ABSTRACT

GILL, APNINDER. Reconstruction of Optical Fiber Bragg Grating Sensor Strain Distributions Using a Genetic Algorithm. (Under the direction of Assistant Professor Kara Peters).

Optical fiber Bragg gratings are unique among embedded strain sensors due to their potential to measure strain distributions with a spatial resolution of a few nanometers over gage lengths of a few centimeters. This thesis presents a genetic algorithm for the interrogation of optical fiber Bragg grating strain sensors. The method calculates the period distribution along the Bragg grating which can then be directly related to the axial strain distribution. The period distribution is determined from the output intensity spectrum of the grating via a T-matrix approach. The genetic algorithm inversion method presented requires only intensity information and reconstructs non-linear and discontinuous distributions well, including regions with significant gradients. The method is demonstrated through example reconstructions of Bragg grating sensor simulated data. The development of this algorithm will permit the use of Bragg grating sensors for damage identification in regions close to localized damages where strong strain non-linearities occur. A second application of the genetic algorithm search procedure is also discussed, the optimization of sensor locations for a particular structural application. An implementation of the genetic algorithm to a discrete sensor location problem is presented and its performance evaluated. A combination of the developed GA with available damage identification systems can be effectively used to find the optimal sensor distribution for damage detection.

Reconstruction of Optical Fiber Bragg Grating Sensor Strain Distributions Using a Genetic Algorithm

by

Apninder Singh Gill

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial satisfaction of the
requirements for the Degree of
Master of Science

Department of Mechanical and Aerospace Engineering

Raleigh

2003

Approved By:

Dr. Eric C. Klang	
Dr. Kara Peters Chair of Advisory Committee	Dr. Jeffrey W. Eischen

To my aunt, Late Mrs. Talwinder Kaur Sandhu for her unconditional love, support and blessings.

Biography

The author was born in Rampura, India on May 3, 1978. He belongs to Muktsar in beautiful state of Punjab in India. He completed his elementary and higher secondary education at Patiala, India, graduating in 1995. He receieved his Bachelor of Engineering degree in Mechanical Engineering from Punjab Technical University, Jalandhar in June 2000.

In the spring of 2002, he joined North Carolina State University, Raleigh to pursue his Master of Science degree in Mechanical Engineering.

Acknowledgements

The author wishes to express his gratitude and sincere appreciation to his advisor Dr. Kara Peters for her invaluable guidance and encouragement in making this study meaningful. Special thanks are extended to Dr. Jeffrey W. Eischen and Dr. Eric. C. Klang, members of advisory committee for their guidance in academics during this study.

The author graciously recognizes the financial support of Department of Mechanical and Aerospace Engineering to conduct his graduate research. Special appreciation to the National Science Foundation for their financial support through grant # CMS 0219690. Special thanks to Dr. Richard Gould for his help during this study. The author is thankful for the constructive suggestions provided by Mohanraj Prabhugoud and Dr. Michel Studer during the course of his research.

Finally the author would like to take this opportunity to express his profound gratitude to his parents and family members for their constant motivation, support and faith that made this study possible. The author is also thankful to all his friends in Raleigh and Patiala for always believing in him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Optical fiber Bragg grating sensor systems have potential for low cost, robust and reliable structural monitoring in the harshest of conditions, where conventional sensors can not operate. They provide improved accuracy and senstivity over existing systems and are immune to electromagnetic and radio frequency interference. Optical fiber Bragg gratings are also unique among embedded sensors due to their potential to measure strain distributions with a spatial resolution of a few nanometers over a gage length of a few centimeters. As a fiber optic analogue to conventional electronic sensors, they can be used as strain-guage sensors in buildings, bridges, airframes or other critical structures. In fiber Bragg grating sensors, the strain monitoring is done by monitoring the wavelength shift of the reflected signal, as a function of a measurand (strain, temperature, pressure etc.).

The goal of this thesis is to reconstruct the applied strain distributions from the knowledge of the output reflected spectrum of the Bragg grating sensor. The genetic algorithm (GA) is used to solve this inverse problem by interrogating the spectra from the Bragg grating sensor. The Transfer matrix method is used to simulate the response of Bragg gratings subjected to strain fields.

