CHAPTER 1

INTRODUCTION

In this research work, heuristic approach, “Simulated Annealing Algorithm” has been implemented for emergency evacuation route planning for traffic management which is one of the most difficult computational problems in the field of applied sciences. The approach uses the numbers of vertices or places of graph of cities. Using the heuristic search method, it finds an initial feasible route for EERP problem. When it is moved according to direction and distance of goal and congestion place, this two-step planning process is activated for it. This chapter introduces some background information about Emergency evacuation route planning, the motivation of the task, research objective, and its significance. This also highlights the contribution of our thesis.

1.1 Background

People along with private and public vehicles around the city, if gathered, form the traffic worst due to some manmade and natural disaster. In such a case, proper management is very necessary, both in normal traffic systems and in emergency periods. Since local optimal routes can be analyzed by it, the concept of heuristicity is applied in the work. This model will enhance the traditional method with the latest technology using computer. Traffic management is the proper arrangement of the traffic using different strategies by using different algorithms. In traditional traffic management system, the traffic system is maintained using traffic police. In this dissertation, a heuristic approach has been studied. For the emergency evacuation route planning of traffic system, the concept of simulated annealing is helpful

Since 30 years, natural and man-made disasters have been increasing. Urban areas are threatened by terrorist attacks, hurricane, earthquakes, volcanic eruptions, power plant or chemical industry incidents, fires, etc. Among them some accidents are predicable and some are non-predictable. Therefore, in these cases, there is a limited time to response effectively and to minimize the fatalities. Then, advance planning is needed for effective solution which should be more flexible and quick. And people should be evacuated in short time. Therefore, transportation networks play a key role in emergency evacuation conditions.

Figure 1.1 Traffic jam near Kathmandu Mall, Kathmandu, Nepal

1.1.1 Evacuation Planning

Evacuation planning is the management of emergency disaster events. In which, action of leaving danger zone as quickly as possible are explained in details. The disaster events may have different types such as natural, technological, and man-made etc. Those disasters such as earthquakes, hurricanes, tornadoes, tsunamis, flood and mudslides are examples of natural disasters. The disasters such as nuclear accidents, industrial accidents and transportation accidents are examples of technological disasters. And terrorism, bombing and shooting are examples of man-made disaster.

1.1.2 Emergency Evacuation Time

At the time of emergency evacuation, the time required to evacuate is important, and it is one of the main factors one should consider when developing a plan for evacuation. Evacuation time includes the time required to configure all traffic control elements on the evacuation routes, initiate the evacuation, and clear the routes of vehicles once all evacuating vehicles reach a destination of safety. Here, evacuation time does not include the time needed for local officials to assemble and make a decision to evacuate. Evacuation time during expected and unexpected emergency events is different, Amani (2009). In general, evacuation time in both expected and unexpected are actually composed of three time subcomponents mobilization time, travel time and queuing delay time.

1.1.3 Graph

A graph data structure consists of a finite set of ordered pairs, called edges or arcs, of certain entities called nodes or vertices. As in mathematics, an edge (x,y) is said to point or go from x to y. The nodes may be part of the graph structure, or may be external entities represented by integer indices. A graph data structure may also associate to each edge some edge value, such as a symbolic label or a numeric attribute (cost, capacity, length, etc.).

Generally graphs are categorized into:

1. Directed graph

The directed graph can be seen as a flow network, where each edge has a capacity and each edge receives a flow. The direction is shown with arrow.

2. Undirected graph

The undirected can't be seen as a flow network, where capacity of flows are indicated but it does not show direction with arrow.

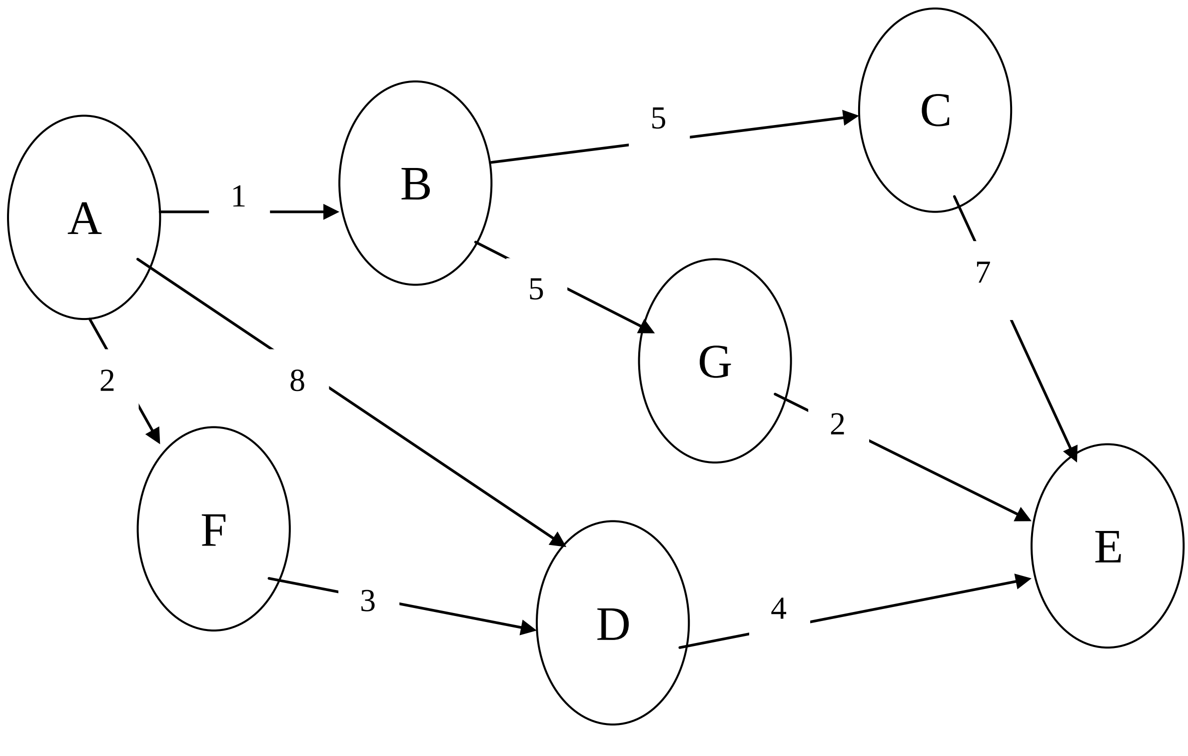

Figure 1.2: Example of directed graph of 7 nodes

1.2 Motivation

At recent time, research on Emergency Evacuation Route Planning [EERP] for traffic management is very hot topic due to increasing disaster accidents. So these facts motivate the need for better EERP approaches in order to minimize the loss of human lives and property. The heuristic approach "Simulated Annealing" has been used for robot route planning with better performance, Hui (2009), so it makes us motivated to study for EERP problem too. Recently, the genetic algorithm based approach has been developed for EERP problem, but the efficiency is not sufficient for large-scale route planning problems. So there is necessary to solve it. The simulated annealing algorithm [SAA] is similar to genetic algorithm in solving general optimization problems. But SAA is used for global optimization with overcoming local optimization.

In recent years, natural, man-made and technological disasters have increased in frequency and in magnitude of occurrence. Specifically, technological and man-made disasters risk have increased due in large part to the evolving and growing threat of terrorist attacks against the US. Natural disaster risk has increased due to progressively changing weather patterns caused partly by global warming. Thousands of people and billion of property has lost due to terrorist attack in Washington DC, United State at date September -11, 2001 and Tsunami in south and southeast of Asia at date December -26, 2004 , Zhou et al. (2010). There are several disasters such as: Hurricane Katrina and Rita, terrorist attack at Mumbai in India and earthquake in Nepal etc which have been losing of people and property due to unmanageability of traffic at emergency periods. Those events encourage us to contribute some works in EERP problem too.

Figure 1.3: Unmanageability of traffic system due to disaster (bomb explosion).

