# UCGE Reports

Number 20267

Department of Geomatics Engineering

# Integration of GA-Based Multiobjective Optimization with VR-Based Visualization to Solve Landuse Problems

(URL: http://www.geomatics.ucalgary.ca/research/publications/GradTheses.html)

by

**Magesh Chandramouli**

August 2007

UNIVERSITY OF CALGARY

# Integration of GA-Based Multiobjective Optimization with VR-Based Visualization to Solve Landuse Problems

by

**Magesh Chandramouli**

**A THESIS**
**SUBMITTED TO THE FACULTY OF GRADUATE STUDIES**
**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE**
**DEGREE OF MASTER OF SCIENCE**

**DEPARTMENT OF GEOMATICS ENGINEERING**

**CALGARY, ALBERTA**

**AUGUST, 2007**

## Abstract

This research integrates GA-based multiobjective optimization with VR-based visualization to select a Pareto-optimal plan for a landuse problem with multiple objectives and given constraints. Even though, seemingly, all the solutions resulting from the process of multiobjective optimization are equally non-dominated, only one single solution can be implemented and from the perspective of the problem on hand, each solution has its own pros and cons. This study proposes the use of visualization a tool o evaluate the Pareto plans. This way the decision makers can select a subset from the Pareto set and evaluate the plans in this subset visually to select the plan that most suits their requirements.

Urban planning problems, especially landuse problems, are inherently multifaceted and involve stakeholders at various levels, thus necessitating multiple objectives that need to be satisfied. Typically, landuse problems involve finding an optimal allocation of zones for given objectives within a given area that satisfies specific constraints. Due to the continual increase in urban population and the associated human activities to meet the heavy demands imposed by the escalating populace, the urban area undergoes continual change. While trying to evolve sustainable landuse patterns, the problem involves objectives that are inherently conflicting in nature. Multiobjective optimization is one tool that can come handy in such urban landuse planning problems. One particular multiobjective optimization (MOO) technique, Genetic Algorithms (GA), is used in this study to arrive at a solution set for the multiobjective optimization problem. GAs are capable of efficiently searching the large solution spaces by performing the alterations in a time saving and computationally efficient manner. Genetic algorithms can generate a set of 'Pareto plans' for a given landuse problem. However, ultimately only a single plan can be implemented. Decision makers still are confronting a set of plans from which they have to choose one plan. Automating this process of selecting one plan from the Pareto set without using a biased approach based on the relative significance of the various objectives is an important research issue. In order to be able to identify the most suitable solution that can be implemented, visualization is used to evaluate the plans.

The results of the study corroborate that visualization is indeed an effective tool for studying the Pareto-optimal plans and assessing the pros and cons of the plans in order to select one final landuse plan for implementation.

**Acknowledgements**

I would like to take this opportunity to express my gratitude to my supervisors, Associate Professor Bo Huang and Professor Yang Gao for their valuable guidance, motivation, and support. The lengthy discussions with A/Prof. Dr. Bo Huang greatly helped me in shaping up the research methodology and in selecting the appropriate tools for carrying out the research. The detailed discussions with Prof. Yang Gao helped me in organizing the research and clearly presenting the findings in a lucid manner. Without the support and timely help of my supervisors, I would not have been able to complete this research with two major components in less than two years.

I am thankful to Prof. Richard Levy who has consented to be the external examiner and my committee members Prof. Yang Gao, A/Prof. Bo Huang, and A/Prof. Alexander Braun. I remain grateful to each and every staff member of the Department of Geomatics Engineering and every single member of the University of Calgary, who, collectively perform the various tasks necessary for the proper functioning of the institution. I am thankful to all my friends who not only helped me in completing my studies successfully, but also made my stay in Calgary a very pleasant experience. I would like to thank the following people, who have helped me in several ways: Wouter, Suren, Balaji, Ahmed, Ivan, Mrs. and Mr. Gnanalingam, Gopi, Poorni, Kumaran, Malarvizhi, Dr. Charles Yeh, Mohannad, Chen Xu, Mike, Kannan, and Vidya. Special thanks are due to the cute little girl Harini for all the happy and innocent moments. I would like to thank the faculty members of my TA courses, ENGO 451 and ENGO 233, as well as the students, from whom I learnt a lot.

I am grateful to the omnipresent and omniscient Lord Almighty and my parents, Mrs. Lakshmi Chandramouli and Mr. Chandramouli Nagarajasastri without whose blessings I can not accomplish anything. I am thankful to my brother, Mr. Aparajithan, and my sister, Mrs. Pavithra Arun Prasad for their love and support. I am grateful to Mrs. Sangeeta Joshi, Mr. Ashok Joshi, Mrs. and Mr. Mahesh Joshi, and Mrs. and Mr. Fulambrikar. My deepest gratitude is due to my wife, Vaishali and our son, Nachiket for their incessant love.

Finally, I would like to thank all those good people who have contributed to the well-being of humanity in one way or the other, thus leading a purposeful and inspirational life.

***To farmers and soldiers all over the world***

**Table of Contents**

## List of Tables

## List of Figures and Illustrations

## List of Symbols, Abbreviations, Nomenclature

| | |
|---|---|
| CGI | Common Gateway Interface |
| CPOP | Competing Pareto-optimal Plan |
| DC | Direct Control |
| DEM | Digital Elevation Model |
| DLG | Digital Line Graph |
| DSS | Decision Support System |
| ESRI | Environmental Systems Research Institute |
| GA | Genetic Algorithm |
| GIS | Geographical Information Systems |
| GPS | Global Positioning Systems |
| IE/NN | Internet Explorer/Netscape Navigator |
| IFS | Indexed Face Set |
| IP | Integer Programming |
| LBS | Location Based Systems |
| LP | Linear Programming |
| LUMOO | Landuse Multiobjective Optimization |
| MADGIC | Maps, Academic Data, Geographic Information Center |
| MATLAB | Matrix Laboratory (Mathworks, Inc.) |
| MOGA | Multiobjective Genetic Algorithm |
| MOO | Multiobjective Optimization |
| NumHU | Number of Housing Units |
| OOP | Object Oriented Programming |

| | |
|---|---|
| OpenGL | Open Graphics Library |
| PCGS | Per Capita Green Space |
| PCPS | Per Capita Space for Public Service |
| PPGIS | Public Participation Geographic Information System |
| PROTO | Prototype File in VRML |
| TIN | Triangulated Irregular Network |
| UR | Urban Reserve |
| URL | Uniform Resource Locator |
| VR | Virtual Reality |
| VRML | Virtual Reality Modeling Language |
| X3D | Extensible 3D |

# Chapter 1: Introduction

## 1.1 Research Background

Landuse planning throughout the world is becoming increasingly complex these days. Cities worldwide are struggling to meet the heavy demands imposed on them by the continually growing populace. Canada is one such nation whose cities are presently undergoing rapid urbanization. On one hand, the cities are becoming flooded with incoming populations on a continual basis, whilst on the other hand, farmlands and other important non-urban landuse types are being lost on a recurrent basis to sporadic urban expansion. At this rate, Canadian cities and the overall landscape might soon be confronting serious problems, if the urban landscape changes are not properly monitored and counteractive measures initiated.

The continually increasing urban sprawl of Canadian cities entails efficient growth management strategies and futuristic measures aimed at sustainable urban landscape development. Albeit these cities act as the engines driving Canada's economy and overall development, the existing policies and the governmental framework are grossly inadequate to manage the tremendous expansion occurring at a geometric rate. Such an alarming situation calls for the judicious integration of the advances in GIS, multiobjective optimization techniques, and visualization to develop a decision support system to serve urban planners and policy-makers in identifying the desirable and undesirable factors among those that influence urban expansion and perform 'proactive interference', where necessary.

Proper planning is inevitable for the sustainable development of urban landuse. Sustainable development implies that there is a proper balance between supply and demand. In other words, the resources available should be properly and evenly distributed for the needs of the society. This means that the various activities such as commercial, industrial, recreational etc. occur at appropriate locations so that neither over utilization nor under utilization occurs. This implies that resources should not be exploited unreasonably at one place so that sustainable development in that area is hampered. A city can not afford to have only commercial activity or only recreational activity. The residents of a city not only need to earn money but also need to relax themselves. Hence, there should be a proper balance among the various landuse zones and these should be distributed appropriately.

From the above discussion, we can see that urban planning problems are inherently multifaceted and involve stakeholders at various levels, thus necessitating multiple objectives that need to be satisfied. The advancements in the geospatial data acquisition and analysis would surely come in handy at such a situation. Advanced spatial data acquisition techniques have resulted in colossal data volumes; however, effectively utilizing such voluminous data to derive useful information still continues to be a daunting task.. Mere data is not sufficient to solve large-scale landuse problems. This data must be transformed into useful information. Multiobjective optimization is one such tool that can come handy in this urban landuse planning problem. One particular technique, Genetic Algorithms, is used in this study to arrive at a solution set for the multiobjective optimization problem. Even though, seemingly, all the solutions from multiobjective optimization are equally good, only one single solution can be implemented and from the perspective of the problem on hand, each solution has its own pros and cons. In order to be able to identify the most suitable solution that can be implemented, a tool such as visualization is used herein to facilitate decision-makers in evaluating the plans.

Decision makers in the domain of land use planning and management are confronting complicated problems wherein multiple objectives have to be satisfied under specific constraints. Hence, we see that an efficient decision support system can aid the process of 'informed decision-making' immensely. Efficient policy making can channelize the process of urban growth and thus prevent urban land eating away into other vital land use types. Besides, unhealthy trends in urban expansion must be inevitably checked in order to achieve sustainable development within urban environments, failing which life in such urban centers will become extremely chaotic in the decades to come. Such efforts to check and channelize urban growth necessarily involve proper policies and resourceful practices that can aid in wholesome development in the long run.

From the above discussion it is evident that there is a pressing need for advanced decision support systems that aid policy makers and planners in the process of informed decision-making. This study focuses on one of the rapidly expanding Canadian cities, Calgary and integrates multiobjective optimization with visualization to arrive at efficient solutions for the aforementioned problems. In particular, the study presents the results in a visual form so that decision makers can compare and contrast the various optimal solutions.

## 1.2 Problem Statement

A vast majority of today's real-world problems entail synchronized optimization of multiple objectives. The key to the problem lies in an efficient trade off among the different objectives thus making a judicious compromise. Evolutionary methods are a group of time-tested techniques in solving problems entailing efficient optimization among various objectives. Evolutionary problem solving techniques are based on the natural evolutionary process wherein the fittest survive. Similarly evolutionary mechanisms contain procedures that execute iteratively with an aim of increasing the overall fitness or suitability of solutions and hence yielding optimal solutions finally. Genetic algorithms are evolutionary algorithms that are extremely efficient tools in performing multiobjective optimization. However, the limitation therein is that GA (Genetic Algorithm) methodology can be used to find 'a set' of non-dominated solutions, namely the Pareto set. Despite excellent multiobjective optimization procedures, it is possible that a comparatively poorer choice from the Pareto-set is made. In other words, in the absence of objective evaluation, decision-makers might choose and implement a solution, which may not exactly be the best fit out of all the Pareto plans available.

Hitherto, spatial optimization models provided a Pareto set, which is not a unique solution, but is a 'solution-set'. Thus, on their own, GAs are quite efficient in providing 'good-enough' solutions, but can not be relied upon to provide the optimum solution, which is the ultimate goal of any optimization process. In other words, even though GAs provide a pool of 'good-enough' solutions, they just stop short of providing a unique solution that can be chosen for implementation. In many applications, for instance landscape planning, planning authorities can not implement all the several hundred solutions in the Pareto set. Just one optimal solution (conceivably 'the best') can be carried out. It is important to scrutinize the variations among the range of candidate solutions and comparing and contrasting them to obtain a superior understanding of the fundamental processes and locations. As no single solution in the non-dominated or Pareto set is totally better than any other solution, any single solution from the Pareto set should be considered an equally acceptable solution. The process of choosing one single solution over others involves in-depth problem knowledge and various other problem related factors. The choice of the solution is usually based on some 'higher level information'. Tools that can help compare and contrast the various solutions in the Pareto set and thus evaluate them before making a decision would make the whole exercise of multiobjective optimization extremely fruitful. Terms such as 'higher level information' and 'various factors' are abstract in nature and refer to vague elements that do not really help in evaluating the solutions in the Pareto set in an objective manner. This might indeed result in under-utilization of the process of optimization using genetic algorithms. Hence, this research integrates multiobjective optimization using GAs with a visual evaluation tool that can help decision makers in comparing the various solutions and perform informed decision-making with consideration of concrete visual representations, rather than relying on abstract factors.

## 1.3 Objectives

This study aims to present a multiobjective optimization approach to generating futuristic landscapes and to integrate it with a visual evaluation tool for assessing the solutions from the Pareto set.

Summarily, the objectives of the study are as follows:

a. Investigate the various factors influencing land use planning whilst considering sustainable development and choose the important objectives to be considered for multiobjective optimization subject to data availability.

b. Formulating the process of modeling futuristic landscapes as an optimization problem wherein spatial configurations are created through the use of evolutionary algorithms in the form of a Pareto set.

c. Designing the evolutionary algorithm for multiple objectives, e.g. maximization of per capita green space, maximization of per capita space for public service, and maximization of number of housing units.

d. Generate a visualization tool that can interface the results of the genetic algorithm based optimization

e. Evolve procedures to generate multiple landscapes from the pareto set using the visualization tool so as to aid in their evaluation and hence in the decision making process

## 1.4 Approach

As multiple objectives are addressed in this study, a multiobjective optimization approach using GA is employed. For GAs, each and every single genotype within the population must include the complete design or specification for a solution. Obviously, in a multiobjective optimization problem, as the name suggests, the solution is made of multiple components. Each individual component is represented by a gene. All together these make up the whole genotype. Manipulations to these are done via what are known as genetic operators or simply, operators. Typically, only genotypes that are not inferior to any other solution within the search space are considered non-dominated (as these are not dominated by any other solution) and these form the constituents of the Pareto-optimal set. Some genotypes may have excellent performance, however, with respect to only one objective, and these obviously are not the best solutions for a multiple objective problem. Hence, there should be no bias towards genotypes that perform well only with respect to one objective. This is achieved by means of the Pareto-ranking method. Once the genetic algorithms provide a Pareto set, the alternate solutions can be visualized using the tool proposed in this study, thereby providing a yardstick that the decision-makers can use for evaluating alternate solutions or scenarios.

This study puts forward the approach of GA-based optimization to handle optimal landscape scenario generation whilst taking into account multiple objective functions and subsequent visualization of the candidate solutions within the Pareto set as a tool for evaluation and informed decision-making. In particular, the proposed GA accommodates three objectives functions – maximization of per capita green space, maximization of urban housing density, and minimization of energy consumption.

### 1.5 Thesis Outline

The outline of the five chapters, including this introductory chapter, is as follows. The introductory chapter contains the research background, problem statement, objectives and brief overview of the methodology employed in this study.

Chapter Two reviews the various urban land use issues and the problems involved in attaining long-term sustainable development. This chapter also covers multiobjective optimization techniques as applied in the field of land use planning and the use of genetic algorithms as a tool for multiobjective optimization. The literature section briefly describes the two fundamental aspects of genetic algorithms, namely 'selection and variation'. Finally, chapter 2 presents a review of visualization tools in the context of this research and their use in the evaluation and hence in the decision-making process.

Chapter 3 covers the methodology for optimal land use planning and presents the overview of the multiobjective optimization and visual evaluation framework as employed in this study. The GA architecture is explained and the various components of the multiobjective GA for land use planning are elucidated. Next, the visualization framework for generating 3D scenarios is explained. The use of virtual reality environments for generating visualization scenarios for the Pareto plans is explained. The sections therein explain the fundamental entities involved in generating virtual reality based 3D visualizations and the process of putting together various scene components to generate the complete virtual environment.

Chapter 4 The case study particularly focuses on the study area, a region of the city of Calgary and covers the implementation of the MOGA for three objectives. It explains the testing parameters for the GA and the fitness function. Subsequently, the design and implementation of the individual components of the visualization framework are covered. Finally, the results are presented and the discussion section covers the comparison among the alternatives and use of an evaluation tool

The fifth and final chapter presents the conclusions, discussing the limitations, and provides the recommendations for improving the existing framework. Future studies along this direction can use these for expanding the results into other domains of interest and further enhancing results in this discipline.

# Chapter 2: Literature Review

## 2.1 Introduction

Typically, landuse problems involve finding an optimal allocation of zones for given objectives within a given area that satisfies given constraints. Due to the continual increase in urban population and the associated human activities to meet the heavy demands imposed by the escalating populace, land uses undergo continual change. Whether or not these changes are in the right direction is of crucial importance. Today, in several urban environments, landuse configurations are being changed without any rational planning and such processes can not lead to the attainment of sustainable long term development. It is imminent to manage land uses systematically to preserve healthy surroundings and aid sustainable development. Landuse planning is a multifaceted problem and due to the intricacy involved in the planning process, appropriate tools and techniques are inevitable for improving the overall quality of the design process. The landuse plan generated as a result of the spatial optimization must meet the objectives and constraints set out at the beginning.

Methods including Linear Programming (LP) and heuristic search methods such as simulated annealing and Tabu-search have been conventionally used in single-objective land use problems. Quite frequently, LP has been combined with heuristic methods to handle specific spatial optimization problems. For multiobjective optimization problems, multi-criteria decision making has been employed, however with serious constraints and limitations. In this chapter, we will discuss multiobjective problems and the various tools used by other researchers in the field to solve such problems. Subsequently, the chapter will elaborate one powerful tool for solving multiobjective problems, namely 'Genetic Algorithms'. This chapter comprehensively reviews the various landuse planning methodologies whilst substantiating the use of Genetic Algorithms (GA) for this study.

The key objective of the work is the design and implementation of a GA-based Decision Support System (DSS) that can aid the process of informed decision making and help in the landuse planning of considerably large urban locales. The system will facilitate administrators, landuse planners, stakeholders, and other decision makers in the process of designing and developing sustainable urban environments.

## 2.2 Multiobjective Optimization: A Brief Overview

The following passages discuss the gist of multiobjective problems. Numerous real-world problems serve as examples for multiobjective optimization problems. 'Design problems' are a particular class of such problems. A design is nothing but a plan. Examples include locomotive design, car design, hardware design, landuse design etc. The reason that a vast majority of the problems are multiobjective is obvious as researchers and designers are aiming for solutions that can satisfy a wide range of demands. For instance, a simple example is that of an automobile design that aims to increase mileage whilst reducing fuel consumption. Similarly, landuse problems aim to increase the housing capacity of an area whilst reducing energy consumption and reducing traffic congestion. It is apparent that the objectives are inherently conflicting in nature. This is because, an increase in housing capacity implies the number or residents of the area will increase, hence the energy consumption and traffic volume will also increase correspondingly. However, a sustainable design will aim to reduce energy consumption and also must reduce urban problems such as traffic congestion, whilst trying to increase the housing capacity of the city to accommodate the continual flow of populace.

Based on the nature of the application, various objectives need to be formulated for a MOP (Multiobjective optimization) problem. Besides, there may also be other objectives that may be of importance, which can be devised as constraints. Thus, a typical MOP consists of a specific number of decision variables and particular number; Say n number of objective functions, which need to be attained under a given set of constraints.

The aim of the optimization process can be expressed as follows:

Maximize (or) minimize $f(x) = y = (f_1(x), f_2(x)... f_n(x))$

Conditional on $C(x) = (C_1(x), C_2(x), \ldots, C_m(x))$

Wherein n is the number of objective functions and m is the number of constraints.

In the above expression the decision vector x is equal to $(x_1,x_2,\ldots x_n)$ and the objective vector y is equal to $(y_1,y_2,\ldots,y_n)$. The set of values $(x_1,x_2,\ldots x_n) \epsilon X$ signifies the decision space and the set of values $(y_1,y_2,\ldots y_n) \epsilon Y$ represents the objective space. The feasible set of solutions is determined by the set of constraints i.e. $C(x) \leq 0$.

In the context of multiobjective optimization, a term that is frequently heard of is 'Pareto' set of 'Pareto-optimal' plans. In order for a plan to be considered part of the Pareto set no other plan should be found, which is superior in all objectives. In other words, a plan may outdo the Pareto plan in one objective and a different plan may be better in another objective; however, a 'single plan' does not outperform a Pareto plan in all the objectives. Balling et al. (1999) correctly point out that the *'Pareto set is independent of the relative importance of the objectives'*. From the above discussion it can be seen that plans that do not belong to the Pareto set (non-Pareto plans) are 'non-dominated', because a Pareto plan that is better (or that which dominates) already exists.

## 2.3 Approaches for Multiobjective Optimization

One very well-known method that has been employed in landuse planning is linear programming (LP) (Arthur 1997). Based on a given set of resources LP optimizes an objective function, conditional on a series of constraints. Conventionally, various MOP approaches involved cumulatively combining all the objectives of multiobjective optimization into a single objective function. The optimization routines are executed numerous times with varied settings of the optimization parameters till a required set of Pareto optimal solutions is obtained. Cohon (1978) employed two techniques namely the Weighting method and the Constraint method, which are discussed here in a succinct manner. Other noteworthy methods include the Goal programming approach by Steuer (1986) and the Minimax approach by Koski (1984).

In the weighting method, the original multiobjective optimization problem is translated into a single objective problem by assigning weights to the various objectives and expressing them as a linear combination. By executing the optimization algorithm for a particular number of times or iterations a solution set is obtained. One major drawback of this method is that it is not capable of generating all the Pareto optimal solutions. In other words, the Pareto set so obtained is incomplete (Stewart et al., 2004). The Constraint method attempts to overcome this shortcoming by transforming all but one of the objectives into constraints. The one objective, that remains, is considered as the objective function of the single objective optimization. To obtain the Pareto-optimal solutions the lower bounds of the constraint set are fluctuated during the optimization procedure. However, in the case of the constraint method, inappropriate choice of the lower bounds results in an empty feasible set. Thus, it is evident that the above methods have limitations when applied to problems that are multiobjective in nature.

By and large, landuse (LU) management problems largely employed linear programming (LP) approaches. LP approaches lack abilities to properly and efficiently handle integer variables and are also not competent in handling spatial coordinates. Also, of late, the complex urban milieu, greater number of stakeholders, and the demand for informed decision-making have necessitated the use of more advanced tools that can better handle multiobjective problems. In addition to the aforementioned reasons, the advancement in spatial data procurement and processing techniques and the use of Geographical Information Systems (GIS) have imposed greater demands on the accuracy and reliability of the results. Consequently, researchers like Stewart et al. (2004) started modifications by converting LU problems in the form of integer programming (IP) problem. This marked the entry of IP approaches in this domain. Still, linear as well as integer programming techniques were inevitably suited for single objective optimization process and this necessitated combining those objectives into a single objective.

