CHAPTER 6

IMPLEMENTATION OF INTELLIGENT TRANSPORTATION ROUTE PLANNING BY MULTI OBJECTIVE GENETIC ALGORITHM FOR OPTIMIZATION USING SPATIAL DATABASES

6.1 INTRODUCTION

Drivers wish to travel in the best conditions in terms of time, safety, economy and comfort. They need up-to-date, accurate, timely and relevant information to enable them to plan and manage their trips, either at the beginning of the journey, or, during travel, by changing their route or simply getting forecasted information concerning the rest of the trip. Efficient traffic control and traffic management are major concerns in any large city. The need for the collection and dissemination of traffic information to drivers and traffic control centres is the basis of modern integrated or dynamic traffic management. At present, most traffic control models utilize historical or averaged traffic flow data. However, for a given set of origin-destination data, some routes are temporarily faster than others due to congestion, construction and accidents in the other routes. This route planned using MOGA system gives the optimized route for the trip (Edward 2001).

Spatial databases are becoming increasingly important and are growing in numbers. It is typical for organizations that manage spatial data to have spatial, especially dedicated database to support this type of information.
Spatial database is used in Multi objective Genetic Algorithm to get
the optimized route for the transportation system.

Spatial data is made up of coordinates. A coordinate is a number
that denotes either:

- A position along an axis relative to an origin, given a unit of
  length.
- A direction relative to a base line or plane, given a unit of
  angular measure.

For example, latitude is a coordinate that denotes an angle relative
to the equatorial plane, usually in degrees. Longitude is a coordinate that
denotes an angle relative to the Greenwich meridian, also usually in degrees.

Figure 6.1 shows a geographic coordinate system where a location
is represented by the coordinate’s longitude 80 degree East and latitude 55
degree North.

![Figure 6.1 A geographic coordinate system](image-url)
A spatial reference system is a set of parameters that includes:

- The name of the coordinate system from which the coordinates are derived.
- The numeric identifier that uniquely identifies the spatial reference system.
- Coordinates that define the maximum possible extent of space that is referenced by a given range of coordinates.
- Numbers that, when applied in certain mathematical operations, convert coordinates received as input into values that can be processed with maximum efficiency.

The definitions of latitude and longitude, their points, lines, and planes of reference, units of measure, and other associated parameters are referred to collectively as a coordinate system. Coordinate systems can be based on values other than latitude and longitude. These coordinate systems have their own points, lines, and planes of reference, units of measure, and additional associated parameters.

The simplest spatial data item consists of a single coordinate pair that defines the position of a single geographic location. A more extensive spatial data item consists of several coordinates that define a linear path that a road or river might form. A third kind consists of coordinates that define the boundary of an area (Hoong Kee 204).

Each spatial data item is an instance of a spatial data type. The data type for coordinates that mark a single location is ST_Point, the data type for coordinates that define a linear path is ST_LineString and the data type for coordinates that define the boundary of an area is ST_Polygon. These types,
together with the other spatial data types, are structured types that belong to a single hierarchy.

Decisions are often evaluated on the basis of quality of the processes behind. It is in this context that geospatial information system (GIS) and spatial decision support system (SDSS) increasingly are being used to generate alternatives to aid decision makers in their deliberations. The intelligent transport systems (ITS) can carry and gather very useful information for delivery to users and perform other useful functions to deal with the travel. The introduction of subsystems that support the decision making to suit changing conditions is a logical important step in providing systems with improved functionalities.

Unfortunately, GIS and SDSS typically lack formal mechanisms to help decision-makers explore the solution space of their problem and thereby challenge their assumptions about the number and range of options available. We describe the use of a genetic algorithm to generate a range of options available. The ability of genetic algorithms to search a solution space and selectively focus on promising combinations of criteria makes them ideally suited to such complex spatial decision problems. Routing problems in car navigation systems are search problems for finding an optimal route from an origin to a destination on a road map within a time limit. In a practical system, when traffic congestion changes during driving, the route should be re-evaluated before the car reaches the next intersection. A representative solution to these search problems is the Dijkstra algorithm (DA). As the DA is an exact algorithm, it always determines the optimal route but cannot guarantee that realistic deadlines will be met.