1.3 Objective

The objectives of this research work are:

1. To minimize the evacuation time for emergency traffic management system.
2. To implement new approaches for constructing the evacuation route using simulated annealing algorithm (SAA).

1.4 Significance and Contribution

The work contributes in implementing a simulated annealing based approach for EERP problem for traffic management. The simulated annealing based approach solves drawbacks of genetic algorithm. Therefore, SAA has given optimal evacuation time for EERP problem. SAA can be used for on- time route configuration of the problem.

1.5 Organization of Thesis

Chapter 2 gives brief information on existing studies in the literature related works with EERP problem and EERP's solution model. Chapter 3 gives the detail about the simulated annealing approach. Chapter 4 gives the details about the simulation results and performance evaluation. Chapter 5 gives results and discussion of proposed model and Chapter 6 gives the conclusion with their limitation and future work. At last, the source codes are appended.

CHAPTER 2

LITERATURE REVIEW

2.1 EERP Problem

Emergency evacuation route planning is the management of emergency disaster which can be explained as the act of leaving from danger zone as quickly as possible with specify route. EERP problem has two parts, which are EERP configuration and capacity constraint of contraflow, Zhou et al. (2010). In recent years, those disasters have increased in frequently and in magnitude of occurrence specially, technological and man-made disaster risk has increased due to large part to evolving and growing threads of terrorist attacks. The disaster will loss more lives of people and property if appropriate EERP is not taken with minimal evacuation time.

2.2 EERP Model

Emergency evacuation route planning can be solved in two models that is, analytical model and simulation model, Hui (2009). Analytical model contains mixed integer programming, non-linear programming and queuing models for EERP. However, simulation model contains macro, meso and micro simulation according size of the graph. The analytical approach has been used for robot route planning in, Kim and Shekhar (2008). Emergency evacuation route planning (EERP) problem with bi-directional flows and contraflow lane reveals is in the class NP due to its combinatorial nature, Hui (2009). So, Heuristic approach for simulated annealing are applied in our work.

2.3 Research Areas in EERP

Lu and George (2007) explained capacity constrained routing planner which was used for evacuation route planning which allows capacity as a time series and found the shortest route. In, Lu and George (2007), Capacity constrained routing planner has been proposed for evacuation route planning which models capacity as a times series and generalizes shortest route algorithms to incorporate capacity constraints and it helps to minimize the evacuation egress time and computation costs of evacuation. But it has done in First In First Out [FIFO] order.

Kim et al. (2007) used bottleneck saturation approach in the Intelligent Load Reduction heuristics for better evacuation route planning. They have proposed two heuristics for contra-flow planning, one heuristic named Flip High Flow Edge [FHFE], which is based on a greedy algorithm with a flow history of edges. And it generates a suboptimal contraflow plan without iteration and another one simulation approach with iteration search, however it has not implement for EERP problem.

Baharnemati and Lim (2011) used maximal dynamic flow model to get maximum number of evacuees through some routes with fixed flow rate. Hui (2009) used genetic algorithm to find optimal route of robot motion. In Baharnemati and Lim (2011), Dynamic network flow model has proposed to achieve the maximum number of evacuees through some routes with a fixed flow rate. However it has not control flow rate in evacuation problem with uniformly.

Kim and Shekhar (2008) used to deal with large numbers of evacuees, they developed bottleneck relief heuristic. Amani (2009) used to formulate the multiple flow emergency evacuation route planning problems in the presence of an unexpected emergency event on the basis of genetic algorithm. Hamza-Lup et al. (2004) explains the smart traffic management system according to accident location and destination place.

Zhou and Chen (2010) have proposed Sequential Monte Carlo Simulated Annealing (SMC-SA) which has helped to use on empirical distribution for global optimization problem. They have improved performance on all test problem of their objective function. Hu and Hu (2010) has proposed Model based Annealing Random Search (MARS), in which it is used to present global convergence properties by exploiting its connection its connection to stochastic approximation.

Liu (2011) has studied and analyzed household pick-up and gathering behavior from interviews, developed models to present the behavior, and integrated the household behavior models with network simulation modeling to examine the effects of household behaviors on network evacuation performance. It has helped to determine performance whether it is correct route at the time of emergency or not. Therefore, it needs direction of goal and distance of goal for correct route finding.

Lu and George (2007) has used a linear programming [LP] based method for the evacuation problem too, which will help for analytical modeling. However, EERP problem with bi-directional flows and contraflow lane reveals is in the class NP due to its combinatorial nature, Hui (2009). Then, Zhou et al. (2010), finding the minimal route becomes more challenging as the network size increases. Therefore heuristic solution approach is necessary to solve such type of problem. There are several heuristic approaches such as Simulated Annealing, Genetic Algorithm, and Ant Colony Optimization etc. Among them, Simulated Annealing will be more appropriate due to its dynamic characteristic, where temperature factor may varies according to their real time information.

2.4 Genetic Algorithm Approach in EERP

The “Genetic algorithm” is meta-heuristic in nature which is used for EERP problem and is important to mention because of using a genetic based solution approach in the study is the wide popularity and accepted for parallel search of solution space, Amani (2009). Genetic algorithm [GA] is the best known and represented for evolutionary heuristic family algorithm for attempting natural selection and reproduction of any survival environments, Hui (2009). GA’s processes the core algorithmic procedure associated with this family of algorithms fitness evaluation, selection, and production, which involves crossover and mutation genetic based operation. A genetic based solution approach is used to find a high quality solution for EERP problem, which maximizes weight flow for evacuees and responders. However, it does not support for the global optimization EERP problem. The implementation code and data set of Amani (2009) are used in our research work for the comparison.

2.5 The Simulated Annealing Approach in EERP

2.5.1 The Simulated Annealing Algorithm

Simulated annealing approach has been widely applied in many industrial as well as robot planning fields. The well-described application of the simulated annealing algorithm is for the traveling salesman problem (TSP). In Hui (2009), SA approach has been proposed for robot route planning However, simulated annealing algorithm is found to be better for EERP problem too, in which the given number of cities and the cost of traveling from one city or place to another, to get the least cost of evacuation time from source to goal with discarding congested place. The simulated annealing algorithm is analogous to metal physical cooling process with each step of the simulated annealing algorithm replacing the current solution by a random “nearby” solution. The probability of choosing the random “nearby” solution depends on the difference between the corresponding function values and a global parameter T which is called the temperature that is decreased during the process. The dependency is such that the current solution changes almost randomly when T is large. Figure 2.1 represents the flow chart of the general steps of the simulated annealing algorithm:

