International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

A Novel Method to Solve Travelling Salesman Problem Using Sequential Constructive Crossover Using Map/Reduce Framework

Anitha Rao[1], Sandeep Kumar Hegde[2]

[1]M.Tech Student, Dept.of Computer Science & Engineering, Mangalore Institute of Technology and Engineering
Mangalore, Karnataka, India

[2]Assistant Professor, Dept.of Computer Science & Engg., Mangalore Institute of Technology and Engineering
Mangalore, Karnataka, India

Abstract: ***The Travelling Salesman Problem (TSP) is well known in the field of combinatorial optimization. Since it is an NP-complete problem, there is no efficient method to solve this problem and give the best result. Many algorithms are used to solve travelling salesman problem. Some algorithms give optimal solution, but some other algorithms give the nearest optimal solution. The genetic algorithm is a heuristic method which is used to improve the solution space for the Travelling Salesman Problem. The genetic algorithm results in nearest optimal solution within a reasonable time. This paper mainly focuses on the various stages of genetic algorithm and comparative study of various methods used in the genetic algorithm. The paper also proposes a method to solve the travelling salesman problem using Sequential Constructive Crossover operator and hence improve the quality of solution space.***

Keywords: Travelling Salesman Problem, Genetic Algorithm, Selection, Sequential Constructive Crossover, Mutation

1. Introduction

Optimization is the process of making something better. An optimization problem is a problem which boosts the solution or finds the better solution from all available solution spaces. The terminology "best" solution implies that there is more than one solution. The Travelling salesman problem also results in more than one solution, but the aim is to find the best solution in a reduced time and the performance is also increased. So the heuristic genetic algorithm is used.

1.1 Travelling Salesman Problem

The Travelling Salesman Problem (TSP) [1] is an optimization problem used to find the shortest path to travel through the given number of cities. Travelling salesman problem states that given a number of cities N and the distance between the cities, the traveler has to travel through all the given cities exactly once and return to the same city from where he started and also the cost of the path is minimized. This path is called as the tour and the path length is the cost of the path.

In a complete weighted undirected graph G (V, E), cities are represented by the vertices and the distance between the cities are represented by the edges. The travelling salesman problem has to find the minimized Hamilton cycle that starts from some specified vertex, visits all other vertices exactly once return to the same specified vertex.

The travelling salesman problem is used in many application areas [2] like planning, logistics, manufacturing of microchips, DNA sequencing, vehicle routing problems, robotics, airport flight scheduling, time and job scheduling of machines.

The travelling salesman problem can be classified as Symmetric Travelling Salesman Problem (STSP), Asymmetric Travelling Salesman Problem (ATSP), and Multi Travelling Salesman Problem (MTSP).

(i) STSP: In STSP the distance between two cities is same in both the directions that mean this will result in an undirected graph.

(ii) ATSP: In ATSP the distance between two cities is not same in both directions. It is a directed graph and distance is different in both the directions.

(iii) MTSP: In a given set of nodes, let there be 'm' salesmen located at a single depot node. The remaining nodes (cities) that are to be visited are intermediate nodes. Then, the MTSP finds the tours for all 'm' salesmen, who all start and end at the place, such that each intermediate node (city) is visited exactly once and the total cost of visiting all nodes is minimized.

Solution of TSP may be of two types. The first will find the optimal solution which is almost nearer to the exact solution. This will guarantee the quality of the solution, but it is very slow. The second will find the nearest optimal solution within a reasonable time. It will not guarantee that the solution is nearer to the exact solution. So to improve the solution space and to increase the performance genetic algorithms are used to solve the travelling salesman problem which also gives the result within a reasonable amount of time.

1.2 Genetic Algorithm

Genetic algorithm [3] is an evolutionary algorithm used to solve the complex problems. It is a searching technique used in the field of problem optimization. It is based on the concept of 'survival of the fittest', which means it will select only the fit ones. Genetic Algorithm finds best or accurate solution with a large number of iterations and it can be easily parallelized to increase its computing ability because of its inherent parallelism. Hence offers great potential towards solving hard problems by minimizing the execution time.