Multi-objective optimization deals with solving optimization problems, which involve multiple objectives. Most real-world search and
optimization problems involve multiple objectives and should be ideally formulated. This should be solved as a multi-objective optimization problem. However, the task of multi-objective optimization is different from that of single-objective optimization. In multi-objective optimization, there is usually no single solution which is optimum with respect to all objectives. The resulting problem usually has a set of optimal solutions known as Pareto-optimal solutions, non-inferior solutions, or effective solutions (Deb 2002).

Moreover, one of the goals of multi-objective optimization is to find as many Pareto-optimal solutions as possible. Classical search and optimization methods usually work with a point-by-point principle and are required to be applied many times, each time finding one Pareto-optimal solution. Further, the efficiency of classical methods largely depends on the shape of the Pareto-optimal region, discreteness of the search space and presence of constraints among others. Population-based evolutionary algorithms have been found to be quite useful in solving multi-objective optimization problems, simply because of their ability to find multiple optimal solutions in a single simulation run (Hanaizumi 1992).

A Pareto optimal set is a set of solutions that are nondominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions. Because, they can be practical, when considering real-life problems, as the final solution of the decision-maker is always a trade-off. Pareto optimal sets can be of varied sizes but, the size of the Pareto set usually increases with the increase in the number of objectives. (Hiroyasu 2000).
Several methods for adapting GAs to cope with the simultaneous optimisation of a problem over a number of dimensions have been proposed, including the use of Pareto-ranking. The MOGA applied in this work uses Pareto-ranking as a means of comparing solutions across multiple objectives. The Pareto-optimal set of a multiobjective optimisation problem consists of all those vectors for which their components cannot be all simultaneously improved without having a detrimental effect on at least one of the remaining components. This is known as the concept of Pareto optimality, and the solution set is known as the Pareto-optimal set (or non-dominated set).

GAs have been recognised to be well-suited to multiobjective optimisation. MOGAs do not impose an ill-informed weighting process on the task of selecting a single optimal solution instead the concept of pareto-ranking can be applied in order to deliver a set of candidate solutions optimised for different combinations of criteria. Also the criteria we wish to optimise are interdependent within the definition of the database. Selecting the correct set of histograms to compactly represent a set of objects for the most reliable and unambiguous recognition results is a computationally demanding problem (Jensen 2003).

Pareto optimality defines how to determine the set of optimal solutions. A solution is Pareto-optimal if no other solution can improve one objective function without a simultaneous deterioration of at least one of the other objectives. A set of such solutions is called the Pareto-optimal front. Figure 6.2 shows the example of a Pareto front when minimizing two objectives f1 and f2. (Nondominated solutions are represented as hollow circles and dominated solution by filled circle) Evolutionary algorithms (EAs) have recently attracted much attention in the exploration of Pareto-optimal fronts. It is claimed that EAs are the preeminent search algorithms for such tasks.
Figure 6.2 Example of Pareto front when minimizing two objectives $f_1$ and $f_2$

6.1.1 Traffic Congestion

Traffic congestion is a condition on networks that occurs as use increases. This is characterized by slower speeds, longer trip times, and increased queueing. The most common example is the physical use of roads by vehicles. When traffic demand is high, the interaction between vehicles slows the speed of the traffic stream and consequently congestion is incurred. As demand approaches the capacity of a road (or of the intersections along the road), extreme traffic congestion sets in. When the vehicles are fully stopped for periods of time, this is colloquially known as a traffic jam. ITS can help to mitigate congestion by helping people plan travel better, by suggesting alternate routes and travel times, by keeping travelers well informed, by leveling traffic loads on roadways, and by helping to respond to and clear incidents more rapidly (Bertin 2005).
6.1.2 Environmental Impact

ITS helps reduce the environmental impact of road travel by optimizing trips, reducing congestion and crashes, improving vehicle and driver performance, and helping to manage the transportation system well. Environmental impact factor is tested with the real routes of Madurai with four vehicles using Multi Objective Genetic Algorithm. The route of the vehicles is decided by the proposed algorithm.

