ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 4, Issue 10, October 2017

Applying Map Reduction Technique with Genetic Algorithm Approach to Solve Travelling Salesman Problem

[1] Rajni, [2] Sunita Rani
Department of Computer Science
School of Computer Science And Engeering
Bhagat Phool Singh Mahila Viashawvidyalya, Khanpur Kalan

Abstract - **Travelling salesman problem is a very old mathematical problem. It models a scenario where a salesman has many cities to visit in shortest possible time. Given the distance among the cities, he must calculate the shortest route. Researchers have been working with Travelling Salesman problem for over centuries. Many models have been introduced to solve this legendary mathematical problem. In this Research Work, genetic algorithm is used to solve Travelling Salesman Problem. Genetic algorithm is a technique used for estimating computer models based on methods adapted from the field of genetics in biology. To use this technique, one encodes possible model behaviors into "genes". We designed a new crossover technique to increase the efficiency of crossover operator of GA for TSP. We also apply map reduce algorithm to efficiently solve the TSP.**

Keywords:--- **Travelling Salesman Problem, Map Reduce, Genetic Algorithm.**

1. INTRODUCTION

The Traveling Salesman Problem (TSP) is a classical combinatorial optimization problem, which is simple to state but very difficult to solve. The problem is to find the shortest tour through a set of N vertices so that each vertex is visited exactly once. This problem is known to be NP-hard, and cannot be solved exactly in polynomial time. Apart from its theoretical approach, the TSP has many applications. Some typical applications of TSP include vehicle routing, computer wiring, cutting wallpaper and job sequencing. The main application in statistics is combinatorial data analysis, e.g., reordering rows and columns of data matrices or identifying clusters. The NP-completeness of the TSP already makes it more time efficient for small-to-medium size TSP instances to rely on heuristics in case a good but not necessarily optimal solution is sufficient.

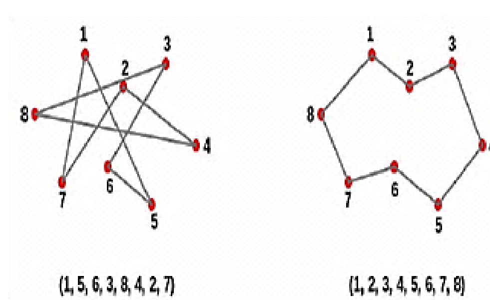
The origin of Travelling Salesman Problem (TSP) dates back to 1759 when the first instance of the travelling salesman problem was from Euler whose problem was to move a knight to every position on a chess board exactly once [1]. Then in 1832 mathematician W.R Hamilton and Thomas Kirkman formulated it. The travelling salesman first gained fame in a book written by German salesman BF Voigt in 1832 on how to be a successful travelling salesman. Though he did not mention TSP by name but suggested that to cover as many locations as possible not visiting any location twice is the key factor of scheduling of a tour. The standard or symmetric travelling salesman problem can be stated mathematically as follows:

Given a weighted graph G = (V, E); where the weight cij on the edge between nodes i and j is a non-negative value, find the tour of all nodes that has the minimum total cost.

In this paper a hybrid solution of genetic algorithm and map reduce algorithm for solving the classical travelling salesman problem is provided.

II. TYPES OF TSP

Considerable number of variations exists for the TSP. Some of these are; asymmetric TSP, multiple TSP, clustered TSP [2] etc. In symmetric TSP (STSP) each city pair have the same distance for both directions. We represent an example of STSP in Figure 1. Figure 1 (a) and (b) show a non-optimal solution and the optimal solution to the example STSP respectively.



IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

Figure 1.1: A visualization of symmetric TSP.
(a) a twisted bad route (b) the optimum route

Asymmetric TSP is similar to the symmetric TSP where the distance between two cities is not equal or no path exists for one direction. In multiple TSP, salesmen collaborate to achieve a target solution. In clustered TSP, there are clusters composed of adjacent cities. These clusters behave as a single city to decrease the number of cities and therefore improve running time. We focus on STSP in this thesis.

III. RELATED WORK

In history, two types of algorithms are designed for this challenge; exact algorithms and heuristic algorithms. Exact algorithms apply brute force search techniques and try to minimize the search space with specific constraints. They may find the optimal tour but the time complexity is not satisfactory especially when the number of cities is large, i.e. N > 1000. Linear programming is an example of exact algorithms. Concorde TSP Solver is arguably the best program [2] to solve TSP using linear programming concepts.