The genetic algorithm first finds all possible solution for the problem and applies various genetic algorithm operators such as fitness evaluation, selection, crossover and mutation operators to optimize the solution for the problem. The all possible solutions to a problem are called population and each individual solution is called as chromosomes. The each value in individual chromosome is called as gene or node.

The genetic algorithm process consists of following steps:

Encoding: Before applying the genetic algorithm to any problem, a method is used to represent the chromosomes or the individual solutions so that the computer can process it. This representation method is called as encoding. There are many approaches for encoding such as Binary Encoding where the sequence of 0's and 1's are used to represent the genes, Value Encoding where the sequence of values is used, Permutation Encoding where every chromosome is a string of numbers and Tree Encoding where every chromosome is a tree of objects or nodes. After the genetic algorithm operators are applied finally the results are converted to the required format. This process is called as Decoding.

Initial Population Generation: The genetic algorithm starts by initialization of population. The initial population is generated randomly by the algorithm. It will encode all possible solutions for the problem. The initial population can be of any size.

Fitness Evaluation: The fitness evaluation phase assigns a fitness value for each individual solution which is produced in the previous step. The fitness value can be calculated in many ways. Based on the user requirement the fitness value is calculated. One common method is given by the equation $F = 1/f$, where 'F' is the fitness value and 'f' is the total path length of the individual. The fitness value shows how fit the chromosome is.

Parent Selection: The parent selection phase selects the fit parents for the further processing of genetic algorithm. This phase observes the fitness values of each individual and selects the individual with best fitness value of the next phase. There are many parent selection methods such as Elitism method, Roulette Wheel method, Tournament selection method and so on. Different selection methods select a different parent and it is given to the crossover process.

Crossover: This is the important stage in genetic algorithms. The result of genetic algorithm mainly depends on how efficient the crossover method is. So, care must be taken while selecting the crossover method. The crossover phase takes two parents and combines them to produce the new child solution. The child solution is also known as the offspring. Crossover can be one point or two point crossover and it may produce the offspring which has the same edges as the parent or it may have some new edges which are not present in the parent solutions.

Mutation: Mutation operator is applied to maintain the uniqueness between the chromosomes. It will make some changes in the solution so that it can generate new values which are not already present and may produce better results. Mutation operator does the modifications to the solution by swapping the values of a solution.

The genetic algorithm is complete if all the above genetic algorithm operators are applied. Genetic algorithm optimizes the initial solution to produce better results for a problem.

1.3 Mapping of Genetic Algorithm to TSP

The main focus of the travelling salesman problem is to find the shortest path to travel through the given cities and to minimize the cost. The genetic algorithm is applied to improve the solution space for travelling salesman problem. For travelling salesman problem generating initial population means finding all possible tours for the problem. Each chromosome represents the path travelled by the traveler. Each gene in the chromosome represents the cities to be travelled. The length of the chromosome is always the total number of cities plus one.

1.4 Parallelizing a Genetic Algorithm

The parallelization of the Genetic Algorithm is an important component of this project. The parallelization of the genetic algorithm should reduce the run time of algorithm significantly. Genetic algorithms do not fall into the Map Reduce framework due to their iterative nature. The technique demonstrates a hierarchical reduction phase in which Genetic Algorithm is reduced to Map Reduce problem. Parallel Genetic Algorithms divide the search space into many smaller pieces to find the near optimal solution. During this process the sub-optimal solutions need to avoid local minima. Different frameworks are available for parallelization such as Hadoop, Open-cl, Parallel Java, and C-MPI. Hadoop is the latest buzz words in cloud computing currently and it employs a map reduce model.