Efforts to find alternate methods that can handle large combinatorial problems rapidly led to non-traditional heuristic approaches such as Tabu search and Simulated Annealing. Such methods were robust, fast and were indeed able to solve large combinatorial problems; however, these methods do not necessarily provide the optimal solution. This is a major drawback, as despite being robust, the inability of the method to assure optimal solution greatly reduces the reliability (Beasley et al. ,1996).

It follows from the above discussion that optimization approaches that were truly multiobjective in nature were not available initially and the lack of such methods especially for large-scale problems is pointed out by Horn (1994). One important aspect is the problem knowledge which may not necessarily be available and even if present, may not be easy to incorporate. Another significant aspect with respect to the execution of the optimization procedures is that the runs are performed independently in the traditional methods. This is not only computationally intensive but the utilization of the positive aspects (traits in the individuals) of each procedure is rendered impossible (Deb 2001). From the above discussion, the need for efficient alternatives to traditional approaches is evident.

Of late, Genetic Algorithms (GAs), a heuristic method, has been found suitable enough to tackle the aforementioned problems (Aerts et al. 2005). In sharp contrast to the other heuristic approaches including those discussed earlier, GAs are wide-ranging search procedures with universal applicability, meaning that they are not restricted to a particular discipline. Genetic algorithms (GA) are a form of evolutionary computation developed by Holland (1975). They were found to be robust tools that can produce exceptional results for MOO problems (Goldberg 1989). GA is rooted in the principles of natural selection and evolution. Landuse problems typically involve large solution spaces and are computationally complex as a large number of iterations are required. GAs are capable of generating optimal solutions in a computationally efficient manner. Most importantly, the evolutionary mechanism surpasses the shortcomings of the single objective optimization approaches. By 80s and 90s, a vast majority of the MOO problems in which the relative importance of the objectives was hard to ascertain started employing GA.

Multiobjective genetic algorithm (MOGA) approaches have from the very rudimentary stages focused on the generation of a set of Pareto-optimal solutions (Pareto, 1896). Beasley et al. (1996) summarize the major elements involved in the GA process as: survival of the fittest, recombination, and mutation. The 'survival of the fittest' step ensures that solutions which are better than the other solutions in a generation are selected for the new generation. Recombination occurs by either random or methodical swapping of values among solutions and mutation by changing the values within a solution. A more detailed discussion of these steps is presented in the subsequent sections.

## 2.4 Genetic Algorithms

Genetic Algorithms are a type of evolutionary computational techniques, a category of stochastic optimization methods imitating the natural evolutionary process. Evolutionary computational techniques can be dated back to the 1970s. Evolutionary algorithms (EAs) start with an initial set of solutions, each member of which is a complete solution in itself to the problem on hand. At the core, two fundamental evolutionary tenets namely 'selection and variation' are employed to generate the subsequent generations. The first principle namely selection is along the Darwinian theory (Darwin, 1859) of 'survival of the fittest'. The other principle, variation, imitates the natural capability of creating living beings with new traits by means of recombination and mutation. The value of the plans in a current generation is gauged by appraising the individual members and allotting scalar values that indicate their fitness. Mutation, typically, is applied to a randomly chosen gene, the value associated with which is randomly modified to a different value from a permissible range of values. For the next iteration, the offspring population thus created substitutes the parent population, and the process is repeated for many generations with the aim of increasing the average fitness of the generations and hence that of the individuals.

During the process of iteration, those individuals which have lower fitness (in other words, which are less suitable) are left behind. Thus with each generation or with each iteration, more fit solutions get selected and the less fit solutions get eliminated or get left behind. Genetic algorithms are particularly suited to the problem of multiobjective optimization owing to their capability of generating a set of Pareto-optimal solutions in a single iteration process. Besides, the variations introduced by recombination and selection make sure that additional traits that are superior have a chance to enter the generations, while selection ensures that the unfit solutions are not carried over to the subsequent generations.

Valenzuela et al. (1997) advocate that genetic algorithms outperform other search techniques while solving multiobjective optimization problems. More and more number of multiobjective optimization problems continue to employ genetic algorithms. Other than the landmark works by pioneers like Goldberg (1989) and Holland (1975), numerous other researchers conducted several successful studies on genetic algorithms and evolutionary computation. Some prominent researchers in the 80s and 90s include Schaffer (1984, 1985), Fonseca et al. (1993, 1995a, 1995b, 1996), Fourman (1985), Hajela (1992); Horn et al. (1994), Kursawe (1991), and Srinivas et al. (1994). All the aforementioned researchers proposed and designed various implementations of GAs. Later, these evolutionary techniques and their variations became highly functional for various multiobjective optimization problems (Cunha et al. 1997; Valenzuela et al. 1997, Fonseca et al.1996, 1998).

Matthews et al. (1999a, 1999b, 2000a) carried out significant studies in the use of GAs in spatial applications, particularly landuse problems. Matthews et al. (2000b) constructed a GA- decision support system (DSS) for a multiobjective problem that identifies the nature of trade-offs among conflicting objectives in landuse and landscape analysis. Matthews et al. (1999b) designed another GA-based DSS that facilitated the exploration of the various land use choices and the possible potential influences of landuse changes. Other people who carried out significant studies include Arika et al. (2000), Stewart et al. (2004), and Seixas et al

(2005). Stewart et al. (2004) designed a GA using a goal programming approach, to a spatial problem consisting of two objectives namely minimization of cost and compactness of each land use. Seixas et al. (2005) implemented a GA to investigate future landuse compositions. The works discussed above and several other significant works clearly demonstrate the capabilities of GAs in solving multiobjective optimization problems (Jaszkiewicz (1998), Zhang et al. (2000), Fonseca and Fleming (1995)). Quite recently, researchers have also started investigating specific areas within multiobjective optimization and search process using GAs, for instance, elitism (Obayashi, Takahashi et al., 1998), Pareto-optimal front convergence (Rudolph 1998), niching (Obayashi, Takahashi et al., 1998). On the other hand, Works by Rudolph et al. (1998), Lis et al (1997) and several others have focused on designing and implementing innovative evolutionary methodologies.

It is lucid that GAs differ significantly from conventional methods in ways more than one. First and foremost, GAs start with a population of solutions and not with a single solution. GAs are very flexible in the sense that they can easily be used for a wide range of problems ranging from air-craft design to medical research. Numerous examples of the various applications of GA can be found in literature (Fogel, 1998), (Rudolph, 1998), and (Pham, 2000).

On the whole, the GA process can be summarized as follows. Figure 2.1 shows the general structure of a typical genetic algorithm. Solutions to the problem are coded as a series of values in a structure referred to as the chromosome. At the start of the genetic algorithm process, a number of random solutions are created and stored in the chromosomes. All the chromosomes are evaluated and given a fitness value based on how good the encoded solution is. These values are then used to populate the next generation of chromosomes so that fitter chromosomes represent a larger proportion of the copied chromosomes than the less fit solutions. These are then subjected to recombination and mutation to create new combinations of values that can then be re-evaluated. This process is repeated for a large number of generations with the goal being that the solutions get better with each successive generation but with the ability to escape local fitness maxima in order to find the best overall solution.

```
Time = 0                                        ; Starting time = 0
Population(T)                                   ; Population at Time T
Evaluate Population(T)                          ; Evaluate Present Population
While (Conditions are met)                      ; Check for Satisfication of criteria
T = T + 1                                       ; Increment Time Step
Create Population(T +1) from Population(T)      ; Create Subsequent generation
Evaluate Population                             ; Again Evaluate Present Population
Loop
```

Figure 2.1 The general structure of a Genetic Algorithm

After a discussion of genetic algorithms and their application to MOO problems, especially landuse and spatial problems, let us proceed to look at the GA itself in detail. The basic building blocks of GAs are ‘genotypes’. A genotype represents a complete specification for the solution to the problem on hand. A genotype is made of genes. Considering a landuse problem, if a master landuse plan for a particular area is to be generated, then one genotype represents a possible solution, i.e. a plan. This plan is in turn made of genes. For instance, if the landuse map is represented as polygons and if it contains n number of polygons, then the genotype consists of n number of genes. If the landuse is represented in the form of a raster grid of size 'n x n' (for e.g. 400 cells), then the genotype consists of 400 genes.

Srinivas et al. (1994) mention that the encoding mechanism is a very fundamental and significant aspect of the GA construction, as this is the means by which the GA represents the variables of the optimization problem. In this study, we use an integer based genetic representation. The core of the GA involves manipulating these genes of a genotype using various operators that perform the required manipulation operations. The very initial forms of GA implementations used a 'fixed-length binary string' representation. In other words, the genotyped was represented in the form of a binary string with a fixed length. Even today, a lot of GA implementations still employ this representation.

While binary representation remained the most common standard for GA representation, Antonisse (1989) disputed this version. Following that, various other representations that can better suit tailor-made applications began to evolve. One of the pioneers, Goldberg (1989) used order-based representations and Michalewicz (1992) used real-coded genes for numerical optimization. Goldberg et al. (1989) proposed a messy representation for solving problems that proved complicated for normal GA optimization. In a further improvement of this, Goldberg (1989) and Deb et al. (1993) used a messy representation, wherein the genes could be amalgamated in any order since the gene itself was tagged thus facilitating its decoding, irrespective of its position on the genotype.

Messy representations offered excellent flexibility as they facilitated both over and under specification of the problem being investigated. In over specification, the representation consisted of more number of genes and in under specification, the representation consisted of too little or less number of genes. These days integer representation is also employed in many studies, as in this study, owing to its simplicity and ease of computation. In the natural genetic process, the genetic constitution is referred to as the genotype and the physical manifestation as the phenotype.

The study by Matthews et al. (1999) on the use of GA for a spatial allocation problem proposed innovative genotype representations for the landuse planning problem. A land allocation problem entails a genotype to code the information for allotting land uses the polygons or cells. Their work examined three alternative representations namely spatially explicit grid representation, land block representation, and percentage and priority representation.

Once the GA representation has been decided, an initial population is generated with a genotype population of a specific number. For instance, 100 genotypes, each representing a possible solution can be generated randomly and this serves as the starting generation. The candidate solutions are called individuals and complete set of candidate solutions is called the population.

Once the initial generation has been crated, GAs typically consist of the following steps:

- Selection process wherein the individuals for the next generation are chosen
- Manipulation, wherein recombination and mutation are performed using genetic operators

In this study, integer based representation has been implemented. The genetic framework for the region is represented by a gene each for every changeable zone. We use integers because, integers are simple and straightforward, and hence easy to handle; Also, integers are probably more efficient in GAs, from the computational perspective. Each gene is an integer that can assume any value from among the various land uses considered in the study. At first, each variable is plotted or mapped to an integer within the range of values and this integer value is encoded using binary bits. However, as there are multiple variables involved optimization problem, the binary codes of all such variables are linked together finally resulting in a binary string. In the beginning (first generation), a random value is assigned by the GA to each gene. Here, the generation size is chosen as 100, as a result of which 100 landscape plans result at the end of the execution of the first generation. The integer representation is also a common method (Srinivas et al. 1994) of encoding used in GAs.

An initial generation is created by a process of random generation in the presence of constraints and the iterations are repeated until a feasible set is obtained. Here, 'feasible set' refers to plans that satisfy the constraints imposed. This process of generation of feasible set is illustrated later in Chapter 3. During iteration, the plans in a generation are checked individually for satisfaction of the minimum requirements for greenspaces, public service space, and housing density. Those that satisfy these requirements are added to the feasible set and the others are discarded. The procedure is repeated till the initial generation with 100 chromosomes is obtained. After the initial generation is obtained, the selection, recombination, and mutation processes are performed to create the subsequent generation.

### 2.4.1 Selection and Variation

As stated earlier, the GA process consists of two very fundamental operations, namely selection and variation. The selection process is the step whereby the individuals that are 'fit enough' to be passed on to the next generations are chosen. Typically, this process is biased by the fitness of the individuals in such a way that individuals with higher fitness have a great probability to make it to the subsequent generation. The selection process can be stochastic or deterministic; the basic objective is to eliminate the poor quality individuals from the population set. The value of an individual member of the population with respect to the optimization process is represented by a scalar quantity known as 'fitness'. The fitness value is calculated based on the objective functions and constraints. After calculating the fitness values of each and every individual in the generation, those members with higher fitness values are selected for the subsequent generation. However, as is obvious, not all the members from the present population can be selected for the next generation. This proportion is called the rate of selection or selection rate. For instance, if the selection rate is .2, then out of a population of 100, 20 individuals will be selected for the next generation.

Assuming that the total number of individuals in a population is ‘n’, if the selected rate is x, then the total number of individuals that are selected from the current generation for being passed to the next generation is (n * x). The next generation now only has (n * x) number of individuals. Hence, the remaining individuals (n – (n * x)) must be generated using recombination and mutation. For instance, if n = 100, and x = 0.4, then 40 individuals are obtained by selection and the remaining 60 are generated by recombination and mutation.

One vital consideration during this step is the choice of the number of chromosomes to retain. If there is a considerable number of poor quality chromosomes in the present population, retaining a large number of chromosomes for the next generation from this generation will facilitate such poor quality chromosomes to contribute immensely to the next generation thus affecting the overall fitness of the generation. On the other hand, if only very less number of chromosomes are retained from the present generation to the next generation restricts the number of genes available in the offspring. This step mimics the natural selection process. Thresholding is a process that is frequently employed in several studies. In this process, chromosomes with a fitness value below the threshold limit are not considered for the next generation.

As seen in the earlier passages, recombination and mutation are the steps following the process of selection, to generate the remaining individuals for the next generation. The new solutions in the search space are generated by altering the existing ones. Simply stated, recombination is the process of merging the genetic information from two parent chromosomes by means of crossover. In the recombination step, a predetermined number of parents are selected and these are recombined using crossover operations to create children. In order that the process remains stochastic, a probability rate known as crossover probability is used along with the crossover operator. Originally single-point crossover was used in earlier GA studies. This was succeeded by multi-point crossover and subsequently uniform crossover came into being, wherein individual genes and not genotype portions are interchanged. Eshelman et al. (1989) found that use of uniform crossover enhanced the search capability of the GA. Also, if the crossover proportion of the offspring from the parent was maintained at 50% the overall efficiency of the GA was enhanced.

For recombining parents, at first the parent chromosomes that form the mating pair must be selected. Two chromosomes must be chosen, which can be combined to generate the offspring. As seen earlier, the process is repeated until the required number of chromosomes are selected to fill the remaining chromosomes in the next generation. Considering the above parameters ( n=100, x =.5), the iterations must cumulatively result in 50 remaining chromosomes for the subsequent generation. A wide range of methods for selecting the mating pair are available. These include random pairing, weighted random pairing, tournament selection etc.

In random pairing, a random number is generated in a uniform manner to select the chromosomes. In this method, the fitness of the individual chromosomes is not taken into account. A more fitness-biased method of pairing is the weighted random pairing. In weighted random pairing, the probability of selection of chromosomes for mating is biased by their fitness. In other words, chromosomes with higher fitness values have a greater chance of getting selected for mating. This way, the chromosomes with good qualities have a greater probability to pass their traits to the next generation. Another widely used method is the tournament selection wherein a small compartment of chromosomes is elected from the original mating pool, and the two chromosomes with the top two highest fitness values are selected as the mating pair. This method is a blend of random as well as fitness-biased methods. This is because, the selection of the subset is random, while within the subset the mating pair is selected based on the fitness values.

Once the pair for mating is selected using one of the aforementioned methods, the chromosomes must be 'mated' to produce 2 offspring. The present members of the chromosome population largely influence the genetic composition of the subsequent generation. The crossover point is where the swapping occurs. This point is chosen randomly and it lies between the first and last genes of the chromosomes. At first, one of the two members of the mating pair, called $Parent_1$ provides the genes to the left of the crossover point to the $Offspring_1$ and the second member of the mating pair, $Parent_2$ provides the genes to the right of the crossover point to the $Offspring_1$. Thus, the $Offspring_1$ now contains material from both the parents. Similarly, the second offspring is generated by combing material from $Parent_1$ and $Parent_2$. The genes to the right of the crossover point from $Parent_1$ and that to the left of the crossover point from $Parent_2$ are combined to produce $Offspring_2$. Other alternative forms of crossover are also available.

Eshelman et al. (1989) delineate a two-point crossover for a binary GA, wherein the crossover operation is performed at two random crossover points. The parents subsequently exchange the genes between the two crossover points chosen randomly. Instead, after selecting two crossover points, one of the three parts of the chromosome is randomly selected by means of which the genes that are to be exchanged can be determined. The use of two-point crossover greatly enhances the range of offspring that can be created by recombination. Eiben (1994a) described a crossover operation involving three parent chromosomes and two crossover points. Uniform crossover considers each gene in the parents and on a random basis assigns each gene to one of the children chromosomes.

Once the recombination step is over and the crossover operations are complete the generation is full with its complete population of chromosomes. At this stage, random mutations are introduced in the population. Random mutations of the chromosomes change a particular proportion of the genes in the chromosomes. Mutation helps the GA process in two ways:

1. Mutation helps prevent premature convergence
2. Mutation aids establishing new traits not present in the original population

By avoiding premature convergence it is meant that if an exceptional solution is attained during the iteration process, mutation makes sure that algorithm does not just converge on that particular solution, but the iterations continue to yield better solutions. In integer-based representation, mutation is exercises by introducing random changes in the gene-wise structure of the chromosomes, based on the mutation probability. This is explained in details in chapter 3.

### 2.4.2 Subsequent Generations and Convergence

After the mutation step is completed, the next generation is completely ready for the iterative process. The steps starting from fitness calculation are started once again for the individual chromosomes of the new generation and this generation undergoes the steps such as selection, recombination, and mutation as before until the subsequent generation is obtained.

The iterative process of the subsequent generations depends on

- Whether specific search criteria have been satisfied or
- Whether a specific number of iterations have been surpassed

As mentioned already, if the mutation step did not exist, after a particular number of iterations the average fitness of the generations would continue to remain the same. The overall success of a GA is to a large extent influenced by its ability to sustain a diverse population so as to preclude untimely or premature convergence. Also this is important to attain an evenly distributed and well spread Pareto-optimal set

After having discussed the GA part in detail, now let us move on to the visualization part. Hitherto, to the authors' knowledge, spatial optimization models provided a Pareto set which is not a unique solution, but is a 'solution-set'.

Dias et al. (2002) clearly point out that as no single solution in the non-dominated or Pareto set is better than any other solution, any single solution from the Pareto set should be considered an equally acceptable solution. Osyczka (1985) mentions that the process of choosing one single solution over others involves in-depth problem knowledge and various other problem related factors. Seixas et al. (2005) opine that choice of the solution is based on some 'higher level information'. In this study, the authors strongly believe that a tool that can help compare and contrast the various solutions in the Pareto set and thus evaluate them before making a decision would make the whole exercise of multiobjective optimization extremely fruitful. Terms such as 'higher level information' and 'various factors' are abstract in nature and do not really help in objectively evaluating the Pareto solutions. This might indeed result in under-utilization of the process of optimization using genetic algorithms.

Hence, the authors propose that multiobjective optimization using GAs is integrated with a visual evaluation tool that can help decision makers in comparing the various solutions and perform informed decision-making based on concrete visual representations, rather than relying on abstract factors or placing weights subjectively. This study puts forward the approach of GA-based optimization to handle optimal landscape scenario generation whilst taking into account multiple objective functions and subsequent visualization of the candidate solutions within the Pareto set as a tool for evaluation and informed decision-making.

Now, let us discuss visualization, its applications, and its usefulness as a tool for evaluating the results of GA-based MOO problems.

### 2.5 Visualization

The need for visualization and its effectiveness in solving practical problems has been emphasized in numerous works by authors from diverse fields. In their works on visualization, several authors (Berry et al. 1998, Tufte 1990, 1992) emphasize the importance and usefulness of visualized data. Recently, 3D visualization of information has turned out to be an essential tool in geological, landuse, infrastructure, geophysical, meteorological, hydrological, and several other environmental applications. Modern sophisticated data acquirement technologies have made it possible to acquire complex data which were primarily much more difficult to procure; nevertheless, to extract useful information from this sea of data volumes is an overwhelming chore. With increasing data accessibility, and the development of a plethora of tools for visualization, there is a mounting need to sensibly and efficiently model geospatial data and phenomena eventually.

Visualization practices not only enable the user to obtain an insight into the data being analyzed, but also facilitate effective presentation of the results of the analytical process (Church et al. 1994, Koppers 1998, McGaughey, R.J. 1998). Visualization enables combining diverse datasets to present an integrated view of the data. Notwithstanding the remarkable research efforts in this direction, photo-realistic modeling of real world objects continues to remain a challenge. Virtual Reality and Web GIS have significantly influenced the process of development of tools (Fairbairn and Parsley, 1997) that facilitate interactive visualization of geospatial data.

Over the past several decades, the evolution of varied image generation techniques and the parallel developments in GIS, image processing software, remote sensing, and CADs (Computer Aided Design) have resulted in colossal volumes of digital spatial data. For issues involving design and decision-making within the realm of urban landscapes and environmental applications, 3D visualization serves as an immensely valuable means for exploring geospatial data (Bonham-Carter, 1994). Shiode N. (2001), in the work on urban modeling, elucidates the function of spatial information database and remote sensing technologies in the development of 3D models. Visual representations are comparatively easier to comprehend and employ, than their analogous tabular or written versions. Contemporary visualization standards such as VRML 2.0 have made gigantic strides from the earlier two dimensional maps and other graphic representations. Doyle et al. (1998) elaborate the prospects of utilizing VRML for rendering complex 3D visualizations. Integrating VRML with Java and CGI (Common Gateway Interface), Huang and Lin (2002) developed a geographic VR toolkit.

Much of the research involving spatial analysis demands that the data be in the 3D form and common sense dictates that spatial data be visualized in the 3D form. Geospatial data is inherently three dimensional in nature since every spatial element has its own position or location in space. In the context of modern research, there is a need to visualize geospatial data in its 3D form in multifarious fields such as geography, civil engineering, hydrology, disaster management, demography, and so on. Chen and Murai (1999) mention a gamut of application domains including oil exploration, mining, and geology, which need three-dimensional visualization. These authors emphasize that in addition to visualization of 3D geospatial data, data manipulation is also important. The choice of application for data visualization depends on the need for data visualization and it might vary among each of the aforementioned fields. The scope of this study does not permit dwelling into the diverse nature of the different 3D geospatial data.

