several times, it will converge each time, possibly at different optimal chromosomes. The schemata which promise convergence are actually indicative of the regions in the search space where good chromosomes may be found.

Typically, the GA is coupled with a local search mechanism to find the optimal chromosome in a region. So, if we use a hybrid algorithm, the problem reduces to ensuring that we run the GA as many times as is needed to pick out all the good regions. If we know before hand the shape of the search space, we can estimate the number of regions we expect to find. We can then repeatedly run the GA until these regions have been found. In most practical problems, however, the shape of the search space is not known before hand.

The systematic approach is then to repeat GA runs until the best chromosomes that are found start to repeat with some regularity. GAs are not good at identifying the optimal value of a chromosome for a problem but do very well in identifying the regions where those optima lie. Therefore, we use a hybrid GA - every ten generations, we anneal the best 1/10th of the population.

This has the effect of moving the top chromosomes in that generation (which are the result of exponential convergence toward the best regions) to the local maximum in their region. In this process of optimal road planning, we use the hybrid genetic algorithm that will reduce the computational of a local search(LS) method without any extra parameters, that is called a best-offspring hybrid genetic algorithm (HGA). This method performs a local search only when the best offspring (solution) in the offspring population is also the best in the current parent population

Our goal is to reduce the total costs associated with the Local Search. It has been noticed that the local search may be repeatedly performed on the same mountain (for finding a maximum) or valley (for finding a minimum). Therefore, it is possible that, after local searching, several chromosomes in a generation are very close to each other, standing on the same top of a mountain or at the same bottom of a valley. This may make it harder for the GA to maintain diversity in its population, an important consideration in avoiding converging to a local optimum.

Therefore, we propose the best-offspring hybrid Genetic Algorithm (BOHGA) where the local Search is only performed on the best offspring in the offspring population when it is also the best overall chromosomes in the current parent population. When such a best offspring appears, it is very likely that the best offspring is located on a new, higher mountain or on a new lower valley.

This action tends to make BOHGA more computationally efficient and helps to prevent converging to a local optimum. Actually, the BOHGA process is a special Hybrid Genetic Algorithm process where a local search is not performed on every new offspring but only on the offspring which are best in both the offspring and the current parent populations. It is possible that not every generation of BOHGA requires a local Search. Once a potential search solution is found by a genetic algorithm a fine tuning search will be conducted by a local Search. Similar to both the GA and the HGA, the whole process is iterated until some appropriate stopping rule is satisfied.

A basic Hybrid Genetic Algorithm procedure has the following steps.

1) Define an objective/fitness function, and set the Genetic Algorithm operators (such as population size, parent/offspring ratio, selection method, number of crossovers, and mutation rate).

2) Randomly generate the initial population as the current parent population.

3) Evaluate the objective function for each individual (chromosome or solution) in the initial population.
4) Generate an offspring population by using GA operators (such as selection/mating, crossover, and mutation).
5) Evaluate the objective function of each individual in the offspring population.
6) Perform a local search on each offspring, evaluating fitness of each new location, and replace the offspring if there exists a locally improved solution.
7) Decide which individuals to include in the next population. This step is referred to as replacement" in that individuals from the current parent population are replaced" by a new population consisting of those individuals from the offspring and/or the parent populations.
8) If a stopping criterion is satisfied, then the procedure is halted. Otherwise, go to Step 4.

A. Features taken into consideration The modes of transportation that need to be considered are bus, subway and on foot

• Bus. The application will consist of a series of buses and lines. Each line consists of several buses that cannot get out of that line. A bus belongs to a single line and a line can have multiple buses

. • Subway. The application will consist of a series of trains and lines. Each line consists of several trains that cannot get out of that line. A train is in a single line and a line can take several trains.

• Foot. Another option is to travel on foot. A user can move between two points in the city on foot. In most cases where we plan a route, the user's point of origin will not match the bus or subway station near you. It will therefore be necessary to plan a path to walk to the bus stop. Another possible option is to get off at any intermediate stops of the tour, either subway or bus, and walk to another stop to get on other transportation.

• Lines. The system consists of several lines for different modes of transport, not for on foot routing. The lines are finite, they have several stops, and can be circular or not. The vehicles move along these lines between different stops. The lines are the sections in charge of merging two stops. When calculating the ideal route these lines are the key decision. More than lines, the weights associated with each of the stages are the key element. You have to mark each section with weight and a long distance. It will take into account the buses or trains time table,as well as the time it takes from the street to the corresponding platform. This time is really necessary, especially when making changeovers.