In general, ITS can provide travelers with better and more current information about the state of the transportation system, both for drivers and for users of public transport. This information will help travelers plan their trips better, make better connections.

Being a population-based approach, GA is well suited to solve multi-objective optimization problems. A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous, and multi-modal solutions spaces.

The crossover operator of GA may exploit structures of good solutions with respect to different objectives to create new nondominated solutions in unexplored parts of the Pareto front. In addition, most multi-objective GA does not require the user to prioritize, scale, or weigh objectives. Therefore, GA has been the most popular heuristic approach to multi-objective design and optimization problems. Jones et al. reported that 90% of the approaches to multi objective optimization aimed to approximate the true Pareto front for the underlying problem. A majority of these used a meta-heuristic technique, and 70% of all Meta heuristics approaches were based on evolutionary approaches (Fonseca 1998).
6.2 METROLOGICAL ATTRIBUTES

The Metrological attributes like Climate, Season and Temperature of Madurai city is given below

Climate – The Climate in Madurai city is generally hot and dry.

Season – The seasons cannot be explicitly being mentioned because rains are mainly obtained through the Northeast monsoon period only while the rains from Southwest Monsoon are erratic and unpredictable. So, Rainy months in a year for this division are three month only (September – November).

Temperature – In summer, the maximum temperature is about 37.5\(^{\circ}\)C and the minimum temperature is about 26.3\(^{\circ}\)C. And in winter, the maximum temperature is about 29.6\(^{\circ}\)C and the minimum temperature is about 20.9\(^{\circ}\)C.

6.2.1 Road Network Data Model

Location based Information

To be able to develop ITS applications location is a key concept because virtually all information used is related to locations. Information about locations and the spatial dimension is complex, and this type of information has traditionally been handled in GIS System. However in many ITS applications have to combine

- Information about locations
- Transport network and traffic information
Road Network

A Road Network is a collection of segments, joined to each other at nodes.

Segments

A segment is the smallest unit of road represented in a road network. A segment is the length of road between two end nodes.

Attributes

A segment may have a set of attributes attached to it. Attributes are name-value pairs, which contain information about the road such as identifier, name, type, surface type, etc. Attributes can be used to aid in making decisions about how to conflate segments.

Nodes

Nodes are the points that define the termini of segments. Start- and end-nodes are the only places at which segments may come into contact with one another.

Vertices

A vertex is any point in the definition of the linear geometry for a segment. Typically vertices are used to indicate changes in direction of the geometry of a segment. The first and last vertices of a segment occur at the end nodes of the segment. Figure 6.3 show Nodes and vertices in the Road network.
In the example, the source table is defined as

```sql
CREATE TABLE roads (  
id1 INT NOT NULL,  
id2 INT NOT NULL,  
latitude DOUBLE,  
longitude DOUBLE,  
name CHAR(30),  
PRIMARY KEY (id1, id2)
)
```

**Graphical Input and Output**

Traditional database systems deal with alphanumeric data types whose values can easily be entered through a keyboard and represented textually within a query result (e.g. a table). For a spatial database system, at least when it is to be used interactively, graphical presentation of SDT values in query results is essential, and entering SDT values to be used as “constants” in queries via a graphical input device is also important. Besides graphical representation of SDT values, another distinctive characteristic of querying a spatial database is that the goal of querying is in general to obtain a “tailored” picture of the space represented in the database, which means that the information to be retrieved is often not the result of a single query but
rather a combination of several queries. For example, for GIS applications, the user wants to see a map built by overlaying graphically the results of several queries.