The discussion in this study particularly revolves around landuse and urban planning applications. When considering landuse applications, conventionally, town or country planning operations relied heavily on drawings and of late, CAD drawings seem to play a major role. However, one major handicap with these forms of data is that they try to represent 3D entities in 2D. Even though these may provide an idea of the place being studied or designed, these cannot substitute a 3D view of the terrain under analysis. Present landscape architecture applications and urban modeling are extremely complex processes. Hence, it is imminent that 3D geospatial data be viewed in their 3D forms in order to gain a better insight.

Visualization facilitates not only presenting information, but also enables seeing and understanding hidden information among datasets. As mentioned in the previous section, huge volumes of data are available today and it is practically impossible to manually sift through these huge amounts of data. Using visualization techniques, data can be presented in a much more organized manner that facilitates understanding the information that may not otherwise be apparent. The advantage of modern visualization is that such visualizations are not mere depictions of scenes, but also interactive environments capable of animating the scenes, and simulating phenomena.

Urban planning authorities and town planners face several problems such as managing water shortages, transportation problems, urban housing and land use problems, natural and man-made disasters, etc. Several of these problems are mutually dependant and trying to solve them in isolation will never lead to a permanent or long-lasting solution. One of the foremost steps in solving these problems is to get a bird's eye view of the problem scenario as a whole, while simultaneously concentrating on the minutiae. This kind of visualization is of immense value to town and country planners and urban infrastructure management in understanding the link among the various components. Also, the influences on the ambient environment as a result of the aforementioned project can be studied by means of the virtual settings. The visual impact of new buildings and surroundings on each other can be vividly seen on the screen.

## 2.6 Visualization in MOO problems

The aim of this study is to employ genetic algorithms (GAs) as a tool to solve a LU MOO (Land Use Multiobjective Optimization Problem) and subsequently use a visual evaluation tool to aid land-managers in selecting appropriate plans. This thesis focuses on the application of GAs to search and identify the landuse plan that is optimal in view of the objective(s) of the decision makers and authorities involved. For this purpose, MOO is integrated with Visualization. Visualization facilitates not only presenting information, but also enables seeing and understanding hidden information among datasets.

By proper use of visualization tools, the same area can be viewed at different scales i.e. a small area in detail or a bird's eye view of a larger area. In order to see the overall landscape of a whole country we need to view the entire country at a glance. However, the advantage with modern visualization is that such visualizations are not mere depictions of scenes, but also enable interacting with the scene and are capable of animating the scenes, and simulating phenomena. Another kind of information that needs to be discussed in this context is associated information. In every society there is an association among the various components. For instance, consumption is related to population, import/export is related to consumption, economy is related to commerce and so on. In order to understand these links among the various components of a system, tools that can reveal the various concurrent processes that occur among the various sub-systems are very essential. Solutions to many complex problems can be found by understanding the link or relationship among the components of a system. Let us consider a sample situation in town or country planning or urban landscape development etc. Places that were once considered unfit for human habitation have been transformed into hubs of economic activity. Several factors influence this transformation of a once desolated place into active urban centers.

Urban planning authorities and town planners face several problems such as managing water shortage, transportation problems, urban housing and land use problems, natural and man-made disasters, infrastructure challenges, etc. Several of these problems are mutually interrelated and trying to solve them in isolation will never lead to a permanent or long-lasting solution. One of the foremost steps in solving these problems is to get a bird's eye view of the problem scenario as a whole, while simultaneously concentrating on the minutiae of the constituent elements. This is a mammoth task, considering the innumerable components and factors that constitute each one of the aforementioned problems. Tools and techniques that can provide a better perspective or a panoramic view of the scenario in its entirety are inevitable. Visualization is a tool of immense value that helps to tackle such problems and to find a feasible, optimal solution, which is both time-saving and economical. In the `figure above, we can see that visualization enables viewing the constituent elements of an urban or rural infrastructure in whole as well as in parts (Figure 2.2)(Chandramouli and Huang, 2006). This kind of visualization is of immense value to town and country planners and urban infrastructure management in understanding the link among the various and evolving a holistic solution to problems. Also, the influences on the ambient environment as a result of the abovementioned project can be studied by means of the virtual settings. The visual impact of new buildings and surroundings on each other can be vividly seen on the screen.

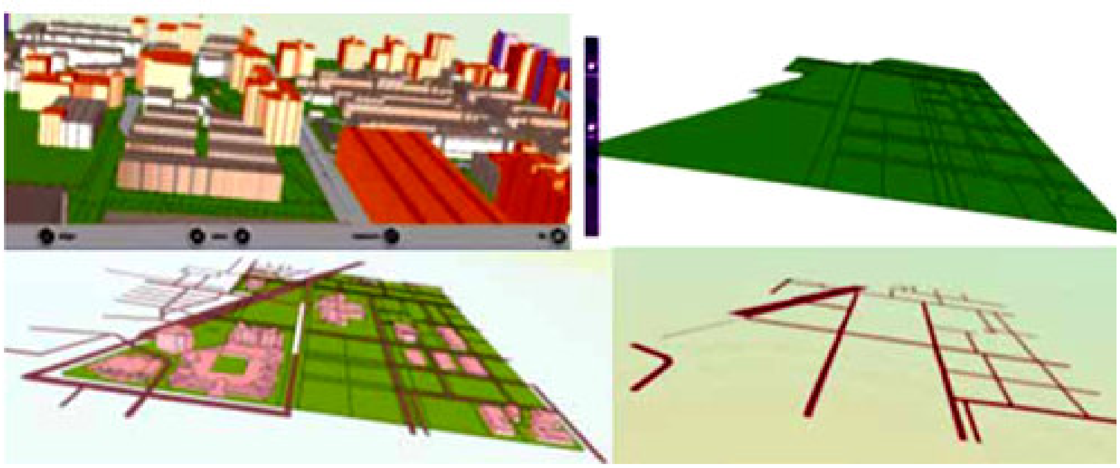


Figure 2.2Visualizing interlinked constituent elements of landuse planning

In this study, landuse design is formulated as a multi-objective optimization problem, which is solved using genetic algorithms. The results, instead of being merely being presented as a Pareto set with a pool of candidate solutions, are evaluated using a visualization tool. This helps in the process of informed decision-making, thereby facilitating the selection of the optimum plan by planners, decision-makers, and administrators.

Even though the Pareto-optimal solutions are available after the multiobjective optimization, urban planning authorities and town planners face problems due to the lack of a proper objective evaluation tool for assessing the plans. One of the foremost steps in solving these problems is to get a bird's eye view of the plans as well as to see the plans in details at varying scales. This way even subjective aspects such as aesthetic scene quality can be ascertained and the plans can be compared visually.

### 2.7 Conclusion

Chapter 3 started with a review of the various urban land use issues and the problems involved in attaining sustainable long-term development. This chapter provided a detailed review of the various multiobjective optimization techniques and discussed works by several researchers in this field. MOO problems as applicable to the field of landuse planning were discussed and the use of genetic algorithms as a tool for Multiobjective optimization was delineated. Finally, the chapter provided a review on the use of visualization and the use of such visualization tools in the context of this research and their use in the evaluation of Pareto-optimal plans.

# Chapter 3: Methodology

## 3.1 Introduction

Chapters 1 and 2 have adequately emphasized the need for sustainable development of a city. Several studies indicate that the most important characteristic of a sustainable city is its ability to sustain its population without causing permanent or irreparable damage to its resources. Among the resources, land resource is of paramount significance as it is extremely limited in today's urban milieu. Typically, the population of cities is always on the rise due to the various reasons such as job opportunities, life style, amenities and so on. Hence, in order for the city to be able to cope with the increasing population and the demands imposed on its resources, proper planning is inevitable. The GA is designed in this research with due consideration to these aspects.

The research methodology in this study consists of two major elements:

1. GA-based multiobjective optimization
2. VR-based visualization

GA is used for solving the multiobjective landuse optimization problem by producing a set of Pareto-optimal plans that are equally optimal from the point of view of all the objectives considered and visualization is used to select one among these plans for implementation.

### 3.2 Software

ESRI's (Environmental System Research Institute) ArcGIS software is used for geospatial data editing and processing. MATLAB is used to program genetic algorithms for multiobjective optimization and VRML is used to create visualization scenarios.

ArcGIS is the collective name of a collection of GIS software created by ESRI. ArcInfo, the advanced version of ArcGIS including added capabilities for data manipulation, editing, and analysis, is used for geospatial data editing and processing in this study. This study uses ArcGIS version 9.1 that consists of an advanced geoprocessing environment for performing spatial manipulations.

The genetic algorithm is programmed using MATLAB version 7.1. The choice of this software for genetic programming was greatly influenced by the simplicity and flexibility of the MATLAB programming and its capability to handle large number of iterations. Matlab is an interpreted language designed for handling voluminous numerical computations. Using Matlab numerical calculations can be performed and the results can be visualized without tedious programming efforts.

Several techniques have been tried and implemented for visualizing 3D geospatial data. This study centers on VRML, one such visualization technique, which has proved to be quite efficient in building 3D scenes (Boyd 1996, Lee et al. 1996) and hosting them on the internet. The origins of the concept of spatial immersion can be dated back to 1965 when

Ivan Sutherland (1965) put forth the ideas of immersion in virtual space in his influential work, “The Ultimate Display”. The VRML Repository defines the Virtual Reality Modeling Language (VRML) as a ‘standard language for generating interactive 3D environments and sharing those worlds across the Internet'. An in-depth description of VRML from the point of view of this study is provided later in this chapter.

### 3.3 GA Formulation

The GA explained here is a tool for generating land use plans for a region represented by polygons, each of which can take one value from among a given range of integer values. Hence, each LU zone is assigned a unique integer value corresponding to the landuse zone it represents (Table 3.1). This value is governed by some constraints, which will be discussed subsequently. The algorithm explores various plans with respect to the set of objectives whilst considering the restrictions/constraints. The goal is to produce a land-use map that will ensure maximum values of per capita green space, urban housing density, and per capita space for public service.

Without loss of generality a maximization problem is considered for this study. .Three objectives and three constraints were specified for this problem. The objectives were maximization of per capita green space, per capita space for pubic service, and housing density. The constraints necessitate that the optimal plans generated have enough housing capacity, sufficient per capita green space, and per capita public service space. The following sections will discuss in detail about objectives and constraints

Table 3.1 Landuse Zones and Corresponding Integer Values

| Landuse Type | LU_Code | Integer corresponding To LU_Code |
|---|---|---|
| Agricultural Zone | AGRI | 0 |
| Commercial Zone | COMM | 1 |
| Direct Control | DC | 2 |
| Industrial zone | IND | 3 |
| Greenspace zone | GS | 4 |
| Pubic service zone | PS | 5 |
| Residential – High Density | RESH | 6 |
| Residential – Low Density | RESL | 7 |
| Residential – Medium Density | RESM | 8 |
| Urban Reserve | UR | 9 |

Each of the 100 plans is made up of genes represented by integers. Each gene corresponds to a land use zone representing the study area.

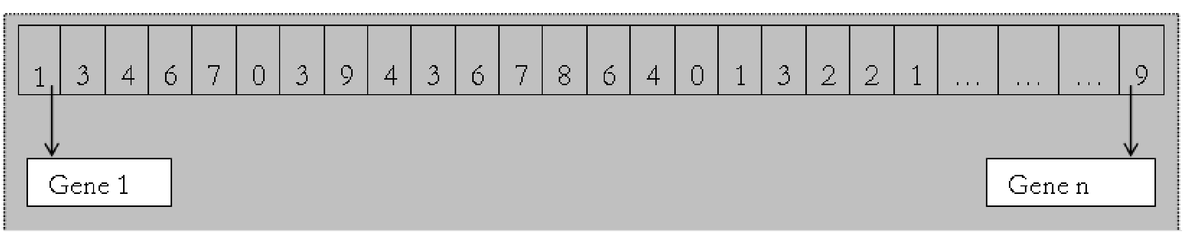



Figure 3.1 A chromosome structure with integer representation

Figure 3.1 illustrates a typical chromosome with n genes, where n is the number of zones or polygons in the study area. ). The 10 landuse zone types are assigned one integer value each from 0 to 9. For instance, integer 9 corresponds to 'Urban Reserve'. The first gene has a value '1'. This means that the corresponding land use zone has a LU_CODE of COMM (commercial). The last gene represents a landuse zone, which is an Urban Reserve (UR, gene value = 9).

The city is divided into zones (with restrictions) and these zones are allowed to assume values from a given set of integers. As the landuse variables can assume one among 10 values, the total set of possible plans is as big as $10^n$, where n is the number of landuse polygons. This signifies an enormously big search space. Only a tool like GA that is robust and efficient can perform multiobjective optimization in such a large search space. The feasible set from this set of solutions can be considered as the collection of the decision vectors that meet the constraint requirements. The objective functions and the constraints are explained as follows:

### 3.3.1 Objective functions

Genetic algorithms typically consist of functions or objectives that they try to maximize or minimize during the process of optimization. 'Sustainability' is the keyword that influenced the selection of objectives for this study.

Three objectives are considered in this study. The objectives ensure that

- The urban dwellers get more green spaces,
- The city is capable of accommodating more residents
- More space for public amenities is available for the residents

The first objective is the maximization of per capita green space (PCGS). Green spaces are inevitable to reduce environmental pollution and to ensure healthy surroundings for the people. Many authors and land use planners consider green spaces as inevitable for attaining sustainable urban environments.

Green spaces such as parks are truly multi-functional in nature, as:

1. Green spaces satisfy the recreational needs of people
2. Green spaces are also used by people practicing exercises and other fitness activities
3. Green spaces / parks are used as water-catchment areas harvesting rain water potential
4. Green spaces serve as places for social mixing
5. Green spaces provide wide open spaces for accommodating public performances
6. Parks also support educational and several other life-long learning events

Gordon (1990), in this work on green spaces, warned that one of the alarming consequences of urbanization is the widening of the rift between human and natural environments. Urbanization world-over, has invariably had a negative impact on the green spaces. Calgary presently has sufficient green spaces for its residents. However, the city's population is growing at a rapid rate and the economic boom continues to trigger the establishment of buildings and infrastructure that eat into open spaces.

The above discussion justifies the first objective for our study namely, maximization of per capita green space. The PCGS (Per Capita Green Space) of a landuse plan is calculated by dividing the total available green space within the study area by the number of residents in the study area.

The PCGS for the hundred plans in a generation is calculated as follows:

```
for i = 1:100
        PCGS(i,1) = ( AreaGS(i,1) ) / Pop ;
end
```

Where,

PCGS(i,1) is per capita green space of plan i,

AreaGS(i,1) is the total green space area of a plan, and Pop is the population of study area,

PCGS is a row matrix with i rows and 1 column (each row corresponds to one plan).

The second objective is the maximization of per capita space for public service (PCPS). Public service includes all amenities that are needed for the daily life of the residents in a city. Most of the cities include transportation facilities such as roads, transit stations, rail road network, and various other public amenities under the category of public service. Just as green spaces are inevitable for recreation sufficient allocation of space for public service facilities ensures that the daily activities of people such as commuting, shopping etc are unhindered. More importantly, if the space available for various public service activities is below the required amount, then even the fundamental daily activities of people become complicated and this greatly affects the overall quality of urban life.

Consequently, the second objective of this study is chosen as maximization of per capita space for public service. The PCPS (Per Capita Green Space) of a landuse plan is calculated by dividing the total available space for public service within the study area by the number of residents in the study area.

The PCPS for the hundred plans in a generation is calculated as follows:

```
for i = 1:100
        PCPS(i,1) = ( AreaPS(i,1) ) / Pop ;
end
```

Where,

PCPS(i,1) refers to the per capita public service space of plan i.

AreaPS(i,1) is the total area available for public service of a plan,

Pop is the population of the study area., and

PCPS is a row matrix with i rows and 1 column (each row corresponds to one plan).

The third and final objective aims to increase the housing capacity by way of increasing the number of housing units in the urban milieu. Housing problems seriously deter the progress of a city. This was particularly so in the city of Calgary where serious housing problems were experienced during the year 2006-2007 and the problem is expected to continue unless drastic measures for housing are taken. Based on the Housing density, the landuse is classified into one of the following three categories:

1. Low Density Residential zone
2. Medium Density Residential zone
3. High Density Residential zone

In this study the following standards are used to determine the number of housing units.

**Table 3.2 Categorization of residential zones - low, medium, and high density**

| Density Range | Land use in Residential Areas in this study shall comply with the land use designations |
|---|---|
| Low Density | In Residential-Low Density areas, single-detached and semi-detached housing units with a density to a maximum of 25 units per net hectare shall be permitted. In addition, other forms of ground oriented housing units with a density to a maximum of 25 units per net hectare may be permitted, provided that these forms are compatible with the scale, urban design and community features of the neighbourhood. |
| Medium Density | In Residential-Medium Density areas, either ground or non-ground-oriented housing units with a density ranging between 26 and 50 units per net hectare shall be permitted. |
| High Density | In Residential-High Density areas, either ground or non-ground-oriented housing units with a density ranging between 51 and 185 units per net hectare shall be permitted. |

Src: Reproduced from www.burlington.ca/Planning/Official%20Plan/Part_III

Based on the above statistics, the number of housing units in a plan is determined as follows for the 100 plans in a generation.

```
for i = 1:100
NumHU(i,1) = round( ( (AreaResL(i,1) * 50) + (AreaResM(i,1)) * 100)  + (AreaResH(i,1) * 185 )
end
```

Where,

NumHU (i, 1) refers to the total number of housing units of plan i.

NumHU (i,1) is the total area available for public service of a plan,

Pop is the population of the study area., and

NumHU is a row matrix with i rows and 1 column.

Thus, we see that there are three objectives of the multiobjective optimization problem, all of which are to be maximized

1. PCGS – Per Capita Green Space
2. PCPS – Per Capita space for Public Service
3. NumHU – Number of Housing Units

Now, let us see the constraints of the multiobjective optimization. Constraints are limitations on the performance of the GA to yield results in such a way that some basic requirements are satisfied.

### 3.3.2 GA Constraints

Among the 10 landuse types seen in this study, two LU types need special consideration. These are,

1. UR (Urban reserve) and
2. DC (Direct Control)

Different set of uses and regulations govern DC landuse types in the urban environment. Simply stated, DC refers to 'tailor-made' areas that have been designed and/or designated for a specific project or purpose. Changes, if any, to such areas involve a whole lot public participation and high-level administrative brainstorming and such changes are rare as the DC areas have been created for specific purposes.

Similar is the case with Urban Reserve landuse designations. The boundary of city includes hundreds or thousands of hectares of area designated as Urban Reserve (UR). Such landuse types are not and will not be considered for landuse planning until a particular time frame is reached. The landuse types categorized as UR areas will continue to be undeveloped till the development of such lands becomes absolutely necessary and sufficient infrastructure facilities such as road network and other utilities are available.

Consequently, in this study, UR and DC are not changed and remain the same. These are imposed as constraints on the GA which will be discussed during the GA implementation. The final plan derived from this study must indicate the optimal combination of landuse values for these zones i.e. for each zone (or polygon) one value from among the integer values from 0 -9 should be specified.

Similar to the constraints imposed for not changing DC and UR landuse types, there are also other constraints with respect to some other LU types. Balling et al. (2004) argue that it is not appropriate to allow the zones to assume any values from this range. There may already be some hospital or public service utility or some educational institution located in several zones. Changing those to any values would mean that an existing hospital must be torn down and that landuse zone should be changed into a commercial or industrial zone. This is obviously unfeasible as much money and effort has already gone into such infrastructures and replacing them with a new landuse will amount to a gross waste of time and resources. Hence, some constraints can be imposed such as some zones can not be allowed to change and some zones can assume only a permitted value or values. For instance, zones designated as direct control (DC) or urban reserve (UR) fall under the purview of the city administration that have been reserved for specific activity. These can not be and must not be changed or designated as another landuse zone. By doing so, the search space is minimized and the plans so generated are practicable.

We discussed about two types of constraints in the earlier passages. In addition, two constraints are imposed which are in line with the objectives; but, are incorporated to make sure that no plan has a green space area or public service area below a minimum threshold.

It would be impracticable to tear down an existing university or hospital or any other such large facility to convert the same into green space or into another landuse. Hence, constraints are imposed within the GA framework to include this. Just as landuse zones that have designations as DC and UR are not allowed to change, those zones with large infrastructure elements such as hospitals and libraries are not allowed to change. These are imposed as constraints.

Also, regarding the green spaces and spaces for public service, in order to ensure that they do not fall below a particular minimum value, constraints are imposed to select only plans that have green space and public service spaces above minimum PCGS and PCPS values set in the constraints. Calgary already has a very high value of per capita green space. It has more than 37,620 hectares of green space for a population of about 1 million people. This amounts to 376 Sq.m. of green space which is much higher than other Canadian cities and other cities of the world as well. Hence, in order that this value does not go down in the process of increasing the housing capacity, this is also imposed as a constraint.

On the whole, the constraints are as follows:

1. Zones designated as DC are not to be changed
2. Zones designated as UR are not to be changed
3. Specific zones with large scale infrastructure developments are not be changed
4. For each plan, PCGS $\geq$100 Sq.m. and PCPS $\geq$ 50 Sq.m.

A zone that has a DC designation in the original landuse plan will continue to have the same value (integer value ‘2’) throughout the GA process and hence, in the final plan resulting from the GA, the corresponding zone will have a DC representation only. Similarly, a zone that has a UR designation in the original landuse plan will continue to have the same value (integer value ‘9’) throughout the GA process and hence, in the final plan resulting from the GA, the corresponding zone will have a UR representation only. All plans will be checked for satisfying the constraints of PCGS and PCPS. Only plans that exceed the minimum values (PCGS $\geq$ 375 Sq.m. & PCPS $\geq$ 100 Sq.m.) are considered as feasible plans and these are added to the starting generation.

### 3.3.3 Fitness Evaluation

Multiobjective problems can be optimized by iterative procedures. The objectives can be normalized using a simple and straightforward procedure that involves scaling. Normalization involves finding the maximum as well as the minimum values for each objective for a set of plans in a generation and then re-scaling using the following formula. The number of objectives is 3 in this study. The normalized objectives scores are given by

$$Obj_n = \frac{(Val - Val_{min})}{(Val_{max} - Val_{min})}$$

The above is a simple and straightforward technique of linear interpolation.