• Stops. A stop is the point where a line is accessed. The user must get to that station on foot and wait for the appropriate transport vehicle. Once this vehicle has arrived, the user must upload this vehicle, the vehicle moves to the next stop. The operations are equal for both, bus and subway, so we only need to kind of one type of stop. To determine whether the stop belongs to subway or bus, we can see the route through which we reach this stop.

B. Steps for genetic algorithm

Representation: A number of methods have been proposed to represent the routing problem. Some of the methods are binary, matrix and path representation. The most basic and obvious method to represent this problem is path representation due to its simplicity and readability.

Initialization: In this step, an initial population is generated from a random selection of solutions(chromosomes). Genetic algorithm performance varies greatly with the size of the population, if the problem becomes more difficult then the size of the population should increase.

Evaluation: After initial population, the next step is to to improve the solutions. In this process, each chromosome in the population is evaluated by using the fitness function. The fitness function that we are using is inverse of the objective function. In this way, we can find the best chromosome that corresponds to the shortest path.

Selection: The process of selection involves choosing two parents from the entire population for the further crossover process. Tournament selection is used for this section, because of its efficiency and simple implementation.

Crossover: Crossover is a process similar to the mating between two chromosomes with the goal of producing offspring. This results to a generation of new offspring via the order crossover (OX) of two parent chromosomes. OX is used to perform the path/permutation representation, because OX rather concerns circular permutations.

Mutation: The function of mutation is to add the diversity of the population and prevent the chromosomes falling into the local minima.. The current implementation uses exchange mutation which randomly selects two genes of a chromosome and exchanges them. After creating new population by crossover and mutation, there is a big chance, that we will lose the best chromosome.. Therefore elitism method (elite selection) is used. The main objective of elitism is to determine the best individuals and immediately copy them over to the next generation.

Termination condition: Termination condition is the method used to stop the evolutionary process. When the evolutionary process reaches the maximum time period, it then stops.

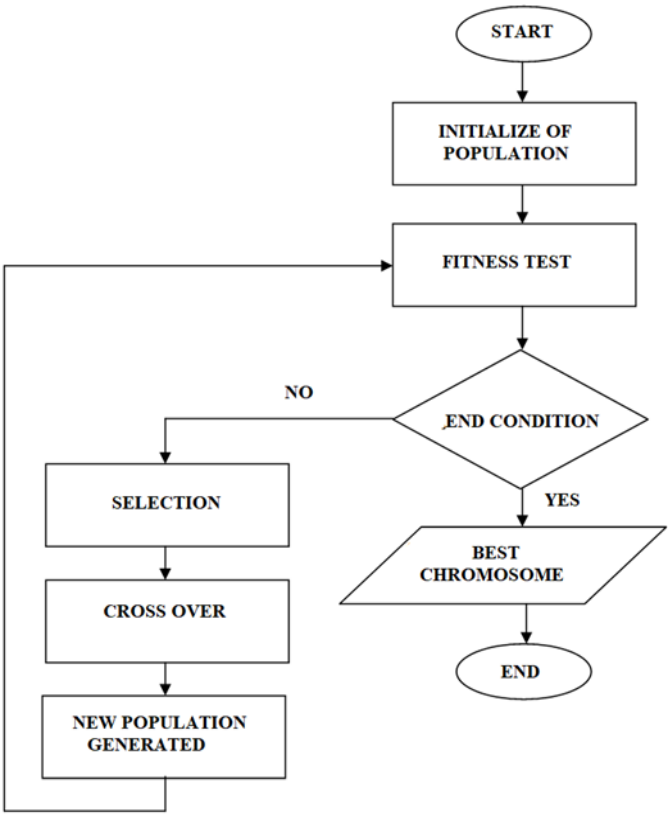

Figure 3 Steps for Genetic Algorithm

5. CONCLUSION

In this proposed approach, a mathematical model for Test Case Optimization problem has been formulated. Hybrid Genetic Algorithm (HGA), a meta-heuristic search approach has been proposed for test case optimization, which produces near global optimal and linear optimal solutions with rapid convergence. The quality of the test cases is improved from generation to generation using mutation score and path coverage based test adequacy criteria. The proposed approach applies two improvement heuristics namely Remove Sharp and Local Opt for guiding the local search procedure. All the three algorithms GA, and HGA have been coded in Python and developed as a tool Hybrid Tester, which is packed as part of the major tool Intelligent Tester. The proposed HGA based test case optimization algorithm has been evaluated for its solution quality by comparing it with other meta-heuristic search approaches such as GA and BA.