Purpose of using DB2 Spatial extender:

Use of DB2 Spatial Extender to generate and analyze spatial information about geographic features, and to store and manage the data on which this information is based. A geographic feature (sometimes called feature in this discussion, for short) is anything in the real world that has an identifiable location, or anything that could be imagined as existing at an identifiable location. A feature can be: An object (that is, a concrete entity of any sort); for example, a river, forest, or range of mountains. Features exist in multiple environments. For example, the objects mentioned in the preceding list river, forest, mountain range belong to the natural environment. Other objects, such as cities, buildings, and offices, belong to the cultural environment. Still others, such as parks, zoos, and farmland, represent a combination of the natural and cultural environments.

6.3 STRATEGIES

To generate an initial population that consists of a small number of high-quality individuals (routes), we use the virus GA and the hybrid GA which are specialized for this problem.

- A part of an arterial road is regarded as a partial solution (call it virus). Only routes that include viruses are generated as the initial population.
• The routes calculated by the DA are used in generating the initial population. This is intended to favor the use of nonarterial roads and provide stable solutions.

Finally, we use the following strategy to generate nondominated solutions in successive generations.

• Offspring’s are generated from individuals with the best value of each objective function in the population.

6.4 DESIGN ISSUES OF MULTI-OBJECTIVE GA

6.4.1 Fitness Functions

The Weighted sum approach to solve a multi-objective optimization problem is to assign a weight \( w_i \) to each normalized objective function \( z_i(x) \) so that the problem is converted to a single objective problem with a scalar objective function as follows:

\[
\min z = w_1 z_1 (x) + w_2 z_2 (x) + \ldots + w_k z_k (x)
\]  

(6.1)

where \( z_i(x) \) is the normalized objective function \( z_i(x) \) and \( \sum wi = 1 \). This approach is called the priori approach since the user is expected to provide the weights. Solving a problem with the objective function (1) for a given weight vector \( w = \{w_1, w_2, \ldots, w_k\} \) yields a single solution, and if multiple solutions are desired, the problem must be solved multiple times with different weight combinations. The main difficulty with this approach is selecting a weight vector for each run.

6.4.2 Pareto-Ranking Approaches

Pareto-ranking approaches explicitly utilize the concept of Pareto dominance in evaluating fitness or assigning selection probability to solutions.
The population is ranked according to a dominance rule, and then each solution is assigned a fitness value based on its rank in the population, not its actual objective function value. Note that herein all objectives are assumed to be minimized. Therefore, a lower rank corresponds to a better solution.

The first Pareto ranking technique was proposed by Goldberg (1989) as follows:

Step 1: Set \( i = 1 \) and \( TP = P \).

Step 2: Identify non-dominated solutions in \( TP \) and assigned them to \( Fi \).

Step 3: Set \( TP = TPF_i \). If \( TP = \emptyset \) go to Step 4, else set \( i = i + 1 \) and go to Step 2.

Step 4: For every solution \( x \in P \) at generation \( t \), assign rank \( r1(x, t) = i \) if \( x \in Fi \).

In the procedure above, \( F1, F2, \ldots \) are called nondominated fronts, and \( F1 \) is the Pareto front of population \( P \). \( TP \) is the Total Population. NSGA also classifies the population into nondominated fronts using an algorithm similar to that given above. Then a dummy fitness value is assigned to each front using a fitness sharing function such that the worst fitness value assigned to \( Fi \) is better than the best fitness value assigned to \( Fi+1 \). NSGA-II, a more efficient algorithm, named the fast non-dominated-sort algorithm, was developed to form non-dominated fronts.

6.5 PROBLEM FORMULATION

A road network \( Net \) with a set of nodes \( N = \{N1, N2, \ldots\} \) and a set of links \( L = \{L1, L2, \ldots\} \). Each node has either with signal or without signal property, and each link has length of link, road class, and number of lanes properties. Examples of road class are listed in Table 6.1. A problem has been
formulated by treating \((\text{Net, S, D, F, C})\), where \(S\) and \(D\) are source and a destination, \(F\) is a set of objective functions, and \(C\) is a set of constraints.