1.5 Hadoop

Hadoop is the most popular open-source implementation of the MapReduce framework. Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation. The current Apache Hadoop ecosystem consists of the Hadoop kernel, MapReduce, the Hadoop distributed file system (HDFS) and a number of related projects such as Apache Hive, HBase and Zookeeper. Its distributed file system facilitates rapid data transfer rates among nodes and allows the system to continue operating uninterrupted in case of a node failure. This approach lowers the risk of catastrophic system failure, even if a significant number of nodes become inoperative.

Hadoop cluster [4] has five main components, namely NameNode, DataNode, Secondary NameNode, JobTracker, and TaskTracker. NameNode is responsible for managing data on HDFS. DataNode, as the name suggests, stores data and interacts with NameNode. Secondary NameNode runs as a back-up for NameNode. TaskTracker communicates with client and runs the client's MapReduce jobs via TaskTracker and coordinates TaskTrackers to complete job consistently. TaskTracker is responsible for the execution of map and reduce tasks which constitute the MapReduce job. NameNode, Secondary NameNode, JobTracker are the master part of MapReduce framework and they all run on a master machine. DataNode and TaskTracker are slave components which run on slave machines. The Hadoop cluster has one NameNode, one Secondary NameNode and one JobTracker, while the cluster can have any number of DataNodes and TaskTrackers.

1.6 Map/Reduce

The MapReduce model [5] provides a parallel design pattern for simplifying application developments in distributed environments. This model can split a large problem space into small pieces and automatically parallelize the execution of small tasks on the smaller space. It was proposed by Google for easily harnessing a large number of resources in data centers to process data-intensive applications and has been proposed to form the basis of a data center computer. This model allows users to benefit from advanced features of distributed computing without worrying about the difficulty of coordinating the execution of parallel tasks in distributed environments.

MapReduce is triggered by map and reduce operations in functional languages, such as Lisp. This model abstracts computation problems through two functions: map and reduce. All problems formulated in this way can be parallelized automatically. Essentially, the MapReduce model allows users to write Map/Reduce components with functional-style code. These components are then composed as a dataflow graph with fixed dependency relationship to explicitly specify its parallelism. Finally, the MapReduce runtime system can transparently explore the parallelism and schedule these components to distributed resources for execution. All data processed by MapReduce are in the form of key/value pairs.

The execution happens in two phases. In the first phase, a map function is invoked once for each input key/value pair and it can generate output key/value pairs as intermediate results. In the second one, all the intermediate results are merged and grouped by keys. The reduce function is called once for each key with associated values and produces output values as final results.

1.7 MapReduce Model

A map function takes a key/value pair as input and produces a list of key/value pairs as output. The type of output key and value can be different from input key and value: map::(key_1,$value_1$) => list(key_2,$value_2$). A reduce function takes a key and an associated value list as input and generates a list of new values as output: reduce::(key_2,list($value_2$)) => list($value_3$).

2. Related Work

a) Varshika Dwivedi, Taruna Chauhan, Sanu Saxena, Princie Agrawal [6].

The work proposed in this paper uses TSP domain and it is solved using genetic algorithm operators. The genetic algorithm is used for the purpose of improving the solution space. The crossover is the important stage in the genetic algorithm. A new crossover method called Sequential Constructive Crossover (SCX) operator is used here. The SCX uses best edges of the parent's structure and produces the new offspring. It is compared against other existing crossover operators and it is proved that SCX results in a high quality solutions. This paper also includes a comparative study on Greedy Approach, Dynamic Programming and Genetic Algorithm for solving TSP.

b) Naveen kumar, Karambir, Rajiv Kumar [7].

In this paper the authors have done the survey on the travelling salesman problem using various genetic algorithm operators. The proposed work solves the travelling salesman problem using various genetic algorithm operators. The various methods for the genetic algorithm operators like selection methods, crossover methods and mutation methods are also mentioned in the paper.

c) Omar M. Sallabi, Younis EI-Haddad [8]

In this paper an improved genetic algorithm is proposed where the new crossover operation, population reformulation operation, multi-mutation operation, partial local optimal mutation operation, and rearrangement operations are used to solve the Traveling Salesman Problem.