Chapter 2 of this thesis provides an introduction to genetic algorithms. Chapter 3 describes an inverse problem regarding the Bragg grating sensors and the previous work that has been done regarding the solution to this problem. Chapter 4 presents analysis approach and the implementation of the GA procedure for the problem of reconstructing the applied strains from the reflected signal of a fiber Bragg grating. The efficiency of the algorithm is demonstrated through the reconstructions of sensor simulated data. In the later part of thesis, the implementation of the GA for the optimization of sensor distributions, given the existence of a diagnosic system, is discussed in chapter 5. Finally, conclusions are presented in chapter 6.

Chapter 2

Genetic Algorithms

2.1 Introduction

Genetic algorithms (GAs) have gained tremendous interest in recent years. GAs were first invented by John Holland in 1960 and were later developed by Holland and his students and colleagues at University of Michigan in the 1960s and late 1970s [1]. GAs are general purpose optimization and search methods based on the principles of natural selection and natural genetics. The most remarkable features of GAs are their ability to search discontinuous, multimodal, non-convex, noisy search spaces which form the crux of many optimization and design problems. During the last few years, genetic algorithms have been employed sucessfully in areas such as biology, computer science, engineering, social sciences, machine learning and finance [2].

2.2 Biological Metaphor

The metaphor underlying the genetic algorithm is a biological one: that of natural selection and natural genetics. All living organisms consists of cells, while each cell contains a set of chromosomes. The chromosome is then composed of genes. The genes are strings of DNA encoding a complete piece of genetic information. Each gene is located at a particular locus (position) on the chromosome. A gene may assume several values or may have several forms. Each value of the gene is referred to as an allele of the gene. The complete genetic information (all chromosomes taken together) is called the genome of an organism. In natural systems the term genotype refers to a particular set of genes conatined in a genome.

In genetic algorithm searches the term 'chromosome'refers to a potential solution to the problem at hand and is encoded as a string comprised of features which can take the form of binary numbers, floating point, real numbers, alphabet characters, etc. An allele in the string therefore refers to the particular value of the feature for one of the potential solutions.

2.3 Mechanics of the GA

Genetic algorithms function by creating an initial population of chromosomes encoding potential solutions and then recombining them in a way to guide the search to the most promising areas of the search space. The simplest type of GA has three types of operators: selection, crossover and mutation [2]. The fitness of a particular chromosome determines its ability to survive and reproduce. Whether or not the chromosome is selected for crossover depends on its fitness value. The probabilistic selection operator ensures that the fittest individual has the highest chance to produce offspring. This selection operator is symbolic of the Darwanian concept 'survival of fittest'. The most important search operator is traditionally considered to be crossover. Crossover is an artificial version of biological recombination between two organisms to produce offspring. The mutation operator involves altering genes in a chromosome. Random mutation of newly generated offsprings induces variability in the population preventing premature loss of important features.

There is always an important concern in optimization problems that a chosen search technique has a balance between exploitation and exploration. If exploitation of obtained solutions is emphasized too much the algorithm may lead to premature convergence, thus heading the search towards unaccepatable solutions. On the other hand if exploration of search the space is over-emphasized, the search procedure may be reduced to nothing else than a random walk. The iterative application of operators in genetic algorithms implies a compromise between exploitation of the best available solutions and a robust exploration of the search space. Exploration and exploitation can also be viewed as the biological concepts of selection pressure and population diversity and are strongly interrelated as described by

Whitley [3].

2.4 Basic Elements of the GA

This section provides an overview of the elements involved in implementation of a genetic algorithms to solve a particular problem.

2.4.1 Representation & Initialization

The representation (encoding) of potential solutions is vital for the performance of a genetic algorithm. Each candidate solution is encoded as an individual (chromosome) of a population and represents (encodes) a point in the search (or representation) space of a particular problem. Holland's original conception of a GA involved encoding candidate solutions as binary strings [1]. In the present work real number encoding will be used for the representation of solutions for the problem of reconstructing Bragg grating sensor strain distributions. Experiments relevant to encoding techniques have shown that real coded representation is faster, more precise and more reliable especially with large domains. Binary representation requires more bits to represent the same search space, which slows down the algorithm. Real coded representations also incorporate problem specific knowledge as compared to their traditional counterparts and are thus more suitable for nontrivial, problem specific constraints. Also, higher precision can be achieved using real coded GAs by designing special operators as compared to traditional operators used in binary representations [4].