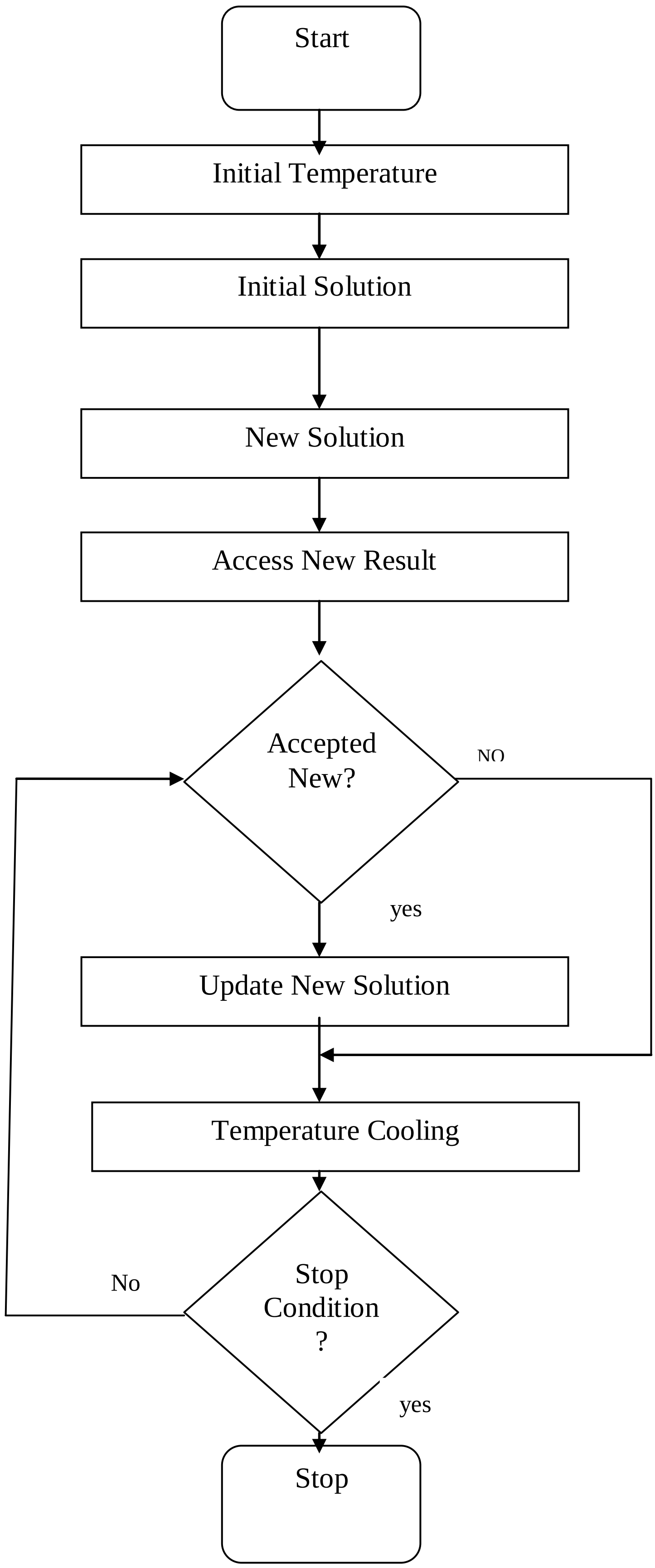

Figure 2.1: Procedure of Simulated Annealing Algorithm

2.5.2 Simulated Annealing Approach

Before the work, the simulated annealing algorithm approach had already been applied to Robot route planning method, Hui (2009). This has proposed a simulated annealing based artificial potential method to deal with the robot route planning issue. The artificial potential method is firstly used to map the environment and find a route for the robot. The simulated annealing algorithm is applied for robot route planning in the work.

In this work,

1. The simulated annealing algorithm is more sensitive to the control parameters such as initial temperature and cooling rate. In his research, only one group of control parameters is set in dealing with three different situations. The performance of simulated annealing approach could be improved if suitable parameters are set.

2. The operator that generates the new solutions is too simple. A swapping operator is used to generate the new solution. The swapping operator only flips some bits of the result to generate a new result. In this way, the possibility of jumping out of the local optimal result is small. The simulation results in this thesis proves that adding more algorithm operators to generate the new result gives the simulated annealing approach better performance. In this thesis, more than one operator are used, the observation results show that the performance is improved.

2.5.3 Simulated Annealing in Heuristic Approach

Simulated annealing method is known that in finding the optimal route, it needs firstly selecting a random feasible route, and then apply a mathematical operator to generate a new route based on the selected feasible route. The initial selected route is the beginning of the search point; the initial solution is replaced by the randomly generated neighbor solution. The search goes along "downhill" movement from the initial solution. Then, as temperature goes down, the algorithm accepts the "up hill" movement to avoid being stuck at local minimum. Therefore, a better initial solution means a better initial searching point, which could enhance the efficiency and the performance of the approach, Fan and Machemehl (2006).

The way of generating the initial route is illustrated as follows:

Firstly, without starting point and end point, a vertex is chosen randomly from the map, use edges or route with those certain heuristic criteria to connect the start point and the selected vertex. If the route exits with any edges of the map, the route is recognized as valid route. Another vertex will be randomly selected again for testing. Then add the vertex into the route. Then, start from the selected vertex to find the next valid vertex. Keep performing the above procedure until the end point is selected and also the route line to the end point is a valid route (i.e., the route to the end point does not intersect with any edges). The method just randomly picks a feasible vertex until the end point is picked. The blindly picking is not an efficient way of producing the initial route.

To improve the efficiency of the initial solution selecting process, a heuristic process is proposed in this Chapter. The process makes a modification on the current method. In current method, a random vertex is selected after one feasible vertex is selected, and then tests the feasibility of the line segment between randomly selected vertex and the feasible vertex. In the proposed heuristic method, after one feasible vertex is selected, an end point feasibility test will be carried out before selecting another vertex. Before selecting next vertex, the method will firstly test whether the line segment between the feasible vertex and the end point is feasible (does not interest with any obstacles). If the segment between the feasible vertex and the end point is feasible, then put the end point as the next vertex and complete the selection process. The following pseudo-codes illustrate the process of the heuristic initial route selection method, Hui (2009). It is modified with applying heuristic information in direction of goal and congested place.

let us consider IP be initial path, route(Vs Ve) be route from Vs to Ve.
initial route IP = {start};

```
while (1)

    if route{IPend, Destination Point} is a feasible route

        IP = {IP, Destination Point};

        break;

    endif;


    randomly select Vs on the basis of direction of goal and congested place;
```

```
    if route{IPend, Vs}meet another node of graph or place;

        continue;

    else

        IP = {IP,Vs};

    endif

    if Vs == destination point;

        break;

    else

        continue;

    end if

end while
```

CHAPTER 3

OVERVIEW OF THE SIMULATED ANNEALING

This chapter explains the methodological approach in detail, including the modeling of environments, structure of the approach/algorithm, the generation of the initial feasible route, the new planner for generating the random route and the procedure of calculation. One of the aims of this thesis is to implement a simulated annealing based approach into emergency evacuation route planning. The simulated annealing based approach is expected to determine quickly the optimal feasible route with moving graph node of cities. The research uses C# programming language to simulate the tasks. The simulated annealing has been implemented for route planning method. Therefore, the entire research work could be separated into the following three phases:

1) Design the algorithms that could find routes and optimize the evacuation time for emergency evacuation route planning.

2) Use C# programming language to implement mathematical model and obtain the observation results proving the feasibility of algorithm.

3) Compare the results of the new approach with those from existing methods, and discuss the performance and the efficiency of the new approach.

3.1 Modified Simulated Annealing Algorithm

Simulated annealing Algorithm (SAA) is a generic probabilistic meta-heuristic algorithm for the global optimization problem of locating a good approximation to the global optimum of a given function in a large search space. Simulated annealing algorithm can be used for evacuation route planning as:

Let us consider feasible solution expressed by a series of vertices linking the start point through to the end point. Each vertex of graph has its series of numbers. Then, we propose the following algorithm:

Let feasible solution X is given by:

$X=\{V_{start}, V_{sart+1}, V_{start+2,} \ldots V_{end-1}, V_{end}\}$;

The evaluation function E_f is given by:

$E_f = \sum D(V_i, V_{i+1})$

Where V represents vertex / node of graph, D (V_i, V_{i+1}) represents the direct distance from V_i to V_{i+1}. The pseudo- code of the algorithm is as following:

Step 1: Set an initial temperature T;

Step 2: Generate a initial route order randomly from the start to the destination and calculate the route length L(initial);

Step 3: Generate a new route which also from the start to the end and calculate route length L(new);

Step 4: If L(new) < L(initial), accept new route. Else possibly accept new order according to heuristic information of goal and congested place.

Step 5: Repeat step 3 and step 4 until the temperature gets down.

Step 6: Down the temperature and return to step 1.