The constraints can be classified into hard constraints, corresponding to traffic regulations, and soft constraints (Table 6.2), corresponding to the ease of driving. Hard constraints must be satisfied, but soft constraints can be violated at the cost of a penalty. We regard the route with the lowest total penalty as the easiest one for drivers.

Consider the set of objective functions

\[
F = \{f_1, f_2, f_3\} \quad (6.2)
\]

Where \(f_1\) and \(f_2\) are respectively the length and the travel time of the route, and \(f_3\) is the congestion. Let \(l_i\) and \(T_i(t)\) respectively be the length and the travel time at time \(t\) of link \(L_i\) along the route \((S, N1, ..., Nn, D)\). Figure 6.4 shows the relationships between these quantities. Functions \(f_1\) and \(f_2\) can be written as follows:

\[
f_1 = \sum_{i=1}^{n-1} l_i
\]

\[
f_2 = \sum_{i=1}^{n-1} T_i(t-1)
\]

where \(t_{i-1} = t_0\) for \(i = 1\)

\[
f_3 = \text{sum} (P_s) + \text{Avg} (P_{lo}) + \text{Avg} (P_{hg}) + \text{Sum} (P_{cong}) + \text{Sum} (P_{ei})
\]

where \(P_s\), \(P_{lo}\), \(P_{hg}\), \(P_{cong}\), \(P_{ei}\) are the penalties listed in the Table 6.2 and \(\text{Sum}(X)\) and \(\text{Avg}(X)\) are respectively the sum and average of the penalty along the route chosen.
Table 6.1 Examples of various types of road

<table>
<thead>
<tr>
<th>Category</th>
<th>Road Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>Local road</td>
</tr>
<tr>
<td>R2</td>
<td>City highways</td>
</tr>
<tr>
<td>R3</td>
<td>National highways</td>
</tr>
</tbody>
</table>

Table 6.2 Constraints and their penalties of road

<table>
<thead>
<tr>
<th>Object of constraint</th>
<th>Constraint</th>
<th>Penalty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
<td>To reduce signals</td>
<td>$P_s$</td>
</tr>
<tr>
<td>Link</td>
<td>To select local road</td>
<td>$P_{lo}$</td>
</tr>
<tr>
<td></td>
<td>To select highways</td>
<td>$P_{hg}$</td>
</tr>
<tr>
<td></td>
<td>To reduce congestion</td>
<td>$P_{cong}$</td>
</tr>
<tr>
<td>Route</td>
<td>To reduce Environmental Impact</td>
<td>$P_{ei}$</td>
</tr>
</tbody>
</table>

From the source to the destination we have ‘N’ numbers of node start from 1 to n. Time varies from $t_0$ to $t_{n+1}$. The relationship between node, link, length, time and Travel time from source to destination is shown in the Figure 6.4.

Node $S \rightarrow N1 \rightarrow \ldots \rightarrow Nn \rightarrow D$

Link $L_1, L_2, \ldots, L_{n+1}$

Length $l_1, l_2, \ldots, l_{n+1}$

Time $t_0, t_1, \ldots, t_{n+1}$

Travel Time $T_1(t0), T_2(t1), \ldots, T_{n+1}(tn)$

Figure 6.4 Relationship between node, link, length, time and travel time
These days, most super computers act as parallel computers. Evolutionary computations (ECs) are the applications that have parallelism implicitly. Therefore, the models of parallel ECs are very important. So a dynamic route planning problem is formulated as a multi-objective problem with predicted traffic and show how it can be solved using a genetic algorithm. A GA-based route-planning algorithm in a dynamic environment has already been reported, but, it gives only one compressed solution because its objective function is the sum of the three objective functions. It also very rapidly generates detours using other routes whenever traffic conditions change. So the route cannot be the global optimum. The method proposed here gives the Pareto-optimal set by using both the predicted traffic and a Multi Objective GA. In present GIS, route finding modules do not concentrate on Improved Safety, Reduced congestion and Reduced Environmental Impact. These objectives are satisfied by using the Multi objective GA. The general procedure of the proposed method is given below. The prediction and the interpolation have already been reported and mutation is not used in this method.

