Practical Training Report

A report submitted in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

Subject Code: - **ET20723**

VII Semester

On

"NETWORK OPTIMIZATION"

Under taken at

NATIONAL INSTITUTE OF TECHNOLOGY RAIPUR

Submitted By-

NITISH KUMAR

Roll No: 16116058

Department of Electronics and Communication Engineering

National Institute of Technology Raipur

G.E. Road, Raipur, CG – 492001, INDIA

Summer Internship Approval Certificate

Department of Electronics and Communications Engineering

National Institute of Technology Raipur

This is to certify that the internship report entitled "Network Optimization" submitted by NITISH KUMAR (16116058) is approved for Summer Internship 2019 program is work done by him from **20/05/2019** to **05/07/2019**, at "NIT RAIPUR" and submitted during 2019–2020 academic year.

Name of Committee Member	**Dr. Alok Naugarhiya**	**Dr. B. Acharya**
Committee Members	**Departmental T&P Coordinator**	**Head of Department, E&C**

ACKNOWLEDGMENT

I would like to express my sincere gratitude to, Mr. Sujay Chakraborty, Assistant Professor, Department of Electronics and Telecommunication Engineering, NIT Raipur for suggesting the problem and providing guidance throughout my research work. His valuable guidance and constant encouragement were of great help in carrying out my internship on “Network optimization using genetic algorithm” successfully.

I also wish to express my profound thanks to my friends and my team member who helped me directly or indirectly. The guidance and support received from all of those who contributed was vital for successful completion of this report.

ABSTRACT

Network optimization is technology used for improving network performance for a given environment. It is considered an important component of effective information systems management. Network optimization plays an important role as information technology is growing at exponential rates with business users producing large volumes of data and thus consuming larger network bandwidths. If proper network optimization is not in place, the continuous growth can add strain to the network architecture of the concerned environment or organization.

The goal of any network optimization is with the given set of constraints; ensuring an optimal network design with lowest cost structure and free flow of data. Network optimization should be able to ensure optimal usage for system resources, improve productivity as well as efficiency for the organization. Network optimization looks at the individual workstation up to the server and the tools and connections associated with it. Large organizations make use of teams of network analysts to optimize networks. Network optimization often makes use of traffic shaping, redundant data elimination, data caching and data compression and streamlining of data protocols. Network optimization must be able to boost network efficiency without acquiring additional or expensive hardware or software.

There are many benefits of network optimization. It can help in faster data transfers including bulk data transfer, disaster recovery capabilities, reducing bandwidth expenses and also improving response times for interactive applications like databases and software applications. It also improves the performance of applications with better bandwidth and helps in maximizing network speeds between remote locations.

In this report we have tried to study about various networking algorithms that that could help us to optimize the shortest path for any given network. Few of the algorithms were analysed for the same which include:

1. Genetic algorithm
2. Dijkstra's algorithm
3. Travelling salesman problem
    - Tabu search
    - Simulated annealing

The major portion of this report is dedicated to analysis and functioning of genetic algorithm. What are the various steps involved in it. In this report we first applied Genetic Algorithm (GA) for finding shortest path routing and then studied about Dijkstra's Algorithm (DA) that finds the shortest path between any pair of nodes in the network. Both the Algorithms provide approximately the same solution. The results affirmed the potential of Genetic Algorithm. GA has the potential to replace DA in finding the shortest path for Network Topologies. Simulation results are performed for both Algorithms on MATLAB. We also dealt with travelling salesman problem and then applied the algorithm on a relatively large network to understand how are the algorithms implemented for real life applications.

INDEX

1. Internship Objectives

- To learn about the necessity of network optimization.
- To learn what are the constraints required to optimize a network.
- To learn about various network optimization algorithms
- To understand various tools and schemes required for implementation of algorithms.
- To get aware of how network optimization is implemented in real life applications.
- To learn Network Optimization using Genetic Algorithm.
- To understand Genetic Algorithm and its implementation.
- To find out shortest path using Dijkstra's Algorithm.
- To understand the Travelling Salesman problem.
- Learn and understand the methods involved in Travelling Salesman problem.

2. Weekly Overview of Internship Activities

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
1ST WEEK	20/05/2019	MONDAY	Basic introduction to Network and Antennas.
	21/05/2019	TUESDAY	Studied about Network Theory, Incidence matrix, Adjacency matrix
	22/05/2019	WEDNESDAY	Studied about network optimization, shortest path routing and various networking algorithms
	23/05/2019	THRUSDAY	Analysis of research paper related to network optimization using genetic algorithm
	24/05/2019	FRIDAY	Analysis of Research Paper related to genetic algorithm for shortest path optimization.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
2ND WEEK	27/05/2019	MONDAY	Analysis of Genetic algorithm and developed understanding about it.
	28/05/2019	TUESDAY	Learnt about various steps involved in genetic algorithm.
	29/05/2019	WEDNESDAY	Studied about basic MATLAB tools for implementation and another codes in C/C++.
	30/05/2019	THRUSDAY	Learning MATLAB software basic and develop understanding about MATLAB code.
	31/05/2019	FRIDAY	Reanalysis of Genetic algorithm code for MATLAB, its implementation and result.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
3RD WEEK	03/06/2019	MONDAY	Analysis of Dijkstra's algorithm.
	04/06/2019	TUESDAY	Understanding the basic steps involved in dijkstra's algorithm.
	05/06/2019	WEDNESDAY	Implementation of matlab code for dijkstra's algorithm
	06/06/2019	THRUSDAY	Analysis of results obtained based on different input constraints..
	07/06/2019	FRIDAY	Comparison of result and understanding the similarity with genetic algorithm.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
4TH WEEK	10/06/2019	MONDAY	Analysis of Travelling Salesman Problem
	11/06/2019	TUESDAY	Analysis of codes of tsp with tabu search and simulated annealing method.
	12/06/2019	WEDNESDAY	Error analysis and error correction in code for TSP tabu search.
	13/06/2019	THRUSDAY	Re-implementation of TSP Tabu search Method
	14/06/2019	FRIDAY	Analysis of the results obtained through implementation of code.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
5TH WEEK	17/06/2019	MONDAY	Analysis of code for simulated annealing
	18/06/2019	TUESDAY	Error correction and analysis of code for TSP simulated annealing method.
	19/06/2019	WEDNESDAY	Re-implementation of TSP Simulated Annealing Method
	20/06/2019	THRUSDAY	Analysis of the result obtained.
	21/06/2019	FRIDAY	Comparison of results of tabu search and simulated annealing.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
6TH WEEK	24/06/2019	MONDAY	Analysis of real life application of travelling salesman problem.
	25/06/2019	TUESDAY	Application of tsp on a relatively large network (54 node network set)
	26/06/2019	WEDNESDAY	Generation of adjacency matrix for 54 node dataset
	27/06/2019	THRUSDAY	Implementation of code of simulated annealing and tabu search for 54 node network.
	28/06/2019	FRIDAY	Analysis of the result obtained for 54 node network.

