Ginni Chawla et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 2, Issue 4, December 2015, pp.73-78

Solving Optimization Problem By Hybrid Genetic Algorithm Using Hill Climbing In Replacement Operator

Ginni Chawla[1], Ms. Yogesh Bala[2]

[1]M.Tech (C.S.E) Student, Panipat Institute Of Engineering & Technology, Kurukshetra University
[2]Assistant Professor (C.S.E), Panipat Institute of Engineering And Technology, Samalkha

Abstract: Genetic algorithm is a population-based search and exploiting objective function. Every basic genetic operators used in a simple GA utilizes "random choice" to an extent or another. Optimization ability can be improved when problem specific knowledge is incorporated and goal oriented operators are used. The population based local refinement mechanism, searches the local area for minima. GA and neighborhood search technique will result in early findings of the optima. In this paper implementation of hill climbing in replacement operator and empirically analyze the convergence rate of hybrid algorithm with simple genetic algorithm. Both algorithms use the complementary property of exploitation to find optimal solution. Memetic algorithm performs good to find optimal result of complex problems. Performance of memetic algorithm is affected by population size and number of generated children. Proposed work also tries to analyze the convergence rate of memetic algorithm on TSP various City problems .

I. INTRODUCTION

Travelling salesman problem

The Travelling Salesman Problem describes a salesman who must travel between N cities. The order in which he does so is something he does not care about, as long as he visits each one during his trip, and finishes where he was at first. Each city is connected to other close by cities, or nodes, by airplanes, or by road or railway. Each of those links between the cities has one or more weights (or the cost) attached. The cost describes how "difficult" it is to traverse this edge on the graph, and may be given, for example, by the cost of an airplane ticket or train ticket, or perhaps by the length of the edge, or time required to complete the traversal. The salesman wants to keep both the travel costs, as well as the distance he travels as low as possible. It is NP hard problem because if there is a way to break this problem into smaller component problems, the components will be at least as complex as the original one.

Optimization

Optimization is a mathematical discipline that concerns the finding of minima and maxima of functions, subject to so-called constraints. Optimization originated in the 1940s, when George Dantzig used mathematical techniques for generating "programs" (training timetables and schedules) for military application. Since then, his "linear programming" techniques and their descendents were applied to a wide variety of problems, from the scheduling of production facilities, to yield management in airlines. Optimization comprises a wide variety of techniques from Operations Research [5], artificial intelligence and computer science.

Hill Climbing

An optimization problem can usually also be modelled as a search problem, since searching for the optimum solution from among the solution space. Without any loss of generality, assuming that our optimization problems are of the maximization category. So, here is the hill climbing technique of search:

1. Start with an initial solution, also called the starting point. Set current point as the starting point
2. Make a move to a next solution, called the move operation
3. If the move is a good move, then set the new point as the current point and repeat (2). If the move is a bad move, terminate. The last current solution is the possible optimum solution.

Genetic Algorithm

A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation ,selection and crossover (also called recombination). The current framework of GAs was first proposed by Holland (1970's) and his student Jong, and was finally popularized by another of his students, Goldberg. In fact the use of evolution to solve problems had been suggested in the 1940s by Alan Turing. Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. The genetic algorithm includes some representation-specific operators and some representation neutral operators. The initialization, mating (typically implemented as crossover), and mutation operators are specific to the representation. Selection, replacement, and termination are all independent of the representation [6].It is based on survival of the fittest.The fitter individual will dominate over the weaker ones and will have more chance to be selected for crossover and mutation. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached.

A typical genetic algorithm requires:

1. Genetic representation of the solution domain,
2. Fitness functions to evaluate the solution domain.

II. RELATED WORK

PHAM Dinh Thanh et al., 2013[1] In this paper Traveling Salesman Problem (TSP) is a well-known NP-hard problem. Many algorithms were developed to solve this problem and gave the nearly optimal solutions within reasonable time. This paper presents a survey about the combination Genetic Algorithm (GA) with Dynamic Programming (DP) for solving TSP. We also setup a combination between GA and DP for this problem and experimented on 7 Euclidean instances derived from TSP-lib. Experimental results are reported to show the efficiency of the experimented algorithm comparing to the genetic algorithm.

