CHAPTER 1

INTRODUCTION

1.1 INTELLIGENT TRANSPORTATION SYSTEMS

The construction and operation of transportation systems is being transformed by computers, sensors, and communication technology - collectively called Information technology (IT). The application of IT to surface transportation is called Intelligent Transport Systems (ITS). ITS provides the ability to gather, organize, analyze, use, and share information about transportation systems. It is a broad range of diverse technology applied to transportation to make the systems safer, more efficient, reliable and environmental friendly, without physically altering the existing infrastructure. ITS is an emerging global phenomenon benefiting both the public and the private sectors (Ben-Activ 1998).

In the modern world, this ability is crucial to the effective and economical construction and operation of transportation systems and to their efficient use.

- IT can be very helpful in conceiving, planning, and building new parts of the transport system. This use of IT is not specifically ITS, but it is very helpful in laying the groundwork for introducing ITS.
- ITS is being incorporated by manufacturers in “intelligent equipment ” that can be installed as part of the transportation infrastructure to gather and disseminate traveller information,
to control traffic signals and variable message signs, to electronically collect tolls, and to help manage the system.

- ITS provides vital support in operating transportation systems, including traffic management, pavement monitoring, oversight of system maintenance, and more effectively and reliably managing public transport.

- ITS can store and evaluate archived data about the transportation system that is useful to planners who are evaluating transportation system improvements or to other evaluating safety aspects of the roadway.

- ITS also provides a wide array of in-vehicle technology to improve the safety, productivity, and comfort of road travel.

1.1.1 Benefits of ITS

Basically, ITS provides two kinds of benefits. One is the resolution of traffic problems which includes traffic congestion, air pollution, and traffic accidents. The other is improved services for users and increased efficiency of the transportation system and its operators.

1.1.2 Route Queries

Transportation systems such as road networks in a country are an important application of a spatial database system. Route queries are the most important classes of queries for such an application. In large metropolitan areas, the common problem is traffic congestion. The ability of finding the fastest or least expensive route from one point to another may prove to be essential in the near future. Finding such a path is an example of route query (Jin Suvng Yoo 2005).
1.1.3 Geographical Information System in Transportation Planning

The road network in India is huge with more than 3.01 million kilometers of road length with 34,608 km of National Highway, 1,28,622 km of State Highway and informal network of 27,37,080 km, operated in vastly different social, economic and climatic environments. The road network planning based on the travel demand requirements in the country could not be adopted merely due to lack of relevant data needed for it.

The major planning in different aspects of road network can be attributed to the lack of availability of large volume of data required for this purpose. Even if this data is made available, the next problem is how to manage and access that data. The valuable information related to existing transport infrastructure is scattered all over the country at different organizations. The attribute data of National Highways and State Highways network is available in pieces in different organizations of the state level system, and it is rarely utilized effectively by planners (Luo Qi 2008).

At present any exercise on sufficiency of the existing network in the regional context or nation wide plan generation for primary network like expressway cannot use any of the existing data. Thus, practically the present available data at a large number of locations in all possible formats are waste, and resources spent for collection and maintenance of this data is draining the economy as a routine ritual and not fulfilling the objectives. Highway networks face deterioration problem due to the lack of funds for infrastructure. The adoption of newly emerging technologies such as Geographic Information System (GIS) can help to improve the decision making process in this area for better use of the available limited funds.

Geographical Information Systems (GIS) are becoming more widely used in transportation planning agencies, especially among
metropolitan transportation organizations. In many developed countries, highway maintenance management is becoming a critical issue. Many more authorities are now able to use GIS for Highways and transport management, due to falling costs. GIS offer transport planners a medium for storing and analyzing data on population densities, land uses, travel behavior and others.

The use of GIS for transportation applications is widespread. Typical applications include highway maintenance, traffic modeling, accident analysis, and route planning and environmental assessment of road schemes. A fundamental requirement for most transportation GIS is a structured road network. The typical GIS represent the world as a map. The major requirements and issues surrounding GIS management technology are building and maintaining a database, selecting and upgrading hardware and software, using the technology to solve problems, funding, networking, providing access, and others. Standard GIS functions include thematic mapping, statistics, charting, matrix manipulation, decision support system, modeling and algorithms and simultaneous access to several databases.