To solve a particular problem an encoding function is defined as :

$$C : S \rightarrow X$$

where S is the solution space of the problem and X is the genetic representation space. Members of the solution space, S, are referred to as phenotypes and individuals from the representation space, X, are known as genotypes. So a genotype is the encoded form of a phenotype and can be related by the encoding function as :

$$c = C(s), s \in S, c \in X.$$

Genetic algorithms evolve an initial set of candidate solutions by iterative application of stochastic operators, thus determining the size of a population pool to initiate a GA is a crucial factor for the performance of the GA. If there are too few individuals in a population, the GA has few choices to perform crossover and only a part of the search space is explored, increasing the risk of converging prematurely to a local minima. On the other hand an excessively large population size finds optimal solution at the expense of a large computation time. For the work described in this thesis, the population size remains constant from generation to generation. Also the simulations performed in this work show that after a limit, it is not desirable to use large populations because they do not solve the problem faster compared to moderate size population sets.

2.4.2 Fitness Function

A fitness (or objective) function is needed to evaluate the performance of individuals in a population. The assignment of a fitness function determines the relative scores (fitnesses) for individuals and helps in driving the search progress. A GA without an appropriate fitness (objective) function deteriorates into a multi-individual blind search. If there is not enough of a fitness difference among the individuals in a population set, it becomes almost impossible for a GA to select individuals for reproduction, therefore stalling the search process [4].

2.4.3 Selection

There are many selection mechanisms available for GAs depending on the nature of problem at hand. The main purpose of the selection operator is to focus the search process into promising regions of the search space leading to the global optimal solution. In order to achieve this, however, the GA must also explore the search space well and must make sure that the search does not become struck at a local minima which is referred to as “premature convergence”in GA. The selection operator depends directly on the measure of fitness (or the value of the objective function) of individuals in a population. The selection operator is designed to ensure that highly fit individuals have higher chances of being selected for reproduction to produce offsprings. The selection operator has two parameters:

- Selection pressure is defined as the degree to which highly fit individuals are allowed to reproduce in next generation.

- Takeover time is the number of generations it takes for a GA to create a final population pool that entirely contains the copies of an optimal individual.

Both the parameters are related in a way that if selection pressure of operator is high, the takeover time is small and vice versa. High selection pressure of operator favors the highly fit individuals and tends to reduce the population diversity quickly.

2.4.4 Crossover

Crossover advances the progress of the search procedure by recombining better individuals to produce offsprings. The crossover operator is applied to the population set of individuals obtained from the selection operator. The crossover operator depends on the representation used and different operators can be used, according to the way problem is encoded, to enhance the performance of GA. Holland's traditional GAs used one-point crossover [1]. In one-point crossover, a random crossover point (locus) is chosen on the chromosome string of each parent organism (individual). Pieces before and after the crossover point are recombined between the two parents to create two offspring. Many other versions such as two-point crossover and multi-point crossover have also been sucessfuly used in GAs [4].

The crossover operator is applied to the intermediate population pool, obtained after applying the selection operator to the initial population. The probability that the crossover operator is applied to each individual is known as crossover probability and is denoted as p_c. So not all the individuals in the intermediate pool are selected for crossover. The crossover probability, p_c, depends on the specific problem and can also be adjusted to enhance the performance of the GA.

2.4.5 Mutation

Mutation is a GA operator which changes the offsprings resulting from crossover. Mutation involves randomly altering one or more gene values in a chromosome string of the offspring. The main purpose of mutation is to maintain diversity within the population and inhibit premature convergence to unacceptable solutions. Like crossover, the mutation operator also depends on the type of encoding used for solving a problem. In binary representations, mutation involves flipping the bits from 0 to 1 and vice versa. Whereas for real number or integer encodings, boundary, uniform and non-uniform mutation operators can be used [4]. The non-uniform mutation is discussed in detail in chapter 4 while the implementation of the GA to the reconstruction of FBG sensor strain distributions and sensor placement problems.

Mutation is a probabilistic operator and the probability of applying the operator is called mutation probability (or mutation rate) and is denoted as p_m. Each gene in a chromosome string of offspring individual has equal chance of being mutated.