III. PROBLEM FORMULATION

This research paper has following objective:

1. To implement hill climbing in replacement operator.
2. To compare the performance of hybrid genetic algorithm with simple genetic algorithm.

Simulation Tool:-MATLAB

MATLAB is a very powerful software package that has many built-in tools for solving problems and for graphical illustrations. The simplest method for using the MATLAB product is interactively; an expression is entered by the user and MATLAB immediately responds with a result. It is also possible to write programs in MATLAB, which are essentially groups of commands that are executed sequentially. MATLAB is a mathematical and graphical software package; it has numerical, graphical, and programming capabilities. It has built-in functions to do many operations, and there are toolboxes that can be added to augment these functions.

IV. PROPOSED WORK

Hybrid algorithm

Moscato & Norman (1992) have introduced the term memetic algorithm to describe evolutionary algorithms in which local search plays a significant part. Heuristic optimization algorithms such as Simulated Annealing or Genetic Algorithms often can locate near optimal solutions but can require many function evaluations [2]. Local search algorithms, including both gradient and non-gradient based methods, are quite efficient at finding the optimal within convex areas of the design space but often fail to find the global optimal in multimodal design spaces. It is possible to combine GAs with a secondary method to create a hybrid GA (also referred to as a Memetic algorithm) . Hybrid GAs usually consist of a GA combined with either a local search (for a general problem solver) or a heuristic (for a more problem dependant solution) [3].

Algorithm I - Memetic algorithm with local Hill Climbing search

Input: TSPLIB Symmetric TSP instance

Parameters: **GN**: Number of Generations, **P** Population Size, **MP**: Mutation probability, **CP**: Crossover Probability

Output: Best Solution, Solution History and Execution Time

Begin

Generate an initial population of individuals of size P;

Compute **fitness function** for each individual using **Algorithm Fitness**;

for i:=1 to **GN** do

Begin

Select the population i from the population i-1 by means of **Roulette Wheel** selection

Cross over each pair Using Crossover Probability **CP**;

Mutate each pair Using Mutation Probability **MP**;

End;

Choose the best individual from the final population using **Hill Climbing Search** in algorithm II;

End;

Algorithm II – Local Hill Climbing algorithm

Begin

```
  currentNode = startNode;
  loop do
    L = NEIGHBORS(currentNode);
    nextEval = 0;
    nextNode = NULL;
    for all x in L
      if (EVAL(x) < nextEval)
         nextNode = x;
         nextEval = EVAL(x);
    if nextEval >= EVAL(currentNode)
      //Return current node since no better neighbors exist
      return currentNode;
    currentNode = nextNode;
```

End

Algorithm: Fitness Function

Input: Permutation Encoding of Cities

Cost = Matrix to Store Distance of the Round Trip from City i

$$\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}.$$

For Each i in Cities

$Cost_i$ = Calculate Cost from city i to all cities

End

Return Cost, Min(Cost)

Here first population is initialized randomly. Then permutation encoding is used which describes the order of cities the salesman visits. Here partially matched crossover is used in which position wise exchanges is done. In between crossover points genes get exchanged[4]. Inversion mutation is used in which end points of the section cut, gets reversed. Roulette Wheel selection is used which is also known as fitness proportionate selection. It is a genetic operator used for selecting potentially useful solutions for recombination. In fitness proportionate selection the chance of an individual being selected is proportional to its fitness, greater or less than it's competitors fitness. Here I have experimented it by taking 500 generations and 100 as population size. Breeding pool is at 100% of population size.