	DATE	DAY	NAME OF THE TOPIC / MODULE COMPLETED
7TH WEEK	01/07/2019	MONDAY	Comparison of the results of various algorithms
	02/07/2019	TUESDAY	Analysis of the code implemented for each algorithm , the similarities and their difference.
	03/07/2019	WEDNESDAY	Verification of the results obtained
	04/07/2019	THRUSDAY	Drawbacks and advantages of algorithms used
	05/07/2019	FRIDAY	Methods for improvising the results obtained.

3. Introduction

3.1 Network Optimization for shortest path problem:

There are many terms when it comes to route optimization and route planning for any network. Routing is the process of finding the best path between two or more locations with a fixed order in a road or rail network. The criterion according to which a path is the best can vary. You may be looking for the shortest path (by distance), the fastest (by travel time), but also the most scenic or the safest path. Anything is possible as long as it can be specified by some quantity, a *generalized cost*, to each road segment, and looking for the *least-cost path*.

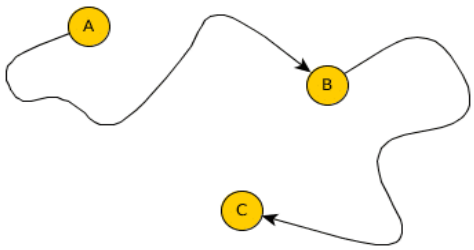

Figure 1

Other terms like journey planning, trip planning or route planning are frequently used, sometimes also itinerary planning. There are a number of algorithms to solve this problem. The examples of such algorithms includes genetic algorithm, dijkstra's algorithm.

When we use the term route optimization, we mean solving vehicle routing problems (VRP) and travelling salesman problems (TSP).The vehicle routing problem (VRP) is a combinatorial optimization and integer programming problem which asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?". It generalises the well-known travelling salesman problem (TSP).

3.2 Shortest Path Routing:

Most people are aware of the shortest path problem, but their familiarity with it begins and ends with considering the shortest path between two points, A and B. However, for computer scientists

this problem takes a different turn, as different algorithms may be needed to solve the different problems.

For simplicity, shortest path algorithms operate on a graph, which is made up of vertices and edges that connect them. A graph may be directed, undirected, weighted, and more. It's these distinctions that determine which algorithm will work better than another for certain graph types.

The shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. In other words, when we have to find a path with minimum cost to go from a place to another place which there are a number of intermediate points in between to travel to with different costs, we are dealing with the shortest path problems. It should be noted that the phrase "shortest path" here does not necessarily mean physically shortest distance, but a path with minimum weight which can be measured in, say, time or monetary cost.

Shortest path algorithms have various uses, most notable being Route planning software such as Google Maps make use of shortest path algorithms to generate your routes.

As there are a number of different shortest path algorithms, we've gathered the most important to help you understand how they work and which is the best.

4. Genetic Algorithm

Genetic algorithm is a heuristic search method used in artificial intelligence and computing. It is used for finding optimized solutions to search problems based on the theory of natural selection and evolutionary biology. Genetic algorithms are excellent for searching through large and complex data sets. They are considered capable of finding reasonable solutions to complex issues as they are highly capable of solving unconstrained and constrained optimization issues.

Genetic algorithms are widely used in many fields such as robotics, automotive design, optimized telecommunications routing, engineering design and computer-aided molecular design.

Genetic Algorithms (GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct

the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection : which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

4.1 Foundation of Genetic Algorithms:

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosome of the population. Following is the foundation of GAs based on this analogy –

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from "fittest" parent propagate throughout the generation that is sometimes parents create offspring which is better than either parent.
4. Thus each successive generation is more suited for their environment.

Search space:

The population of individuals are maintained within search space. Each individual represent a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).

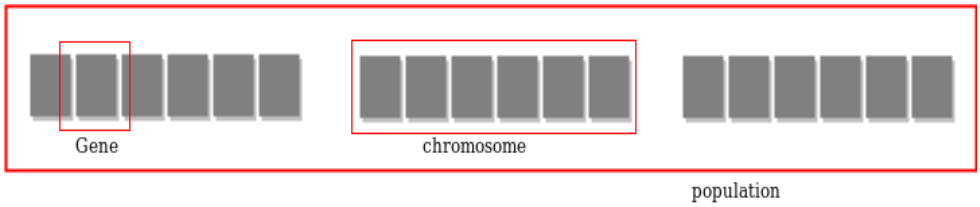

Figure 2

Fitness Score:

A Fitness Score is given to each individual which shows the ability of an individual to "compete"**.** The individual having optimal fitness score (or near optimal) are sought.