For each objective in the study, considering all the plans in a generation,

Val is the current value of the corresponding objective

$Val_{min}$ is the least value of all Val values of the plans in the current generation

$Val_{max}$ is the highest value of all Val values of the plans in the current generation

Thus, considering the three objectives concerning PCGS, PCPS, and NumHU, the normalized scores can be obtained as follows.

$Obj_1$ is maximization of Per Capita Green Space (PCGS)

$Obj_2$ is maximization of Per Capita Space for Public Service (PCPS)

$Obj_3$ is maximization of housing capacity by maximizing number of housing units (NumHU)

The plans need to be compared with other plans in the generation to find the fit ones in the generation.

Hence when a plan i is compared with a plan j, plan j is better than plan i if the difference between j and i is positive, as follows:

$$Plan_j - Plan_i = \left[\frac{(Val_j - Val_{min})}{(Val_{max} - Val_{min})}\right] - \left[\frac{(Val_i - Val_{min})}{(Val_{max} - Val_{min})}\right]$$

$$= \frac{(Val_j - Val_i)}{(Val_{max} - Val_{min})}$$

If there are three objectives, let $Range_1$ denote the difference in values between the maximum and minimum values of objective1, $Range_2$ denote the difference in values between the maximum and minimum values of objective2, and $Range_3$ denote the difference in values between the maximum and minimum values of objective3.

Hence,

$$Range_1 = PCGS_{max} – PCGS_{min}$$

$$Range_2 = PCPS_{max} – PCPS_{min}$$

$$Range_3 = NumHU_{max} – NumHU_{min}$$

The above comparison can now be rewritten as follows:

$$Plan_j – Plan_i = \left[ \frac{(Val_j – Val_{min})}{Range_1} \right] - \left[ \frac{(Val_i – Val_{min})}{Range_1} \right]$$

$$= \frac{(Val_j – Val_i)}{Range_1}$$

For measuring the fitness of the plans, the Maximin fitness function employed by Balling et al. (1999) is used. The fitness of each plan in a generation is calculated relative to that of the other plans in the same generation. The higher the values of PCGS, PCPS, and NumHU of a plan, the higher the fitness of the plan in comparison with the other plans of the generation.

Considering two plans $Plan_j$ and $Plan_i$, $Plan_j$ is superior to $Plan_i$ if $PCGS_j$, $PCPS_j$, and $NumHU_j$ are all greater than the corresponding objective values, namely $PCGS_i$, $PCPS_i$, and $NumHU_i$.

$PCGS_j > PCGS_i$, $PCPS_j > PCPS_i$, and $NumHU_j > NumHU_i$

i.e. $Plan_j$ is superior to $Plan_i$ if it exceeds it in all the three objectives.

This can be restated as follows:

$$\min (PCGS_j - PCGS_i,\ PCPS_j - PCPS_i,\ NumHU_j - NumHU_i) > 0$$

i.e. if the minimum of the above three differences is greater than 0, then $Plan_j$ is superior to $Plan_i$.

Each plan in a generation must be compared with all the other plans in the generation. If it is to be found whether a $Plan_i$ is dominated or not it is compared with all other plans using the aforementioned principle.

$$\max_{j \neq i}( (\min (PCGS_j - PCGS_i,\ PCPS_j - PCPS_i,\ NumHU_j - NumHU_i)) > 0$$

The fitness of the $i^{th}$ plan is obtained as follows:

$$f_i = \left[1 - \max_{j \neq i}\left[\min\left(\frac{PCGS_j - PCGS_i}{Range_1},\ \frac{PCPS_j - PCPS_i}{Range_2},\ \frac{NumHu_j - NumHU_i}{Range_3}\right)\right]\right]^p$$

Where,

$$Range_1 = PCGS_{max} – PCGS_{min}$$

$$Range_2 = PCPS_{max} – PCPS_{min}$$

$$Range_3 = NumHU_{max} – NumHU_{min}$$

$Range_1$, $Range_2$, and $Range_3$ represent the scaling factors for the three objectives PCGS, PCPS, and NumHU respectively, for all the plans in the generation. However, it should be noted that this value has to be computed for each iteration for every single generation. This is so because, the maximum and minimum values of each objective varies during each generation. Based on the fitness formula described above, it is possible to identify the Pareto-optimal plans from the fitness values obtained. While dominated plans have a fitness value between 0 and 1, Pareto-optimal plans have fitness values greater than 1.

In this study, a p value of 21 is employed. This is done in order to pursue Pareto-optimality more vigorously. This way, the fitness of those plans with $f_i$ more than 1 gets further higher, and the fitness of those plans with $f_i$ values less than 1 gets further lower (Balling et al., 1999).

### 3.4 GA Implementation

The GA is implemented with the objective of searching and finding a set of landuse plans, which meet the constraints imposed on the GA and maximize the objectives pertaining to green space, public service space, and housing capacity. The constraints are in place to make sure that a plan meets the minimum requirements for sustainable development as discussed in this study. Plans that satisfy the constraints are called 'feasible plans'. The final plans obtained from the GA must be Pareto-optimal with respect to the multiple objectives. Pareto-optimal plans are both 'feasible' and non-dominated. The word non-dominated implies that no other feasible plan in the generation is better than this plan in all objectives.

### 3.4.1 Starting Generation and Feasible set

The population of the initial generation is chosen as 100 in this study. This means that the initial generation will contain 100 plans each of which is a possible solution to the given problem. However, it is to be noted that this solution may not be optimal. The plans in the starting generation are generated by a random process wherein integer values are assigned to each of the 100 chromosomes containing 135 genes. In other words, 100 x 135 integer values within the range of 0 – 9 are generated at random and assigned to the chromosomes.

It should be recalled that among the set of constraints there are constraints that require the values of some land use zones to remain constant. For instance, an Urban Reserve in the original landuse zone will always have the value of 9 in the corresponding gene throughout the genetic algorithm process. Similarly, Direct Control zones will continue to remain the same throughout the iterations and in the final plan also the LU_CODE of the corresponding zone will be equal to DC. There are other constraints that require the per capita green space and per capita public service space to be above a minimum threshold.

Hence, the randomly generated plans will have to be checked to see if they satisfy the constraints.

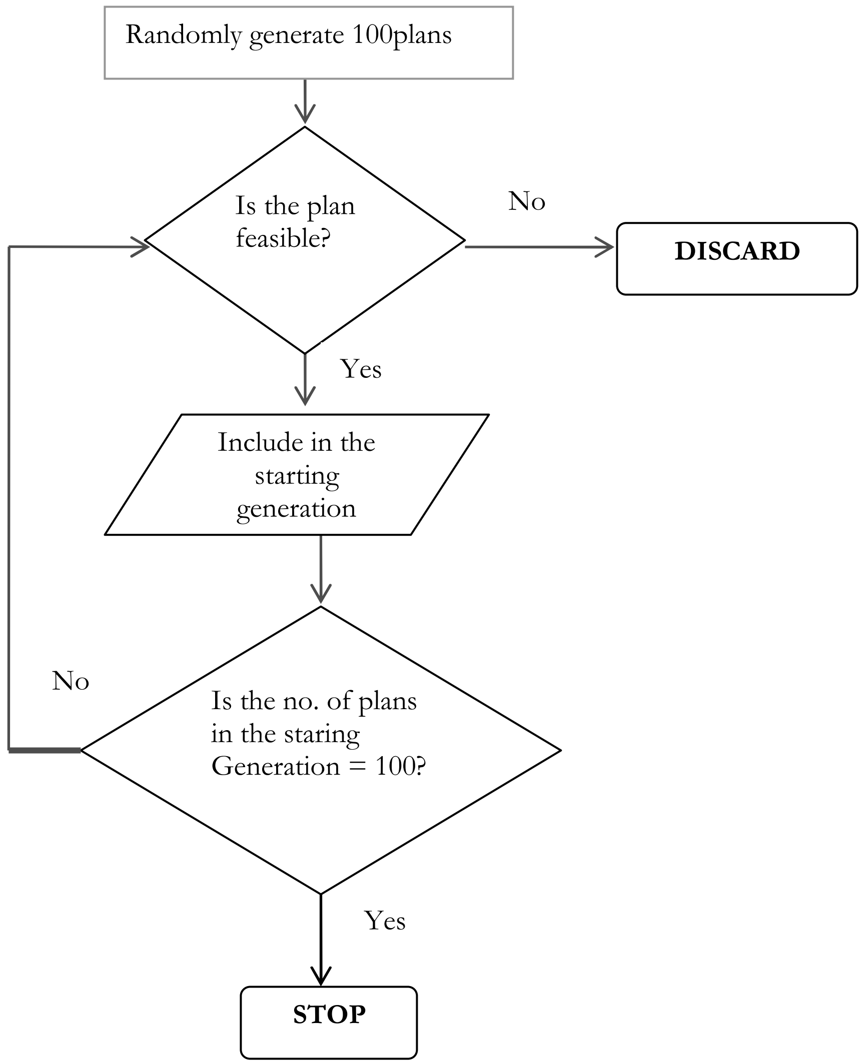



**Figure 3.2 Generation of 100 'feasible plans' for starting generation**

Figure 3.2 shows the framework for generating 100 feasible plans for the starting generation. During each iteration, the plans in the current generation are checked for feasibility based on the constraints. Plans that do not satisfy the constraints are discarded and those that satisfy the constraints are included in the starting generation. This iterative procedure is carried on till the starting generation has a total of 100 'feasible' plans.

The feasible plans for the initial generation were generated as mentioned above and a total of 1,213,400 plans were generated and tested in order to obtain these 100 feasible plans.

From this starting generation, the second generation is constructed using the GA methodology. The third is generated from the second, the fourth from the third and so on for a total of 100 iterations at the end of which a generation with 100 final, feasible plans results.

### 3.4.2 Subsequent generations

The process of generation of subsequent generations is explained here. While a portion of the total 100 plans is directly copied from the previous generation, the remaining plans are generated using the processes of selection, crossover, and mutation.

#### 3.4.2.1 Natural Selection

Evolutionary computation is inspired by the natural evolutionary processes. Hence, in line with the Darwinian theory of 'survival of the fittest', the fit ones from the previous generation are selected for the subsequent generation. The fraction of the total population of plans that is selected to the next generation is determined by the 'rate of selection'. In other words, the rate of selection refers to the percentage of the parent population that is retained for the subsequent generation. The number of chromosomes to be selected is an important decision. Retaining a lot of chromosomes from the parent population is not advisable as the bad traits from the parent population will continue to be passed to the inheritance. On the other hand, keeping only a very small proportion of the parent population will limit the total available genes in the subsequent generations.

In this study 20% of the parent population is kept for the subsequent generation.

$$Num_{Ret} = SelRate * Num_{ChrPop}$$

Where,

$Num_{Ret}$= No. of chromosomes retained,

SelRate = Rate of Selection, and

$Num_{ChrPop}$= No. of chromosomes in a generation

Thus we see that not all the plans, but only 20% (for a population of 100, 20% is 20) makes it to the next generation. Also, these 20 plans are the fittest plans from the generation. In order to the select the top 20 plans, the plans in the generation are sorted based on their fitness values. After sorting the plans according to their fitness, the best 20 plans are selected for the next generation. Now, of the 100 plans in the next generation, only 20 are available. The remaining 80 plans must be generated using the processes of selection, mating, and mutation.

#### 3.4.2.2 Selection for Pairing

This step involves choosing chromosomes from the parent population in order to generate offspring. In other words, this is the process of pairing or combining two parent chromosomes to produce two offspring or children. This section does not actually discuss the pairing or mating process, but is about the process of selecting parents for mating. Different kinds of selection methods namely random pairing, top to bottom pairing, weighted random pairing exist. In this study, tournament selection is used. Tournament selection strongly imitates the natural process of mating. This method of selection of parents for mating involves randomly selecting a small subset of chromosomes (5 in this study) and from this selection, the chromosome with the highest fitness is selected as one parent. The same procedure is repeated to select the other parent. One prominent advantage of tournament selection is that the population need not be sorted. This is particularly useful in the case of large populations as in this study. The Matlab code to accomplish the process of selection of parents for the mating process to generate offspring for subsequent generation is illustrated subsequently (Figure 3.3).

```
% Select Parents for Crossover or Pairs for mating
% Here two chromosomes are needed, Parent1 and Parent2


        Parent1=randint(1,135);

        Parent2=randint(1,135);

        Parent1(:,:)=0;

        Parent2(:,:)=0;

        Pick1=randint(5,1,100);

        Pick2=randint(5,1,100);

% As in Tournament selection a random set of 5 numbers   for    the
% chromosomes are selected and the maximum     number is chosen for
% the parent chromosome number1 and similarly for number 2

% The maximum number is chosen as the plans in the
% previous generation has already been sorted

        Pick1a=0;
        Pick1a=max([Pick1]);
        Pick2a=0;
        Pick2a=max([Pick2]);
        Parent1=NextGen(Pick1a,:);
        Parent2=NextGen(Pick2a,:);
```

**Figure 3.3 Matlab code for selecting two parents for mating**

### 3.4.2.3 Mating

Using the parents selected in the selection process, two offspring are created via the process of mating. Mating, in its simplest and most common form, involves two parents that mate to produce two offspring. Let us now look at this simplest form of crossover known as single-point crossover. In this type of crossover, as the name implies, the crossover occurs at a single point known as the crossover point. The crossover point is where the swapping occurs. This point is chosen randomly and it lies between the first and last genes of the chromosomes. At first, one of the two members of the mating pair, called $Parent_1$ provides the genes to the left of the crossover point to the first Offspring and the second member of the mating pair, $Parent_2$ provides the genes to the right of the crossover point to the $Offspring_1$. Thus, the $Offspring_1$ is now contains material from both the parents. Similarly, the second offspring is generated by combing material from $Parent_1$ and $Parent_2$. The genes to the right of the crossover point from $Parent_1$ and that to the left of the crossover point from $Parent_2$ are combined to produce $Offspring_2$. Other alternative forms of crossover are also available.

### 3.4.2.4 Mutation

After mating, mutation is performed to introduce qualities that are not originally present in the parent population. Mutation involves randomly changing a selected number of genes in specific chromosomes obtained from the earlier process. In this study, the mutation probability is chosen as .05 (5). Mutation is typically applied to the offspring generated from the earlier step, subject to the mutation probability. Now a random number between 0 and 1 is generated for each gene in the two offspring. If the random number is less than the above probability of mutation (.05), then the integer value of the gene is changed to another random value between 0 and 9.

The above processes cumulatively represent the complete process of creating a new generation from an earlier generation. This constitutes one sequence of iteration. The whole GA process involves 100 iterations at the end of which the Pareto set containing the Pareto-optimal plans is obtained.

The fitness values of the individual plans in the generation are calculated using the fitness formula described earlier. Plans with higher fitness values have higher Pareto-optimality an d hence are more 'fit' than the rest of the plans in the generation. The plans altogether constitute the Pareto set.

### 3.4.3 Pareto plans

Plans belonging to the Pareto set are called non-dominated plans. This is because no other plan exceeds the Pareto plan in all the objectives. A plan may outdo the Pareto plan in one objective and yet another plan may outperform the Pareto plan in another objective; however, no single plan surpasses the Pareto plan in all the objectives. The Pareto set is devoid of the influence of the relative significance of the various objectives. Hence, plans not belonging to the Pareto set are called dominated plans since Pareto plans that surpass these plans have been found. Pareto plans significantly aid the process of decision-making as planners and administrators need not sift through hundreds of thousands plans; but, they can merely search the Pareto set to find an optimal plan.

However, there is still one shortcoming. Decision makers still are confronting a set of plans from which they have to choose one plan. This process can not be automated as now the relative significance of the various objectives based on the ultimate development goals should be considered. This study proposes the use of an objective evaluation tool, visualization, to evaluate the Pareto plans. This way the decision makers can select a subset from the Pareto set and evaluate the plans in this subset visually to select the plan that most suits their requirements.

### 3.5 Visualization

This study uses an innovative approach combining a genetic algorithm with a visual evaluation tool to generate sustainable future landscape scenarios. The proposed method will aid decision-makers and planners in the process of informed decision-making as a visualization tool is used to evaluate the solutions provided by the Pareto set.

This chapter will now discuss about virtual reality, object oriented programming (OOP) for visualization, hierarchical framework of landuse scene representation, component diagram, and building the visualization scenario for Pareto optimal plans.

### 3.5.1 VR-Based Visualization

When talking about 3D visualization another term that needs to be discussed is 'Virtual Reality'. There exist numerous definitions for 'Virtual Reality'. Plainly stated, Virtual Reality is a tool for 3D data visualization that helps visualizing and interacting with data. Much of the 3D visualization in today's applications is done in a virtual space that is often described as 'virtual worlds'. The reason they are called virtual worlds is that they are not actually 3D worlds in real space, but they are digital or cyber worlds that have their own coordinate systems and define a 3D virtual coordinate space within which applications can be built. The users can navigate within these virtual worlds, move the objects in the worlds, rotate or scale them, and transform them in multiple ways. These virtual worlds facilitate user interaction with the 3D objects and provide a sense of immersion.

In this study visualization is used as a tool to evaluate the Pareto plans objectively. Many GAs generate optimal solutions via iterative optimization procedures. However, the work stops there and then subjective measures are employed to select one plan for implementation. The justification is provided clearly by Balling et al. for the need of an additional tool. Balling et al. (2004) state that *'A plan is a member of the Pareto set if no other single plan has been found that is better in all objectives. The Pareto set's beauty is that it is independent of the relative importance of the objectives'*. Nevertheless, for a certain combination of weights on the objectives, the optimum plan will be a particular constituent of the Pareto set. For yet another arrangement of weighting factors, the optimum plan will be another Pareto plan belonging to the Pareto set.

As on date, numerous tools are available for visualization. One among them is VRML, a standard specification that has been adopted world-wide by numerous visualization programmers to create virtual worlds. VRML is the acronym for Virtual Reality Modeling Language, the latest standard to display 3D models on the web. There are several standards and specifications to describe 3D scenes and objects contained in these scenes. VRML is regarded by many as the origin for several basic notions of three dimensional modeling and the latest standards like X3D are believed to be extensions of the ideas. X3D is regarded as the XML based revised version of VRML. Other 3D formats that are quite popular include Java3D and OpenGL which contain libraries of 3D classes. VRML is a file system for creating 3D virtual objects and environments. Just as Microsoft word files have a .doc extension, VRML files have a .wrl extension. The origins of concept of spatial immersion can be dated back to 1965 when Ivan Sutherland (1965) put forth the ideas of immersion in virtual space in his influential work, "The Ultimate Display". The VRML Repository defines the Virtual Reality Modeling Language (VRML) as a 'standard language for generating interactive 3D environments and sharing those worlds across the Intern

Virtual Reality Modeling Language (VRML) is a file format for creating interactive 3D worlds and is presently also considered as a universally accepted format for multimedia applications and 3D graphics utilities. Many of the present notions for modeling 3D worlds are derived from VRML, these schemes are currently being expanded in X3D, XMT, MPEG4/BIFS etc. The fundamental arrangement of the language is based on object oriented programming concepts. A VRML scene is defined by a group of objects. The scene is defined by "nodes". These nodes are analogous to what are called objects in OOP terminology.

### 3.5.2 Object Oriented Programming for Visualization

Just like there is a classification of programming languages as high and low level programming languages, there are also classifications that categorize programming languages as procedural and object oriented programming (OOP) languages. In procedural programming, lines in a program, called statements, are grouped together to form pieces of code called 'procedures'. These procedures, (also identified as functions sometimes) perform specific tasks. For instance, performing a simple calculation as addition, or doing advanced computational tasks.

On the other hand, object oriented programming languages are composed of 'components', which are software 'Objects' that can be used again or called again later when needed. This 'reusability of objects' is of tremendous advantage and saves a whole lot of coding efforts and of course, precious time. Just like the built-in or user-defined functions in programming languages which the user can use as and when required in programs, these objects once created can be used multiple times.

Object oriented programming revolves around objects and their properties. In a way, this is very much similar to visualizing a scenario. A visualization scene can be considered to be composed of objects with properties. In the preceding sentence, the phrase 'objects with properties' is highlighted or emphasized, since these properties determine how an objects looks and/or behaves. Let us consider a sample scene. Say a bus-stop near a building in an urban locality. If we further break this down into smaller fragments, we will find that there must be a bus-stop, a building, roads, lamp-posts etc. Further the bus-stop must contain signboards, passenger seating facilities etc. The furniture may be of a particular material, color, and dimensions. All these are the attributes of the furniture. Similarly, each element has its own characteristic features or attributes. Thus a scene is composed of elements or objects, each of which has its own properties or attributes. A parent object can include any number of children, which can be grouped or assembled to function as one single entity. Figure 3.4 illustrates the representation of the above scene hierarchy. It can be seen that one parent object contains several children. As the user may desire or as may be required, any number of children nodes can be included within the parent node.

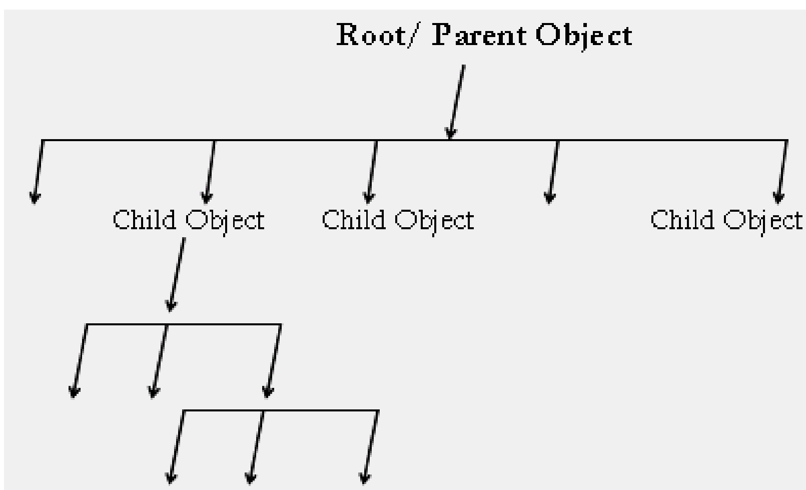


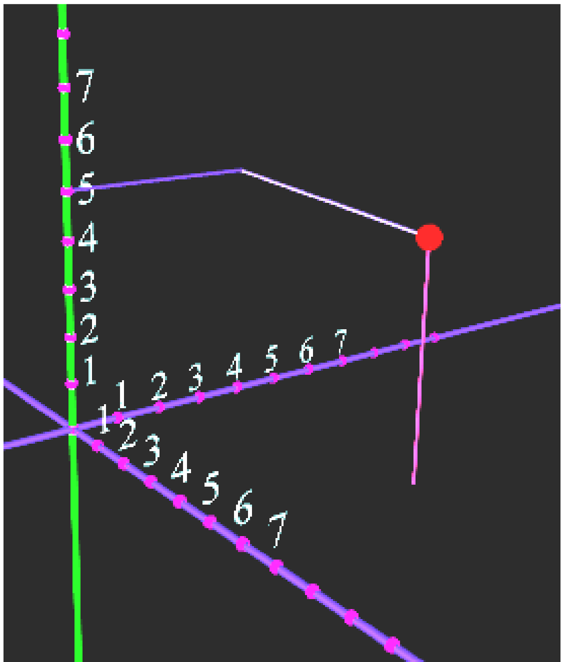

Figure 3.4 Hierarchical structure of scene representation

Employing object-oriented features makes things easier in the course of devising virtual worlds. The elements in a scene can be grouped in the form of a hierarchical structure in the form of parent child relationships. For instance, 'the wall' object can be grouped as a child of 'the house' object. This sort of hierarchical approach is particularly useful in animation of a scene when it is required to animate different objects differently. The word hierarchy refers to a categorization or order. In the simplest form, members of a hierarchy are shown one below the other in their order. This sort of hierarchical arrangement helps in the step-by-step design of the object and also understanding the framework at any later stage.