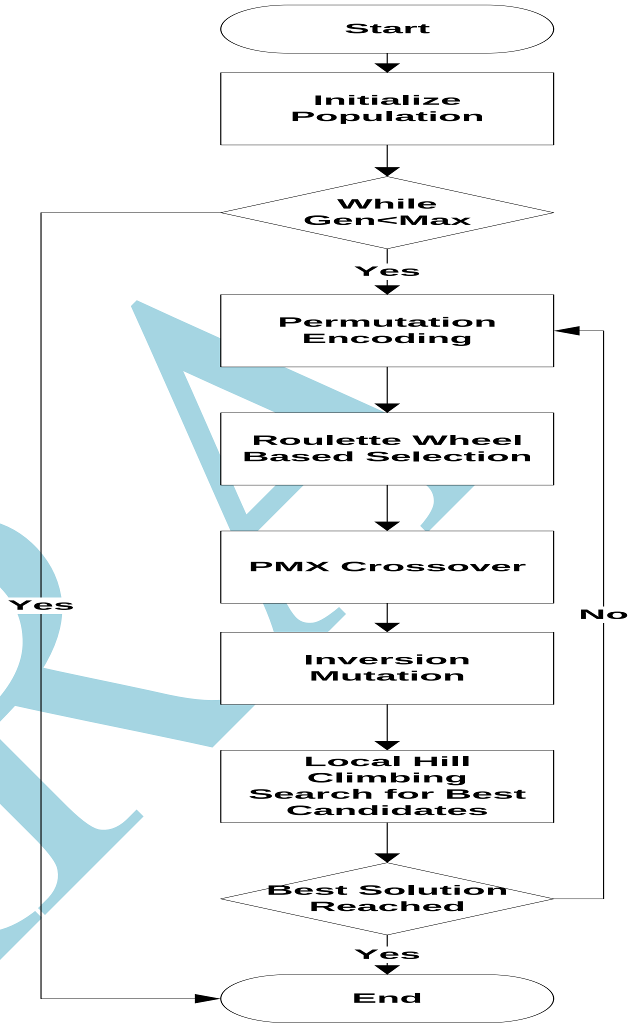

Figure1 : Proposed workflow of HC Replacement based Hybrid GA for TSP

V. RESULT AND ANALYSIS

To analyse the optimization ability of the algorithm it is experimented on various tsp problem instances like Eil51, Eil76, Oliva30. The general parameters used in my experiment are:

- Random initialization
- Permutation encoding
- PMX crossover
- Inversion mutation
- Selection: Roulette Wheel
- 0.8 crossover probability
- Breeding pool at 100% of population size
- mutation probability .01
- For 500 Generations.
- Population size of 100.

Results

Eil51 Instance

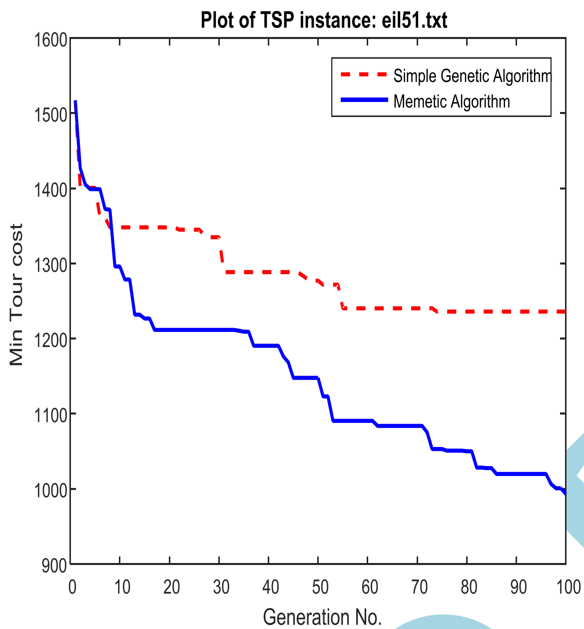

Figure 2: Eil51 Results with population size 100

Figure 2 shows the performance of Genetic algorithm and Memetic algorithms on eil51 with 100 population size and 500 generations and shows more closeness to optimum with memetic algorithm in comparison to simple genetic algorithm.