The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores. The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce better offspring by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

Each new generation has on average more "better genes" than the individual (solution) of previous generations. Thus each new generation have better "partial solutions**"** than previous generations. Once the offspring produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

4.2 Operators of Genetic Algorithms

Once the initial generation is created, the algorithm evolves the generation using following operators: –

1) Selection Operator: The idea is to give preference to the individuals with good fitness scores and allow them to pass their genes to the successive generations.

2) Crossover Operator: This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring).

For example –

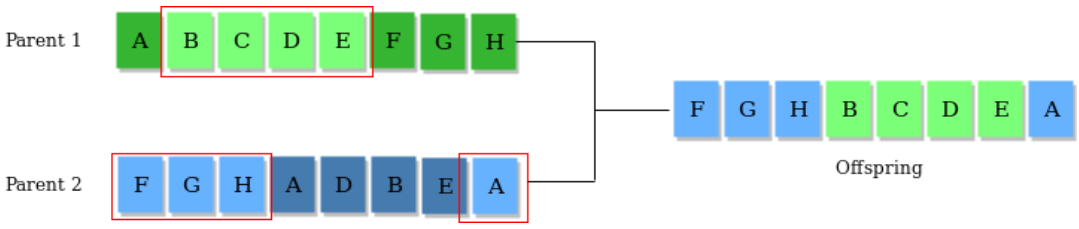

Figure 3

3) Mutation Operator: The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For example –

Figure 4

4) Fitness score is the number of characters which differ from characters in target string at a particular index. So individual having lower fitness value is given more preference

5) Terminating Condition: terminating condition is a predefined number of iterations. Because, in the network topology, the goal is not to find the global optimum, but to find a path with a reasonable cost in a limited time

4.3 Study of shortest path using Genetic algorithm:

Following result has been obtained on applying the code of genetic algorithm for the given input constraints:

Code reference: https://github.com/amarjitdhillon/Find-Shortest-Path-using-Generic-Algorithm-in-MATLAB

INPUT:

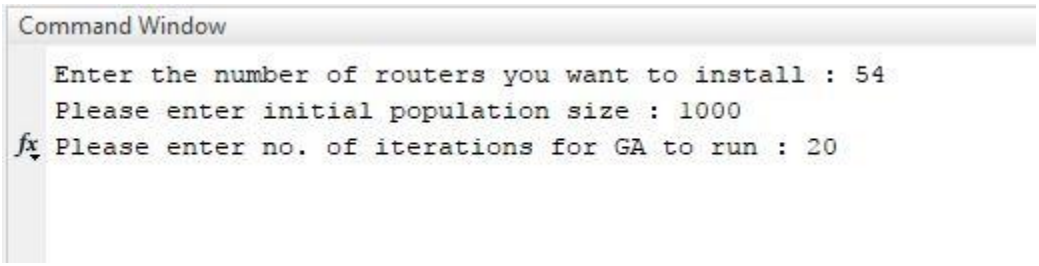

Figure 5

OUTPUT:

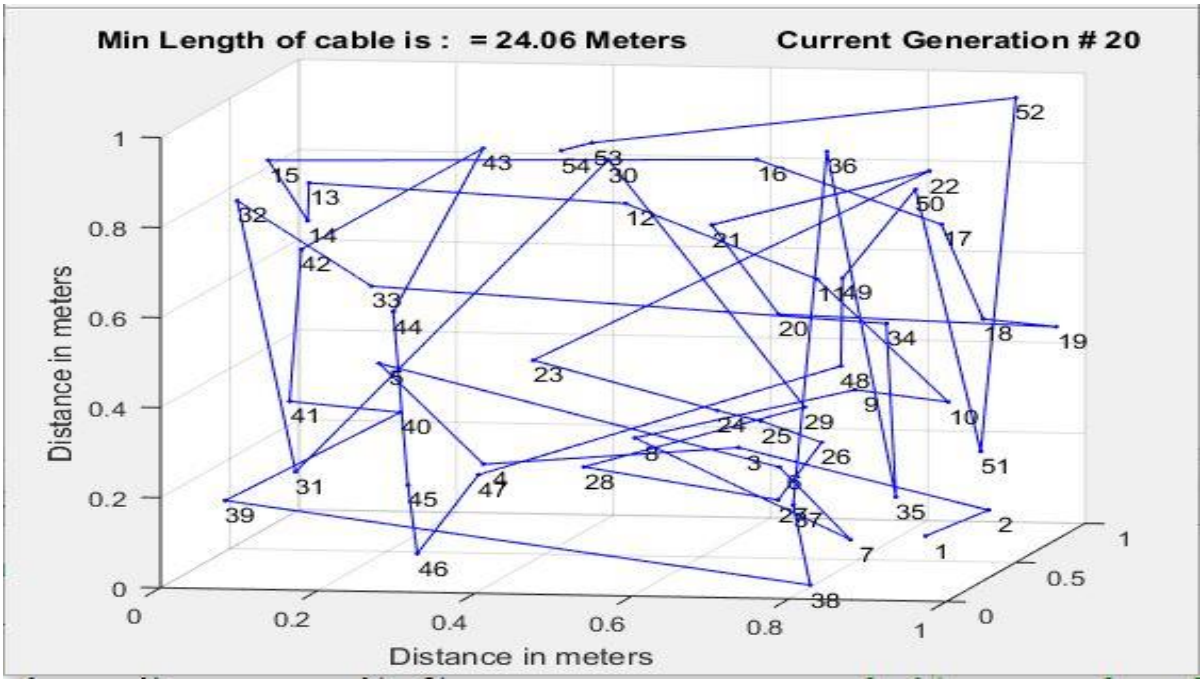

Figure 6

5. Dijkstra's Algorithm for Shortest Path Routing

Dijkstra's Algorithm stands out from the rest due to its ability to find the shortest path from one node to every other node within the same graph data structure. This means, that rather than just finding the shortest path from the starting node to another specific node, the algorithm works to find the shortest path to every single reachable node – provided the graph doesn't change.

