Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-5, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

A Hybrid Approach to solve Travelling Salesman Problem in Map Reduce Framework using Parallel Genetic Algorithm

[1]Shishirram Borker, [2]Shobana Markeshan,[3]Shreekanth mayya
[4]Shreya Suvarna J & [5]Mr Vasanth Nayak
[1,2,3,4,5]Information Science & Engg Canara Engineering College

Abstract : *Given a list of cities and distance between them. The Travelling Salesman Problem (TSP) is to find the shortest tour which salesman visits all the cities exactly once and ends the city which he started. Since TSP is NP Hard, there are many methods and solutions for this problem. One of them is Parallel Genetic Algorithm (PGA). In this paper we took Parallel Genetic Algorithm to solve Travelling Salesman Problem on MapReduce framework. MapReduce framework is a framework which is used to implement Parallel Genetic Algorithm and to solve travelling Salesman Problem. We used free licensed Hadoop as MapReduce framework.*

Keywords: *Travelling Salesman Problem, Genetic Algorithm, MapReduce Framework, Hadoop.*

1. INTRODUCTION

The process of making something better is called Optimization. The Optimization problem is the one which finds the better path within the available solution. TSP results in more than one solution. In this paper, we find better solution in less time and also with increased performance.

a) Travelling Salesman Problem

TSP is an important combinatorial optimization problem. In the given list of cities, our goal is to find the shortest tour that visits all the cities exactly once and ends the city which was the starting city. Since TSP is a NP-Hard problem, it is difficult to solve.

b) Genetic Algorithm

There are many approaches to solve TSP. In this paper we have used Genetic Algorithm (GA) to solve TSP. Genetic Algorithm is a heuristic solution which gives nearly optimal solutions within a reasonable time. Although Genetic Algorithm is time efficient and a good solution for TSP, sometimes sequential GA gets stuck with local optima or time consuming when the number of cities increases.

The first step of GA is to find all possible solutions for the given vertices and applies various operators such as fitness evaluation, selection, crossover and mutation operators to optimize the solution. Population is defined as all the number of possible solutions which have obtained from the given vertices and the chromosomes are the each individual. The each value in individual solution is called as gene or node.

The following are the steps of GA process:

Encoding: Before applying the genetic algorithm, the individual solutions should be represented in such a way so

that the computer can process it. This method is called as encoding.

Few methods of Encoding are:

- The sequence of 0 s and 1" s are used for representing the genes called Binary Encoding.
- The sequence of values is used which is called Value Encoding.
- Every chromosome is a string of numbers called Permutation Encoding.
- Every chromosome is a tree of objects or nodes called Tree Encoding.

After the genetic algorithm operators are applied, the results are converted to the required format which is called as Decoding.

Initial Population Generation: Initially, the GA starts with initial population. This initial population will be generated randomly by the GA.

Fitness Evaluation: After the Initial population is generated, each individual solution will have a fitness value. This fitness evaluation process can be done in many ways. The calculation is done based on the user requirement. This process will show how fit the chromosome is.

Selection: The selection process selects the fitness path for the further process of GA. In this process selection perceive the fitness values of each individual and selects the individual with best fitness value for the next process. Different types of selection methods are Elitism method, Roulette Wheel method, Tournament selection method and so on. The selected path will be given to the crossover process.

Crossover: This is an important stage in GA. The result of GA mainly depends on the outcome of the crossover method. So, more importance must be given for the crossover method. In this process, two paths are taken and combined to produce the new child path. The child process is known as offspring.

Mutation: The uniqueness between the chromosomes is maintained in this process. Some changes will be made to the path so that new values are generated which are unique from other paths and may produce better results. Mutation operator will modify the path by swapping the values of a solution.
The GA will be completed only if all the above processes are applied.

c) Parallelizing a Genetic Algorithm

The important component of this project is parallelization of the Genetic Algorithm. GA is an Iterative process. So, it does not fall into Map Reduce framework. A hierarchical reduction phase is the one in which GA is reduced to Map Reduce problem. PGA divides the search space into many smaller pieces to find the nearly optimal solution. The sub-optimal solutions need to avoid local minima during this process. Different frameworks such as Hadoop, Open-cl, Parallel Java, and C-MPI are available for parallelization. The latest buzz words in cloud computing currently is Hadoop which employs a map reduce model.