A new crossover method used here is Swapped Inverted Crossover (SIC) to produce better tours. The rearrangement operation is used to find the maximum value for cost among all adjacent cities on the tour. In multi mutation operation the selected individuals are copied several times and each individual of these selected copies are mutated using different neighbors. Partial local optimal mutation operater selects the sub tour of individuals randomly. Then find the tour that produces the local minima of this sub tour and exchange it with the original sub tour. In population reformulation operation, the population is reformulated without changing its fitness value. All these operations together produce the better result.

d) Chetan Chudasama, S. M. Shah, Mahesh Panchal [9]

In this paper the travelling salesman problem is solved using genetic algorithm operators. The paper also includes a comparative study on various parent selection methods such as Roulette Wheel, Elitism and Tournament selection methods for Travelling Salesman problem.

The roulette wheel selection method selects the individual which is proportional to its fitness value, whereas in tournament selection every individual is paired with another individual in the population randomly. In the Elitism selection method the individuals are selected based on their fitness value. This paper concludes that all the three selection methods give similar solution when the population size is small but when the population size is large Elitism method gives the better result.

e) Kanchan Rani, Vikas Kumar [10]

In this paper the authors used Roulette Wheel method for the selection and also compared it with other selection method called Stochastic Universal Selection (SUS) method where N equally spaced pointers are used to select the parents. The SUS selection method gives better result when the population size is small but when the problem size increases Roulette Wheel gives the better result.

The crossover method used here is variation of Order Crossover (OX). In the new crossover method two cut points are selected. The nodes between the two cut points are copied and the rest of the nodes are selected from the second parent in relative order omitting the existing nodes and it is found that this method gives the better result than the existing crossover methods. The authors even mentioned that the Elitism method also gives the better result when the problem size is large.

f) Md. Lutful Islam, Danish Pandhare, Arshad Makhthedar, Nadeem Shaikh [11]

The work proposed in this paper solves the travelling salesman problem using genetic algorithm and it is parallelized using Hadoop Map/Reduce framework. Here three crossover methods are used such as Order Crossover, Two Point Crossover and Partially Matched Crossover. A Genetic algorithm for the travelling salesman problem is parallelized to produce the solution in a reasonable amount of time. Since the Map/Reduce framework is used where each Mapper and Reducer functions will be run in parallel on different nodes. Hence it will reduce the time taken by the algorithm. The Hadoop Map/Reduce also deals with the large problem size. The Hadoop Distributed File System (HDFS) is used for the larger data storage. The Map/Reduce functions demonstrated that it will increase processing speed and save more time. The paper also gives the overview of Hadoop and Map/Reduce.

g) A. Arananayakgi [12]

In this paper the travelling problem is solved using genetic algorithm operators to reduce the total distance and time. This is achieved by generating the fittest criteria using selection, crossover and mutation operators. The main aim of the proposed method is to produce the high quality solutions in reasonable time. So a new crossover method, the Sequential Constructive Crossover method is used. This method will selects the better edges from the parent chromosome and produce a new offspring which may have same edges as the parents or it may have new edges which is not present in the parent chromosomes. This study also proposes a binary matrix representation of chromosomes.

h) Mouhammd Al kasassbeh, Ahmad Alabadleh, Tahsen Al-Ramadeen [13]

In this paper a new crossover method is used and it is called as a Shared Crossover method. This method is simple and fast and the main aim is to reduce the execution time. This new technique depends on passing as many as possible of the shared paths between cities to the next generation with a guarantee that none of the cities will appear in the child chromosome more than once after applying the crossover method. The experimental results also show the greater reduction in the time consumption.

i) Zakir H. Ahmed [14]