The algorithm runs until all of the reachable nodes have been visited. Therefore, you would only need to run Dijkstra's algorithm once, and save the results to be used again and again without re-running the algorithm – again, unless the graph data structure changed in any way.

In the case of a change in the graph, you would need to rerun the graph to ensure you have the most updated shortest paths for your data structure.

Let's take a routing example , if you want to go from A to B in the shortest way possible, but you know that some roads are heavily congested, blocked, undergoing works, and so on, when using Dijkstra, the algorithm will find the shortest path while avoiding any edges with larger weights, thereby finding you the shortest route.

The Dijkstra's algorithm is an algorithm that can find the shortest path for a single source graph. The Dijkstra's algorithm adopts the concept of greedy approach. First, we will have to initialize the temporary "estimated time needed" from the starting point (the source which is MW in this case) of every checkpoint.

Obviously, the time needed to go to the starting point from the starting point is 0. However we do not know the time needed to go to other checkpoint from the starting point yet so we may take them as infinity. Then we can choose a checkpoint which take the minimum time to go from the starting point (choose any one if there are more than 1 checkpoint needed same minimum time to go to) and update the neighbouring checkpoints' "estimated time needed", i.e. if the time needed to go to next checkpoint(CPnext) via the chosen checkpoint(CPchosen) is less, we will go to CPnext via the CPchosen instead and replace the time needed to go to CPnext by the sum of the time needed to go to the CPchosen from starting point and the time needed to go to CPnext from CPchosen. And we repeat the above process until all checkpoints are considered.

So, we know that the Dijkstra's algorithm is useful in solving this kind of shortest path problem. Then can we apply the Dijkstra's algorithm to every situation to solve the shortest path problems? Unfortunately, the answer is no. The Dijkstra's algorithm will fail to find the path with minimum weight if the graph consists of some negative weight. At first glance, this situation seems impossible to happen as you may wonder since there will never be a path that will reduce the time needed for the journey. Yet, according to Robert Sedgewick, "Negative weights are not merely a mathematical curiosity; they arise in a natural way when we reduce other problems to shortest-paths problems"

5.1 Study of Dijkstra's algorithm:

Following result has been obtained on applying on applying code (http://www.mathworks.com/matlabcentral/fileexchange/20025-advanceddijkstras-minimum-path-algorithm) of Dijkstra's algorithm for the given input constraints:

INPUT:

```
>> dijkstra2([0 10 3 2 0;0 0  1 2 0;0 4  0 8 2;0 0  0 0 7;0 0  0 9 0])
```

Figure 7

OUTPUT:

```
ans =

     0     7     3     2     5
   Inf     0     1     2     3
   Inf     4     0     6     2
   Inf   Inf   Inf     0     7
   Inf   Inf   Inf     9     0
```

Figure 8

6. Travelling Salesman Problem

The **Traveling Salesman Problem** (often called **TSP**) is a classic algorithmic problem in the field of computer science and operations research. It is focused on optimization. In this context, better solution often means a solution that is cheaper, shorter, or faster. TSP is a mathematical problem. It is most easily expressed as a graph describing the locations of a set of nodes

The traveling salesman problem consists of a salesman and a set of cities. The salesman has to visit each one of the cities starting from a certain one (e.g. the hometown) and returning to the same city. The challenge of the problem is that the traveling salesman wants to minimize the total length of the trip. The traveling salesman problem can be described as follows:

$$\begin{aligned}\text{TSP} = \{(G, f, t):\ & G = (V, E) \text{ a complete graph,} \\ & f \text{ is a function } V \times V \rightarrow \mathrm{Z}, \\ & t \in \mathrm{Z}, \\ & G \text{ is a graph that contains a traveling salesman tour with cost that does not} \\ \text{exceed } t\}.\end{aligned}$$

TSP can be modelled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's weight. It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once. Often, the model is a complete graph (*i.e.* each pair of vertices is connected by an edge). If no path exists between two cities, adding an arbitrarily long edge will complete the graph without affecting the optimal tour.

6.1 Tabu Search for TSP

Tabu Search is a heuristic that, if used effectively, can promise an efficient near-optimal solution to the TSP. The basic steps as applied to the TSP are presented below:

1. Solution Representation: A feasible solution is represented as a sequence of nodes, each node appearing only once and in the order it is visited. The first and the last visited nodes are fixed to 1.

The starting node is not specified in the solution representation and is always understood to be node 1.

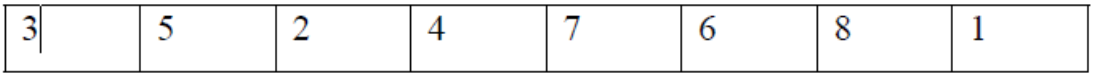

Figure 9: Solution Representation

2. Initial Solution: A good feasible, yet not-optimal, solution to the TSP can be found quickly using a greedy approach. Starting with the first node in the tour, find the nearest node. Each time find the nearest unvisited node from the current node until all the nodes are visited.