Subsequently, the very fundamental entities used to build 3D objects in virtual reality are discussed. These include points, lines, and faces.

### 3.5.3 Basic Entities for VR-Based Visualization: Points, Lines, and Faces

A point represents a specific location in 3D space. Every point occupies a well defined location in space. For instance, the coordinates (4, 5, 6) represent a point that is 4 units from the origin along the positive x-axis, 5 units from the origin along the positive y axis and 6 units from the origin along the positive z-axis (Figure 3.5a). In virtual world conventions, an increasing value of z moves the point towards the viewer and a decreasing value of z takes the point away from the viewer. The equidistant yellow dots on the axes show the increasing distance in meters. The point is 4m from YZ–plane; 5m from XZ; 6m from XY.



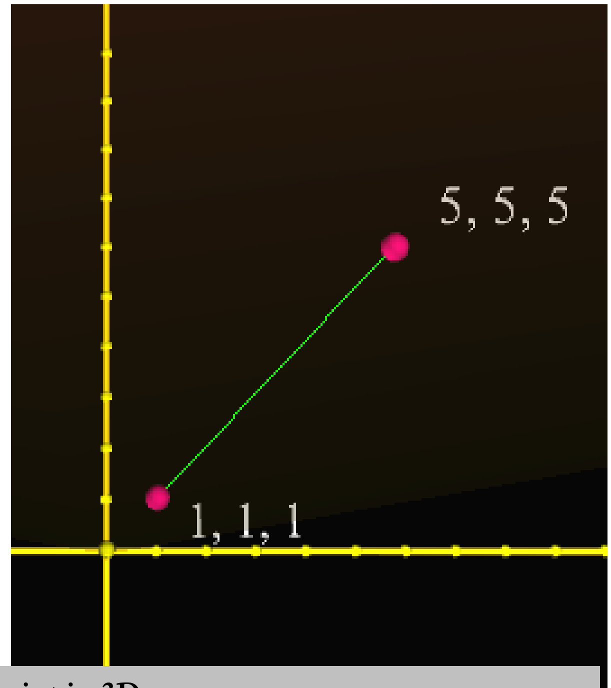




**Figure 3.5a A point in 3D space**
**Figure 3.5b A line between points**

Having discussed about the 'point', the next entity in that order is the line. Mathematically, a straight line is the shortest distance between two points. The line in the above figure connects two points, (1,1,1,) and (5,5,5,) (Figure 3.5b).

The next item of discussion is a face or polygon. Simply stated, a face means a flat surface. For instance, XY, YZ, XZ are all planes. The plane XY is perpendicular to Z-axis, the plane YZ is perpendicular to X-axis, and the plane XZ is perpendicular to Y-axis. Lines or points lying on the same plane are called co-planar. A face can also be considered as a plane or as being in a plane. Figures 3.6a and 3.6b illustrate the vertices and the generation of a face using the vertices.

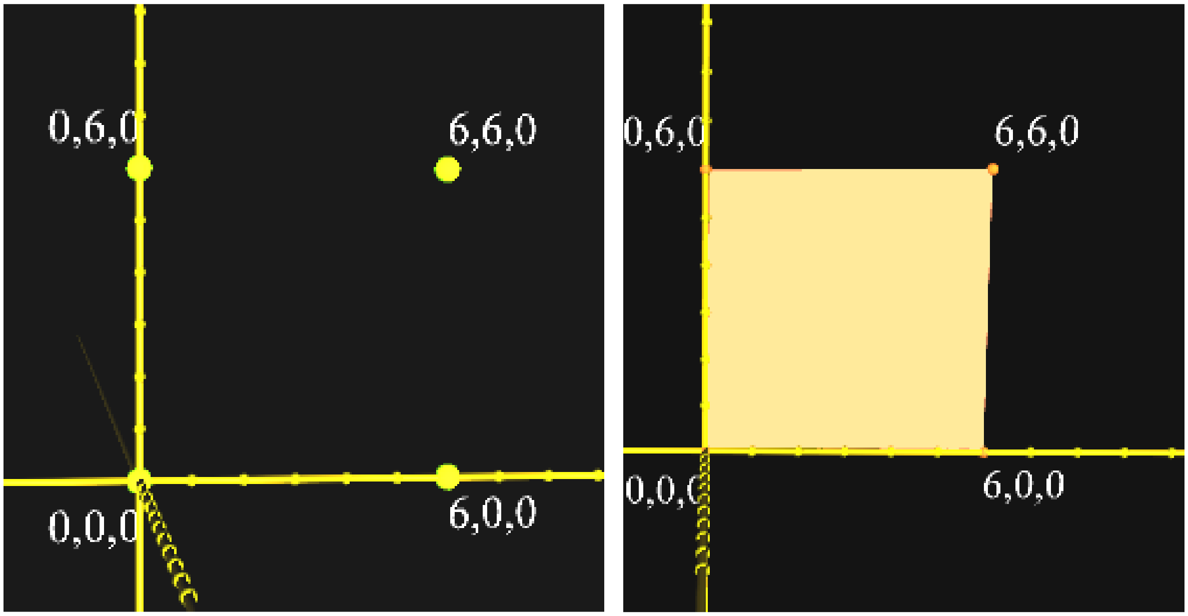



Figure 3.6 a Vertices of a face in 3D coordinate space
Figure 3.6b A face or polygon corresponding to the above vertices

The words plane and face are used synonymously on several occasions. In simple terms a plane is a flat surface. Many other words that are used synonymously with side include surface, face, area, region etc.

The fundamental elements of describing a 3D scene in VRML were discussed above. The subsequent sections discuss some of the specific elements of VRML for modeling 3D geospatial data such as landuse modeling and 3D landscape generation.

### 3.6 Landuse Modeling in 3D VRML

VRML 2.0 describes real world scenarios in the form of scene that consists of a hierarchical scene graph. VRML is based on a notion similar to the concept of OOP. Real world objects are described as shapes with geometry and appearance. All features such as buildings, roads, trees, and rivers can be modeled as shapes which can be grouped together and transformed (translated or rotated) within the coordinate system within which they are built.

By and large, a vast number of objects, however complex they might be, are built using the fundamental shape node (Figure 3.7) with the principal fields namely geometry and appearance. The geometry field is used to describe the geometric properties of the object and the appearance field is used to describe how the object looks. Realistic environments can be built by judicious use of textures and scaling them to accurately match the faces. The appearance node in VRML includes a texture factor that refers to the URL containing the image to be superimposed or overlaid on a particular face of the VRML object.

```
#VRML V2.0 utf8

Shape {
    geometry Box {
    }
    appearance Appearance {
                    material Material {
                    diffuseColor 1 0 0
                }}
}
```

Figure 3.7 Shape Node with Geometry and Appearance Nodes

Invariably, geospatial data fall under either of the two broad categories namely vector and raster, which in a sense correspond to features that can be represented as either discrete or continuous. The former represents spatial features in the form of points, lines, and polygons, while the latter represents features in the form of continuously varying values in grid cells namely pixels. DLGs (Digital Line Graphs) and ArcInfo Coverages are examples of vector formats, while scanned maps and DEMs are examples of raster format. In VRML terminology, an Elevation Grid can be considered the equivalent of a DEM, while an IndexedFaceSets might be thought as the equivalent of a TIN, just as a face in VRML corresponds to a polygon in GIS terminology. The IndexedFaceSet node used to represent TINs consists of coord and color Exposedfields, and coordIndex and colorIndex fields. The website http://www.vrml.org/ can be referred for further details on the node syntax and other standard VRML 2.0 representations.

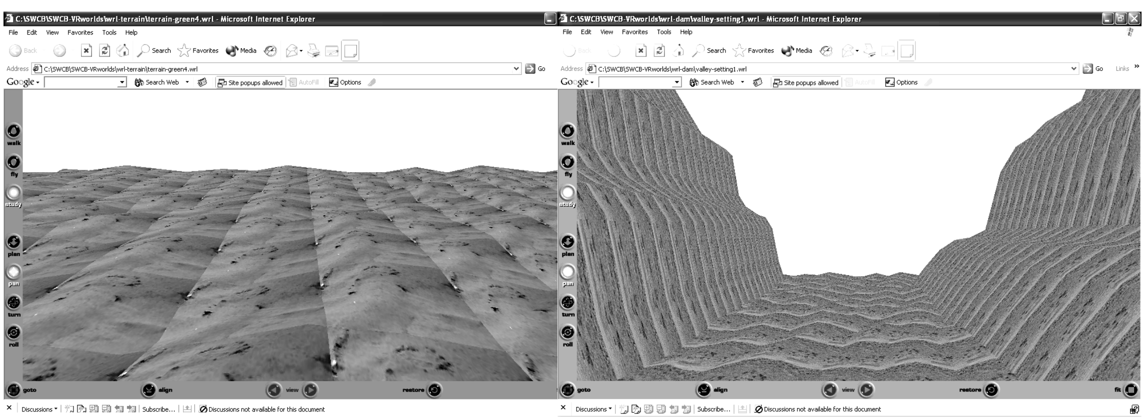


Figure 3.8 Terrain Generation using VRML Elevation Grid
a. A considerably flat terrain - few undulations b A Valley Setting
(*Chandramouli and Huang 2006)

The ElevationGrid node used to represent the GRID model is composed of xSpacing, zSpacing, height fields and the color ExposedField. Thus, VRML supports 2 types of modeling appropriate for terrain elevation data: IndexedFaceSet and ElevationGrid, which in GIS terminology are referred TINs and gridded DEMs, respectively (Chandramouli et al. 2004). In case of ElevationGrids the height values are shown along the y axis with eastings being drawn in the positive x plane and northings sketched the negative z plane. However, huge arrays of grids adversely affect the system performance since the movements with the world become slow to as the node specification consists of an intricate geometric configuration. Terrains generated using the ElevationGrid nodes are illustrated in Figure 3.8.

Nevertheless, several VRML world authors regard IndexedFaceSets as a primary means of geometric modeling of objects in VRML. IndexedFaceSets are a type of geometry node characterized by a list of x, y, z coordinate points. Next to the list of coordinates is an index containing the details of the order of connectivity of these coordinates to build a face. IndexedFaceSets are considered to be an efficient way to increase performance in VRML browsers while building realistic and efficient VR environments. Building realistic terrains continues to be a daunting task. In addition to efficiently scaling the texture images and manipulating the texture transform operations, creative employment of lighting function within the virtual world tremendously adds to the reality of the world. Another important feature geometry functionality in this context fall under the group of lattices, which are shown as IndexedFaceSets and are obtained from DEMs (Digital Elevation Models).

### 3.7 Component Diagram for visualizing Pareto-optimal plans

Visualization scenarios corresponding to the Pareto solutions need to be generated to evaluate them. Several programming paradigms are available to render 3D scenes and objects. We already discussed the notion of Object Oriented Paradigm. Accordingly, the fundamental idea is to split any object into its component parts and to decide how to model these components. Among the component parts there may parts which are exactly similar but are at different locations or are oriented differently. A landuse scenario contains various elements such as terrain, trees, road network, buildings, and other infrastructure elements.

As mentioned previously several programming paradigms are available to render 3D scenes and objects. We have already discussed the Object Oriented Paradigm in detail. Accordingly, the fundamental idea is to split any object into its component parts and to decide how to model these components (Figure 3.9). Among the component parts there may parts which are exactly similar but are at different locations or are oriented differently.

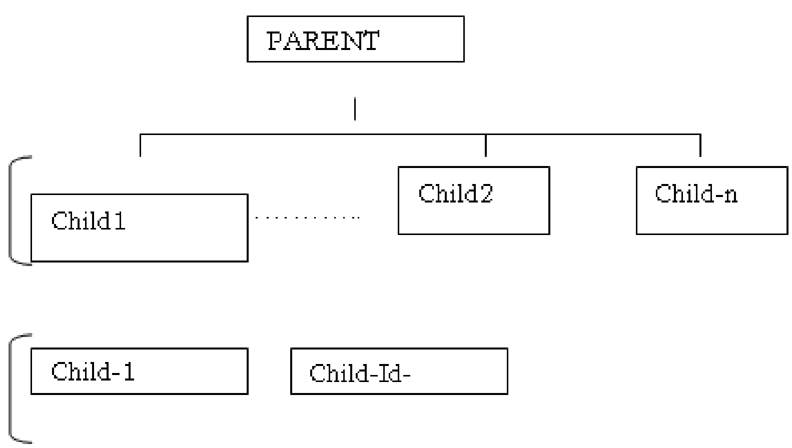



Figure 3.9 Component Diagram for visualizing Pareto-optimal plans

For the sake of simplicity and in order to better elucidate the concept of scene hierarchy within VRML, let us consider a table and a chair and break down the constituents (Figure 3.10).

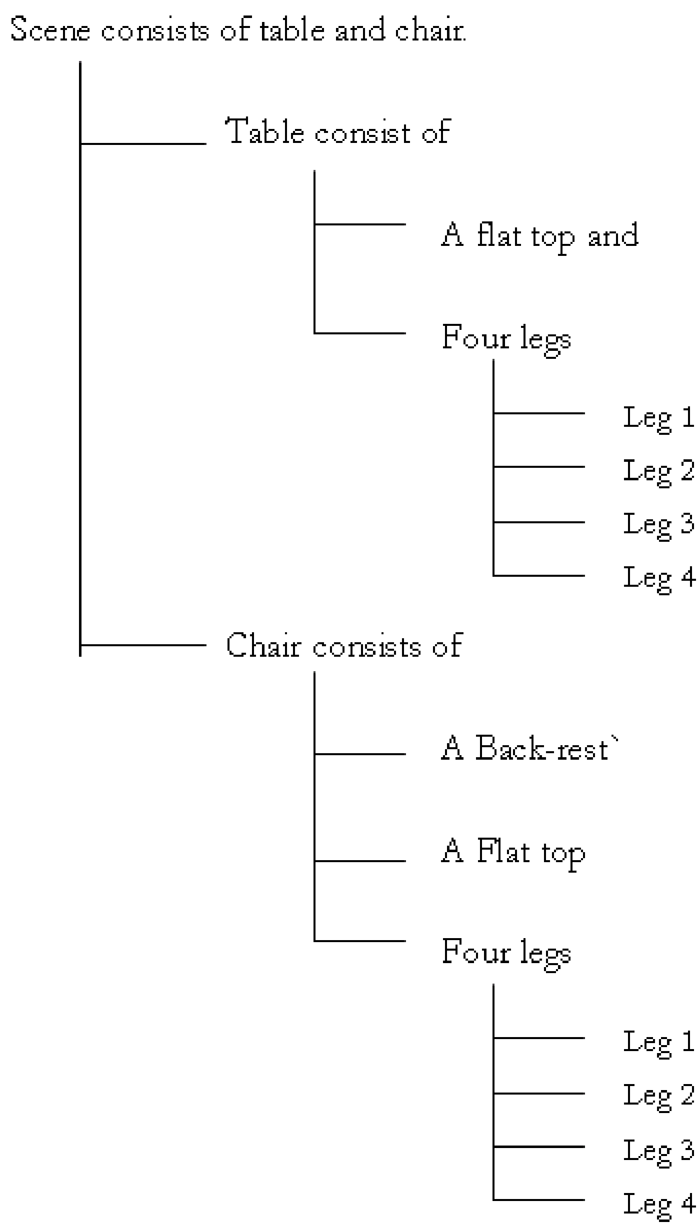



Figure 3.10 Parent-child relationship for a sample 3D scene with table and chair

If we draw a hierarchical framework for the same, it will be as follows:

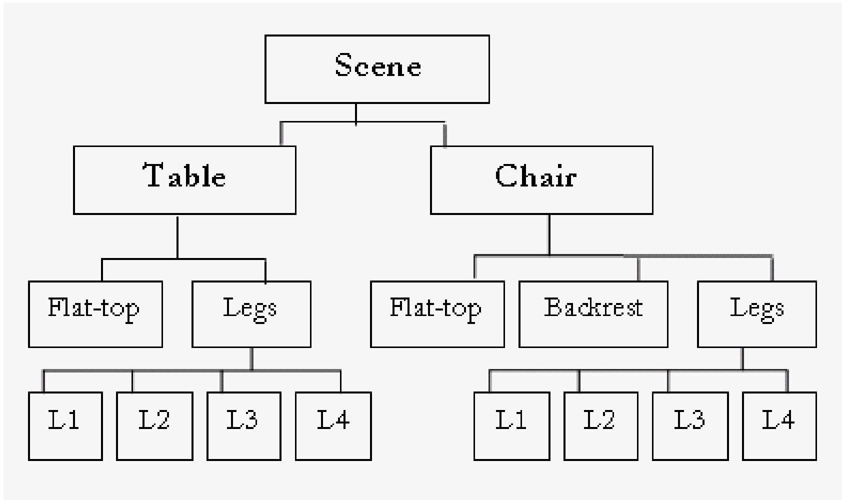



Figure 3.11 Hierarchical representation of the Table-Chair Scene

The parent object or the scene consists of two children namely Table and Chair. The Table object in turn consists of two children and there are four constituents under the parent object 'Legs' (Figure 3.11). Actually, all the four legs have the same appearance or properties, except that their location is different. So, only one time it needs to be created and it can be used four times at different positions.

In this study, real world scenarios are described in the form of a hierarchical scene tree structure as described in the previous paragraphs. VRML is based on a notion analogous to the OOP (Object Oriented Paradigm). The object-oriented approach models real world objects as shapes with geometry and appearance. All features such as buildings, roads, trees, and rivers can be designed and modeled as shapes which can be grouped together and transformed (translated or rotated). By and large, a vast number of objects, however complex they might be, are built using the fundamental shape node with the principal fields namely geometry and appearance. The geometry field is used to describe the geometric properties of the object and the appearance field is used to describe how the object looks. Realistic environments can be built by judicious use of textures and scaling them to accurately match the faces. The appearance node in VRML includes a texture factor that refers to the URL containing the image to be superimposed or overlaid on a particular face of the VRML object.

A landuse scenario can be depicted using VRML as follows:

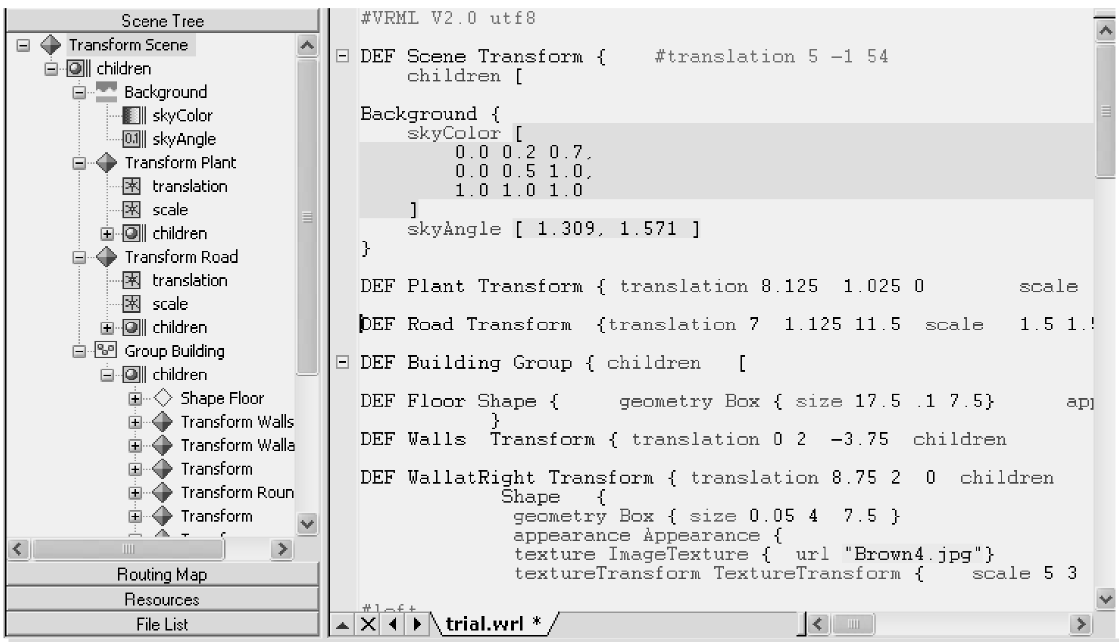



Figure 3.12 Scene Tree Structure in VRML

From the code (Figure 3.12), it is evident that various elements are grouped under the whole scene or root object, and each of these element has further ramifications based on the complexity of the object. For instance, a building has components such as floor, walls, etc. which have further divided into appropriate elements.

Using the same principles described in this chapter, more complex objects are generated by grouping smaller objects. The following figure (Figure 3.13) shows buildings modeled within VRML using indexed face sets.