d) Hadoop

In this paper, Licensed Hadoop[4] is used which is an open-source which implemens MapReduce framework. Hadoop supports in processing large set of data in distributed computing environment. It is sponsored by the Apache Software Foundation.
The five main components of hadoop cluster are, namely NameNode, DataNode, Secondary NameNode, JobTracker, and TaskTracker. The data on HDFS are managed by NameNode. The data's are stored in DataNode and interacts with NameNode. The back-up for NameNode is done by Secondary NameNode. Communicates with client is done by TaskTracker and runs the client's MapReduce jobs by means of TaskTracker and coordinates TaskTrackers to complete job consistently. The TaskTracker responsiblility is the execution of map and reduce tasks which are the part of whole MapReduce job. NameNode, Secondary NameNode, JobTracker will run only in master machine because they are the master part of MapReduce framework. DataNode and TaskTracker will run in slave machine because they are slave components. The Hadoop cluster can have one NameNode, one Secondary NameNode and one JobTracker, while the cluster can have any number of DataNodes and TaskTrackers.

e) MapReduce Framework

MapReduce parallelization is another way of parallelization which is studied in this paper.
Map-Reduce is a framework which contains maper and reducer functions. Applications developed on MapReduce framework are self-fault tolerant. MapReduce framework is implemented in software implementer called Hadoop. The basic approach of Hadoop is transferring of program code to the data node instead of transferring data across the network.

2. Related work

A. Formulation

To formalize the definition of the TSP, many basic computer scientific terms must be defined.The term graph is used to explain a set of vertices, or nodes, and a set of edges that connect the vertices. A complete graph consists of an edge between all pairs of vertices. It's possible that the edges can be directed or undirected. It can have weights related with them. Sequence of edges that begins at one vertex and ends at another vertex .the edge is a path between two vertices. A cycle is a path that starts and

ends at the same vertex. Hamiltonian cycle is a cycle that covers all nodes in the graph exactly once.

The TSP can now be defined as follows:

Determine the shortest Hamiltonian Cycle in a complete weighted graph.

B. Solution Algorithms

As it is explained, 3 types of approach for solving NP-complete problems are as follows:

- Devising algorithms for finding correct solution (they will work only for relatively small problem sizes)
- Devising "sub-optimal" or heuristic algorithms which deliver seemingly or provably good solutions, but that could not be proved as optimal.
- To find special instances for the problem which either exact or better heuristic are possible?

In exact here are the Solution algorithms for TSP:

1. Exact Algorithms

To find an optimal solution, the exact algorithms are guaranteed but it may take an exponential number of iterations.

This means that you have to generate all possible routes and takes the shortest. This becomes impractical as the number of towns increases for the reason that the number of possible routes is !(N-1). The cutting –plane or facet –finding algorithms are the most potential exact algorithms. These algorithms are quite complicated, and they are very demanding of computer power.

2. Heuristic Algorithms

They are much faster and they provide good solutions, but there is no guarantee of the optimal solution. Some of algorithms give solutions which on average differ only by a few percent (2-3%) from the optimal solution. Hence, if a small deviation from optimum solution can be accepted, it may be appropriate to use a Heuristic algorithm.

The heuristic algorithms are categorized into three classes:

• Tour construction algorithms

• Tour improvement algorithms

• Composite algorithms

• **Tour Construction Algorithms**

In the construction algorithm cities are built one-by one and tour is built. And the best tour is selected. The algorithm is 11-14% optimal.

An Example of a tour constructor algorithm is:

The Nearest Neighbor Algorithm

It is simple approach and takes less time. The main aim of this algorithm is to visit the nearest city. Select the unvisited cities from the starting point, travel the nearest cities that has not been visited yet and return back to the first city.

Nearest Insertion

Nearest insertion is simple and straightforward. The ideology of this algorithm is to select the shortest edge and make a sub-tour of it, then select the city that is not present in sub-tour, and check for the shortest distance. Repeat this until no city left.

• Tour Improvement Algorithms

The tour construction process is a greedy approach algorithm. During tour construction process already constructed tour remains unchanged. This is contradictory to the tour improvement process which changes the settings or configurations of tour during the iterative improvement process till the shortest tour is found. A simple example of this type of algorithm is the:

2-Opt Algorithm

This algorithm starts from either the tour that resulted from the nearest neighbor process or random tour. Here we replace two links of the tour with the two other links such that new tour shortest than the previous. Repeat this until no improvements possible.