2.5 GA Procedure

The steps followed to obtain an optimal soluation using a GA are outlined here:

1. Create a random initial population P(t) of individuals (potential solutions) at time t.

2. The fitness of each individual is evaluated using fitness (or objective) function.

3. Individuals of the population pool $P(t)$ are selected according to their measure of fitness. The fittest individuals have a higher probability of selection for reproduction. The selected individuals form an intermediate population pool P_1. Now the individuals are chosen from pool P_1 with probability known as the crossover probability to act as parents for recombination to produce offsprings. This pool of parent individuals is called a mating pool.

4. The individuals from the mating pool undergo crossover to form new intermediate population pool P_2 containing offsprings.

5. The resulting offspring pool is further modified by introducing variability into the members through mutation to form population set P_3.

6. Finally the next generation population P$(t+1)$ is obatined by combining the individuals from P_3 and the individuals from P(t) that were not selected for crossover.

A detailed discussion of the implementation of the GA developed for this thesis to calculate the strain distribution profile by inverting the spectral response of Bragg grating sensors is given in chapter 4.

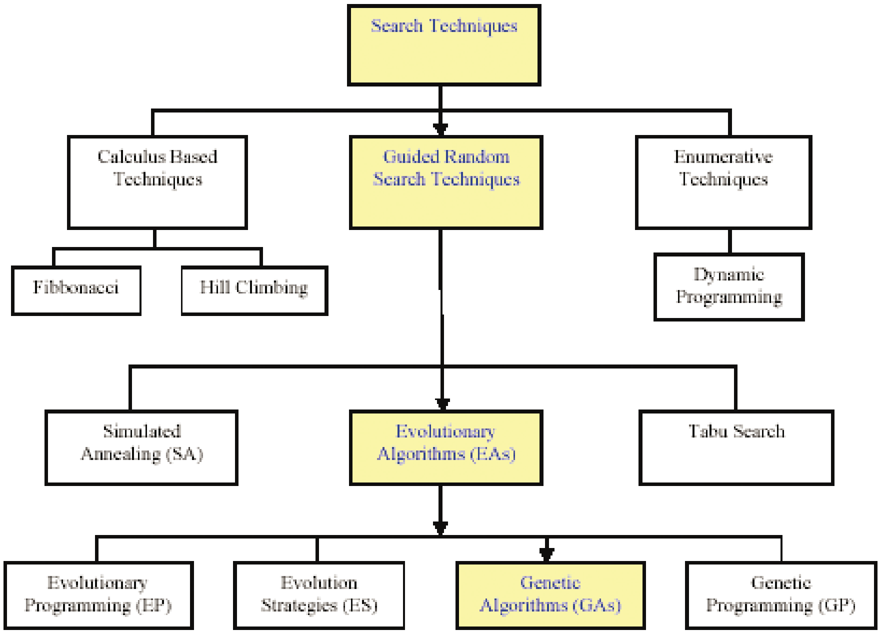

Figure 2.1: Classification of GAs among various search and optimization techniques.

2.6 GA Among Various Search and Optimization Methods

GAs are placed in a class of search and optimization techniques called Evolutionary Algorithms (EAs) which are a subset of Artificial Intelligence (AI), as seen in figure 2.1. The most important characteristic of evolutionary algorithms is the intensive use of randomness and genetics-inspired operations to evolve a set of candidate solutions to a given problem. However, genetic algorithms have important differences as compared to other evolutionary algorithms such as evolutionary programming (EP), evolution strategies (ES) and genetic programming (GP). An in-depth comparision among various evolutionary computational models is given in [5]. The main difference between GAs and ES or EP is the way new solutions (offsprings) are produced from existing members. In GAs crossover is the main operator, whereas in ES and EP mutation is employed as the main search operator. Genetic programming (GP) differs from GAs in the way the potential solutions are represented, GP evolves a population of programs (candidate solutions) to solve problems therefore each individual is a program.