Heuristic algorithms traverse the search space randomly and try to approach the global optimum. For these approaches, we are less likely to reach the global optimum as exact algorithms do. On the other hand, time complexities of the heuristic algorithms are far better than the exact algorithms. Ant colony optimization and simulated annealing algorithms are well-known examples of heuristic approaches. They can compete with genetic algorithms and can also perform as internal GA elements.

IV. GENETIC ALGORITHM

Genetic algorithm [3][4] is a technique used for estimating computer models based on methods adapted from the field of genetics in biology. To use this technique, one encodes possible model behaviors into "genes". After each generation, the current models are rated and allowed to mate and breed based on their fitness. In the process of mating, the genes are exchanged, crossovers and mutations can occur. The current population is discarded and its offspring forms the next generation.

In a way of using GA in solving TSP, The following methods are used;

● Simulated Annealing, based on natural annealing processes.

● Artificial Neural Networks, based on processes in central nervous systems.

● Evolutionary Computation based on biological evolution processes.

The algorithms inspired by Evolutionary Computation are called evolutionary algorithms. These evolutionary algorithms may be divided into the following branches: genetic algorithms (Holland 1975), evolutionary programming (Fogel 1962), evolution strategies (Bremermann et al. 1965), classifier systems (Holland 1975), genetic programming (Koza 1992) and other optimization algorithms based on Darwin's evolution theory of natural selection and "survival of the fittest". In this thesis, we will only examine one of the above mentioned types of algorithms: genetic algorithms, although some of the exposed mutation operators have been developed in relation to evolutionary programming.
We consider these algorithms in combination with the Travelling Salesman Problem (TSP). The TSP objective is to find the shortest route for a travelling salesman who, starting from his home city has to visit every city on a given list precisely once and then return to his home city. The main difficulty of this problem is the immense number of possible tours: (n-2)! /2 for n cities.

Evolutionary algorithms are probabilistic search algorithms which simulate natural evolution. Holland (1975) introduced genetic algorithms. In these algorithms the search space of a problem is represented as a collection of individuals. These individuals are represented by character strings which are often referred to as chromosomes. The purpose of using genetic algorithm to find the individual from the search space with the best "genetic material". The quality of an individual is measured with an evaluation function.

A. Control parameters

These are the parameters that govern the GA search process [6]. Some of them are:

(a) Population size: - It determines how many chromosomes and thereafter, how much genetic material is available for use during the search. If there is too little, the search has no chance to adequately cover the space. If there is too much, the GA wastes time evaluating chromosomes.

(b) Crossover probability: - It specifies the probability of crossover occurring between two chromosomes.

(c) Mutation probability: - It specifies the probability of doing bit-wise mutation.

(d) Termination criteria: - It specifies when to terminate the genetic search.

B. Structure of genetic algorithms

GAs may be summarized as follows:

GA ( )

{

Initialize random population;

Evaluate the population;

Generation = 0;

While termination criterion is not satisfied

{

Generation = Generation + 1;

Select good chromosomes by reproduction procedure;

Perform crossover with probability of crossover (Pc);

Select fitter chromosomes by survivor selection procedure;

Perform mutation with probability of mutation (Pm);

Evaluate the population;

}

}

The algorithm consists of the following fundamental steps

Initialization: Chromosomes are randomly created. At this point it is very important that the population is diverse otherwise the algorithm may not produce good solutions.

Evaluation: Each chromosome is rated how well the chromosome solves the problem at hand. A fitness value is assigned to each chromosome.

Selection: Fittest chromosomes are selected for propagation into the future generation based on how fit they are.

Recombination: Individual chromosomes and pairs of chromosomes are recombined and modified and then put back in the population.

V. MAP REDUCE

Map Reduce is programming model which has proven to be very effective to run a query on big data. Generally speaking, it works like this [5]:

▪ The data is partitioned across multiple computer nodes.

▪ A map function runs on every partition and returns a result.

▪ A reduce function reduces 2 results into one result. It is continuously run until only a single result remains.

The figure 2 below shows map reduction on TSP. Given the cities, we can scenario into four possible co-ordinates and solve them separately.