1.2 SPATIAL DATABASES

A spatial database is a database that is optimized to store and query data related to objects in space, including points, lines and polygons. While typical databases can understand various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types. These are typically called geometry or feature. The Open Geospatial Consortium created the Simple Features specification and sets standards for adding spatial functionality to database systems (Guenther 1990).

Spatial databases provide structures for storage and analysis of spatial data. Spatial data comprise objects in multi-dimensional space. Storing spatial data in a standard database would require excessive amounts of space.
Queries to retrieve and analyze spatial data from a standard database would be long and cumbersome leaving a lot of room for error. Spatial databases provide much more efficient storage, retrieval, and analysis of spatial data.

The drawbacks of traditional approach in handling geographic data and computer aided design data are the cost of transforming data from one form to another form for processing; and the need to read the entire data even if only parts of it are required and it is lacking of user interaction.

Spatially related collections of objects are separated as partitions and networks. A partition can be viewed as a set of region objects that are required to be disjoint. Partitions can be used to represent thematic maps. A network can be viewed as a graph embedded in the plane, consisting of a set of point objects, forming its nodes, and a set of line objects describing the geometry of the edges. Systems of spatial data types, or spatial algebras, can capture the fundamental abstractions for point, line, and region, together with relationships between them and operations for composition (e.g., forming the intersection of regions) commonly called as geometric data.

1.2.1 Geographic Information System

Geographic Information System (GIS) is a relational database manager specialized for handling spatial data (Abel 1989). In a Relational Database Management System (RDBMS), information is stored in tables that are linked in an intuitive way. GIS products tend to be an additional layer "on top" of a commercial RDBMS. The GIS then provides applications programmers with operations useful for analysis on the spatial data stored in the underlying RDBMS. One example would be the ability to define basic graphic entities common in spatial analysis and to manage the storage and retrieval of these entities, which are referred to as points, lines, and polygons.
Another example would be the ability to compute the areas of geographic features stored as polygons.

1.2.2 Spatial Database Management System

Spatial Database Management System (SDBMS) provides the capabilities of a traditional database management system (DBMS) while allowing special storage and handling of spatial data (Shekhar 1997). SDBMS can be characterized as follows:

- SDBMS is a software module that can work with an underlying database management system, for example, an object-relational database management system or object-oriented database management system.

- SDBMSs support multiple spatial data models, commensurate spatial abstract data types (ADTs), and a query language from which these ADTs are callable.

- SDBMSs support spatial indexing, efficient algorithms for spatial operations, and domain-specific rules for query optimization.

1.2.3 GIS and SDBMS

The Geographic Information System (GIS) is the principal technology motivating interest in Spatial Database Management System (SDBMS). GIS provides a convenient mechanism for the analysis and visualization of geographic data. Geographic data is spatial data whose underlying frame of reference is the earth’s surface. The GIS provides a rich set of analysis functions which allow a user to affect powerful transformations on geographic data. The rich array of techniques that geographers have
packed into the GIS is the reason behind its phenomenal growth and multidisciplinary applications. SDBMSs are also designed to handle very large amounts of spatial data stored on secondary devices, using specialized indices and query processing techniques. Finally, SDBMSs inherit the traditional DBMS functionality of providing a concurrency control mechanism to allow multiple users to simultaneously access shared spatial data, while preserving the consistency of that data (Chawla 2001).

1.2.4 Spatial Representation

For modeling single objects, the fundamental abstractions are point, line, and region. A point represents an object for which only its location in space, but not its extent, is relevant. For example, a city may be modeled as a point in a model describing a large geographic area. A line (in this context always to be understood as meaning a curve in space, usually represented by a polyline, a sequence of line segments) is the basic abstraction for facilities for moving through space, or connections in space (roads, rivers, etc.). A region is the abstraction for something having an extent in 2d-space, e.g. a country, a lake, or a national park. A region may have holes and may also consist of several disjoint pieces (Danzhou Liu 2002). Figure 1.1 shows the three basic abstractions for single objects.

![Figure 1.1 The three basic abstractions of point, line and polygon](image)

The two most important instances of spatially related collections of objects are partitions (of the plane) and networks. A partition can be viewed as a set of region objects that are required to be disjoint. The adjacency
relationship is of particular interest, that is, there exist often pairs of region objects with a common boundary. Partitions can be used to represent thematic maps. A network can be viewed as a graph embedded into the plane, consisting of a set of point objects, forming its nodes, and a set of line objects describing the geometry of the edges. Networks are ubiquitous in geography, for example, highways, rivers, public transport, or power supply lines. Partitions and Networks show in the Figure 1.2.