Goldberg presents some important differences between genetic algorithms and tra-

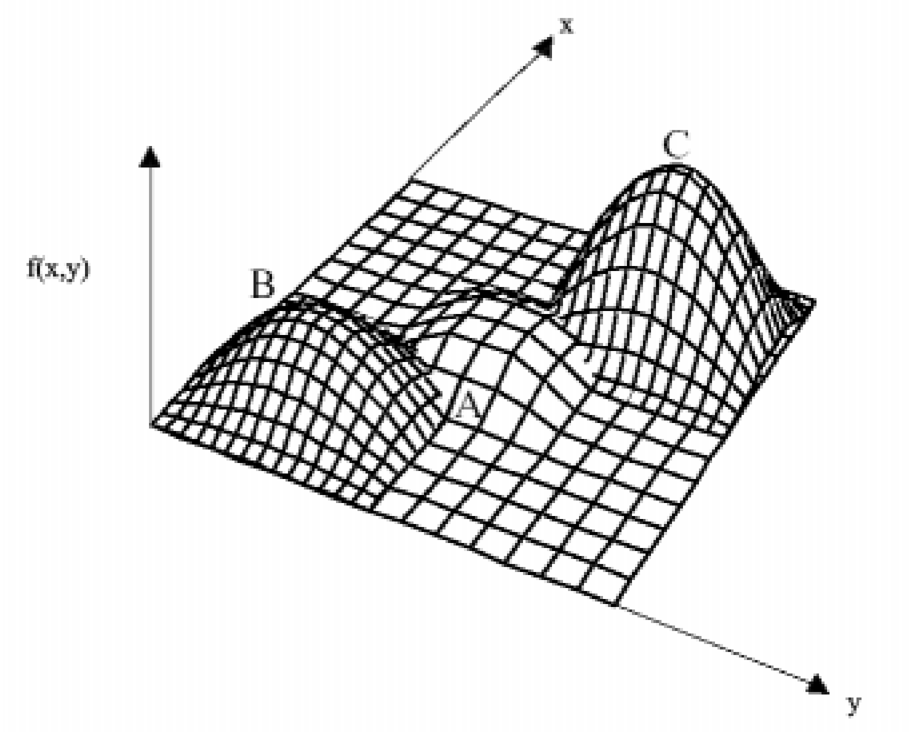

Figure 2.2: Function with several local optimum points such as A, B & C.

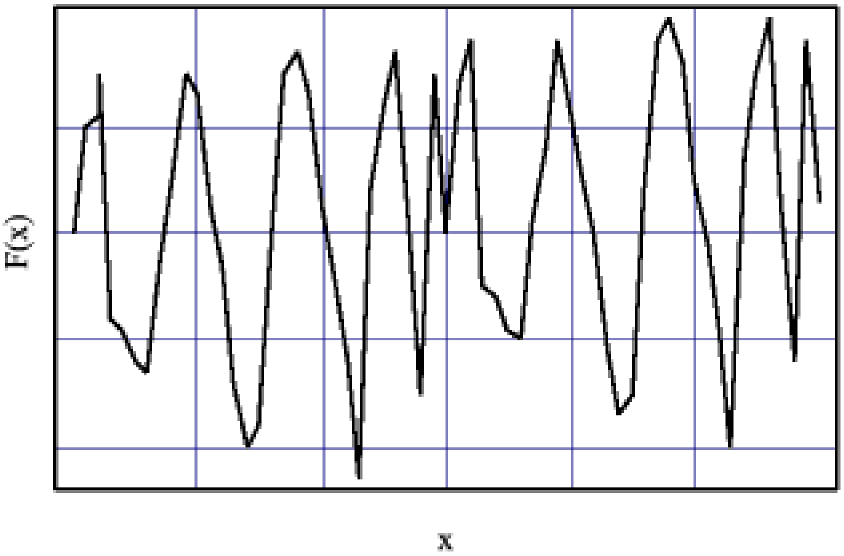

Figure 2.3: Function not suitable for derivative based optimization methods.