3. Neighborhood: A neighborhood to a given solution is defined as any other solution that is obtained by a pair wise exchange of any two nodes in the solution. This always guarantees that any neighborhood to a feasible solution is always a feasible solution (i.e, does not form any sub-tour). If we fix node 1 as the start and the end node, for a problem of N nodes, there are C(N-1,2) such neighborhoods to a given solution. At each iteration, the neighborhood with the best objective value (minimum distance) is selected.

4.

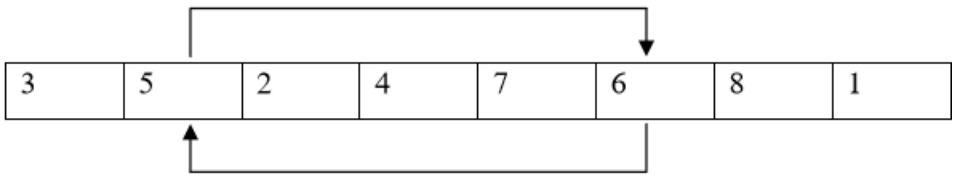

Figure 10: Neighborhood solution obtained by swapping the order of visit of cities 5 and 6

5. Tabu List: To prevent the process from cycling in a small set of solutions, some attribute of recently visited solutions is stored in a Tabu List, which prevents their occurrence for a limited period. For our problem, the attribute used is a pair of nodes that have been exchanged recently. A Tabu structure stores the number of iterations for which a given pair of nodes is prohibited from

exchange. Aspiration criterion: Tabus may sometimes be too powerful: they may prohibit attractive moves, even when there is no danger of cycling, or they may lead to an overall stagnation of the searching process [3]. It may, therefore, become necessary to revoke tabus at times. The criterion used for this to happen in the present problem of TSP is to allow a move, even if it is tabu, if it results in a solution with an objective value better than that of the current best-known solution.

7. Diversification: Quite often, the process may get trapped in a space of local optimum. To allow the process to search other parts of the solution space (to look for the global optimum), it is required to diversify the search process, driving it into new regions. This is implemented in the current problem using "frequency based memory". The use of frequency information is used to penalize nonimproving moves by assigning a larger penalty (frequency count adjusted by a suitable factor) to swaps with greater frequency counts. This diversifying influence is allowed to operate only on occasions when no improving moves exist. Additionally, if there is no improvement in the solution for a pre-defined number of iterations, frequency information can be used for a pair wise exchange of nodes that have been explored for the least number of times in the search space, thus driving the search process to areas that are largely unexplored so far.

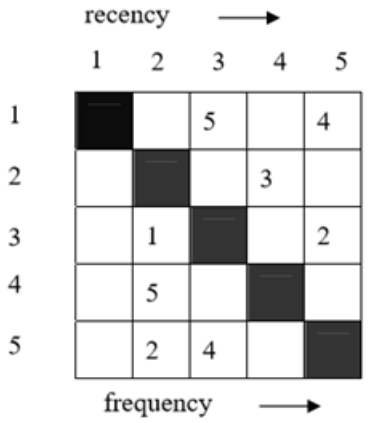

Figure 11: Tabu Structure

8. Termination criteria: The algorithm terminates if a pre-specified number of iterations is reached

6.2 Simulated Annealing for TSP

The basic steps of Simulated Annealing (SA) applied to the TSP are described below. The solution representation and the algorithm for initial solution for the SA are same as that for Tabu Search described above.

1. Neighborhood: At each step, a neighborhood solution is selected by an exchange of a randomly selected pair of nodes. The randomly generated neighbor solution is selected if it improves the solution else it is selected with a probability that depends on the extent to which it deteriorates from the current solution.

2. Termination criteria: The algorithm terminates if it meets any one of the following criteria: a. It reaches a pre-specified number of iterations. b. There is no improvement in the solution for last pre-specified number of iterations. c. Fraction of neighbor solutions tried that is accepted at any temperature reaches a pre-specified minimum. The maximum number of iterations is kept large enough to allow the process to terminate either using criterion b or c.

Thesis referred: A Comparative Study of Tabu Search and Simulated Annealing for Traveling Salesman Problem.

7. Real life application (using a 54 node network):

Image Source: Intel Berkeley research lab (http://db.csail.mit.edu/labdata/labdata.html)