![Figure 1.2 Partitions and networks](image)

The base geometry subtypes, and the homogeneous collection subtypes are two categories of spatial data (David 1992).

The base geometries include:

Points: A single point. Points represent discrete features that are perceived as occupying the locus where an east-west coordinate line (such as a parallel) intersects a north-south coordinate line (such as a meridian). For example, suppose that the notation on a world map shows that each city on the map is located at the intersection of a parallel and a meridian. A point could represent each city.

Linestrings: A line between two points. It does not have to be a straight line. Linestrings represent linear geographic features (for example, streets, canals, and pipelines).
Polygons: A polygon or surface within a polygon. Polygons represent multisided geographic features.

The homogeneous collections include:

Multipoints: A multiple point geometry collection. Multipoints represent multipart features whose components are each located at the intersection of an east-west coordinate line and a north-south coordinate line (for example, an island chain whose members are each situated at an intersection of a parallel and meridian).

Multilinestrings: A multiple curve geometry collection with multiple strings. Multilinestrings represent multipart features that are made up (for example, river systems and highway systems).

Multipolygons: A multiple surface geometry collection with multiple polygons. Multipolygons represent multipart features made up of multisided units or components (for example, the collective farmlands in a specific region, or a system of lakes).

1.2.5 Spatial Data Types

Systems of spatial data types, or spatial algebras, can capture the fundamental abstractions for point, line and region together with relationships between them and operations for composition (e.g. forming the intersection of regions). As an example spatial algebra we briefly consider the ROSE algebra. The ROSE algebra offers three data types called points, lines, and regions (Ralf Hartmut Guting 1994).

1.2.6 Spatial Relationships

Among the operations offered by spatial algebras, spatial relationships are the most important ones. For example, they make it possible
to ask for all objects in a given relationship with a query object, e.g. all objects within a window. One can distinguish several classes (Aref 1991).

- Topological relationships, such as adjacent, inside, disjoint, are invariant under topological transformations like translation, scaling, and rotation.
- Direction relationships, for example, above, below, or north_of, southwest_of, etc.
- Metric relationships, e.g. “distance < 100”.

1.2.7 Advantages of Spatial Databases

Able to treat your spatial data like anything else in the DB

- transactions
- backups
- integrity checks
- less data redundancy
- fundamental organization and operations handled by the DB
- multi-user support
- security/access control
- locking

1.3 HIGH PERFORMANCE CLUSTER

The name of the game in high-performance computing is parallelism. Parallelism is the quality that allows something to be done in parts that work independently rather than a task that has so many interlocking dependencies that it cannot be further broken down. Parallelism operates at two levels: hardware parallelism and software (algorithmic) parallelism.
Hardware parallelism deals with the CPU of an individual system and how one can squeeze performance out of sub-components of the CPU that can speed up the code (Aaron 2003).

At another level, there is the parallelism that is gained by having multiple systems working on a computational problem in a distributed fashion. This may be either fine grained or coarse grained. Whereas software parallelism is the ability to find well-defined areas in a problem one wants to solve that can be broken down into self-contained parts. These parts are the program elements that can be distributed and gives the big speedup that one wants to get out of a high-performance computing system (Buyya 1999).

Clusters are based on the concepts of both Hardware and software parallelism. It exploits the advantages of both. Clusters are in fact quite simple. They are a bunch of computers tied together with a network working on some large problem that has been broken down into smaller pieces. There are a number of different strategies you can use to tie them together.

As cluster computing and related technologies became mature, more and more High Performance Computing systems were built by this paradigm. Intel offers a broad range of processors and standards based building blocks for building HPC clusters that can meet high-performance computing needs and project budget constraints – from low-cost, workstation clusters to high-end performance clusters.

1.3.1 Cluster Functionality

A cluster can offer high performance, high throughput, and high availability. A cluster can be expanded and is, therefore, scalable. Cluster computing enables an organization to expand their processing power using standard existing components, i.e. PCs and workstations, which are off-the-
shelf commodity hardware and software devices available at low cost. The organization need not procure proprietary hardware with high performance capabilities. More importantly, the organization can conserve and preserve the existing hardware stock and yet assemble them into a cluster for higher performance computing, adding equivalent additional commodity components procured readily from the market. Thus, the user organizations can themselves leverage on their existing hardware to build high performance clusters from their own resources a 10 or 100 factor cost reduction in comparison with high performance supercomputing mainframes (Mark Baker 2000).