6.6 METHODOLOGY

Route planning problem in a real road network was a multi objective optimization problem for obtaining the optimal routes from an origin to a destination. Three objective functions are used to optimize simultaneously in this problem: Route length, travel time and congestion. Congestion was calculated based on the constraint and the penalty of the road. Fitness function was calculated based on the weighted sum approaches to solve a multi-objective optimization problem and assign a weight to each normalized objective function.

Crossover was performed by randomly selecting a pair of sub-populations and two crossover points. The population was ranked according
to a dominance rule, and then each solution is assigned a fitness value based on its rank in the population. Whenever traffic conditions change, that rapidly generates detour using other routes. Pareto ranking technique was used to find the rank in the population. Pareto-ranking approaches explicitly utilize the concept of Pareto dominance in evaluating fitness or assigning selection probability to solutions.

6.6.1 Pseudo Code

procedure MOGA
Input map and map database;
Input origin and destination;
Input traffic data;
Perform prediction and interpolation of the travel time for all links;
Generate a population of viruses;
Generate a population of individuals using the Dijkstra Algorithm;
for generation = 1 to Number of Generation (repeat until meeting deadline)
begin
MPI is initialized;
If (rank = 0) // ie, the node is considered as master
begin
For (all ranks other than zero)
begin
MPI_Send(“specific fitness function & some part of population”);
end;
For (all ranks other than zero)
begin
MPI_Recv(“the evaluated results”);
end
end
end
else // ie, the node is considered as slave
begin
    MPI_Recv ("specific fitness function & some part of population");
    Calculate fitness values of individuals;
    Select two individuals at random;
    single-point crossover;
    MPI_Send ("the evaluated results");
end
end
Select nondominated solutions from the population;
Output them as recommended routes;
end procedure;

6.6.2 Crossover Operator

In order for crossover to function as desired we needed to ensure that two good sub-populations produced fit/fitter offspring sub-populations. Crossover was performed by randomly selecting a pair of sub-populations and two crossover points (corresponding to rows of the arrays). Suppose we had $N$ individuals in each sub-population. Two crossover points were used, the first was $N/2$ and the second was generated randomly (from the range $(N,N/2)$). This was to ensure a balance between randomly crossing and selectively crossing fit groups of individual.

6.7 RESULTS AND DISCUSSION

The performance of the proposed method was evaluated using road data. The map of Madurai used in the experiment includes 678 nodes and 1200 edges that represent highway and freeway segments for a 16 square mile section of Madurai area. The Multi-Objective GA without predicted traffic
(MOGA) and the Single Objective GA without predicted traffic (SOGA) are compared and shown in Table 6.3. The objective functions are Travel time (measured in second), the Route length (measured in Kilometer), reduced congestion (measured in meter) and reduced environmental Impact (measured in litre) and their MOGA and SOGA algorithm are compared.

Table 6.3 Objective function and performance indices of routes

<table>
<thead>
<tr>
<th>Objective Function</th>
<th>MOGA</th>
<th>SOGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel time (sec)</td>
<td>219</td>
<td>224</td>
</tr>
<tr>
<td>Route length (km)</td>
<td>19.57</td>
<td>19.57</td>
</tr>
<tr>
<td>Reduced congestion (m)</td>
<td>462</td>
<td>481</td>
</tr>
<tr>
<td>Reduced Environmental Impact (litre)</td>
<td>3.8</td>
<td>4.2</td>
</tr>
</tbody>
</table>

From Table 6.3, it is found that the proposed method is better than the route found by SOGA from the perspective of Travel time, Congestion and Environmental impact. Thus, it is clear that the proposed route consumes less fuel and less travel time when compared to the ordinary route.