The experiments are conducted on various test beds ranging from simple to complex. From the evaluation results, it has been inferred that, GA produces non linear suboptimal solution and usually strikes up at local optima, BA provides linear and optimal solutions, and HGA produces near global optimal and linear optimal solutions with rapid convergence.

Figure 4 : Route Map

Even though there is an overhead involved in the local search procedure, this can be compensated when the quality of the solution is taken into consideration.

The number of test cases needed to exercise the SUT is reduced to an amount of up to 80.6% (approx.) based on coverage and mutation score when compared to the other approaches. Hence, comparatively Hybrid Genetic Algorithm produces near optimal solutions than GA and BA. Finally, this work concluded that, for test case optimization, Hybrid Genetic algorithm is better than Genetic and Bacteriologic algorithms.

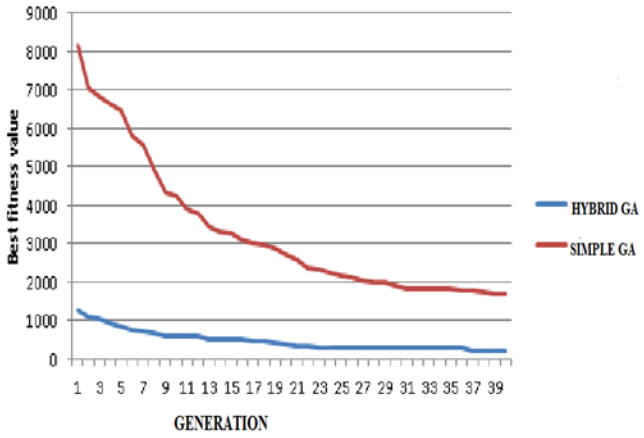

Figure 5: Hybrid GA vs Simple GA

REFERENCES

[1] F. Y. Wang, S. Tang, Y. Sui, and X. Wang, Toward intelligent transportation systems for the 2008 Olympics, IEEE Intell. Syst., vol. 18, no. 6, pp. 811, Nov. 2003.

[2] R. Barrero, J. V. Mierlo, and X. Tackoen, Energy savings in public transport, IEEE Veh. Technol. Mag., vol. 3, no. 3, pp. 2636, Sep. 2008

[3] S. Tsugawa and S. Kato, Energy ITS: Another application of vehicular communications, IEEE Commun. Mag., vol. 48, no. 11, pp. 120126, Nov. 2010.

[4] S.HartwigandM.Buchmann,Emptyseatstraveling,NokiaRes.Center Bochum, Bochum, Germany, Feb. 2007.

[5] R. Fagin and J. H. Williams, A fair carpool scheduling algorithm, IBM J. Res. Develop., vol. 27, pp. 133139, Mar. 1983.

[6] Vidal T, Crainic TG, Gendreau M, Prins C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. Computers Operations Research 2013; 40(1):47589.

[7] B. Yu, Z.Z. Yang, B.Z. Yao. A Hybrid Algorithm for Vehicle Routing Problem with Time Windows. Expert Systems with Applications 38 (2011) 435441.K. Elissa, Title of paper if known, unpublished.

[8] Keivan Ghoseiri, Seyed Farid Ghannadpour. Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. Applied Soft Computing 10 (2010) 10961107.

[9] Collective Intelligence Sentimental Analysis of Twitter Data By Using Sta Chi-Bin Cheng, Keng-Pin Wang. Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. Expert Systems with Applications 36 (2009) 77587763.

[10] Ziauddin Ursani, Daryl Essam, David Cornforth, Robert Stocker. Localized genetic algorithm for vehicle routing.

[11] CVRPdataset. http://neumann.hec.ca/chairedistributique/data/vrp/. Retrieved at 24 November 2014.

[12] Dantzig, G.B. and J.H. Ramser. 1959. The truck dispatching problem. Management Science.