1.3.2 Categories of Cluster

Classification of clusters can be made on the basis of various independent criteria such as functionality, performance, availability, node ownership, type of node hardware, node operating system, node configuration, and also level or layering of clustering.

In terms of performance, a cluster can be classified into either high performance clusters or high availability clusters. In terms of nodes, clusters can be classified into dedicated clusters and non dedicated clusters. In the case of dedicated cluster, all the nodes are dedicated fully for the cluster, with no independent usage. In the case of non-dedicated or shared clusters, the nodes are used independently by the respective end users and simultaneously CPU cycles are stolen for cluster functionality. This is possible because most of the time the CPU is not fully utilized – most CPU cycles can be stolen for cluster functionality purposes. Non dedicated node hardware is further classified on the basis of whether the nodes are PCs or Workstations. Classification based on operating system is on the basis of whether the node has LINUX, Windows and others.
1.4 GENETIC ALGORITHMS

Genetic Algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics. As such they represent an intelligent exploitation of a random search used to solve optimization problems. Although randomized, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, specially those follow the principles first laid down by Charles Darwin of "survival of the fittest". Since, in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones (Prinetto 1993).

The GA simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

The GA is a search procedure that optimizes some objective function $f$ by maintaining a population $P$ of candidate solutions and employing operations inspired by genetics (called crossover and mutation) to generate a new population from the previous one. Generally, the candidate solutions are encoded as bit strings.

The three most important aspects of using genetic algorithms are: (1) definition of the objective function, (2) definition and implementation of the genetic representation, and (3) definition and implementation of the genetic operators. Once these three have been defined, the genetic algorithm
should work fairly well. Beyond that one can try many different variations to improve performance, find multiple optima (species - if they exist), or parallelize the algorithms (Nowostawsk 1999).

The most important part of a GA is the fitness function which gives an evaluation for each candidate as to how good a solution it is. When candidates are chosen for crossover they are picked in proportion to their relative fitness to the entire population. Crossover and mutation are the operators of GA.

GAs is based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations:

- Individuals in a population compete for resources and mates.
- Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.
- Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
- Thus each successive generation will become more suited to their environment.

1.4.1 Search Space

A population of individuals is maintained within search space for a GA, each representing a possible solution to a given problem. Each individual is coded as a finite length vector of components, or variables, in terms of some alphabet, usually the binary alphabet {0, 1}. To continue the genetic
analogy these individuals are likened to chromosomes and the variables are analogous to genes. Thus a chromosome (solution) is composed of several genes (variables). A fitness score is assigned to each solution representing the abilities of an individual to `compete'. The individual with the optimal (or generally near optimal) fitness score is sought. The GA aims to use selective `breeding' of the solutions to produce `offspring' better than the parents by combining information from the chromosomes (Tan 2001).

1.4.2 GA Benefits

Nearly everyone can gain benefits from Genetic Algorithms, once they can encode solutions of a given problem to chromosomes in GA, and compare the relative performance (fitness) of solutions. An effective GA representation and meaningful fitness evaluation are the keys of the success in GA applications. The appeal of GAs comes from their simplicity and elegance as robust search algorithms as well as from their power to discover good solutions rapidly for difficult high-dimensional problems. GAs is useful and efficient when

- The search space is large, complex or poorly understood
- Domain knowledge is scarce or expert knowledge is difficult to encode to narrow the search space
- No mathematical analysis is available
- Traditional search methods fail

The advantage of the GA approach is the ease with which it can handle arbitrary kinds of constraints and objectives; all such things can be handled as weighted components of the fitness function, making it easy to adapt the GA scheduler to the particular requirements of a very wide range of possible overall objectives.
1.4.3 Parallel Genetic Algorithm

Genetic algorithms in general, and especially parallel genetic algorithm, have been universally considered to be very important in the development of the new generation of IT applications. Genetic algorithms - inspired by the analogy of evolution and population genetics - have demonstrated to be particularly successful in the optimization, classification and control of very-large-scale and varied data. The parallel nature of genetic algorithms suggests parallel processing as the natural route to explore. Parallel genetic algorithms not only provide the basis for tackling problems in a wider range of fields which were previously considered to be virtually unsolvable but also creating a new paradigm within this area and thus establishing a new and promising field of research.