• Composite Algorithms

To obtain initial solution for tsp use construction algorithm. And then use improvement algorithm to improve it. An example of such algorithm is:

Simulated Annealing Algorithm

The stimulated annealing algorithm ideology is statistical mechanics and the heat bath is present so that the behavior of physical systems which is an

analog is motivated. To obtain a better solution, by going from one solution to the next we use this algorithm.

3. METHOD

In this paper, we apply PGA to solve TSP. GA consist of Fitness Evaluation, Selection, Crossover, Mutation.

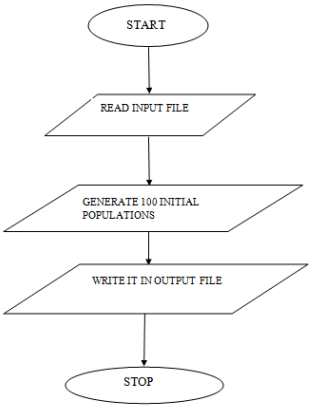

Fig3.1: Initial Population

STEP-1: Initially, the input is sent in the format of vertices and edges in a file which consists of G (v,e) where v is the vertices which represents city and e is the edges which represents the distance between the cities.

STEP-2: Randomly hundred initial path is created from the given set cites and result is stored in a file which is called initial population.

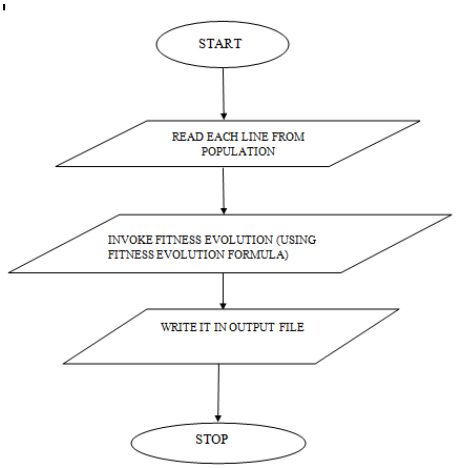

Fig3.2: Fitness Evaluation

STEP-3: Each path is selected form the initial population files for evaluation fitness value by using distance matrix formula. The fitness value is generated for each path of the generated populations and the result will be stored in HDFS.

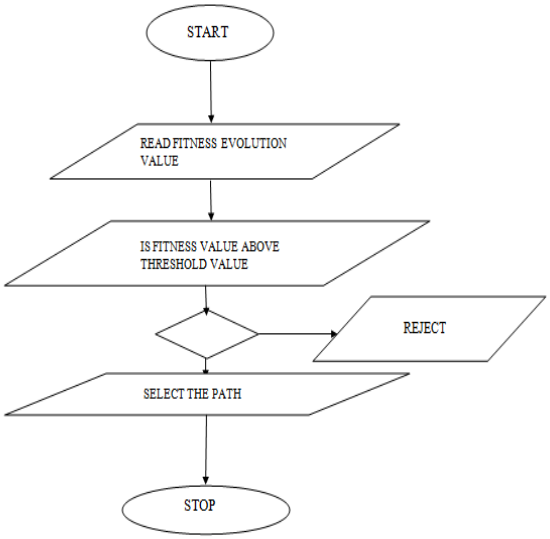

Fig3.3: Selection

STEP-4: The path which has the Fitness value greater than the threshold value will be rejected and the rest will be selected for further process called selection process.

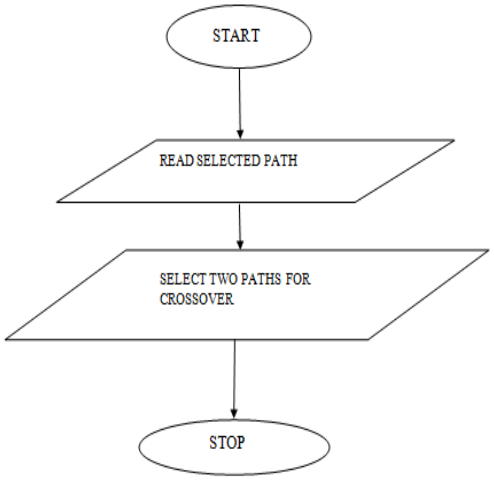

Fig3.4: Crossover

STEP-5: From the selected path two paths are selected which are called chromosomes and used for the further process called crossover in GA.