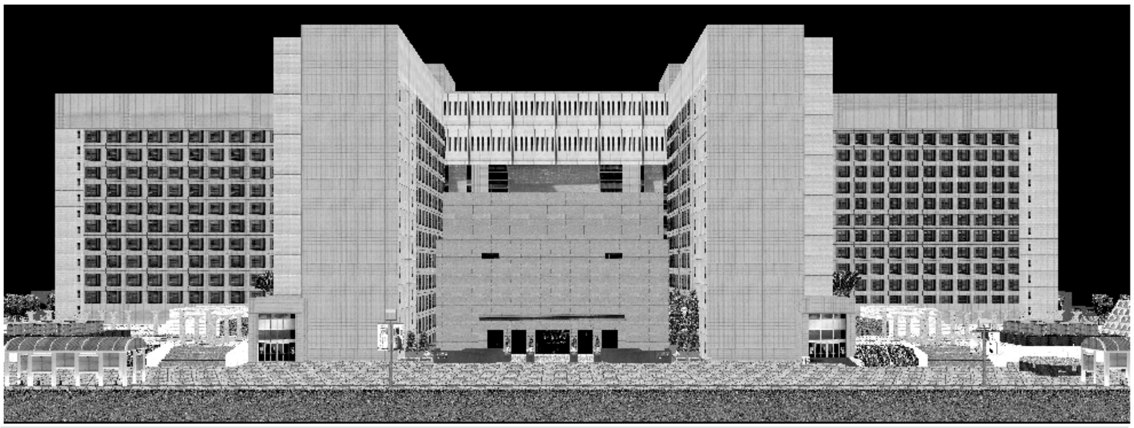


Figure 3.13 Building modeled for a Pareto-optimal plan using Indexed Face Sets

While an object itself might contain several parent-child relationships, the whole scene consists of numerous parent-child relationships among its various components.

While visualizing VRML scenes, a plug-in is necessary to view the scenes in a standard browser such as IE or NN. In this study, Cortona Client VRML plug-in is employed, which can be downloaded for free from www.parallelgraphics.com. This study uses a combination of IE (Internet Explorer 6.0) with Cortona VRML client to visualize the 3D worlds.

Once the 3D component of the model generation is done, any descriptive information or annotation can be associated with the corresponding object. This may include any type of information as the application demands. For instance, an application might demand the linking of GPS positional information and another application might require facilitating user-input by means of visual basic forms. This is done by means of the 'Anchor Node' construct (Figure 3.14). Similar to the Anchor node functionality, another node that is used in the construction of databases or 3D worlds is an Inline node illustrated above. A VRML scene can be divided into a set of files. This not only simplifies the design of the VR world, but also facilitates reusing parts of the world already built in one scene in many other worlds. In the case of a building, a set of shapes that draw a room are grouped together and stored under a single node. This group node can be given a specific name, say 'Room1', and reused later in another world. The Inline node permits the specification of a URL from where a specific file or data can be retrieved and reused in another file.

```
DEF Scene Transform {
    children [
# Hyperlink
Anchor {
url ["Object1.wrl "]
description "Hyperlinking"
}  ]}
```

```
DEF Scene Transform {
    children [
# Reusing software object using INLINE
        Inline {
            url "Object2.wrl"
            bboxCenter 0 0 0
            bboxSize -1 -1 -1
        } ]}
```

Figure 3.14 Anchor node and Inline node functionalities

## 3.8 Conclusion

This study uses visualization as a tool for evaluating the Pareto optimal plans generated using GA-based multiobjective optimization. While GA is useful in the generation of a set of Pareto-optimal plans that meet multiple objectives stated in the study, visualization using virtual immersive models facilitates selecting one plan that is most appropriate for the problem on hand. Such immersive models are of immense value in the planning and decision-making processes involving terrain and building databases. Such models serve as valuable tools in areas such as landscape studies, land use change detection, and urban-rural conversions, and sewage and water supply schema. Several techniques have been tried and implemented for visualizing 3D geospatial data. This study centers on VRML, one such visualization technique, which has proved to be quite efficient in building 3D scenes. One important aspect is the problem knowledge and experts with problem knowledge may not necessarily able to interpret the results of optimization procedures easily. Using such 3D visualization landuse scenarios, experts may be able to better understand the results of the optimization process as now the results are shown in a visual form rather than tables and statistical data. Thus problem knowledge can be used for evaluation and visual tool facilitates this. Experts with domain specific knowledge and expertise can easily analyze the solutions by means of the visual Evaluation tool.

# Chapter 4: Results and Discussion

In Chapter 3, the basic methodology of using genetic algorithms for performing multiobjective optimization as applied to a landuse problem was discussed in detail. Subsequently, Chapter 3 discussed the use of virtual reality to generation 3D visualizations for visualizing Pareto-optimal plans. In this chapter, the results of processes of multiobjective optimization and visualization will be discussed.

The GA-based MOO integrated with VR-based visualization described in the previous chapter was tested on a study area. In this study, the study area selected was central Calgary region.

## 4.1 Study area

The cities of Canada are home to more than 70% of the nation's total population. Hence, in this study, one such metropolitan city is chosen for the multiobjective optimization process. The city of Calgary was chosen as the city for the study. The city of Calgary in the province of Alberta is a rapidly growing city. Calgary is no exception to the universal trend of geometric population growth and especially, with the province of Alberta experiencing a massive boom in its economy, Canadians from elsewhere and other nationalities are continuing to immigrate to the City. The population of the city of Calgary as on April 2006 was 991,759, which has now exceeded a million. With the province of Alberta experiencing a massive boom in economy in the recent years, the population growth rate is expected to escalate in the next several years. Hence, this study has selected such a rapidly growing city. This study illustrates the generation of optimal futuristic landscape scenarios using the integration of GA with a visual evaluation tool.

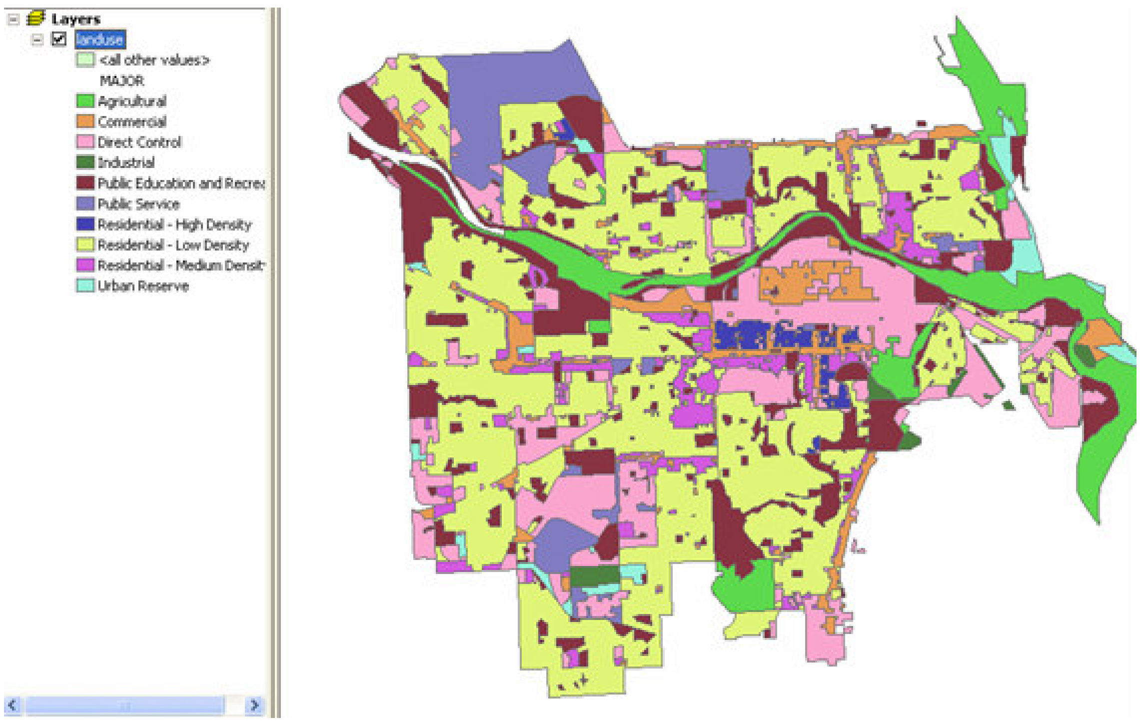



Figure 4.1 Study Area – Central Calgary Region

The landuse maps (Figure 4.1) were obtained from MADGIC, a geospatial data repository in the library of the University of Calgary and from the City of Calgary website. Data were basically in ArcGIS formats (.shp, .shx, .sbn, .dbf, .prj).

The study area was selected after consideration of the following facts:

- The study area had a decent mix of all common landuse zones such as commercial, industrial, residential etc.
- Within residential category, the study area consisted good proportions of low density, medium density, and high density residential zones
- The study area still has considerable room for modifications and if necessary expansion

To start with, the initial data consisted of the entire landuse map of the city of Calgary. In this study, polygon based (shape file) data is used. From this the central region of the city of Calgary is selected as the study area. Subsequently, this region is divided into 135 polygons representing 135 zones. The custom zoning map created as above has an associated attribute table that contains various details such as the land use, area, and population corresponding polygon. A polygon that contains high density residential buildings is categorized as Res-H zone. A polygon containing green spaces is categorized as GS and so on. Zoning determines the size and use of buildings, where they are located and, in large measure, the densities of the city's diverse neighborhoods. Zoning is a key tool for carrying out planning policy. The zoning (land use designation) is the primary legal control on the use and intensity of development on a parcel of land (not on the design of a project). If a proposed use for a parcel of land is not allowed in the land use district, it is possible to apply to re-designate the parcel.

### 4.2 GA Parameters for Experiments

Tournament selection was employed with a tournament size of 5. 5 plans were selected at random from the parent generation. Two feasible plans with high fitness values were selected as parents and mating was performed to generate two children chromosomes. A random set of 5 numbers for the chromosomes are selected and the maximum number is chosen for the parent chromosome number 1 and similarly for number 2. The maximum number is chosen because as the plan in the previous generation has already been sorted according to fitness.

In order that the same chromosome is not selected both the times, the algorithm performs a check. This makes sure that the same chromosome is not selected as Parent1 and Parent2 for mating operation. If this was the case, then the resulting two children chromosomes will be identical and will be the same as the parent chromosomes, as the swapping is between the two segments of the same chromosome (as both parents are same).

In this study, single-point crossover was employed. Single point crossover involves generating a random integer value between 1 and the total number of genes in the chromosome representing the landuse plan. In this case, the random number is between 1and 135, as the chromosome has 135 genes. By performing crossover two new offsprings are produced. The two parents are cut after the crossover point and the portions are exchanged to create the offsprings. The first offspring has values similar to the first parent till the crossover point and after that it has values similar to the second parent. The Second

Offspring has values similar to the second parent till the crossover point and after that it has values similar to the first parent.

The crossover point is chosen as a random value as follows:

CrossOverPoint= (randint(1,1,135) )+1 ;

The addition of 1 ensures that the crossover point is not zero. Once the crossover point is chosen, the pairing is done as mentioned above. For instance, let us assume that the random number generated is 75. Then, the first child chromosome will have genes identical to Parent1 till the $75^{th}$ gene, and will take the gene values of Parent2 from 76 to 135. The second child chromosome will have genes identical to Parent2 from 1 to 75 and will take the gene values of Parent1 from 76 to 135.

Subsequently, mutation is performed. Mutation involves introducing random changes in the gene structure of chromosomes. This is similar to the mutation that occurs in the natural processes of evolution. After the pairing process, the resulting chromosome population must be mutated, subject to results of the comparison of the random value generated with the mutation probability.

In this study, a mutation probability of 0.05 is employed. In real-world, mutations are a rare phenomenon and hence a low-value for the mutation probability. Subsequently, a random number between 0 and 1 is generated for each gene in the two offsprings. If the random number is less than the probability of mutation, then the integer value of the gene is changed to another random value between 0 and 9.

```
ColIndex=1;
for ColIndex=1:135
    RandVal = rand;
    NumIntValuesChanged1=0;
     if (RandVal <=  MutProb )
         OffSpring1(1,ColIndex)=randint(1,1,10);
         NumIntValuesChanged1=NumIntValuesChanged1+1;
     end
```

Figure 4.2 Gene-wise mutation:- Matlab Code

As gene-wise mutation (Figure 4.2) is employed here, a random probability value is generated for each gene in the chromosome obtained after pairing and if it is less than the mutation probability, then the corresponding gene is changed.

### 4.3 Variation of GA parameters

The efficiency of the genetic algorithm is influenced strongly by the parameters of the algorithm (Deb, 2001). These include.

- Generation size
- Tournament size
- Crossover probability
- Mutation probability

Of these, the tournament size and the mutation probability are of paramount importance. Hence, the genetic algorithm was executed for varying values of tournament size and mutation probability. These values were classified in the range of low, medium, and high.

For low, high, and medium values of tournament size, the GA was executed for low, high, and medium values of mutation probability (Table 4.1). For instance, for a low tournament size, say 3, the mutation probability is varied as .05, .1, and .2 and the GA is executed and the results compared. One very important parameter for comparing the efficiency of the GA is the increase in the Pareto plans. This ratio is a significant indicator of the performance of the genetic algorithm.

Table 4.1 Repeated algorithm executions for varying values of Parameters

| Tournament size | Mutation Probability | Generation size | Ratio of Pareto plans Starting :Final |
|---|---|---|---|
| Low | Low 0.05 | 100 | 1:3 |
| | Medium 0.1 | 100 | 1:2.5 |
| | High 0.2 | 100 | 1:2 |
| Medium | Low 0.05 | 100 | 1:4.2 |
| | Medium 0.1 | 100 | 1:3 |
| | High 0.2 | 100 | 1:3 |
| High | Low 0.05 | 100 | 1:3 |
| | Medium 0.1 | 100 | 1:3.25 |
| | High 0.2 | 100 | 1:3 |

The above experiments with the varied set of values for mutation probability and tournament size corroborate the fact that the values suggested by pioneers in the domain are indeed correct (Deb, 2001). It can be seen from the above table that the number of pareto plans increase four-fold when a medium tournament size and low mutation value is employed. Hence, the results are discussed with a medium tournament size of 5 and a low mutation value of 0.05 as this is more similar to the natural evolutionary process.

### 4.4 Results of Multiobjective Optimization

The genetic algorithm was executed for 100 generations maximizing three objectives namely

- Per capita Green space (PCGS)
- Space for public service (PCPS)
- Housing capacity (NumHU)

The execution of the GA for a single future plan for the study area required about 10 seconds on a LG desktop computer with a Pentium 4 CPU (3.20 GHz). The execution of the GA for 100 generations required approximately 300 seconds on an average, which is a considerable improvement over previous genetic algorithms dealing with landuse problems. From a very high proportion of infeasible solutions in the first random generation, the algorithm seems to converge towards a feasible set towards the last generations. On the whole, the average time consumed for the experiments that were performed on a 3.2 GHz Pentium-IV machines with 512 MB RAM was 800 seconds for one execution of 100 generations.

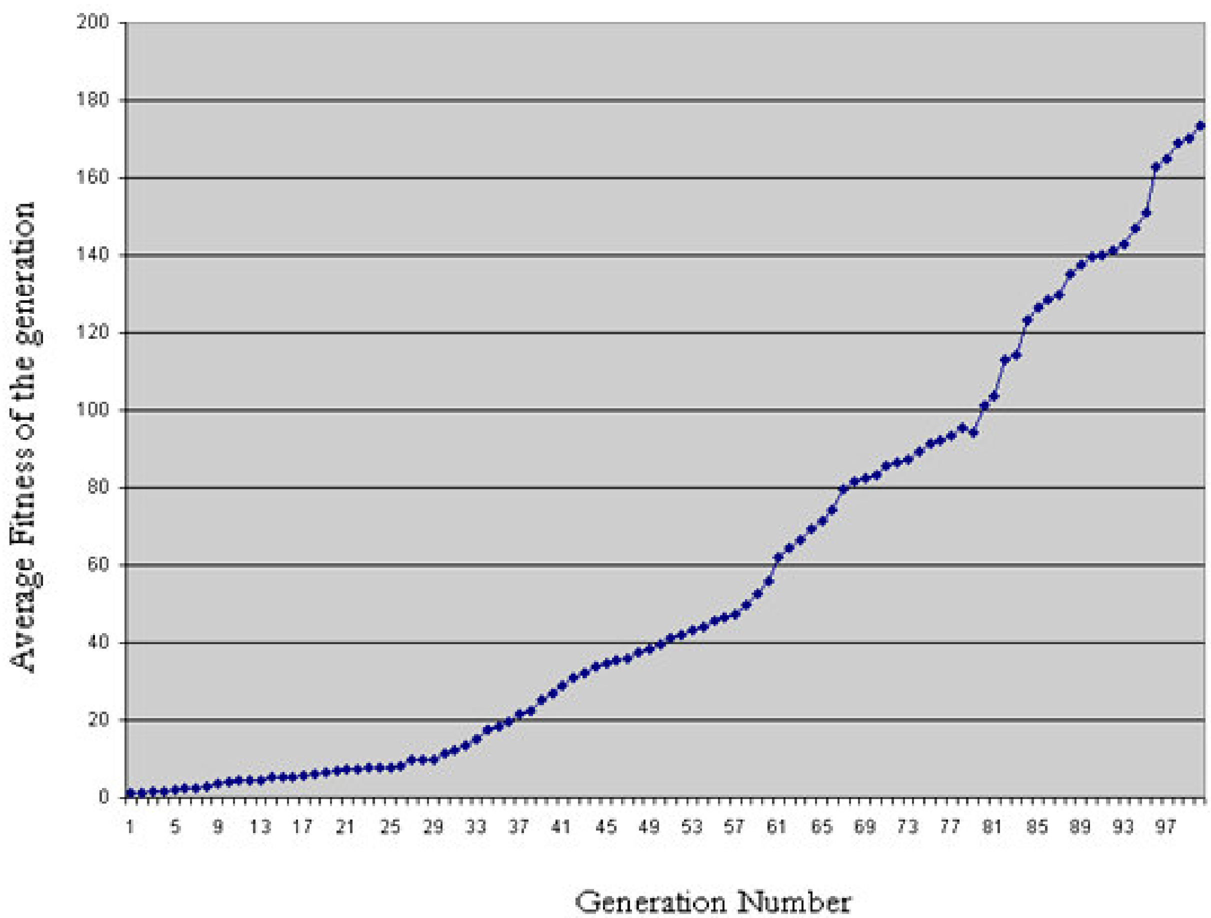



Figure 4.3 Improvement of the average fitness over the generations

The above chart (Figure 4.3) shows the average fitness of the generations. It is evident that the GA has significantly improved the fitness of the plans. While the fitness of the initial generation was 1.23, the fitness of the final generation has improved to 173, which is more than 100 times the overall fitness of the initial generation. This implies that the final plans, whilst satisfying the constraints, have maximized the per capita green space, per capita space for public service, and the housing capacity.

### 4.5 Performance metrics of the GA

From the following table that provides the generational minimum and average values for the three objectives the performance of the GA is obvious. When comparing these values for the starting and final generations (Table 4.2), it is obvious that the GA has undoubtedly improved all the three objectives.

Table 4.2 Table showing the increase in objective values after GA iterations

| Parameter | PCGS (Sq.m./ resident) | PCPS (Sq.m./ resident) | Housing Capacity (Number of Housing Units) |
|---|---|---|---|
| Average- Initial Gen. | 376.56 | 115.59 | $2.569 * 10^{7}$ |
| Minimum-Initial Gen. | 375.2 | 105.36 | $1.536 *10^{6}$ |
| Average- Final Gen. | 405.2 | 175.36 | $5.412 * 10^{8}$ |
| Minimum-Final Gen. | 400.12 | 169.36 | $2.356 * 10^{7}$ |

Also, another important point to be noted is that the Pareto set for the starting generation included a mere 10 of the 100 feasible plans, whereas the Pareto set for the final generation included 40 of the 100 feasible plans. This was determined after an average run of at least 10 times of the 100 generations. A maximum of 64 was achieved i.e. 64 Pareto-optimal sets were obtained in one generation.

Let us take an in-depth look at the Pareto-optimality criteria and investigate how the process of multi-objective optimization improved the Pareto-optimality on a generation-by-generation basis. As the GA fitness function compares the fitness values of the plans within one generation, and it is not possible or sensible to compare fitness values between plans in different generations. Therefore, new "global The "global fitness" for each of the 10,000 plans in the global generation according to the fitness equations described earlier in chapter 3.

As it follows from the previous discussion, the global generation also must have a "global Pareto set". From the total 10,000 plans in the global generation, there were 483 distinctive plans in the global Pareto set. The average value of the global fitness over the 100 plans in each generation is plotted and shown earlier and the following figure (Figure 4.4) shows their distribution.

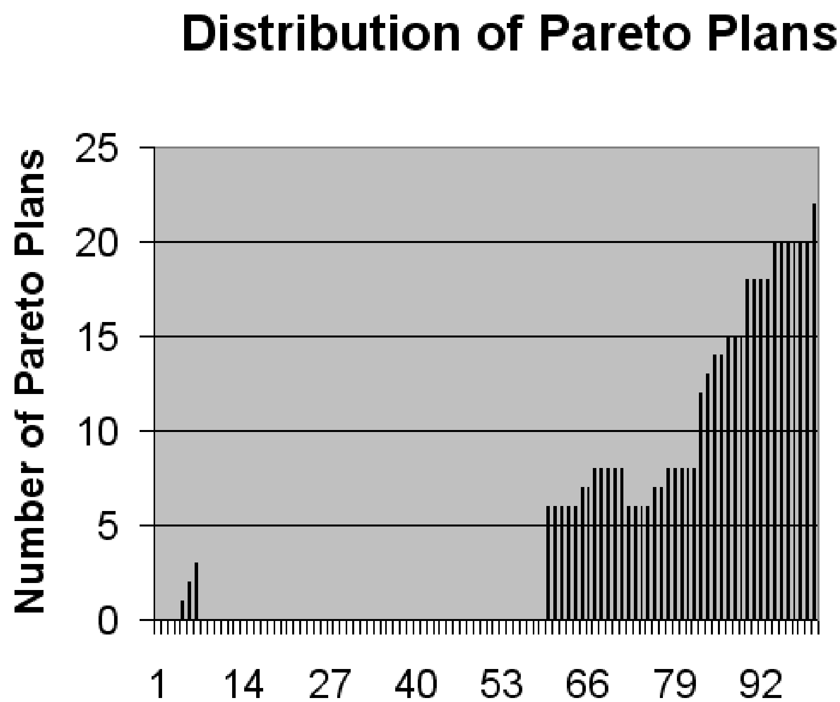


Figure 4.4 Distribution of Global Pareto Plans

The total number of plans in each generation that belong to the global Pareto set has also been plotted (Figure 4.4). This plot clearly shows that the genetic algorithm improved global Pareto optimality.