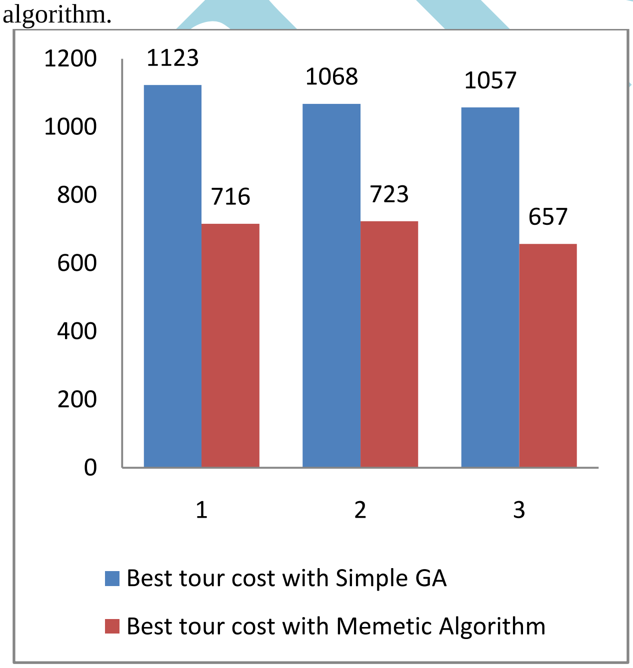

Figure 3: Results for Eil51 with 51 cities and 426 known optimum

Sr. No.	Best tour cost with Simple GA	Best tour cost with Memetic Algorithm
1	1123	716
2	1068	723
3	1057	657

Table 1: Results for Eil51 with 51 cities and 426 known optimum

Eil76 Instance

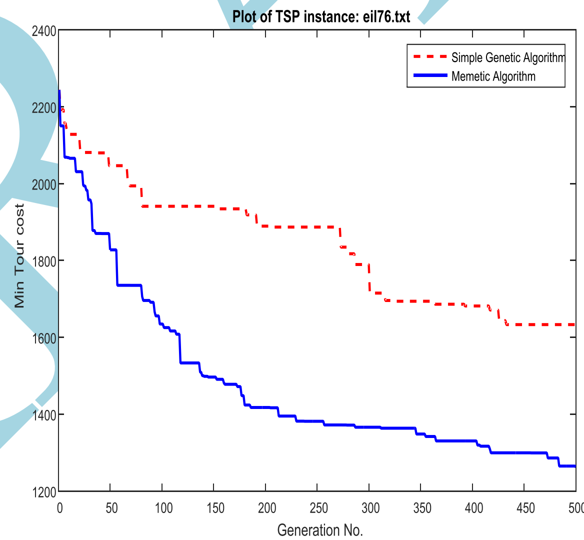

Figure 4 : Eil76 Results with population size 100

Figure 4 shows the performance of Genetic algorithm and Memetic algorithms on eil76 with 100 population size and 500 generations and shows more closeness to optimum with memetic algorithm in comparison to simple genetic algorithm.

Sr. No.	Best tour cost with Simple GA	Best tour cost with Memetic Algorithm
1	1779	1315
2	1636	1217
3	1651	1237

Table 2 : Results for Eil76 with 76 cities and 538 known optimum

Oliva30 Instance

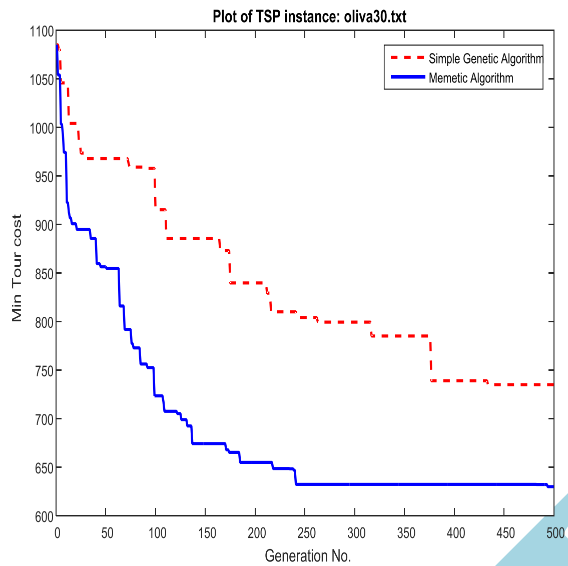

Figure 5: Oliva30 Results with population size 100

Sr. No.	**Best tour cost with Simple GA**	**Best tour cost with Memetic Algorithm**
1	**734**	**629**
2	**781**	**547**
3	**706**	**567**

Table 3 : Results for Oliva30 with 30 cities and 426 known optimum

Figure 5 shows the performance of Genetic algorithm and Memetic algorithms on oliva30 with 100 population size and 500 generations and shows more closeness to optimum with memetic algorithm in comparison to simple genetic algorithm.