The following are pseudo-codes for the simulated annealing algorithm:

$T = T_{init;}$

while ($T>T_{terminate}$)

```
  randomly generate one feasible solution Xs;

  evaluate Xs, Ef = f(Xs);

  count = 1;

  while (count < Threshold)

      generate a new feasible solution Xn base on Xs;

      evaluate Xn, En = f(Xn);

      if f(Xn) < f(Xs)

            Xs = Xn;

      else if rand(1) < (exp((f(Xs)- f(Xn))/T)

              Xs = Xn;

       count =count +1;

   endwhile

   T = cool_rate * T;

  update Xs at each reduction of temperature T according to direction and distance of goal
endwhile
```

3.2 Mathematical Concepts of SAA

It is known from Figure1 that after each reduction of initial temperature T, a new feasible solution Xn is selected in each new round. It is essential for the algorithm to quickly and correctly generate a random feasible route in each round. For this purpose, the edges of the graph should be specified first. If an edge or route is exist between two points on the edge of or within the node of graph with fulfilling certain heuristic criteria, then. Any route is defined as a valid route. In the proposed program, a separate array is used to store all the edges of the map. At the initial stage of the program, starting point and end point; a vertex is chosen randomly from the map, then use a edge to connect the start point and the selected vertex. Another vertex will be randomly selected again for testing.

If route with direction of goal to the selected vertex any edges in the map, which is recognized as part of a valid route. If it's route is without direction of goal to the selected vertex any edges in the map, which is recognized as part of an invalid route. Then add the vertex into the route. After that, start from the selected vertex to find next valid vertex. Keep doing the above procedure until the end point is selected and also the route line to the end point is a valid route.

3.2.1 Initial Route Selection Process

The heuristic simulated annealing algorithm is implemented in our thesis. In the tasks, initial route selection is the first step in which the four components has played important role for route configuration of EERP problem. The four major components are as follow:

Initial state: The place where vehicles are located

Final state/Goal state: The destination of vehicles

Congestion place and direction of it: The place where disasters occurs.

Calculation of temperature factor: The factor which is used to help for calculating shortest distance on basis of heuristic information. It will be changed in every stages of transportation.

In the initial route selection, first of all, distance and direction of goal and congested place has to be determined. On the basis of those heuristic information, initial route are selected. Figure 3.1 illustrates the process of the initial route selection process.

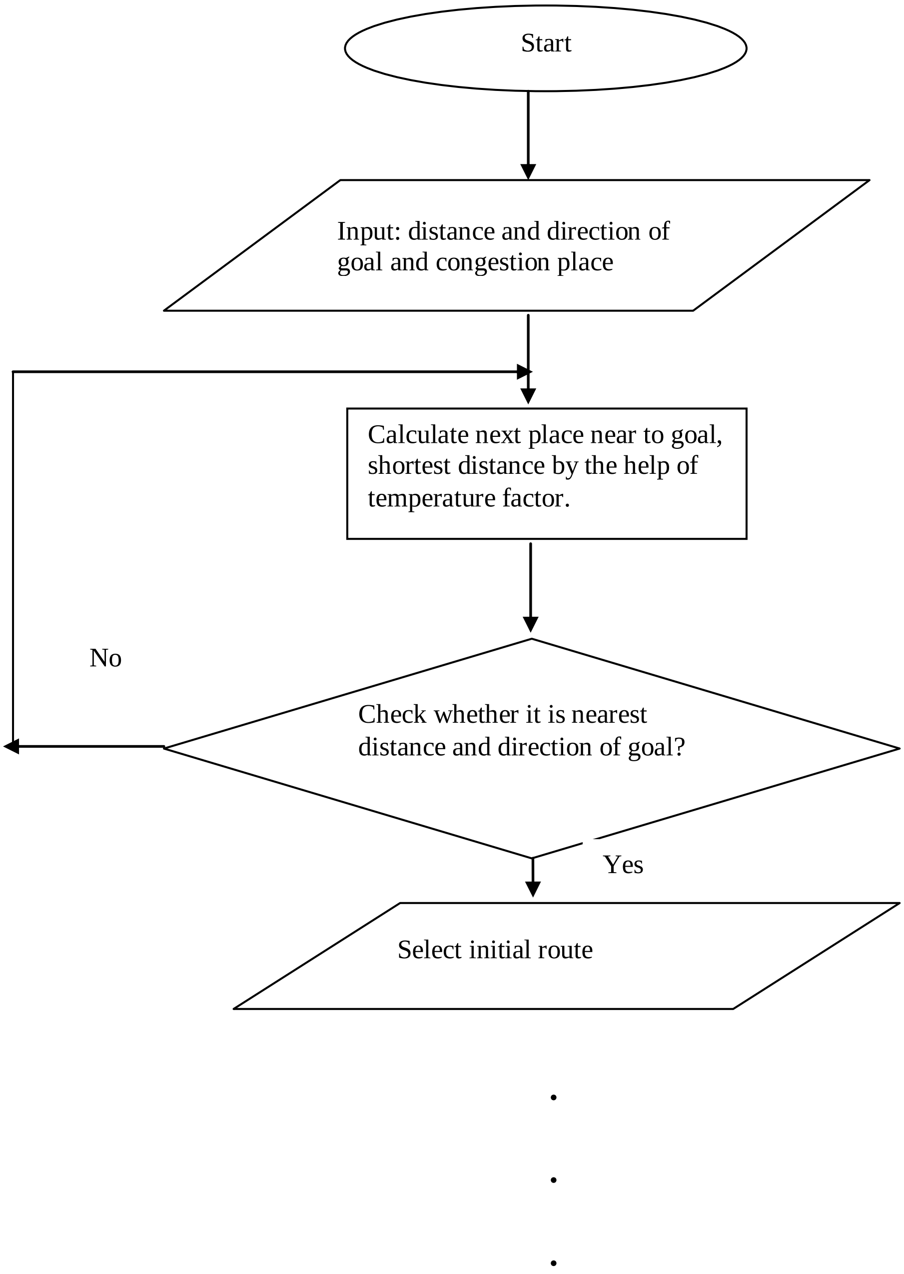

Figure 3.1: Initial Route Selection

3.2.2 Random Route Planning

After the initial route selection, the next step is to find the next node which should be near the goal and far from congested place. Therefore, random route planning is based on the two factors which are:

1. Minimal distance from goal and direction of goal
2. Far from Congested place and its direction.

3.3 Platform for Implementation

The platform of our research task is categorized as:

Algorithm simulating language: C# programming language

System OS: Windows 7

Processor: Pentium(R) Dual-Core CPU T4400 @2.20 GHz

Memory (RAM): 2GB

System type: 32-bit Operating System.

CHAPTER 4

OBSERVATION RESULTS

This chapter presents the simulation results. With these results, the performance of the simulated annealing based approach is evaluated. Four cases are tested using the developed approach. The simulation results are compared with those presented in; Amani (2009) to see whether the new approach is efficient for route planning in such environments. The evaluation results clearly illustrate the first main contribution of the thesis, that the simulated annealing based approach successfully configure collision free route for EERP problem. The approach gives improved performance over existing methods. The calculation time for obtaining the optimal route is better than genetic algorithm based approach in, Amani (2009).

4.1 Observation and Algorithm Parameters

The approach is tested in four cases of graph, in which route is optimized for EERP problem. The solution derived from the algorithm is optimal or near-optimal. The numbers of nodes of graph of cities in the four testing environments are summarized in Tables. The different cases are taken on basis of numbers of congestion nodes, numbers of sources of nodes and numbers of goals of nodes. The direction of goals and congestion places helps for selecting route at a random time and moves forward or backward. The parameters of the SA algorithm are temperature factor, direction and distance of goal and congested place.