This clearly shows that the Pareto-optimality has been improving continuously after a particular generation number has been reached and after that particular number the improvement has been continual as well as consistent. The millions of infeasible plans that did not provide enough housing to accommodate the projected growth were aborted as already mentioned during the algorithm description process.

### 4.6 Results of Visualization

Two Pareto-plans with the highest fitness values were selected and visualization plans generated for these. In order to visualize landscapes, different kinds of information must exist and must be processed: Terrain data (e.g. digital elevation models) or land use data (eg. GIS data of the vegetation), or data about existing or planned changes in landscape (eg. CAD construction data, buildings, streets, bridges). Using texture mapping, surface attributes of three-dimensional objects, such as color and transparency can be manipulated in a variety of ways or customized according to specific applications. Highly customizable IDEs are available these days, using which it is possible to accomplish similar operations within a wide range of GIS applications.

Overall, the visualization procedure can be divided into three fundamental steps (Lim et al. 2002). In the first step, the 3D digital data of the terrain are obtained from the shape file from the ArcGIS. In the second step, a conversion program is used to convert the data on the terrain and vegetation in the landscape into VRML format. In the final step, modifications to alignments and other changes including transformation and orientation, if necessary, are made and the 3D VRML image of the landscape is generated on the computer.

A particular software belonging to the ArcGIS family of software, ArcScene is used to perform this conversion. The appropriate landuse shape file (.shp) is imported within the ArcScene environment (Figure 4.5) and it is exported using the 'Export' functionality into 3D (.wrl format). This can be then used in the VRML worlds along with the buildings and other infrastructure components to generate the 3D landscape scenario.

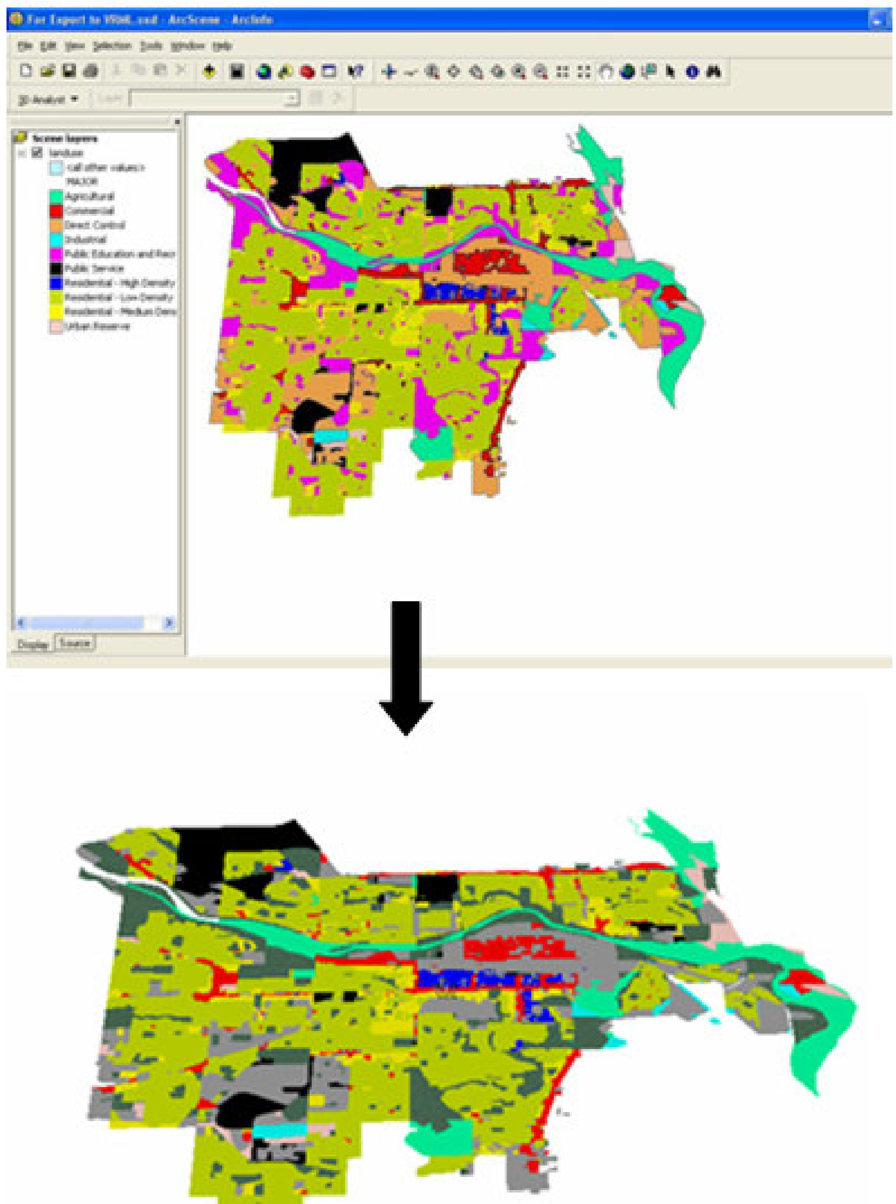
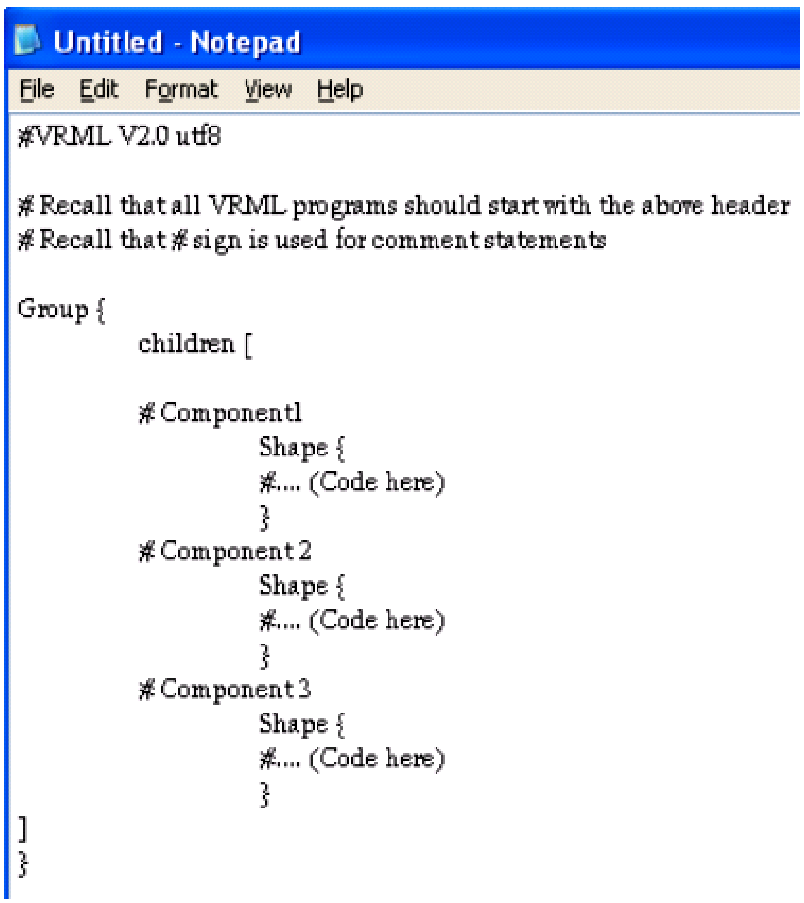


Figure 4.5 ArcScene Document: Conversion to VRML format

Using the object-oriented principles delineated in the previous chapter, the landscape scenarios are generated. Following shows a small code-snippet (Figure 4.6) showing the grouping of components under a group to create bigger objects. Similarly, various object parts are grouped to form objects, objects to form bigger objects, and all the objects together with the LU VRML file to form the complete 3D scene.




```
#VRML V2.0 utf8

# Recall that all VRML programs should start with the above header
# Recall that # sign is used for comment statements

Group {
        children [

        # Component1
                Shape {
                #.... (Code here)
                }
        # Component 2
                Shape {
                #.... (Code here)
                }
        # Component 3
                Shape {
                #.... (Code here)
                }
]
}
```

Figure 4.6  A sample VRML code showing scene components

The difference between a scene and object must be understood. A scene typically represents a larger picture such as a university campus or an urban landscape etc. Mathematically, a scene can be considered as the sum total of all the scene objects, where the scene itself is an object. Each object is a result of the function of its properties.

$$\text{Scene} = \sum_{i=1}^{n} \text{Scene Objects} \quad , \text{Objects} = \text{fn.(properties)}$$

Where, n is the number of objects making up the scene

The number of objects making up a scene may vary from scene to scene depending on the purpose of visualization and scene complexity. To create a ball, a single sphere object might be sufficient, while to generate a 3D building model, hundreds of objects with varying properties may be required. On the whole, when we are talking about a 3D scene, we are actually discussing about an object or a group of objects assembled together that function in unison to create a 3D representation of some real-world entity. Hence, it is important to understand some of the fundamental concepts of object oriented programming.

An object is the critical feature in the object-oriented programming paradigm. An object contains data and it can be acted upon and it can act accordingly. A wide range of things that we deal in our day to day life can be considered as Objects. A computer, a cycle, a bus, a washing machine, an electric stove, an iron box, a watch, a clock, a hair dryer, a television, can all be considered objects. This is because, they all contain something (data) and upon being initiated (or put to use) they perform specific tasks. In other words they have their own properties and they can 'act' or behave.  An object may be acted upon and it reacts in accordance with the action. An object may be made of several other objects and it can also be a part of another object.  If a PC (Personal Computer) is considered an object, then the monitor can be thought of as an object that is a component object of the PC object.

Let us consider a piece of code which, albeit small, has the basic elements required to create a 3D object in the virtual world. This small piece of code (Figure 4.7) provides an insight into the fundamental concepts behind building a 3D object in the virtual coordinate space. This is a VRML code.

```
#VRML V2.0 utf8

Shape {
        geometry Sphere {radius 1 }

        appearance Appearance {
            material Material {
                diffuseColor    1 0 0 }
         }}
```

Figure 4.7 A piece of VRML Code used to generate a Red colored sphere object

Using the above notions, the following objects (Figure 4.8) have been modeled within VRML 2.0 for generating the landuse scene for the Pareto plan.

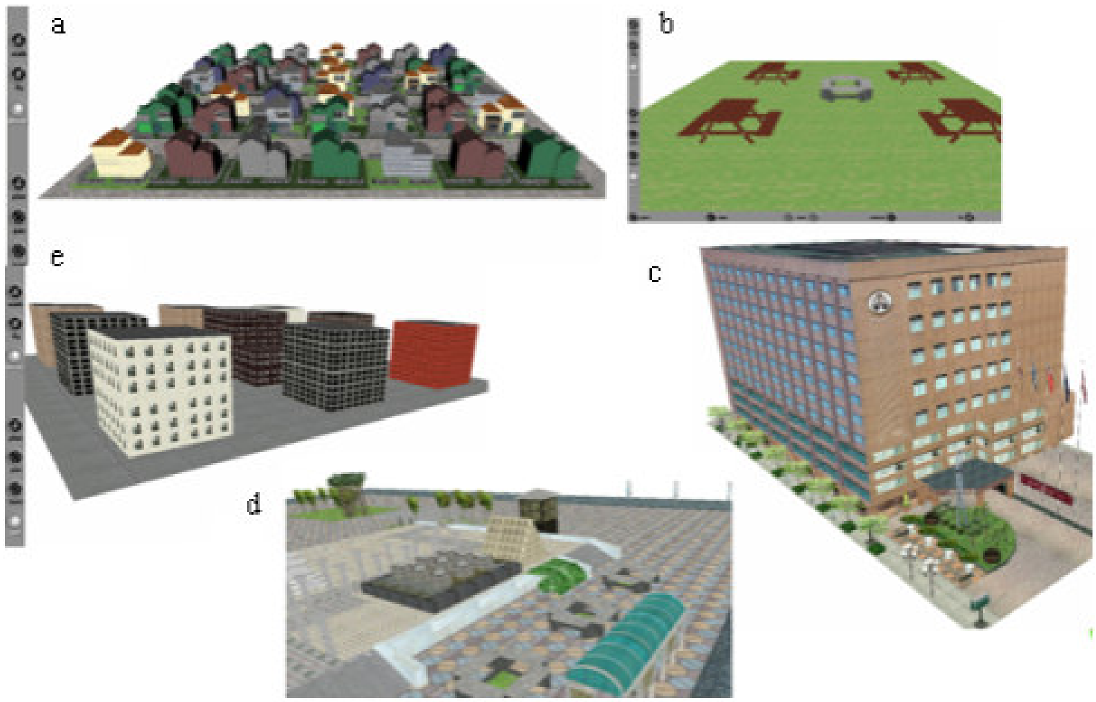



Figure 4.8 Individual components of the virtual world scene
Clockwise from top left – a. Residential – Low density housing b. Green space – park c. Commercial d. Community space e. Residential – Medium density housing

There are various elements of the scene that need to modeled if the scene is to be realistic and textures need to draped or mapped on to the objects thus modeled to incorporate photo-realism in the scene. The term photo-realism refers to the degree to which a rendered scene resembles the real-world. Some important elements that need to modeled are things that we come across in our daily life such as lamp-posts, plants, side-walks, telephone booths, roadside benches, sign boards, signals etc (Figure 4.9).

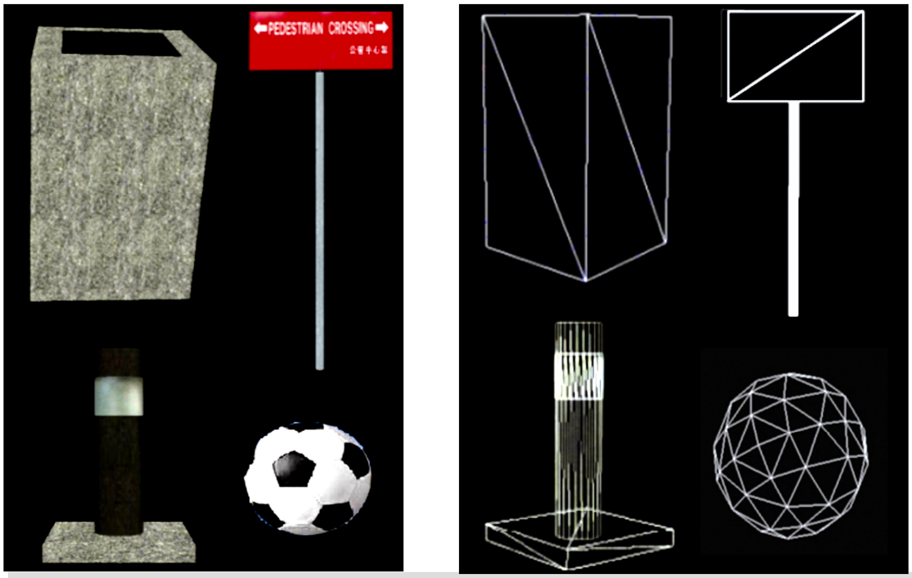



Figure 4.9a. Different kinds of objects in 3D format
b. Wire Frame Model

All the elements of a scene are built and are grouped together hierarchically and added to the scene as shown below (Figure 4.10).

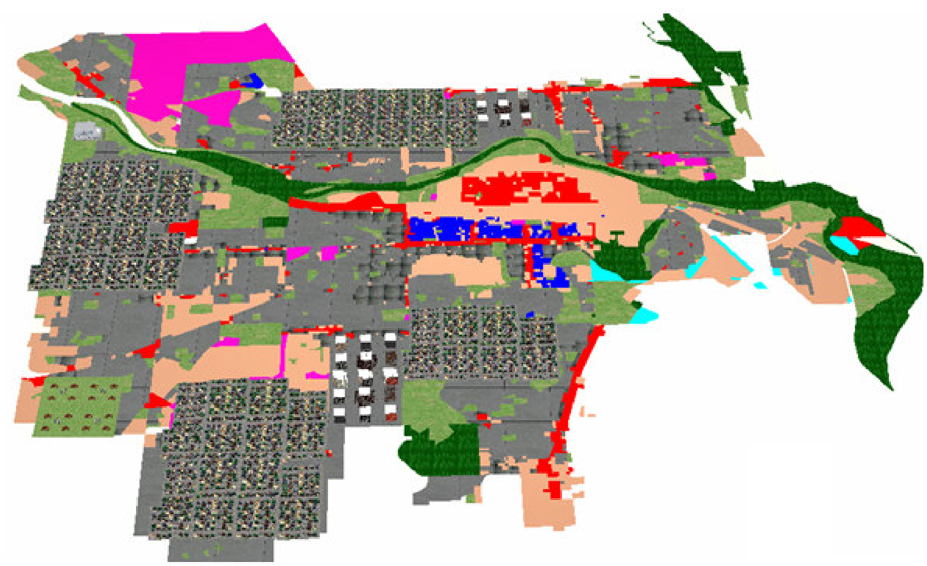


Figure 4.10 A screen-shot of the landscape scenario with components added

Shown below are the visualizations of two Pareto-optimal plans selected for visualization.

Figure 4.11 Visualization for Pareto-optimal plan no.1
Green spaces distributed properly around housing areas

Figure 4.12 Visualization for Pareto-optimal plan no.2
Green spaces not distributed properly

From the figures shown here (Figure 4.11 and Figure 4.12), the two plans can be compared in a very systematic manner. Evaluating them the point of view of the objectives considered in this study, in plan no.1 (Figure 4.11), it can be seen that the green spaces are properly distributed around the housing areas. The green spaces are evenly distributed so that the residents from the high-density residential area can access the green spaces without much commuting. On the other hand, in the second plan (Figure 4.12), it can be seen that there are three residential regions (two high-density and one low-density) competing for meager green space which is also not evenly distributed.

It should be noted that both these are Pareto-optimal plans with very fitness values. They satisfy the constraints that the green spaces must be more than the threshold values. However, it is not necessary, that the green spaces must be distributed evenly all over the region. Hence, we can see that there are a lot of practical constraints, all of which may not be included in the GA, but can be observed using visualization plans.

Consequently, plan no.1 can be selected for implementation. Thus, the usefulness of visualization in evaluating CPOPs (Competing Pareto-optimal plans) is evident. Similarly, when decision makers need to evaluate several Pareto plans to select the most appropriate one for implementation, they can use 3D visualizations corresponding to the Pareto-optimal plans to evaluate the plans from various perspectives and select the one that is most suitable for the problem on hand.

### 4.7 Discussion

The use of visualization in this study is triggered by the increasing need for tools or indicators, which can efficiently depict landuse scenes as one comprehensive screenshot rather than a series of non-coherent data layers. It is essential to capture the links between the various dimensions of a landuse scenario. Whilst GIS provide the analytical tools and methodologies for spatial integration of the different scientific areas and sub-models, unless the data is transformed into the 3D format it is not of significant use to planners and decision makers as interpreting voluminous statistical data is a mammoth and cumbersome task. The Visualization aids to understand the overall composition of the landscape and understand its functioning holistically (Dramstad et al., 1996; Turner et al., 2001; Nagendra et al., 2004). One prominent advantage of using visualization models, is that even a bird's eye view can provide enormous details to the observant eye. For instance, planners who are experts in the field of landuse can find out desirable or undesirable patterns using visual scene renderings.

One very obvious advantage of using virtual worlds for visualizing the competing Pareto-optimal plans (CPOP) is that patterns can be easily found. Using varying LODs and by studying the same scene from multiple viewpoints, numerous aspects that might not be otherwise be obvious can be found. Subjective features such as scene quality can be studied in a more reliable manner.

In the subsequent paragraphs, some specific uses of visualization from the perspective of landuse modeling and for evaluating the various plans are discussed. From the following figure (Figure 4.13), the use of visualization to study the same scenario from various levels of detail is obvious. The same scene can be built with varying levels of detail. For instance, when viewing from a distance, the finer details are not obvious. This notion can be used to efficiently model the scene. Based on the viewer's position in a scene, the objects can be rendered accordingly.

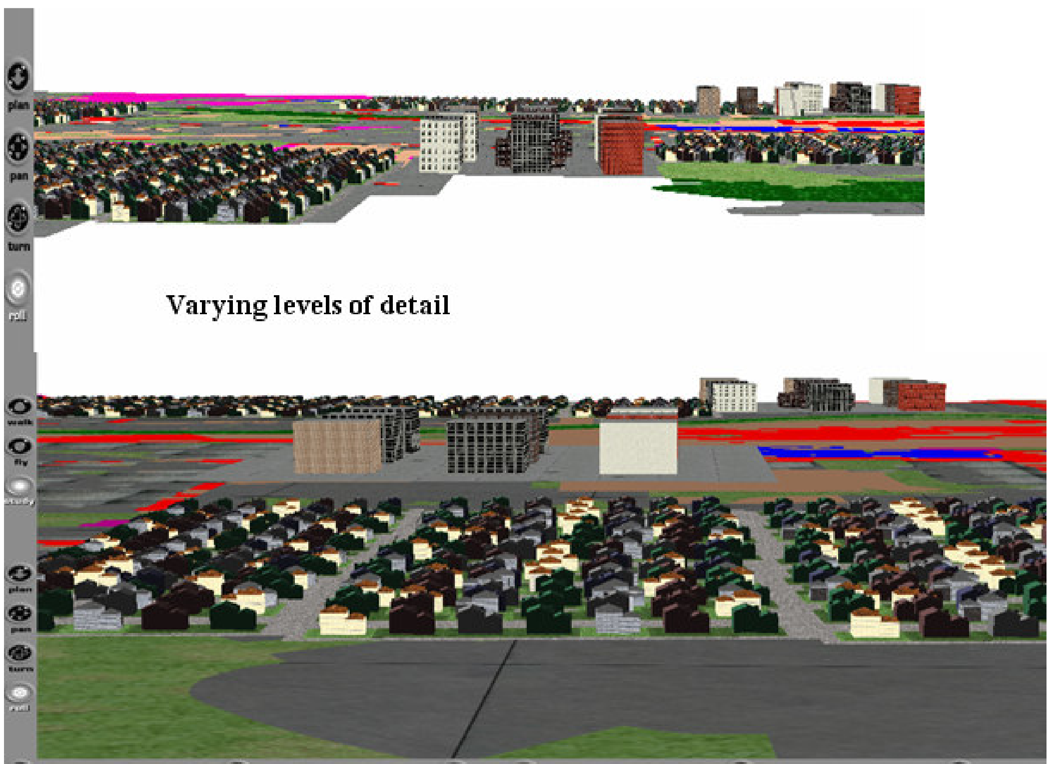



Figure 4.13 Using 3D Visualization to study the same scene from varying LOD

When a viewer is at quite a distance from buildings, the finer details can be hidden and only the coarser details that need to efficiently project the object can be shown. This will greatly reduce the number of faces that need to be rendered by the VRML browser and hence will significantly enhance performance. Another prominent advantage is that the same scenario can be seen from multiple viewpoints.