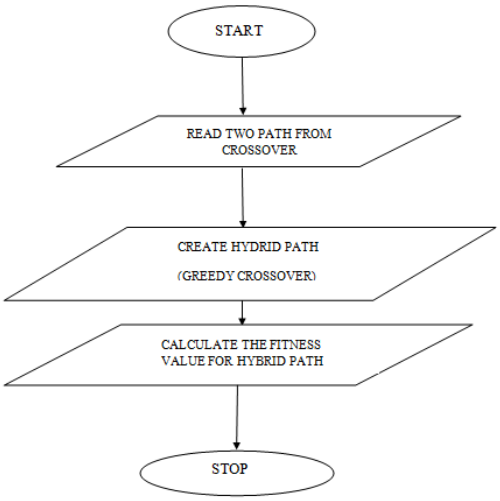

Fig 3.5: Mutation

STEP-6: Many paths will be generated by applying crossover and mutation. The best path will be selected from further process and the GA will be an iterative process until it finds the best path.

4. PERFORMANCE ANALYSIS

We compare the parallel implementation with a sequential implementation of genetic algorithm to get the performance of it. SGA is also compared with other implementation. The problem is the distance from node *i* to node *j* and the distance from node *j* to node i could be different. The algorithms are executed for at least 10 times to get the results. Population size of Parallel Genetic Algorithm in MapReduce is set to 100.

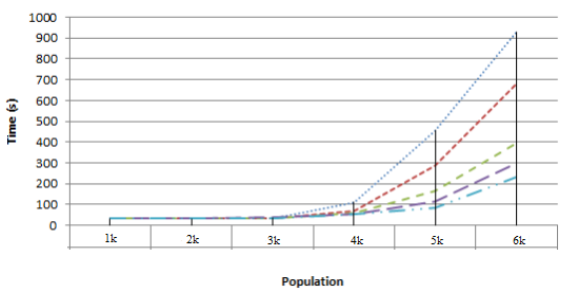

Fig 6.1: Time Vs Population

The above fig 6.1 explains the relationship between time and populations generated. Each population will have either 1k, 2k, 3k, 4k, 5k or 6k number of paths generating (As we assign). The time increases as the number of populations vary.

5. RESULT

In this paper, we have taken single node hadoop with all the back ground process executing on single machine. The installation of the hadoop is done on the virtual machine. Twenty cities are taken for the analysis purpose. In the initial stage, hundred populations are created and the calculation of the fitness evaluation is done. In the later step, to obtain the optimized result, the initial population will undergo several of generation. The best chromosome will be selected for the further crossover.

6. CONCLUSION

MapReduce Framework implementation for GA with large population is shown in this paper. The algorithms performance suggests that MapReduce is a very less effort and more productivity parallel framework which is better performance improvement compared to serial GA. The improvement of performance is up to 5 million populations which is a feat for any computing resources are proposed in this algorithm. The performance pattern also suggest that more population can be needed when more processing resources is made available. The fact that Hadoop MapReduce Framework requires input data which also creates an I/O overhead for the proposed algorithm. The use of MapReduce Framework such as Hadoop may give an even better overall performance.

7. REFERENCE

[1] Anitha Rao1 and Sandeep Kumar Hegde,"Novel Method to Solve Travelling Salesman Problem Using Sequential Constructive Crossover Using Map/Reduce Framework", International Journal of Science and Research,Volume 4 Issue 5, May 2015.

[2] Noor Elaiza Abd Khalid, Ahmad Firdaus Ahmad Fadzil and Mazani Manaf, "Adapting MapReduce Framework for Genetic Algorithm with Large Population", IEEE Conference on Systems on 13 - 15 December 2013.

[3] Harun Raşit Er and Prof. Dr. Nadia Erdoğan," Parallel Genetic Algorithm to Solve Traveling Salesman Problem on MapReduce Framework using Hadoop Cluster" International Journal on march 2013.

[4] http://hadoop.apache.org

[5]http://www.iwr.uni-eidelberg.de/groups/comopt/software/TSPLIB95