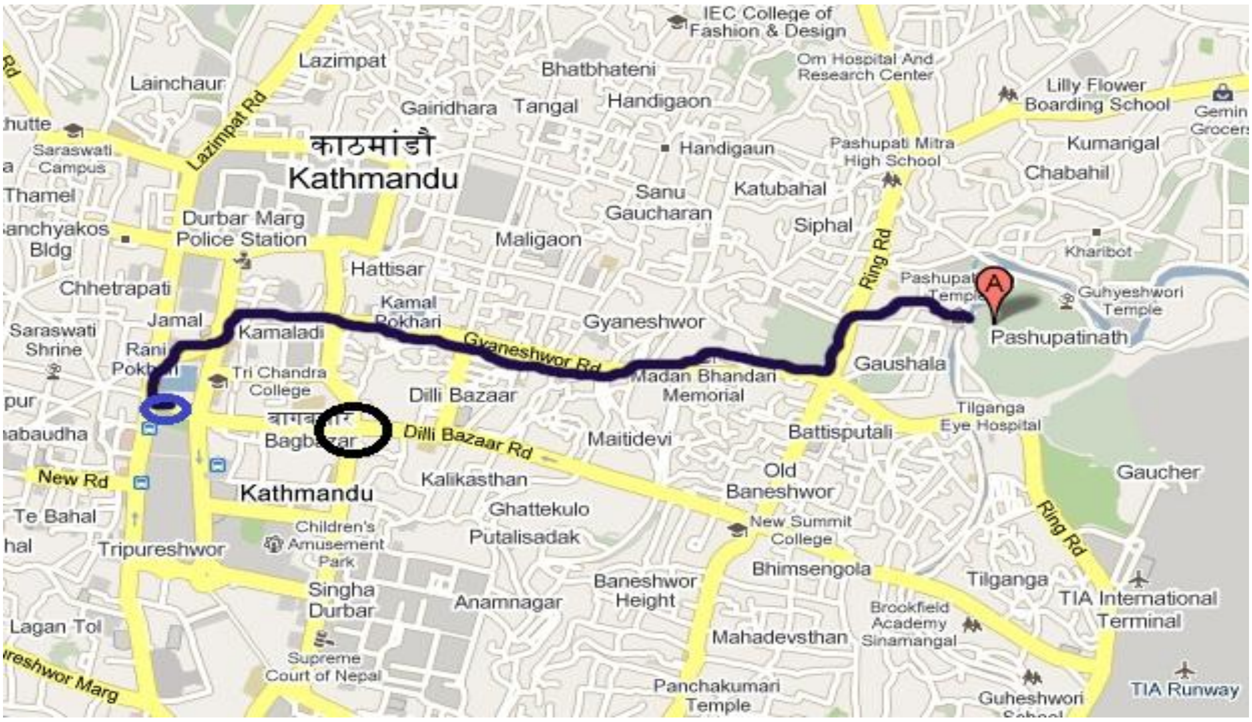

Figure 4.1: Finding rough shortest route from Ratnapark to Pashupatinath when disaster occurred at Dilli Bazar chowk using simulated annealing.

In the above Figure 4.1, shortest route is calculated by the help of heuristic information using simulated annealing.

Under this work, a simulated annealing algorithm is implemented and the main job of this algorithm is to find those routes from the place where the person or vehicle is residing. Then it finds the route and shortest route from that places where the entity is residing. Firstly, the data are taken by GIS [Geographic Information System] or any communication media. Then, that data are processed using our approach so as to calculate the optimal route for achieving the goal.

4.2 Observation Results

In the implementation of those algorithms, the number of vertices varies from 10 to 90 in numbers. Each vertices corresponds the city which may be source or destination or congestion place or other cities. In our work, congestion place is discarded in graph under implementation.