This paper mainly focused on various crossover methods. Since Crossover is the important step in the genetic algorithm, efficient crossover methods should be chosen. So a comparative study of various crossover methods such as Sequential Constructive Crossover (SCX), Generalized N-point Crossover (GNX) and Edge Recombination Crossover (ERX) is done here. The experimental result shows that the sequential constructive crossover gives the high quality results for the travelling salesman problem.

j) Saloni Gupta, Poonam Panwar [15]

The work proposed in this paper solves the TSP using genetic algorithm. The Euclidian distance between each city is given in the matrix format and the initial population is generated in random order. As a next step selection, crossover and mutation operators are applied. A Two point crossover is used where two positions in the chromosomes are chosen and then replaces the gene with each other in both chromosomes randomly. The process is continued until the termination condition is satisfied.

k) Harun Rasit Er, Prof. Dr. Nadia Erdogan [16]

In this paper the travelling salesman problem is solved using parallel genetic algorithm and Map/Reduce Framework on Hadoop cluster. The main aim of this method is to reduce the execution time. The parallel genetic algorithm processes the fitness evaluation phase in parallel so that time consumption is reduced and hence execution speed is increased. The Map/Reduce framework is also used to run the genetic algorithm in parallel so that it can handle the large amount of data.

The Travelling salesman problem can be solved using different algorithms and different algorithms give different solutions. The genetic algorithm is used to improve the quality of solution so that it can produce better results. There are various kinds of methods for genetic algorithm operators and each has its own advantages and disadvantages.

The proposed method mainly concentrates on the quality of the solution and large problem size for TSP. In the works proposed above, the Hadoop Map/Reduce framework [16] is used to deal with larger problem size and sequential constructive crossover method [14] is used to produce the high quality results. By comparing all the works explained above, these two techniques gives the better results when the problem size is large and to store large amount of data Hadoop Distributed File System (HDFS) is used.

The Map/Reduce functions run on multimode Hadoop clusters to execute the genetic algorithm in parallel. In the proposed method the Sequential Constructive Crossover generates the better results by producing new edges which are not present in the parent tours. Hence, by combining the two techniques better results can be achieved.

3. Methodology

Genetic algorithm has five stages, initial population generation, fitness evaluation, parent selection, crossover and mutation.

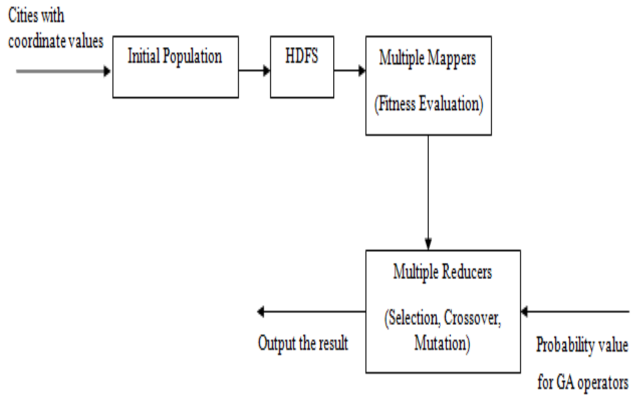

Figure 1: TSP using genetic algorithm and Map/Reduce

Stage 1: Firstly, an initial population is generated that means for Travelling Salesman Problem, population generation means finding all possible paths which visits all the cities given exactly once and return to the starting city. The individuals in the population should be different from each other in order to maintain the diversity.

Stage 2: After initial population is created the next step is the fitness evaluation of individuals. For the fitness evaluation first find the path length of all individuals. The fitness value of each individual is divided by the total path length.

Stage 3: Elitism selection method will be used here. Selection operator is applied to individuals in the population. According to the fitness values of the individuals, good ones are selected for the reproduction of the new population.

Stage 4: After selection operation, crossover between selected individuals is carried out to produce the next generation. The Sequential Constructive Crossover [14] method will be used. This method constructs an offspring using better edges on the basis of their values present in the parents' structure.