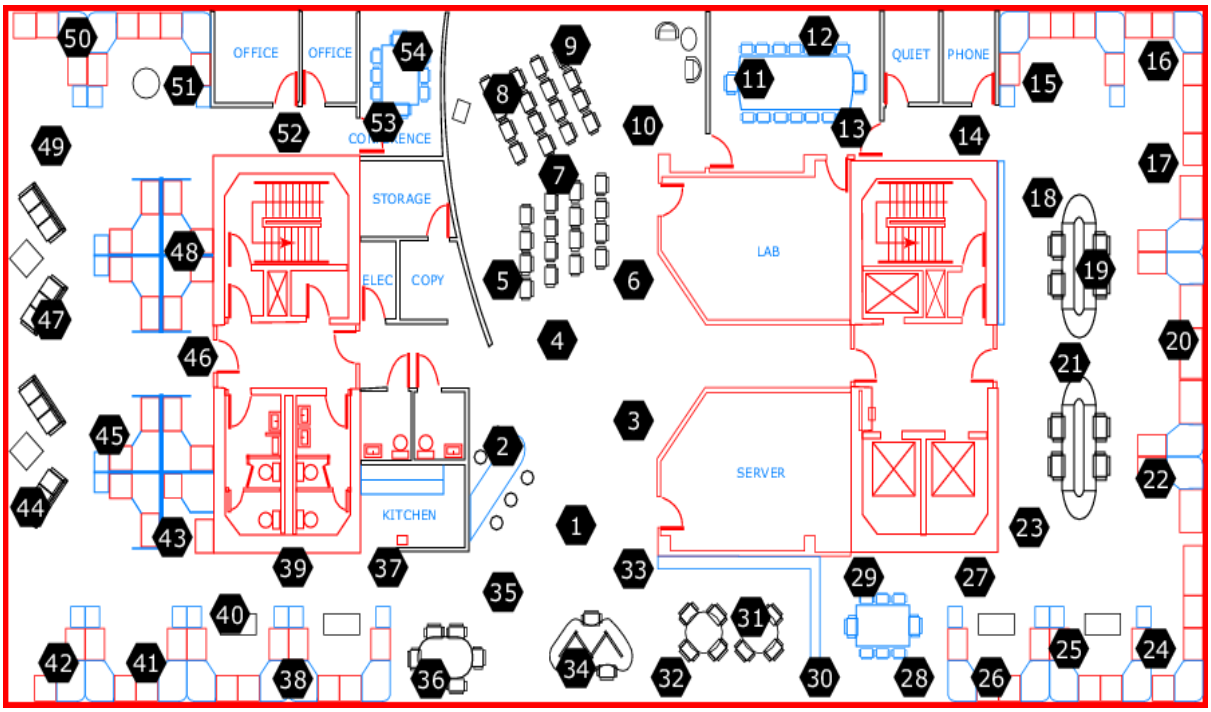

Figure 12

We made a 54×54 adjacency matrix for above dataset with approximates distances between each pair of nodes and then analysed the result of travelling salesman problem on it.

In matrix, the cells with 40,000 in them represent that there is no direct link between those nodes. Because, 40,000 is too big compared to other small costs, therefore my implementation ignore those big numbers, and pick the links with small costs, instead.

Code Reference: Report on 'A Comparative Study of Tabu Search and Simulated Annealing for Traveling Salesman Problem' by Sachin Jayaswal, University of Waterloo.

7.1 54×54 adjacency matrix:

Start: Node-0 ; End: Node-54

Column 1to 26:

W29 10

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	0	12	13	17	20	20	80	100	120	90	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	220	210	200
2	12	0	13	17	20	20	80	100	120	90	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	220	210	200
3	13	13	0	17	20	20	80	100	120	90	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	220	210	200
4	17	17	17	0	10	12	14	25	30	22	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	220	210	200
5	20	20	20	10	0	10	8	12	15	12	25	28	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	220	210	200
6	20	20	20	12	10	0	8	10	12	8	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
7	80	80	80	14	8	8	0	4	6	4	12	15	20	25	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
8	100	100	100	25	12	10	4	0	10	12	13	14	15	18	22	32	30	18	40000	40000	40000	40000	40000	40000	40000	40000
9	120	120	120	30	15	12	6	10	0	6	7	10	15	18	20	20	20	40000	40000	40000	40000	40000	40000	40000	40000	40000
10	90	90	90	22	12	8	4	12	6	0	5	6	10	15	25	40	42	40000	40000	40000	40000	40000	40000	40000	40000	40000
11	40000	40000	40000	40000	25	40000	12	13	7	5	0	5	5	10	18	25	28	40000	40000	40000	40000	40000	40000	40000	40000	40000
12	40000	40000	40000	40000	28	40000	15	14	10	6	5	0	5	10	15	20	20	40000	40000	40000	40000	40000	40000	40000	40000	40000
13	40000	40000	40000	40000	40000	40000	20	15	15	10	5	5	0	12	18	20	12	40000	40000	40000	40000	40000	40000	40000	40000	40000
14	40000	40000	40000	40000	40000	40000	25	18	18	15	10	10	12	0	4	8	8	4	10	20	23	26	32	40	32	40000
15	40000	40000	40000	40000	40000	40000	40	22	20	25	18	15	18	4	0	5	7	5	10	15	20	25	30	35	35	38
16	40000	40000	40000	40000	40000	40000	80	32	20	40	25	20	20	8	5	0	5	15	18	20	18	22	25	30	30	32
17	40000	40000	40000	40000	40000	40000	40000	30	20	42	28	20	12	8	7	5	0	10	10	15	18	20	25	30	30	35
18	40000	40000	40000	40000	40000	40000	40000	18	40000	40000	40000	40000	40000	4	5	15	10	0	5	5	5	10	15	20	25	28
19	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	10	10	18	10	5	0	5	5	10	15	20	20	25
20	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	20	15	20	15	5	5	0	6	6	12	18	16	22
21	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	23	20	18	18	5	5	6	0	8	10	12	14	18
22	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	26	25	22	20	10	10	6	8	0	10	10	12	15
23	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	32	30	25	25	15	15	12	10	10	0	10	8	9
24	220	220	220	220	220	40000	40000	40000	40000	40000	40000	40000	40000	40	35	30	30	20	20	18	12	10	10	0	5	10
25	210	210	210	210	210	40000	40000	40000	40000	40000	40000	40000	40000	32	35	30	30	25	20	16	14	12	8	5	0	5
26	200	200	200	200	200	40000	40000	40000	40000	40000	40000	40000	40000	40000	38	32	35	28	25	22	18	15	9	10	5	0
27	180	180	180	180	180	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	30	32	25	20	22	14	12	5	12	8	8
28	180	180	180	180	180	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	23	22	18	10	15	10	10
29	150	150	150	150	150	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	32	40000	18	10	15	15	15
30	120	120	120	120	120	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	38	40000	20	15	20	18	18
31	80	80	80	80	80	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	25	18	25	20	20
32	80	80	80	80	80	80	40000	40	40	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	30	20	28	25	25
33	80	8	8	8	8	80	40000	38	38	38	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	32	40000	32	20	28	28
34	10	10	10	10	100	40000	40	40	38	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	35	30	30
35	3	3	3	3	3	30	40000	36	35	30	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40	35	35
36	14	14	14	14	14	140	40000	40	40	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	45	38	38
37	14	14	14	14	14	140	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	48	40	40
38	15	15	15	15	15	150	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	49	45	45
39	20	20	20	20	20	200	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	52	45	45
40	22	22	22	22	22	220	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	55	50	50
41	40	40	40	40	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	58	52	52
42	80	80	80	80	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	52	55	55
43	43	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	52	52
44	44	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	55	55
45	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
46	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
47	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
48	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
49	40000	40000	40000	40000	40000	40000	40000	40000	30	42	45	45	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
50	40000	40000	40000	40000	40000	40000	40000	36	30	40	45	40	42	52	55	80	80	40000	40000	40000	40000	40000	40000	40000	40000	40000
51	40000	40000	40000	40000	40000	40000	40000	28	25	35	30	40	40	48	50	50	50	40000	40000	40000	40000	40000	40000	40000	40000	40000
52	40000	40000	40000	40000	40000	40	40	23	20	29	20	35	38	46	45	45	40	40000	40000	40000	40000	40000	40000	40000	40000	40000
53	22	22	22	22	22	22	40	10	10	25	15	30	36	42	42	40	38	40000	40000	40000	40000	40000	40000	40000	40000	40000
54	25	25	25	25	25	25	40	6	10	20	15	30	35	41	40	38	40	40000	40000	40000	40000	40000	40000	40000	40000	40000
55																										
56																										