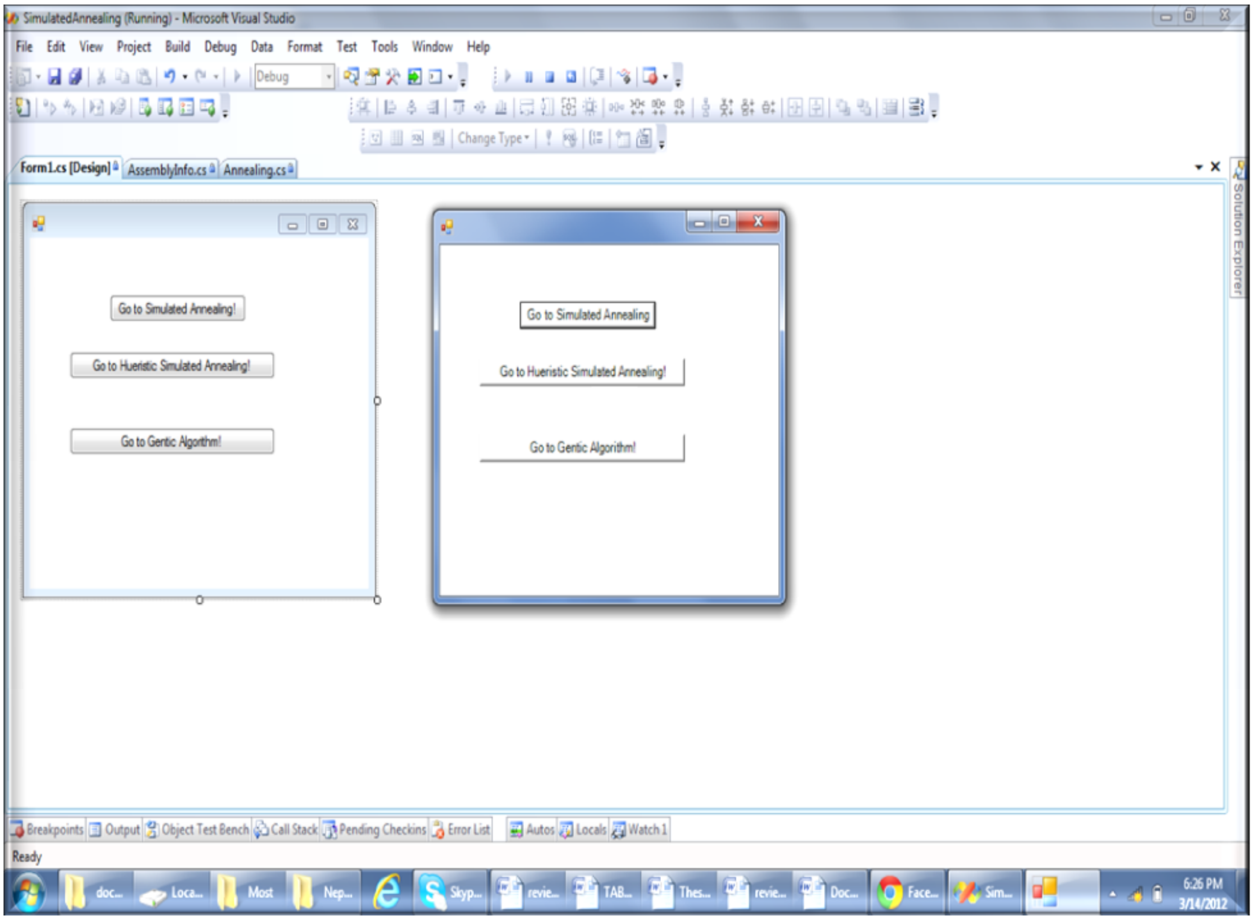

Figure 4.2: Snapshot of simulated annealing with heuristic algorithm

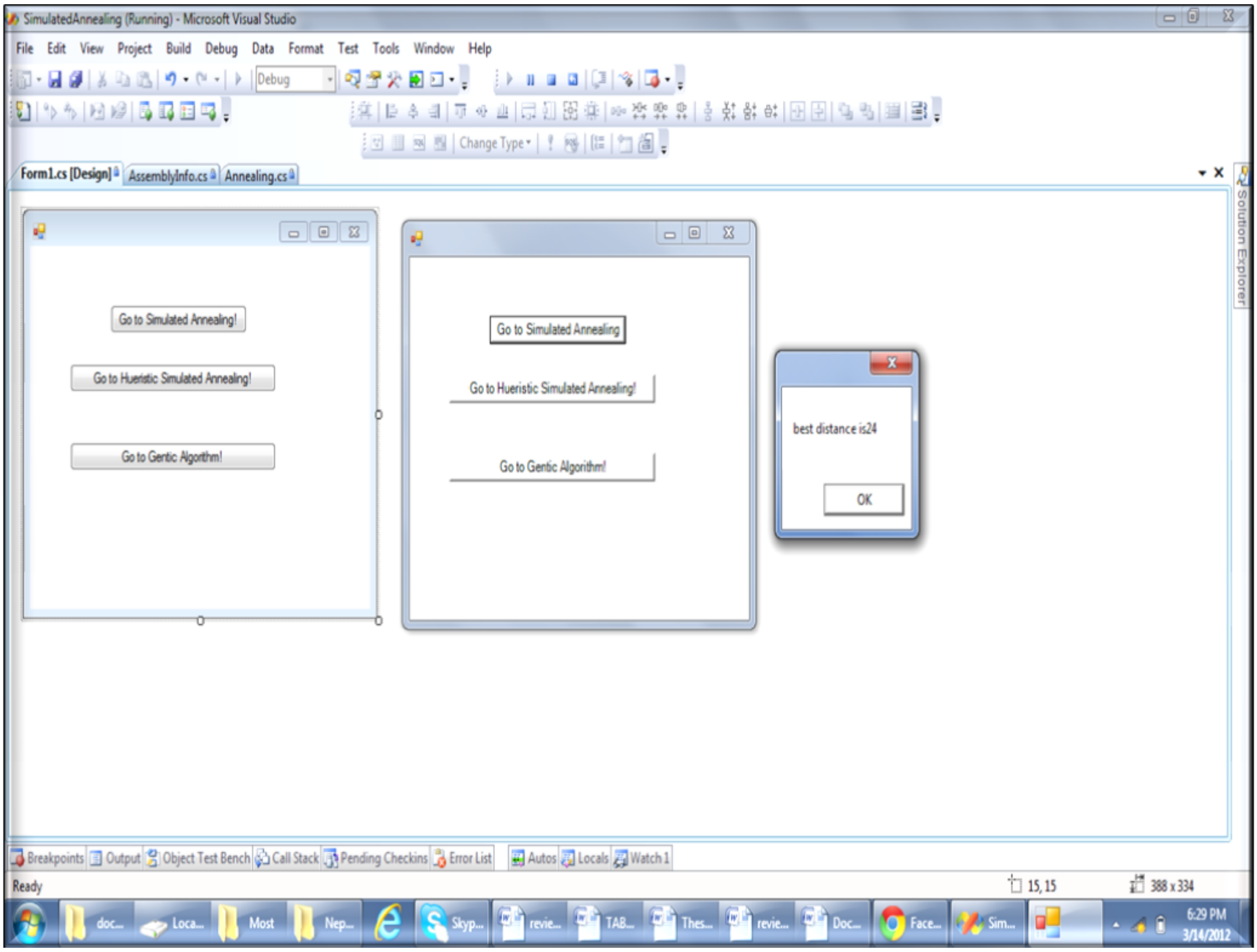

Figure 4.3: Snapshot of simulated annealing with heuristic algorithm output

4.2.1 Static Simulated Annealing:

For 10 nodes

Source Node	Destination Node	Congested Node	Processing Time (sec)
1	10	4	0.58
2	9	5	0.53
3	7	6	0.52
4	6	7	0.60
5	1	8	0.58
6	2	9	0.64
7	3	10	0.51
8	4	1	0.57
9	5	2	0.59
10	8	3	0.53
Average Processing Time			0.56

Table 4.1: Static Simulated Annealing for 10 nodes

In this way, processing time of static simulated algorithm for 25, 56, and 90 vertices of graph are calculated as:

No. of Nodes	10 vertices	25 vertices	56 vertices	90 vertices
Processing time (sec)	0.56	1.331	5.739	15.21

Table 4.2: Static Simulated Annealing

4.2.2 Genetic Algorithm

For 10 vertices

Source Node	Destination Node	Congested Node	Processing Time (sec)
1	10	4	0.48
2	9	5	0.53
3	7	6	0.52
4	6	7	0.47
5	1	8	0.56
6	2	9	0.45
7	3	10	0.55
8	4	1	0.44
9	5	2	0.57
10	8	3	0.43
Average Processing Time			0.50

Table 4.3: Genetic Algorithm for 10 nodes

In this way, processing time of genetic algorithm for 25, 56 and 90 vertices of graph are calculated as:

No. of Nodes	10 vertices	25 vertices	56 vertices	90 vertices
Processing time (sec)	0.50	1.246	4.923	14.50

Table 4.4: Genetic algorithm

4.2.3 Heuristic Simulated Annealing Algorithm

For 10 nodes:

Source Node	Destination Node	Congested Node	Processing Time (sec)
1	10	4	0.43
2	9	5	0.52
3	7	6	0.53
4	6	7	0.45
5	1	8	0.58
6	2	9	0.45
7	3	10	0.55
8	4	1	0.44
9	5	2	0.57
10	8	3	0.48
Average Processing Time			0.50

Table 4.5: Heuristic Simulated Annealing for 10 nodes

In this way, processing time of heuristic simulated annealing for 25, 56 and 90 vertices of graph are calculated as:

No. of Nodes	10 vertices	25 vertices	56 vertices	90 vertices
Processing time (sec)	0.50	1.20	4.80	13.90

Table 4.6: Heuristic Simulated Annealing

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Performance Evaluation

It is seen in Figure 3 that among three algorithms the performance of heuristic simulated annealing algorithms is found better.

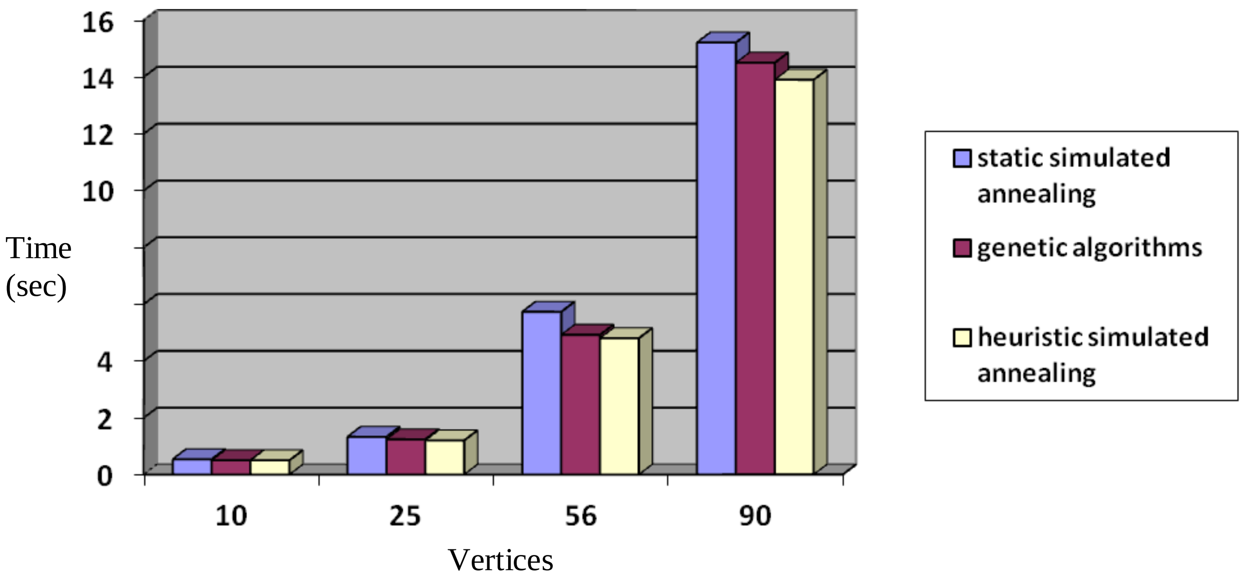

Figure 5.1 A comparative studies of three algorithms

It can be seen in another way through another graphical way as follow in Figure 4.