Stage 5: The last step is to mutate the individuals with a given probability for diversity of search directions.

Parallel genetic algorithm uses master-slave parallelization and multiple populations with migration parallelization method. In master-slave parallelization method the fitness evaluation stage is calculated in parallel. All the phases except fitness evaluation are performed on a master node in a sequential manner. The master node sends individuals to the slave node, which calculates the fitness values for given individuals and return the result back to the master node. Multiple populations with migration parallelization used migration operator in order to share search direction between subpopulation.

Iterative MapReduce is used to implement parallel genetic algorithm. Each phase is implemented as a MapReduce job and each job has N number of subpopulations. At the end of the job individuals are written back to Hadoop Distributed File System (HDFS).

The main components of this Mapreduce application are: Driver Class, Mapper function, Reducer function and Partitioner Class

The Driver Class is the entry point of the program. It is used to set the initial Hadoop environment parameters such as the number of map and reducer tasks, input/output format etc. The initial population is generated in this class.

Map function is used to read individuals from HDFS file system with their population identifier and group individuals according to their population identifier.

The Partitioner Class will shuffle the individuals based on their population identifiers so that all the individuals with same population identifier are sent to the same reducer class.

Reducer receives the individuals that belong to same population. Selection, crossover and mutation operations are done in this class. After these phases new population will be generated that is written back to the HDFS. After iteration has completed, all Reducers write the best individual in their sub populations to the file system and it is determined whether the convergence is obtained or not. If the solution satisfies the termination condition the algorithm is terminated. Otherwise selection, crossover and mutation stages are repeated until convergence is obtained.

4. Results and Analysis

We have used single node hadoop cluster with all the daemons are running on single machine. Hadoop is deployed on the virtual machine and cloud era's open source hadoop platform is used. For the analysis purpose an input file with only twenty cities is taken. Initially hundred populations are created and fitness value is calculated. In the next step the initial population will undergo number of generations to get the optimized result.

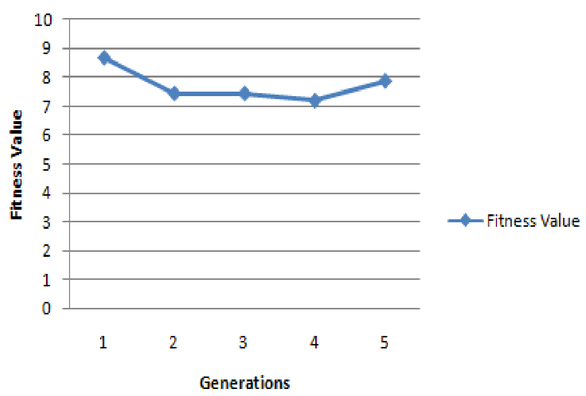

Figure 2: Optimization of fitness value as the number of generation increases.

The above graph fitness value versus generations shows that the fitness value of a chromosome decreases as it undergoes the various evolution of genetic algorithm hence optimizing the result.

5. Conclusion

Travelling salesman problem can be solved using different types of algorithms. Genetic Algorithm is used to improve the solution for Travelling Salesman Problem. The Parallel Genetic Algorithm produces the solution in less time by parallelizing the tasks. The Parallel Genetic Algorithm on MapReduce framework provides better solution for travelling salesman problem by reducing the time consumption and MapReduce framework is used to deal with larger problem size. The Elitism selection method and Sequential Constructive Crossover method is used to get better results.

6. Acknowledgment

I am very thankful to my guide Mr. Sandeep Kumar Hegde Assistant Professor, Department of Computer Science and Engineering, Mite for his cordial support, valuable information and guidance, to prepare this paper and also thankful to Prof. Dr. Nagesh H R, Head of the Department, Computer Science and Engineering, for his valuable and constructive suggestions during the planning and development of this work.