Figure 13

:

Column 27 to 54

AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB
180	180	150	120	80	80	80	10	9	14	14	15	20	22	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	25
180	180	150	120	80	80	8	10	9	14	14	15	20	22	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	25
180	180	150	120	80	80	8	10	9	14	14	15	20	22	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	25
180	180	150	120	80	80	8	10	9	14	14	15	20	22	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	25
180	180	150	120	80	80	8	10	9	14	14	15	20	22	40	80	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	25
40000	40000	40000	40000	80	80	80	100	90	140	140	150	200	220	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40	22	25
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40	40	40
40000	40000	40000	40000	40000	40	38	40	36	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	36	28	23	10	6
40000	40000	40000	40000	40000	40	38	40	35	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	30	30	25	20	10	10
40000	40000	40000	40000	40000	40	38	38	30	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	42	40	35	29	25	20
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	45	45	30	20	15	15
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	45	40	40	35	30	30
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	42	40	38	36	35
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	52	48	46	42	41
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	55	50	45	42	40
30	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	80	50	45	40	38
32	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	80	50	40	38	40
25	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
20	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
22	23	32	38	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
14	22	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
12	18	18	20	25	30	32	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
5	10	10	15	18	20	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
12	15	15	20	25	28	32	35	40	45	48	49	52	55	58	52	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
8	10	15	18	20	25	28	30	35	38	40	45	45	50	52	55	52	55	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
8	10	15	18	20	25	28	30	35	38	40	45	45	50	52	55	52	55	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
0	10	15	18	20	25	28	30	35	38	40	45	45	50	52	55	52	55	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
10	0	5	5	10	12	15	18	17	25	28	30	30	32	35	38	35	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
15	5	0	5	5	12	18	24	30	32	34	36	36	38	42	44	36	48	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
18	5	5	0	4	6	8	10	14	18	20	22	22	24	28	32	26	40	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
20	10	5	4	0	4	6	8	10	14	16	20	20	24	28	32	28	38	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
25	12	12	6	4	0	4	4	8	10	12	18	18	20	24	28	24	32	40000	40000	40000	40000	40000	40000	40000	40000	52	52
28	15	18	8	6	4	0	6	6	10	11	14	14	16	18	22	16	24	26	40000	40000	40000	40000	40000	40000	40000	42	38
30	18	24	10	8	4	6	0	5	10	12	15	15	20	22	25	22	30	35	40000	40000	40000	40000	40000	40000	40000	60	62
35	17	30	14	10	8	6	5	0	5	5	10	10	12	15	20	15	22	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
38	25	32	18	14	10	10	10	5	0	5	5	10	12	15	20	15	25	28	40000	40000	40000	40000	40000	40000	40000	40000	40000
40	28	34	20	16	12	11	12	5	5	0	8	5	8	10	12	8	15	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
45	30	36	22	20	18	14	15	10	5	8	0	5	5	7	10	7	15	15	18	20	22	25	30	28	40000	40000	40000
45	30	36	22	20	18	14	15	10	10	5	5	0	5	8	10	5	15	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000
50	32	38	24	24	20	16	20	12	12	8	5	5	0	5	10	5	15	15	17	20	22	30	35	32	40000	40000	40000
52	35	42	28	28	24	18	22	15	15	10	7	8	5	0	5	8	10	15	20	22	25	30	35	32	40000	40000	40000
55	38	44	32	32	28	22	25	20	20	12	10	10	10	5	0	10	10	15	20	20	25	30	38	35	32	40000	40000
52	35	36	26	28	24	16	22	15	15	8	7	5	5	8	10	0	10	8	10	12	15	22	25	30	40000	40000	40000
55	40	48	40	38	32	24	30	22	25	15	15	15	15	10	10	10	0	5	10	10	15	18	22	20	25	40000	40000
40000	40000	40000	40000	40000	40000	26	35	40000	28	40000	15	40000	15	15	15	8	5	0	5	5	8	12	20	18	18	40000	40000
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	18	40000	17	20	20	10	10	5	0	8	5	10	20	18	40000	40000	40000
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	20	40000	20	22	20	12	10	5	8	0	10	12	18	15	18	40000	20
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	22	40000	22	25	25	15	15	8	5	10	0	10	15	12	40000	40000	40000
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	25	40000	30	30	30	22	18	12	10	12	10	0	5	8	10	50	20
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	30	40000	35	35	38	25	22	20	20	18	15	5	0	5	10	12	15
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	28	40000	32	32	35	30	20	18	18	15	12	8	5	0	5	10	20
40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	32	40000	25	18	40000	18	40000	10	10	5	0	5	10
40000	40000	40000	40000	40000	52	42	60	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	50	12	10	5	0	5
40000	40000	40000	40000	40000	52	38	62	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	40000	20	40000	20	15	20	10	5	0