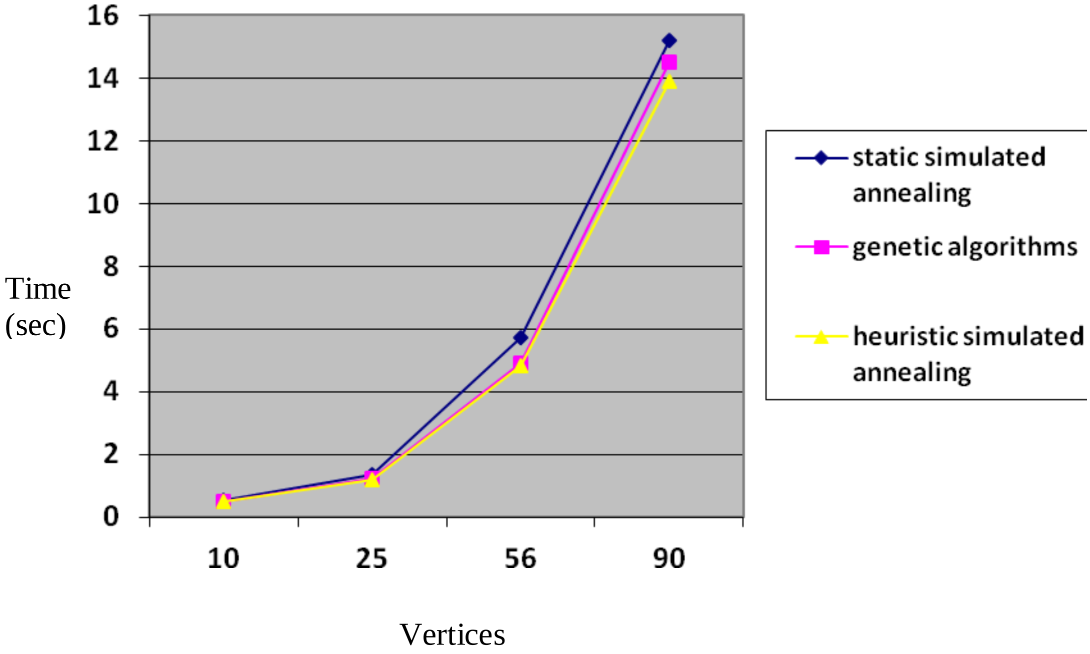

Figure 5.2 Comparative studies of three algorithms

The lowest line is the heuristic simulated annealing algorithms which have showed better performance among other threes. In which in small size of nodes, the performance are like similar. But when size of nodes are increased, heuristic simulated annealing algorithm has shown better than others two. The drawback of genetic algorithm has been improved by removing local minima with global minima.

CHAPTER 6

CONCLUSION AND FUTUER WORKS

6.1 Summary of the Study

This thesis has developed and implemented a simulated annealing based approach to deal with evacuation route planning for emergency traffic management. This is our main contributions of the thesis. The approach uses the vertices of graph of cities as search space to obtain the optimal route for EERP problem, the approach searches the initial feasible route for any vehicle to configure emergency evacuation route using simulated annealing algorithm in the map. Comparison with the genetic based approach in, Amani (2009) and the observations results in Chapter 4 show that the simulated annealing based approaches have provided a better performance in processing time, and is crucial for EERP problem. The heuristic method has been incorporated into the developed simulated annealing algorithm based approach to greatly enhance the performance in EERP problem. The approach is simpler and easier to implement than existing genetic algorithm based methods. As the method, it is able to quickly determine the optimal feasible route for EERP problem. Furthermore, the heuristic selecting method can also be used in other optimization methods which need to generate an initial feasible solution for calculation.

6.2 Limitations and Future Works

This new approach has promising results and the result can further be improved with the following future work:

1. Due to lack of resources, real data could not be used.
2. More algorithms can be compared.

3. Due the lack of real updated, heuristic data also causes some problem to configure emergency evacuation route.

4. The concept can be applied to design computer network configuration of any company according to their demand.

5. The thesis could not be solved multi variant configuration and contraflow. So it will be guideline for the future work.

REFERENCES

[1] A. Amani (2009), *Modeling Lane-Based Traffic Flow in Emergency Situation in the Presence of Multiple heterogeneous flows*, PhD Dissertation, University of Central Florida, USA

[2] M.R. Baharnemati and G.J. Lim (2011), Hurricane Evacuation Planning: A Network Flow Optimization Approach, *Proceeding of the 2011 Industrial Engineering Research Conference T. Doolen and E Van Aken eds*, pages 108-118.

[3] W. Fan and R. B. Machemehl (2006), Using a Simulated Annealing Algorithm to Solve the Transit Route Network Design Problem, *ASCE Journal of Transportation Engineering*, pages 122-132.

[4] M. Hui (2009), *Robot Route Planning in Dynamic Environment using a Simulated Annealing Based Approach,* Master thesis, Queensland University of Technology, Australia.

[5] G.L. Hamza-Lup, K.A. Hua, M. Le and R. Peng (2004), Enhancing Intelligent Transportation Systems to Improve and Support Homeland Security, *IEEE Transaction on Knowledge and Data Engineering*, pages 1-26.

[6] J. Hu and P. Hu (2010), An Approximate Annealing Search Algorithm to Global Optimization and Its Connection to Stochastic Approximation, *IEEE Proceeding of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Hugon, and E. Yiicesan, eds*, pages 1223-1234.

[7] S. Kim, S. Shekhar and B. George (2007), Evacuation Route Planning: Scalable Heuristics, *ACM*.

[8] S. Kim and S. Shekhar (2008), Contraflow Transportation Network Reconfiguration for Evacuation Route Planning, *IEEE Transaction on Knowledge and Data Engineering*, pages 1115-1129.

[9] L. Liu (2011), *Analysis and Evaluation of Household Pick-up and Gathering Behavior in No-Notice Evacuations*, PhD Dissertation, Virginia Polytechnic Institute and State University, Virginia.

[10] Q. Lu and B. George (2007), Evacuation route planning: a case study in semantic computing, *Evacuation planning*, pages 1-55.

[11] E. Zhou and X. Chen (2010), A New Populated-Based Simulated Annealing Algorithm, *IEEE Proceeding of the 2010 Winter Simulation Conference B. Johansson, S. Jain, J. Montoya-Torres, J. Hugon, and E. Yiicesan, eds,* pages 1211-1222.

[13] Wikipedia, the Free Encyclopedia (https://en.wikipedia.org 2012).

[12] X. Zhou, B. George, S. Kim, J.M.R. Wolff, Q. Lu, S. Shekhar (2010), Evacuation Planning: A Spatial Network Database Approach, *IEEE Computer Society Technical Committee on Data Engineering*, pages 1-6

APPENDICES

Source code of simulated annealing (heuristic)