References

[1] Miguel A. S. Casquilho, "The Travelling Salesman Problem", Technical University of Lisbon, Ave. Rovisco Pais, 1049-001 Lisboa, Portugal.
[2] Rajesh Matai, Surya Prakash Singh and Murari Lal Mittal, "Traveling Salesman Problem:An Overview of Applications, Formulations, and Solution Approaches".
[3] Darrell Whitley, "A Genetic Algorithm Tutorial", Computer Science Department, Colorado University, Fort Collins, CO 80523, whitley@cs.colostate.edu.
[4] http://hadoop.apache.org
[5] J. Dean and S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, Proc. of the 6th Symposium on Operating System Design and Implementation, USA, 2004.
[6] Varshika Dwivedi, Taruna Chauhan, Sanu Saxena and Princie Agrawal, "Travelling Salesman Problem using Genetic Algorithm", National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI) 2012.
[7] Naveen kumar, Karambir and Rajiv Kumar, "A Genetic Algorithm Approach to Study Travelling Salesman Problem", Journal of Global Research in Computer Science, Volume 3, No. 3, March 2012, ISSN-2229-371X.
[8] Omar M. Sallabi and Younis EI-Haddad, "An Improved Genetic Algorithm to Solve the Traveling Salesman Problem", World Academy of Science, Engineering and Technology Volume 3, 2009.
[9] Chetan Chudasama, S. M. Shah and Mahesh Panchal, "Comparison of Parents Selection Methods of Genetic Algorithm for TSP", International Conference on Computer Communication and Networks CSI-COMNET-2011.
[10] Kanchan Rani and Vikas Kumar, "Solving Travelling Salesman Problem Using Genetic Algorithm Based on Heuristic Crossover and Mutation Operator", International Journal of Research in Engineering and Technology, ISSN(E): 2321-8843; ISSN(P): 2347-4599 Vol. 2, Issue 2, Feb 2014.
[11] Md. Lutful Islam, Danish Pandhare, Arshad Makhthedar and Nadeem Shaikh, "A Heuristic Approach for Optimizing Travel Planning Using Genetics Algorithm", International Journal of Research in Engineering and Technology eISSN: 2319-1163, pISSN: 2321-7308, Volume: 03, Issue: 01, Jan-2014.
[12] A. Arananayakgi, "Reduce Total Distance and Time Using Genetic Algorithm in Traveling Salesman Problem", International Journal of Computer Science & Engineering Technology, ISSN : 2229-3345, Vol. 5 No. 08 Aug 2014.
[13] Mouhammd Al kasassbeh, Ahmad Alabadleh and Tahsen Al-Ramadeen, "Shared Crossover Method for Solving Traveling Salesman Problem", IJICS Volume 1, Issue 6 September 2012.
[14] Zakir H. Ahmed, "Genetic Algorithm for the Traveling Salesman Problem using Sequential Constructive Crossover Operator", International Journal of Biometrics & Bioinformatics (IJBB) Volume (3): Issue (6).
[15] Saloni Gupta and Poonam Panwar, "Solving Travelling Salesman Problem Using Genetic Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 6, June 2013.
[16] Harun Rasit Er and Prof. Dr. Nadia Erdogan,"Parallel Genetic Algorithm to Solve Travelling Salesman Problem on MapReduce Framework using Hadoop Cluster", The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, March 2013.

Author Profile

Anitha Rao completed the Bachelor's Degree in Computer Science & Engineering from Visvesvaraya technological University (VTU). Currently pursuing M.Tech degree in Computer Science & Engineering at Mangalore Institute of Technology, Mangalore, Karnataka, India.

Mr. Sandeep Kumar Hegde received Master Degree in Computer Science and Engineering. He has three years of industrial experience as software engineer in Tata Consultancy Services and currently he is working as Assistant Professor in the department of computer science and engineering at Mangalore Institute of Technology, Mangalore, Karnataka, India.