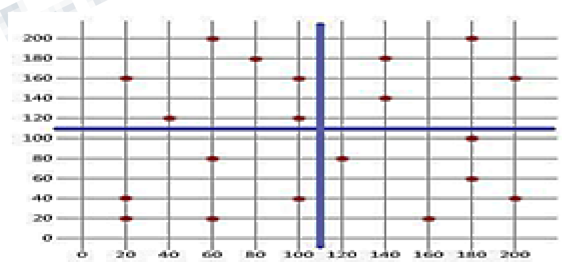


Figure 2: Map reduction

As, TSP is a NP- hard problem, researchers often tried to reduce in complexity by dividing it into pieces of several problems. It is established that Map Reduce can solve any planning problem optimally.

VI. FINAL IMPLEMENTATION

Below given figure shows thatt how we finally implement the complete procedure

IFERP
connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 10, October 2017

.

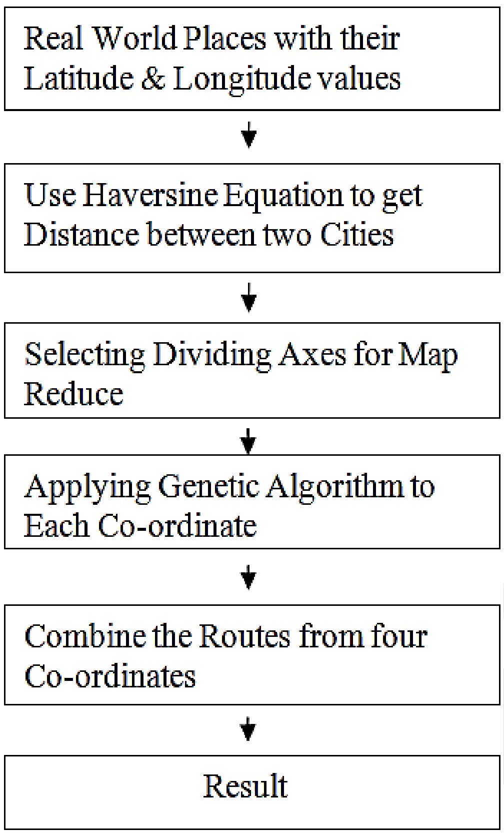
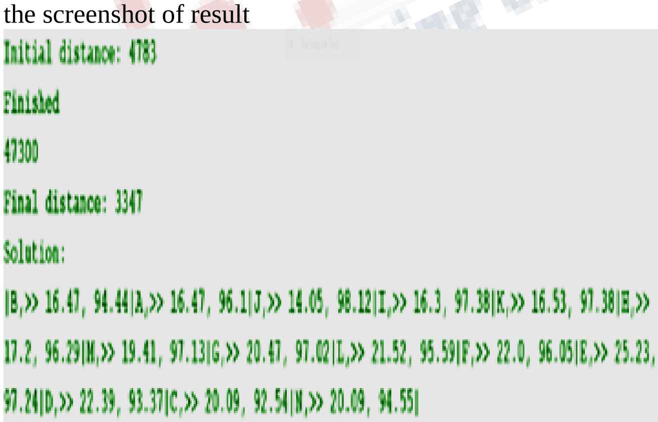


Figure 3: Final Impelmentation Steps

V. RESULT

The result we found from using the above defined methodology we get optimal result. Below we are showing the screenshot of result



Initial distance: 4783

Finished

47300

Final distance: 3347

Solution:

|B,>> 16.47, 94.44|A,>> 16.47, 96.1|J,>> 14.05, 98.12|I,>> 16.3, 97.38|K,>> 16.53, 97.38|H,>> 17.2, 96.29|M,>> 19.41, 97.13|G,>> 20.47, 97.02|L,>> 21.52, 95.59|F,>> 22.0, 96.05|E,>> 25.23, 97.24|D,>> 22.39, 93.37|C,>> 20.09, 92.54|N,>> 20.09, 94.55|

Figure 4: Screen shot of result

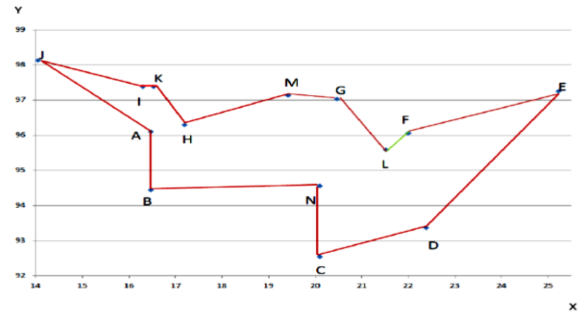


Figure 5: Graph of Result

When we do implementation we found that we get different result from every different starting node to different ending node. This shows by following table.