Simulatedannealing.cs

```
using System;
using System.Collections;
namespace SimulatedAnnealing
{
	/// <summary>
	/// Simulated annealing c#
	/// created by Bal Krishna Subedi
	/// </summary>
	public class Annealing
	{
		public Annealing()
		{

		}

		void assign(int[]c, int[]n)
		{
			for(int i=0;i<c.Length;i++)
				c[i]=n[i];
		}

		Random rnd = new Random();
		/// <summary>
		/// compute a new next configuration and save the old next as
		current
		/// </summary>
		/// <param name="c">current configuration</param>
```

```
        /// <param name="n">next configuration</param>
        void computeNext(int[] c, int[] n)
        {
                for(int i=0;i<c.Length;i++)
                        n[i]=c[i];
                int i1 = (int)(rnd.Next(14))+1;
                int i2 = (int)(rnd.Next(14))+1;
                int aux = n[i1];
                n[i1]=n[i2];
                n[i2]=aux;
        }


public string StartAnnealing()
        {
                EERPDataReader.computeData();
                ArrayList list = new ArrayList();
                //primary configuration of cities
                int [] current={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14};
                //the next configuration of cities to be tested
                int []next=new int[15];
                int iteration =-1;
                //the probability
                double proba;
                double alpha =0.999;
                double temperature = 400.0;
                double epsilon = 0.001;
                double delta;
                double distance = EERPDataReader.computeDistance(current);

        //while the temperature didnt reach epsilon
                while(temperature > epsilon)
                {
                        iteration++;
```

```
            //get the next random permutation of distances
            computeNext(current,next);
            //compute the distance of the new permuted
            configuration
delta = EERPDataReader.computeDistance(next)-distance;
            //if the new distance is better accept it and assign
                        it
            if(delta<0)
            {
                        assign(current,next);
                        distance = delta+distance;
            }

            else
            {
                        proba = rnd.Next();
                        //if the new distance is worse accept it but
                        with a probability level
                        // if the probability is less than E to the
                        power -delta/temperature.
                        //otherwise the old value is kept
                        if(proba< Math.Exp(-delta/temperature))
                        {
                                    assign(current,next);
                                    distance = delta+distance;
                        }
            }
            //cooling proces on every iteration
            temperature *=alpha;
 //print every 400 iterations
            if (iteration%400==0)
            Console.WriteLine(distance);
}
```

```
			try
			{
				return "best distance is"+distance;
			}
			catch
			{
				return "error";
			}
		}
	}
}
```

SimulatedAnnealing.form1 code:

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.Data;

namespace SimulatedAnnealing
{
	/// <summary>
	/// created by Bal Krishna Subedi
	/// 2012
	/// </summary>
	public class Form1 : System.Windows.Forms.Form
	{
		private System.Windows.Forms.Button BtnAnnealing;
		/// <summary>
		/// Required designer variable.
		/// </summary>
		private System.ComponentModel.Container components = null;
        private Button button1;
        private Button button2;

Annealing ann = new Annealing();
		public Form1()
		{
			//
			// Required for Windows Form Designer support
			//
			InitializeComponent();

			//
```

```
            // TODO: Add any constructor code after InitializeComponent
            call
            //
        }


        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        protected override void Dispose( bool disposing )
        {
            if( disposing )
            {
                if (components != null)
                {
                    components.Dispose();
                }
            }
            base.Dispose( disposing );
        }

        #region Windows Form Designer generated code
        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
this.BtnAnnealing = new System.Windows.Forms.Button();
this.button1 = new System.Windows.Forms.Button();
this.button2 = new System.Windows.Forms.Button();
this.SuspendLayout();
//
// BtnAnnealing
```

```
//
this.BtnAnnealing.Location = new System.Drawing.Point(88, 48);
this.BtnAnnealing.Name = "BtnAnnealing";
this.BtnAnnealing.Size = new System.Drawing.Size(149, 23);
this.BtnAnnealing.TabIndex = 0;
this.BtnAnnealing.Text = "Go to Simulated Annealing!";
this.BtnAnnealing.Click += new
System.EventHandler(this.BtnAnnealing_Click);
//
// button1
//
this.button1.Location = new System.Drawing.Point(44, 96);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(225, 23);
this.button1.TabIndex = 1;
this.button1.Text = "Go to Hueristic Simulated Annealing!";
//
// button2
//
this.button2.Location = new System.Drawing.Point(44, 160);
this.button2.Name = "button2";
this.button2.Size = new System.Drawing.Size(225, 23);
this.button2.TabIndex = 2;
this.button2.Text = "Go to Gentic Algorithm!";
//
// Form1
//
this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
this.ClientSize = new System.Drawing.Size(372, 296);
this.Controls.Add(this.button2);
this.Controls.Add(this.button1);
this.Controls.Add(this.BtnAnnealing);
this.Name = "Form1";
this.ResumeLayout(false);
```

```
		}
		#endregion

		/// <summary>
		/// The main entry point for the application.
		/// </summary>
		[STAThread]
		static void Main()
		{
			Application.Run(new Form1());
		}

	private void BtnAnnealing_Click(object sender, System.EventArgs e)
		{
			string distance=ann.StartAnnealing();
		MessageBox.Show(distance);
		}
	}
}
```

EERP Data Reader Code:

```
using System;
using System.IO;

namespace SimulatedAnnealing
{
	/// <summary>
	/// Summary description for EERPDataReader.
	/// </summary>
	public class EERPDataReader
	{

				private static string file="EERP.txt";
				private static int dim=15;
				private static double[,]data;

			public static double computeDistance(int[] t)
			{
				int dist=0;
				for(int i=0;i<dim-1;i++)
					dist +=Convert.ToInt32(data[t[i],t[i+1]]);
				dist+=Convert.ToInt32(data[t[dim-1],t[0]]);
				return dist;
			}




	public EERPDataReader()
		{
```

```
            //
            // TODO: Add constructor logic here
            //
        }
        public static double[,] getData()
        {
            return data;
        }



        public static void computeData()
        {
            String line;
            data=new double[dim,dim];

            try
            {
             StreamReader str = new
    StreamReader(System.Windows.Forms.Application.StartupRoute+"\\EERP.txt");
                for(int i=0;i<dim;i++)
                {
                    line=str.ReadLine();
                string[]st=line.Split(' ');
                    for(int j=0;j<dim;j++)
                        data[i,j]= Double.Parse(st[j]);
                }
                str.Close();
            }
            catch(FileNotFoundException e)
            {
                Console.WriteLine(e.ToString());
            }
            catch(IOException e)
```

```
                {
                        Console.WriteLine(e.ToString());
                }

            }
        }
}
```

Input file for 10 nodes:

EERP.txt:

```
0.0 1.0 2.0 4.0 9.0 8.0 3.0 2.0 1.0 5.0 7.0 1.0 2.0 9.0 3.0
1.0 0.0 5.0 3.0 7.0 2.0 5.0 1.0 3.0 4.0 6.0 6.0 6.0 1.0 9.0
2.0 5.0 0.0 6.0 1.0 4.0 7.0 7.0 1.0 6.0 5.0 9.0 1.0 3.0 4.0
4.0 3.0 6.0 0.0 5.0 2.0 1.0 6.0 5.0 4.0 2.0 1.0 2.0 1.0 3.0
9.0 7.0 1.0 5.0 0.0 9.0 1.0 1.0 2.0 1.0 3.0 6.0 8.0 2.0 5.0
8.0 2.0 4.0 2.0 9.0 0.0 3.0 5.0 4.0 7.0 8.0 3.0 1.0 2.0 5.0
3.0 5.0 7.0 1.0 1.0 3.0 0.0 2.0 6.0 1.0 7.0 9.0 5.0 1.0 4.0
2.0 1.0 7.0 6.0 1.0 5.0 2.0 0.0 9.0 4.0 2.0 1.0 1.0 7.0 8.0
1.0 3.0 1.0 5.0 2.0 4.0 6.0 9.0 0.0 3.0 3.0 5.0 1.0 6.0 4.0
5.0 4.0 6.0 4.0 1.0 7.0 1.0 4.0 3.0 0.0 9.0 1.0 8.0 5.0 2.0
7.0 6.0 5.0 2.0 3.0 8.0 7.0 2.0 3.0 9.0 0.0 2.0 1.0 8.0 1.0
1.0 6.0 9.0 1.0 6.0 3.0 9.0 1.0 5.0 1.0 2.0 0.0 5.0 4.0 3.0
2.0 6.0 1.0 2.0 8.0 1.0 5.0 1.0 1.0 8.0 1.0 5.0 0.0 9.0 6.0
9.0 1.0 3.0 1.0 2.0 2.0 1.0 7.0 6.0 5.0 8.0 4.0 9.0 0.0 7.0
3.0 9.0 4.0 3.0 5.0 5.0 4.0 8.0 4.0 2.0 1.0 3.0 6.0 7.0 0.0
```