Figure 14

7.2 Study of Taboo search (TSP) on 54 node Intel Berkeley research lab :

```
best neighbor cost greater than current tour cost

current tour cost = 1382    best obj =375
best tour

best_tour =

  Columns 1 through 24

     2     3    34    32    31    30    28    29    23    26    25    24    22    20    18    14    15    16    17    19    21    27    33     4

  Columns 25 through 48

    36    37    39    38    40    41    42    43    44    45    47    46    48    49    50    51    52    53    54     8     7     9    11    13

  Columns 49 through 54

    12    10     6     5    35     1

best obj =375

time taken = 127.484375 >>
```


Figure 15

7.3 Study of Simulated Annealing (TSP) on 54 node Intel Berkeley research lab:

```
iter# = 1173    , T = 1.69  , obj = 429 , accpt ratio=0.01
iter# = 1175    , T = 1.67  , obj = 434 , accpt ratio=0.01
iter# = 1177    , T = 1.66  , obj = 437 , accpt ratio=0.00
iter# = 1179    , T = 1.64  , obj = 426 , accpt ratio=0.01
iter# = 1181    , T = 1.62  , obj = 421 , accpt ratio=0.00
iter# = 1183    , T = 1.61  , obj = 420 , accpt ratio=0.00
iter# = 1185    , T = 1.59  , obj = 420 , accpt ratio=0.00
best obj = 407
 best tour

time taken = 41.000000  >>
```

Editor - D:\dijikstra\tspannealing.m

tspwaterloo01.m | tspannealing.m

```
1     % **********Read distance (cost) matrix from Excel sheet "data.xls"******
2 -   d = xlsread('Book1.xlsx');
```

Figure 16

8. Conclusion

During 6 week internship period in NIT Raipur it has been an excellent and rewarding experience. I have practically gained knowledge of Networking and its optimization. Through this internship report we came into the following conclusions that helped us understand network optimization better. Different network optimization algorithms may have their own strengths and drawbacks e.g. Dijkstra's algorithm is faster and simpler but cannot deal with graph with negative weights and we should not be constrained to use 1 only in all situation. The advantage of Genetic Algorithms are that they are Robust, they Provide optimisation over large space state and unlike traditional AI, they do not break on slight change in input or presence of noise.The results obtained for the implementation of both SA and Tabu Search for TSP show that they provide good solutions for small size problems tested. The shortest-path problems and traveling-salesman problems have certain similarities because in each of them one has to walk a graph and find a path between two nodes. The difference is the constraint on the solution. The shortest-path requires just a path between two points, while the traveling salesman requires a path between more points that returns to the first point. . Further improvements can be obtained by fine tuning the various parameters in Simulated Annealing and by better implementation of diversification and intensification in Tabu Search.

9. Bibliography

[1] Y. Leung, G. Li, and Z. B. Xu, "A genetic algorithm for the multiple destination routing problems," IEEE Trans. Evol. Comput., vol. 2, pp. 150–161, Nov. 1998.

[2] Z. Xiawei, C. Changjia, and Z. Gang, "A genetic algorithm for multicasting routing problem," in Proc. Int. Conf. Communication Technology (WCC-ICCT 2000), 2000, pp. 1248–1253.

[3] D. E. Goldberg and M. Rundnick, "Genetic algorithms and the variance of fitness," Complex Syst., vol. 5, no. 3, pp. 265–278, 1991.

[4] Gihan Nagib and wahied G. Ali," Network Routing Protocol Using Genetic Algorithms". International journal of electrical & Computer sciences, IJECS-IJENS vol:10 No: 02,march,2010.

[5] D.EGoldberg, "*Genetic Algorithms in search, optimization and machine Learning".* Addison-Wesley publishing company,1989.

[6] Dijianh huang, "secure link state Routing protocol."AIML 06,International conference,13-15 june,2006,Sharm El Sheikh, Egypt.

[7] Reeves C. R., 1993, Modern heuristic techniques for combinatorial problems, Orient Longman, Oxford

[8] Pataki G., 2003, Teaching integer programming formulations using the traveling salesman problem, Society for Industrial and Applied Mathematics, 45 (1), 116-123

[9] Gendreau M., 2002, An introduction to Tabu Search, http://www.ifi.uio.no/infheur/Bakgrunn/Intro_to_TS_Gendreau.htm

[10] Battiti R., Tecchiolli G., 1994, The Reactive Tabu Search, ORSA Journal on Computing, 6 (2), 126-140

[11] Glover F., Tilliard E., Werra D.E., 1993, A user's guide to Tabu Search, Annals of Operations Research, 41, 3-28

[12]https://en.wikipedia.org/wiki/List_of_genetic_algorithm_applications

[13]https://en.wikipedia.org/wiki/Genetic_algorithm

[14] https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html