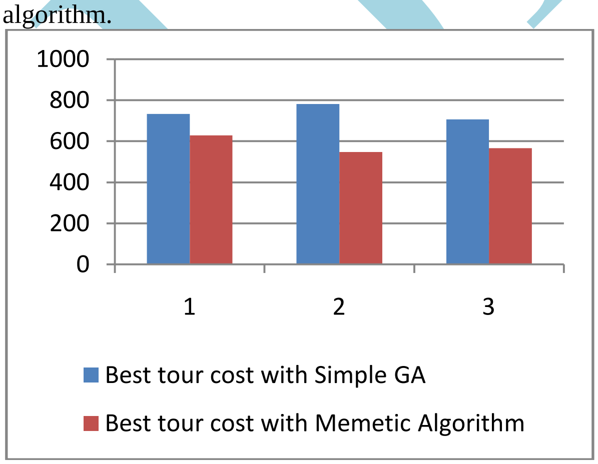

Figure 6 Results for Oliva30 with 30 cities and 426 known optimum

VI. CONCLUSION AND FUTURE WORK

This Work is about to implement and analyze the optimization ability of Hill climbing in replacement. The memetic algorithm ability depends on the way of utilizing the information from both the searching mechanism i.e. Genetic algorithm and local search. Here the optimization of different tsp instances is implemented to evaluate the general computational behavior of Genetic and memetic algorithm. At the initial stage, the genetic algorithm is implemented as the basic architecture on this algorithm. Here, replace all and hybridization of hill climbing in replacement Algorithm was discussed comparatively to identify the convergence scenario. In future we can work on following:

1. We will try to implement GA with HC replacement over VRP
2. Instead of Genetic algorithm we can also work on Monte Carlo based Approximation method for Solving TSP by gaining Parallelism.
3. To enable Global search using Entropy based methods, as more entropy means bad solutions and less means more optimal.

REFERENCES

[1]. PHAM Dinh Thanh, HUYNH Thi Thanh Binh, and BUI Thu Lam. "A survey on hybridizing genetic algorithm with dynamic programming for solving the traveling salesman problem." In *Soft Computing and Pattern Recognition (SoCPaR), 2013 International Conference of*, pp. 66-71. IEEE, 2013.

[2]. MuzaffarEusuff, Kevin Lansey& FayzulPasha ,“Shuffled Frog leaping Algorithm: a memetic meta-heuristic for discrete optimization.”,Engineering OptimizationVol. 38, No. 2, March 2006.

[3]. J. Digalakis and K. Margaritis, “Performance Comparison of Memetic Algorithms” Appl. Math. Comput. 158 (2004).

[4]. N Kumar, and R. Kumar Karambir. "A comparative analysis of PMX, CX and OX crossover operators for solving traveling salesman problem." International journal of Latest Research in science and technology 1 (2012).

[5]. Hacker, Kurt A., John Eddy, and Kemper E. Lewis. "Efficient global optimization using hybrid genetic algorithms." In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pp. 4-6. 2002.

[6]. H. A. Sanusi, A. Zubair , R. O. Oladele,“Comparative Assessment of Genetic and Memetic Algorithms”,Journal of Emerging

Trends in Computing and Information Sciences, VOL. 2, NO. 10, October 2011.

[7]. M. A. H. Akhand, Pintu Chnadra Shill, Md Forhad Hossain, A. B. M. Junaed, and K. Murase. "Producer-Scrounger Method to Solve Traveling Salesman Problem." (2015).

[8]. Li-Juan Zheng, De-Cun Dong, and Dong-Yun Wang. "A hybrid intelligent algorithm for the vehicle scheduling problems with time windows." In Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on, pp. 2756-2761. IEEE, 2014.

[9]. Rahul Chaudhary and Arun Prakash Agrawal. "Regression Test Case Selection for Multi-Objective Optimization Using Metaheuristics." (2015).

[10]. PoonamGarg,"A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm",International Journal of Network Security & Its Applications (IJNSA), Vol.1, No 1, April 2009.