SNo.	Starting Node	Ending Node	Initial Distance	Final Distance
1	F	E	5461	3384
2	L	N	5181	3487
3	K	H	4828	3450
4	E	F	5427	3592
5	F	L	5464	3388
6	D	E	4966	3611
7	C	N	5580	3347
8	C	G	5505	3611
9	B	A	5302	3347
10	C	B	4794	3450

Table 1: Variation due to no. of cities

Here it is clearly shown that different combination of paths has different costs. So we choose that path which we have minimum cost.

When we change no. of nodes then also variations in result occur. This variation is shown in following table.

SNo.	Nodes	Distance	Execution Time
1	14	4827	0.061
2	16	5633	0.068
3	18	6924	0.077
3	20	8121	0.100

Table 2: Variation due to starting node

ALGORITHM	TIME	DISADVANTAGE
Genetic with map reduce	Less than 1 sec.	More complex
Genetic	Exponential time	Not so much optimal
Greedy	5 sec for 15 cities	No Accuracy
Dynamic programming	9 sec for 15 cities	Expensive for memory and time

Table 3: Comparison of Various Algorithms[15]

VII. CONCLUSION

Genetic algorithms appear to find good solutions for the traveling salesman problem, however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. It seems that the methods that use heuristic information or encode the edges of the tour (such as the matrix representation and crossover) perform the best and give good indications for future work in this area.

Overall, it seems that genetic algorithms have proved suitable for solving the traveling salesman problem. As yet, genetic algorithms have not found a better solution to the traveling salesman problem than is already known, but many of the already known best solutions have been found by some genetic algorithm method also.

In this paper, we designed a new crossover technique to increase the efficiency of crossover operator of GA for TSP. Then we divided the map of given city into four equal co-ordinates such that no city rest on the axes using map reduce algorithm. Then GA was run for each four co-ordinates and not only the best path, but also second or third best paths were recorded. Then we used simple combination to determine the 4 paths of four co-ordinates & merging in between them.

REFERENCES

[1] Zbigniew Michalewicz. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, 2nd edition, 1994.

[2] Concord TSP Solver. http://www.math. uwaterloo. ca/tsp/concorde.html.

[3] Seniw, D., A Genetic algorithm for the Travelling Salesman Problem, MSc Thesis, University of North Carolina, at Charlotte. http://www.heatonresearch.com / articales/ 65/page1. html. 1996.

[4] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[5] http:// Amsterdam optimization .com / pic / tspmip . png

[6] Davis, L. (ed.) (1991). Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold.

[7] Grefenstette, J. J. (ed.) (1987a). Genetic Algorithms and Their Applications: Proceedings of the Second International Conference. Hillsdale, New Jersey: Lawrence Erlbaum.

[8] Homaifar, A. & Guan, S. (1991). A New Approach on the Traveling Salesman Problem by Genetic Algorithm. Technical Report, North Carolina A&T State University.

[9] Kylie Bryant, Arthur Benjamin, Advisor, “Genetic Algorithms and the Travelling Salesman Problem”, Department of Mathematics, December 2000.

[10] Adewole Philip, Akinwale Adio Taofiki, Otunbanowo Kehinde “A Genetic Algorithm for Solving Travelling Salesman Problem”,(IJACSA) International Journal of Advanced Computer

Science and Applications, Vol. 2, No.1, January 2011.

[11] Abdollah Homaifer, Shanguchuan Guan, Gunar E. Liepins "Schema Analysis of the Travelling Salesman Problem Using Genetic Algorithms".

[12] Milena Karova Vassil Smarkov Stoyan Penev,"Genetic operators crossover and mutation in solving the TSP problem", International Conference on Computer Systems and Technologies - CompSysTech‟ 2005.

[13] Fogel, D. B. (1993). Applying Evolutionary Programming to Selected Traveling Salesman Problems. Cybernetics and Systems 24: 27–36.

[14] H. Braun, "On Solving Travelling Salesman Problem by Genetic Algorithm", in Parallel Problem- Solving from Nature, Lecture Notes in Computer Science 496, H.P. Schwefel and R. Manner Eds, Springer-Verlag, app. 129-133.

[15] Varshika Dwivedi Taruna Chauhan Sanu Saxena Princie Agrawal Travelling Salesman Problem using Genetic Algorithm National Conference on Development of Reliable Information Systems, Techniques and Related Issues (DRISTI) Proceedings published in International Journal of Computer Applications® (IJCA) 2012