Table 6.4 Travel time and performance indices of routes

<table>
<thead>
<tr>
<th>Iterations</th>
<th>MOGA</th>
<th>SOGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>219</td>
<td>224</td>
</tr>
<tr>
<td>2</td>
<td>234</td>
<td>245</td>
</tr>
<tr>
<td>3</td>
<td>227</td>
<td>236</td>
</tr>
<tr>
<td>4</td>
<td>222</td>
<td>231</td>
</tr>
<tr>
<td>5</td>
<td>225</td>
<td>234</td>
</tr>
</tbody>
</table>
The travel time for the route proposed by MOGA from a given source to destination is relatively less than the travel time for the route proposed by SOGA in order of Seconds. Table 6.4 shows that the Travel time for MOGA and SOGA and performance indices of routes for five iterations. Based on the route length from a given source to destination the MOGA performs better than the SOGA. Comparisons of MOGA and SOGA based on Travel time is shown in the Figure 6.5. Here the X –axis represents the number of iterations taken for the computation and Y- axis represents the travel time from the source node to destination node.

![Figure 6.5 Comparison of MOGA and SOGA based on Travel time](image-url)
Figure 6.6 Comparison of MOGA and SOGA based on Length

Table 6.5 Length and performance indices of routes

<table>
<thead>
<tr>
<th>Iterations</th>
<th>MOGA</th>
<th>SOGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.21</td>
<td>22.55</td>
</tr>
<tr>
<td>2</td>
<td>20.36</td>
<td>21.36</td>
</tr>
<tr>
<td>3</td>
<td>19.12</td>
<td>20.85</td>
</tr>
<tr>
<td>4</td>
<td>18.84</td>
<td>19.21</td>
</tr>
<tr>
<td>5</td>
<td>18.65</td>
<td>18.96</td>
</tr>
</tbody>
</table>

The length for the route proposed by MOGA for a given source to destination is relatively less than the one proposed by SOGA in order of Seconds. Table 6.5 shows the Length for MOGA and SOGA and performance indices of routes for five iterations. Based on the route length for a given source to destination the MOGA performs better than the SOGA. Comparison of MOGA and SOGA based on Length is shown in the Figure 6.6. Here the
X axis represents the number of iterations taken for the computation and Y axis represents the distance traveled in Kilometers from the source node to destination node.

**Table 6.6 Reduced congestion and performance indices of routes**

<table>
<thead>
<tr>
<th>Iterations</th>
<th>MOGA</th>
<th>SOGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>459</td>
<td>478</td>
</tr>
<tr>
<td>2</td>
<td>462</td>
<td>481</td>
</tr>
<tr>
<td>3</td>
<td>465</td>
<td>483</td>
</tr>
<tr>
<td>4</td>
<td>460</td>
<td>485</td>
</tr>
<tr>
<td>5</td>
<td>463</td>
<td>484</td>
</tr>
</tbody>
</table>

**Figure 6.7 Comparisons of MOGA and SOGA based on reduced congestion**
The reduced congestion for the route proposed by MOGA for a given source to destination is relatively less than the route proposed by SOGA in order of Seconds. Table 6.6 shows that the reduced congestion for MOGA and SOGA and performance indices of routes for five iterations. Based on the reduced congestion for a given source to destination the MOGA performs better than the SOGA. From the Table 6.6, it is clear that the MOGA performs better than SOGA. Comparison of MOGA and SOGA based on reduced congestion is shown in the Figure 6.7. Here the X axis represents the number of iterations taken for the computation and Y axis represents the congestion from the source node to destination node.

6.8 CONCLUSION

Car navigation equipment in practical use has treated the route planning problem as a single-objective problem, and users of this equipment have had to decide the priority of objective functions before planning. In this chapter, the problem was described as a multi-objective problem so that a user can chose a favorite route after looking at feasible ones. Furthermore, the route planning in three-dimensional travel time space including predicted traffic was performed in a wide area road network. The proposed MOGA method is better than the route found by SOGA in terms of Travel time, Congestion and Environmental impact using multi objective.