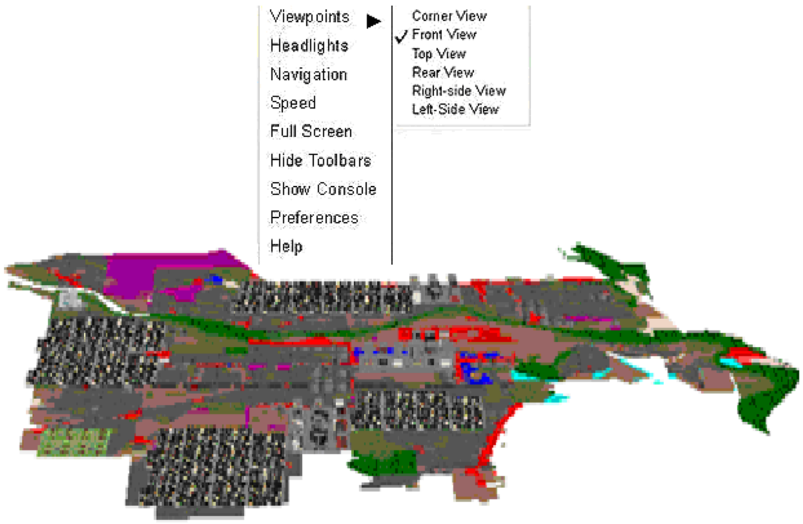



Figure 4.14 Viewpoints within a Pareto-optimal plan visualization

For the scene shown above (Figure 4.14), two viewpoints (Figure 4.15) are shown below

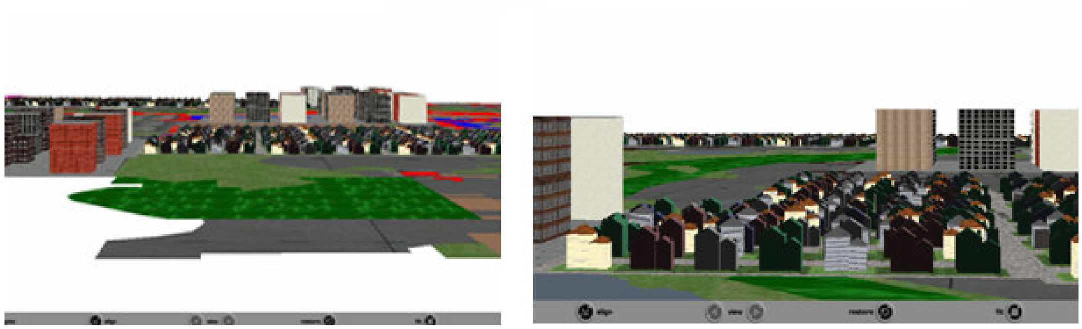


Figure 4.15 Viewpoints within a Pareto-optimal plan visualization

The various advantages are summarized as follows:

- Desirable or undesirable patterns or developments, can be easily observed by the planner. The success of the visual display is in the fact that this is quite obvious to the planner which may not be that obvious if only the Pareto plan was evaluated after the MOO process only.
- The per capita green space per population might be more than the required value, however, the distribution of the green spaces within the study area is an important factor. A resident might not be in a condition to travel several kilometers to reach a green space. In order to ensure that the purpose of green spaces is satisfied, they must not be in the required quantity, but must also be well distributed.
- Similarly, the distribution of various other landuse types can be easily studied using varying levels of detail (LOD) within the virtual worlds
- Access to public service facilities from various housing types can be studied and understood
- Plans can be compared and contrasted in a more objective manner as a visual tool provides a more reliable benchmark for evaluation rather than relying on abstract measures
- Aesthetic view quality or scene quality is another element of significant importance in urban design these days. Visualization can greatly facilitate studying the aesthetic quality of a plan.
- The virtual worlds can be navigated and it can found as to whether the placement or location of various buildings and infrastructure elements is

appropriate. For instance, an office building blocking the view of a monumental structure of some other feature of prominence is undesirable and hence such a landuse cannot be allocated for high-density residential or high-rise buildings etc.

### 4.8 Conclusion

This chapter provided the results and also a detailed interpretation of the results from the perspective of this study. In this study, a system for optimizing multiple objectives of a landuse design and selecting one plan from among the various CPOPs is proposed. Whist the multiobjective optimization part optimizes multiple competing objectives and provides a set of CPOPs, the visual evaluation tool, namely scene visualizations aid the process of selecting the appropriate plan from among the CPOPs. Landscape visualizations facilitate walk-through functionalities and simulate the future landscape photo-realistically. Thus, we see that when studying geospatial data that is inherently 3D in nature, combining GIS with visualization facilitates generating realistic 3D landscapes and understanding the 3D topologies. Also, with PPGIS (Public Participation GIS) becoming the norm of the day, such visualizations can be hosted online and the feedback of the general public, who are the ultimate consumers, can be obtained. Such visual representations are excellent tools to overcome the barriers of scale and those imposed by viewpoints. In other words, the same scene can be studied at various viewpoints which may not be accessible from the real-world

## CHAPTER 5: Conclusion and Recommendations for Future Study

### 5.1 Conclusion

The goal of this study was to design an innovative framework integrating visualization with MOO to generate sustainable landuse designs. This study used an area from central Calgary as the study-area and employed three objectives and four constraints. The study tested the GA-framework integrated with Visualization on this study area and found that the results were extremely satisfactory both from the perspective of the performance of the GA and the performance of the visualization tool.

With urban environments becoming increasing complex by the day, the process of land use planning and future allocation is getting increasingly multifaceted. As landuse planning engages larger landscape scales more complex design/planning conditions need to be met. The effective resolution of such multifaceted problems will require the synchronization among various disciplines and exploiting the advances in various related disciplines.

Over the past several decades, information presentation has inspired the development of several new tools and techniques. The information revolution has resulted in vast amounts of data volumes that are far too complex, both in quality and quantity, to be handled by conventional tools and techniques. Recent technological advances in the realm of remote sensing have dramatically increased the amount of geospatial data available.

Producing 3D visualizations is no longer considered a cumbersome task. Using judicious methods and the latest developments, 3D scenes can be rendered in a time and cost efficient manner. The overall costs involved in producing very high quality and photo-realistic visualizations have gone down considerably and visualization is increasingly becoming a geospatial domain. Various fields such as Global Positioning Systems (GPS), Location Based Services (LBS), Geodesy, etc. are beginning to embrace visualization. VRML possesses excellent capabilities for visualization and interaction when extended using a scripting language such as JavaScript or Java. While JavaScript can be included using the url field within a VRML file itself, a Java class file can be used to externally manipulate the animation within a virtual world.

The experiments performed using genetic algorithms showed that the overall fitness of the plans from the point of view of the objectives has increased significantly. From the increase in overall fitness of the generations and from the four-fold increase in the quantity of Pareto-optimal sets, it can safely be concluded that genetic algorithms can indeed serve as useful tools in developing good solutions to an allocation problem with multiple and contradicting objective functions.

However, for long, GAs has only provided a solution-set and not a unique solution. In order to overcome this impediment this study proposed the integration of optimization and geospatial visualization. The virtual worlds served as excellent tools for evaluating the competing Pareto-optimal solutions and in selecting the most appropriate one. Stakeholders and decision makers are no longer compelled to rely on subjective measures to evaluate the CPOPs. Instead, 3D scene rendering can be employed for the evaluation purpose. The CPOPs can be explicitly studied and the various aspects, both positive and negative, of the plans can be clearly outlined. This way, the approach is transparent to all stakeholders since there can be no subjective assumptions and hence no false judgments. Thus, the possibility of a good or more appropriate CPOP being not selected or an inappropriate CPOP being selected is greatly reduced.

The important results are summarized as follows:

- A comprehensive GA-based multiobjective optimization framework was designed to maximize three objectives under four given constraints. From the experimental results it is evident that the GA is robust and functions effectively to optimize given objectives. The GA is efficient from the programming perspective and is time-saving as well.
- The MOO method significantly enhances the fitness of the plans and results in enhanced Pareto-optimality.
- A novel approach, integration of visualization with MOO was proposed in this research to evaluate the Pareto plans resulting from the GA optimization process.
- The Visualization framework is based on virtual reality and 3D environments corresponding to Pareto plans with high fitness values were generated using Virtual Reality Modeling Language (VRML 2.0)
- The results demonstrate that visualization serves as a valuable tool in evaluating the CPOP (Competing Pareto-optimal Plans) and selecting the plan that is most suitable for the problem on hand.

In particular, the advantage of the virtual environments can be summarized as follows:

- Multiple scenarios can be evaluated
- Infinite viewpoints can be generated
- A guided tour of buildings can be generated by linking using scripts
- Decision-makers can view the finished product before hand
- The virtual models are extremely time-saving and economical
- Visibility of various elements in the scene can be tested
- Aesthetic view quality can be evaluated and modified if needed
- All plans/modifications can be done at the convenience of desktop
- Facilitates proactive interference as future scenarios can be built
- Facilitates non-destructive testing as product is available beforehand

### 5.2 Limitations

This study involves two major components: designing the GA for MOO and designing a visualization tool to evaluate the results of the solutions from the Pareto set. This section discusses limitations with respect to both the components.

The overall design of the aforementioned is a technically complex and time-consuming process. This is a prototype study whose primary objective is to combine multiobjective optimization with visual evaluation techniques. Hence, the emphasis here is on the multiobjective optimization and the subsequent visualization and not on the various data-related and other peripheral issues.

Hence, owing to the scope of this research and inherent time limitations, the following issues are not included in this research:

- Not all the zones, but only the required number of zones from the city of Calgary are considered for this study
- Only three objectives are considered for the multiobjective optimization process. Other objectives that are equally important to those considered here surely do exist. However, being a prototype study only three objectives have been considered with the view of sustainable development
- Visualization models are generated only for a select subset from those among the pareto set

One noteworthy issue is that is here in visualizing real-world scenarios, there is an inevitable trade-off that many browsers are forced to make amid performance and resolution. Considerable complexity is involved in the generation the VRML IndexedFaceSets from typical TIN models, necessitating transformation prior to placing in the VRML world. Talking of data quantity, several factors need to be considered during visualization such as the type and volume of data to be visualized, memory constraints, and system performance. Scenes with a greater number of polygons decelerate the system and adversely affect the system performance. Creating a photorealistic environment might make the interactivity poor, as the system performance goes down. A carefully structured VRML file can enhance the rendering performance of the browser for displaying a terrain model. However, a carefully structure VRML file does not necessarily mean controlling and manipulating the LOD nodes only. But, the VRML scene author must take into consideration the viewpoint and navigation type into consideration so that unnecessary facets are not drawn. For instance, once a building has been generated and is viewed in the scene, the floor of the building is unnecessary.

### 5.3 Recommendations for future study

Further experimentation and studies investigating the integration of spatial visualization with multiobjective optimization can serve as an efficient tool to solve various other geospatial problems. Particularly, such tools are of immense potential when multiple objectives and multiple stakeholders are involved. The integration of visualization with optimization will greatly facilitate decision makers in developing sustainable solutions for land use allocation. There will be significant improvements in geospatial analysis and the overall knowledge of landuse can be greatly enhanced.

Another important technical aspect that can be further studied is the automation of the process of generating landuse scenarios from the CPOPs. This can be achieved by a combination of the PROTO functionality in VRML with an object oriented language such as Java. Further research in this direction must be aimed at creating a standard VRML animation library using the PROTO and EXTERNPROTO functionalities. This will facilitate creating simple and real-looking animation on the fly and such a library can also be equipped with standard building types so that when generating huge scenes considerable time can be saved.

## References


Aerts J.C.J.H., Herwijhen M. van , Janssen R., and T.J. Stewart (2005), Evaluating spatial design techniques for solving land-use allocation problems, Journal of Environmental Planning and Management 48 (2005) (1), pp. 121–142.

Arika Ligmann-Zielinska, Richard Church, and Piotr Jankowski (2005) Sustainable Urban Land Use Allocation with Spatial Optimization, Geocomputation, 2005

Arthur, J. L. (1997) Clarification on the use of linear programming and GIS for land-use modelling, International Journal of Geographical Information Science 11(4).

Balling R.J., Taber J.T., Brown M.R and Day K. (1999) Multiobjective urban planning using genetic algorithm, Journal or Urban Planning and Development 125 , 1999(2), pp. 86–99

Balling, R., Powell, B. & Saito, M. (2004) Generating future land-use and transportation plans for high-growth cities using a genetic algorithm, Computer-Aided Civil and Infrastructure Engineering, **19**(3), 213–222.

Beasley et al., "Genetic Algorithm for the Set Covering Problem," European Journal of Operational Research, vol. 94, No. 2, Oct. 25, 1996, pp. 392-404.

Berry, J.K., Buckley, D.J. and Ulbricht, C. 1998. "Visualize Realistic Landscapes." GIS WORLD, Vol. 11, No. 8, 42-47.

Chandramouli M, Chung L.K., Chou T.Y., Kuo C.Y., Design and Implementation of Virtual Environments for Visualizing 3D Geospatial Data, TGIS Conference, Oct 28-29 2004 http://gisclass.geo.ncu.edu.tw/93gisapp/GIS2004/PAPER/ALL.htm

Chandramouli M**,** Huang B. (2006) Virtual Environments for Geospatial Applications. Applications, Methods and Issues: Encyclopedia of Geoinformatics, (In Print) http://gis.sis.pitt.edu/encygeo/

Chen Xiaoyong and Shunji Murai 1999, Analysis and Visualization of 3D Geospatial Data by Using Delaunay Tetrahedral Tessellation , Proceedings ACRS, 1999

Church, R.L., Murray, A.T., Figueroa, M.A., Ager, A.A., McGaughey, R.J. and Merzenich, J. (1994) Artificial Landscape Visualization of Ecosystem Management Plans. Submitted to AI Applications, June 1994

Cohon, J. L. (1978). Multiobjective Programming and Planning. New York: Academic Press.

Cunha, A. G., P. Oliviera, and J. Covas (1997). Use of genetic algorithms in multicriteria optimization to solve industrial problems. In T. Back (Ed.), Proceedings of the Seventh International Conference on Genetic Algorithms, San Francisco, California, pp. 682–688. Morgan Kaufmann.

Darwin Charles (1859), The Origin of Species, 1859

Deb, 2001 K. Deb, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, Chichester (2001).

Dias A. H. F. and Vasconcelos J. A. (2002) Multiobjective Genetic Algorithm applied to solve Optimization Problems, IEEE Trans. Magnetics, v. 38, pp. 1133-1136, March 2002. http://citeseer.ist.psu.edu/dias02multiobjective.html

Doyle, S., Dodge, M., & Smith, A., (1998): The potential of web-based mapping and virtual reality technologies for modeling urban environments. Computers, Environment and Urban Systems, 22: 137 – 155.

Eiben A. E., Paul-Erik Raué, and Zsófia Ruttkay (1994a) Solving constraint satisfaction problems using genetic algorithms. International Conference on Evolutionary Computation 1994: 542-547

Eiben A. E., Paul-Erik Raué, Zsófia Ruttkay (1994b) Genetic algorithms with multi-parent recombination. PPSN 1994: 78-87

Eshelman, L.J., R.A. Caruana and J.D. Schaffer (1989) Biases in the crossover landscape, in Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann, George Mason University, 4–7 June 1989, pp. 10–19.

Fairbairn, D., Parsley, S., 1997 The use of VRML for Cartographic representation. Computer and Geosciences 23(4), 475-481.

Fogel, D. B. (1995). Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Piscataway, NJ: IEEE Press.

Fonseca, C. M. and P. J. Fleming (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In S. Forrest (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, pp. 416–423. Morgan Kaufmann.

Fonseca, C. M. and P. J. Fleming (1995a). Multiobjective genetic algorithms made easy: Selection, sharing and mating restrictions. In First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA 95), London, UK, pp. 45–52. The Institution of Electrical Engineers.

Fonseca, C. M. and P. J. Fleming (1995b). An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3(1), 1–16.

Fonseca, C. M. and P. J. Fleming (1996). On the performance assessment and comparison of stochastic multiobjective optimizers. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel (Eds.), Fourth International Conference on Parallel Problem Solving

Fonseca, C. M. & Fleming, P. J. (1998). Multi-objective optimization and multiple constraint handling with evolutionary algorithms -part 1: A united formulation. IEEE Transactions on systems Man and Cybernetics-Part A: Systems and Humans 28(1): 26{37.

Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 141–153. Sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

Goldberg, D. E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, Massachusetts: Addison-Wesley.

Gordon, D. Editor, (1990) Green Cities: Ecologically Sound Approaches to Urban Space, Black Rose, Montreal.

Hajela, P. and C.-Y. Lin (1992). Genetic search strategies in multicriterion optimal design. Structural Optimization 4, 99–107.

Holland, J. (1975). Adaptation in natural and artificial systems, second edn, The University of Michigan Press, Ann Arbor. Second edition, 1992, MIT Press.

Horn, N. Nafpliotis and D.E. Goldberg (1994) A niched Pareto genetic algorithm for multiobjective optimization, Proceedings of the first IEEE conference on evolutionary computation, IEEE world congress on computational intelligence Vol. 1, IEEE Service Center, Piscataway, NJ (1994), pp. 82–87.

Huang, B. and Lin, H., 2002. A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet. Computers & Geosciences, 28(1): 13-19

Jaszkiewicz A. (1998) Genetic local search for multiple objective combinatorial optimizations. Technical Report RA-014/98, Institute of Computing Science, Poznan University of Technology, 1998.

Koppers Lothar (1998) 3D-Map - Virtual Reality and Geo-Data http://www.sbg.ac.at/geo/eogeo/authors/koppers/koppers.html

Koski, J. (1984).Multicriterion optimization in structural design. In E. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz (Eds.), New Directions in Optimum Structural Design, pp. 483–503. Wiley.

Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In H.-P. Schwefel and R. Manner (Eds.), Parallel Problem Solving from Nature — Proc. 1st Workshop PPSN, Berlin, pp. 193–197. Springer.

Lis, J. and A. E. Eiben (1997). A multi-sexual genetic algorithm for multiobjective optimization. In Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), Piscataway, NJ, pp. 59– 64. IEEE.
Matthews, K.B., Craw, S., MacKenzie, I., Sibbald, A.R. and Elder, S. (1999a) Applying Genetic Algorithms to Land Use Planning, in G. Petley et al. Proceedings of the 19th Workshop of the UK Planing and Scheduling Special Interest Group, University of Salford, Manchester, pp109-115.

Matthews, K.B., Sibbald, A.R. and Craw, S. (1999b) Implementation of a spatial decision support system for rural land use planning: integrating GIS and environmental models with search and optimization algorithms, Computers and Electronics in Agriculture, 23:9-26.

Matthews, K.B., Craw, S., Elder, S., Sibbald, A.R. and MacKenzie, I. (2000a) Applying Genetic Algorithms to Multiobjective Land Use Planning, in D. Whitley et al.(eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2000), Morgan Kaufmann, San Francisco, pp613-620.

Matthews, K.B., Craw, S., Elder, S. and Sibbald, A.R. (2000b) Evaluating Multi- Objective Land Use Planning Tools Using Soft Systems Methods, in M. Garagnani (ed.) Proceedings of the 19th Workshop of the UK Planing and Scheduling Special Interest Group, Open University, Milton Keynes, pp109-120.

McGaughey, R.J. 1998. Techniques for Visualizing the Appearance of Forestry Operations. J. Forestry, Vol. 96, No. 6, 9-14.

Michalewicz Z. (1992) Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York.

Obayashi, S., Takahashi S., and Takeguchi, Y. (1998) Niching and elitist models for MOGAs. In A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel (Eds.), Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V), Berlin, Germany, pp. 260–269. Springer.

Osyczka, A. (1985) Multicriteria optimization for engineering design. In Design Optimization, (Gero, J.S., Editor). Academic Press, New York, 193–227

Pareto, V. (1896) Cours D'Economie Politique, Volume 1. Lausanne: F.Rouge

Pham, D., and Karaboga, D. (2000): Intelligent Optimization Techniques, London: Springer.

Rudolph, G. (1998) On a multi-objective evolutionary algorithm and its convergence to the pareto set. In IEEE International Conference on Evolutionary Computation (ICEC'98), Piscataway, NJ, pp. 511–516. IEEE.

Schaffer, J. D. (1984) Some experiments in machine learning using vector evaluated geneticalgorithms, Doctoral dissertation, Vanderbilt University: Nashville, TN.

Schaffer, J. D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette (Ed.), Proceedings of an International Conference on Genetic Algorithms and Their Applications, Pittsburgh, PA, pp. 93–100. sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

Seixas, J., Nunes, J.P., Louren_co, P., Lobo, F., & Condado, P. (2005) GeneticLand: Modeling land use change using evolutionary algorithms, In Proceedings of the 45th Congress of the European Regional Science Association, Land Use and Water Management in a Sustainable Network Society, Vrije Universiteit Amsterdam, 2005, 23-27.

Shiode, N. (2001) 3D urban models: recent developments in the digital modeling of urban environments in three-dimensions, GeoJournal, 52(3), 263-269.

Srinivas, N. and K. Deb (1994) Multiobjective optimization using non-dominated sorting in genetic algorithms. Evolutionary Computation 2(3), 221–248.

Steuer, R. E. (1986) Multiple Criteria Optimization: Theory, Computation, and Application. New York: Wiley

Stewart T.J., Janssen R. and Herwijnen M. van (2004) A genetic algorithm approach to multiobjective land use planning, Computers & Operations Research 31 (2004), pp. 2293–2313.

Sutherland, I.E. (1965) The ultimate display. In the proceedings of the IFIPS Congress, New York City, NY, May 1965, Vol. 2. , pp. 506 – 508.

Tufte, E. R. (1990) Envisioning information. Cheshire, Connecticut: Graphics Press.

Tufte, E.R. (1992) The Visual Display of Quantitative Information. Graphics Press.

Valenzuela-Rendon, M. and Uresti-Charre, E. (1997) A non-generational genetic algorithm for multiobjective optimization, in Thomas Back et al. (eds), Proceedings of the Seventh International Conference on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA., pp. 658-665.