1.5 Multi Objective Genetic Algorithm

In Multiple-objective problems, the objectives are generally conflicting, preventing simultaneous optimization of each objective. Many, or even most, real engineering problems actually do have multiple objectives, i.e., minimize cost, maximize performance, maximize reliability, etc. These are difficult but realistic problems. GA are a popular meta-heuristic that is particularly well-suited for this class of problems. Traditional GA are customized to accommodate multi-objective problems by using specialized fitness functions and introducing methods to promote solution diversity (Amar 2004).

There are two general approaches to multiple-objective optimization. One is to combine the individual objective functions into a single composite function or move all but one objective to the constraint set. In the former case, determination of a single objective is possible with methods such as utility theory and weighted sum method, but the problem lies
in the proper selection of the weights or utility functions to characterize the
decision-maker’s preferences. In practice, it can be very difficult to precisely
and accurately select these weights, even for someone familiar with the
problem domain. Compounding this drawback is that scaling amongst
objectives is needed and small perturbations in the weights can sometimes
lead to quite different solutions. In the latter case, the problem is that to move
objectives to the constraint set, a constraining value must be established for
each of these former objectives. This can be rather arbitrary. In both cases, an
optimization method would return a single solution rather than a set of
solutions that can be examined for trade-offs. For this reason, decision-makers
often prefer a set of good solutions considering the multiple objectives
(Fonseca 1993).

The second general approach is to determine an entire Pareto
optimal solution set or a representative subset. 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 since 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.

1.6 PROBLEM FORMULATION

Decisions are often evaluated on the basis of quality of the
processes behind them. It is in this context that Geospatial Information
System (GIS) and Spatial Decision Support System (SDSS) are being
increasingly used to generate alternatives to aid decision-makers in their
deliberations. The intelligent transport system (ITS) can carry and deliver
very useful information to users and can perform other useful functions to deal with the travel. The principle of a traditional route search is unsuitable for the decision making to find the optimal paths. The route planning includes multiple objectives such as length of the route, time for the travel and the ease of overcoming the temporary obstacles. While constructing a route, there are two conflicting goals. On one hand, the route should be as short as possible. On the other hand, the route should go via the most relevant entities for the travel. It must also concentrate on the uncertainties.

Network analyses in GIS provide strong decision support for users in searching optimal route, finding the nearest facility and determining the service area. Searching optimal path is an important advanced analysis function in GIS. In the present GIS route finding modules, heuristic algorithms have been used to carry out their search strategy. A representative solution to these search problems is the Dijkstra’s Algorithm (DA). DA is an exact algorithm which always determines the optimal route. But it cannot guarantee that realistic deadlines will be met. Owing to the lack of global sampling in the feasible solution space, this algorithm has reasonably sufficient possibility of being trapped into local optima. As Genetic Algorithms always have solutions in a population during a search, they can provide alternative routes using other solutions in the shortest time.

Traditional database systems deal with alphanumeric data types and the values of which can easily be entered through a keyboard and represented textually within a query result, whereas Spatial Databases represent spatial data, which consist of location, shape, size, orientation and spatial relationships. Spatial data are generally multi-dimensional and auto correlated. A customized approach, based on Spatial Databases and Genetic Algorithm, has been proposed in this thesis for formulating the effective route planning.
1.7 OUTLINE

The remainder of the dissertation is divided into six chapters. Chapter 2 reviews some of the literature from the existing route computation methods and multi objective Genetic algorithm computing techniques. Chapter 3 discusses the concept and implementation of Route optimization methods in Intelligent Transportation Systems using the Spatial Network data model. Chapter 4 discusses the concept and implementation of Minimal Generation Gap – Genetic Algorithm for ITS. Chapter 5 discusses the concept and the implementation of High Performance Computing cluster techniques in Intelligent Transport Route planning strategy using MPI. Chapter 6 discusses the concept and the implementation of Intelligent Transportation Route planning by Multi Objective Genetic Algorithm for optimization using Spatial Databases. Chapter 7 presents the conclusion and future work.