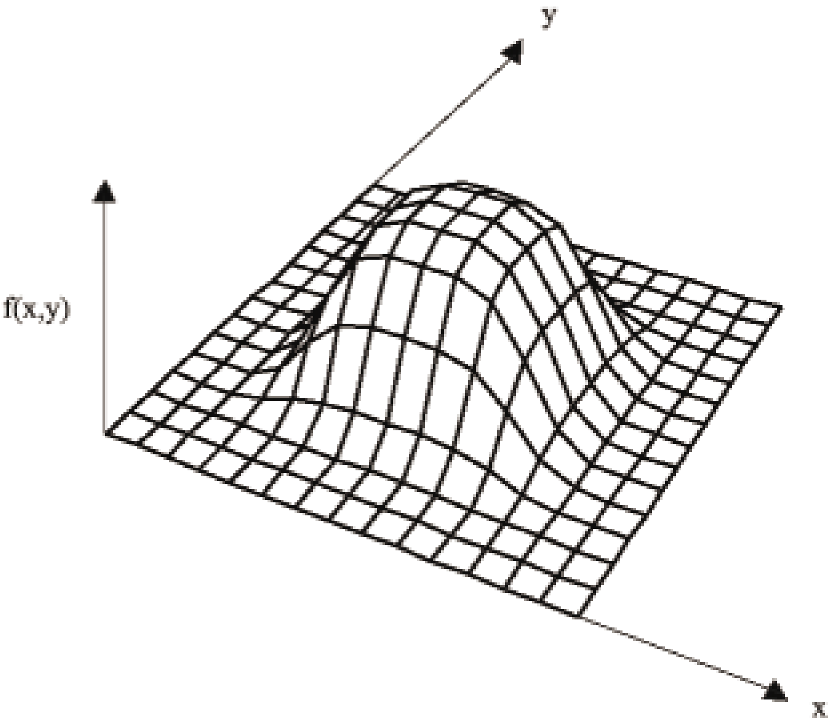

Figure 2.4: Function with single optimum point.

ditional search methods, summarized here [2]. GAs work with a fitness (or objective function) and do not require any derivative information about the fitness function. Thus GAs work well when the search space is discontinuous and multimodal as shown in figures 2.2 and 2.3. Calculus based methods are ideal for only the unimodal, continuous search spaces, as in figure 2.4, due to their dependence on derivatives. At a given time GAs work with a population of points in the search space instead of a single point. With this feature GAs can search many regions of the space at once. While a region rich in information is exploited, other regions are explored thus maintaining a trade-off. Working with the population of points makes GAs robust in tackling complex, noisy spaces. Calculus based methods (e.g. hill climbing) are vulnerable to become struck in local optimum points. Other traditional techniques such as enumerative methods are also inefficient in large, complex spaces since to find optimal solutions they look at function values of every point in the space. Since GAs are implicitly parallel algorithms they can test many regions of a space without having to evaluate every point in the space. Genetic algorithms also employ probabilistic rules instead of deterministic ones. This allows them to use randomness during the search procedure and therefore they can easily avoid local optima points. However, GAs are not merely random searches but use randomness to guide a search process towards an optimum point of the coded parameter space.

Chapter 3

Fiber Bragg Grating Sensors

3.1 Background

Fiber Bragg grating sensors are the most commonly used fiber optic sensors for strain sensing applications in structural health monitoring. Optical fiber Bragg gratings are unique among embedded sensors due to their potential to measure strain distributions with a spatial resolution of a few nanometers over a gage length of a few centimeters. A fiber Bragg grating (FBG) is a periodic perturbation of the index of refraction along a certain length of the core of the optical fiber. When light propagates through the periodically alternating regions of higher and lower refractive index within the fiber, it is reflected by successive, coherent scattering from the index variations. When the reflection from a crest in the index modulation is in phase with the next one, we have maximum mode coupling or reflection. The strongest mode coupling occurs at the Bragg wavelength, λ_B, (see figure 3.1) given by,

$$\lambda_B = 2n_{eff}\Lambda \tag{3.1}$$

where n_{eff} is the effective mode index of refraction of the fiber and Λ is the grating period. The Bragg gratings are fabricated by exposing an optical fiber core to a spatially varying pattern of ultraviolet intensity to produce the alternate high and low-refractive index regions along a certain length of optical fiber. Typical values for the index change as a result of fabrication process are of the order of 10^{-4} to 10^{-3}, dependent on the UV-exposure and the dopants (germanium, molecular hydrogen etc.) used in the optical fiber core. Therefore, when broadband light is launched into the fiber Bragg grating, only a narrow band centered at wavelength, λ_B, is reflected.

The most common mathematical model that governs wave propagation in gratings is the coupled-mode theory [6]. It describes a relation between the spectral dependence of a fiber grating and the corresponding grating structure. The grating is an intrinsic sensor which changes the spectrum of incident signal by coupling energy to other fiber modes. The sensing principle underlying the fiber Bragg grating is that any change in strain, stress or temperature which affects the modal index, n_{eff}, grating period Λ, will also change the the Bragg wavelength λ_B. Therefore by determining the wavelength at maximum reflectivity peak we can get information about the measurand, which in this work is a strain distribution. If the strain is uniform along the grating, all the grating periods experience same change and there is simply a wavelength shift in the reflected spectrum.

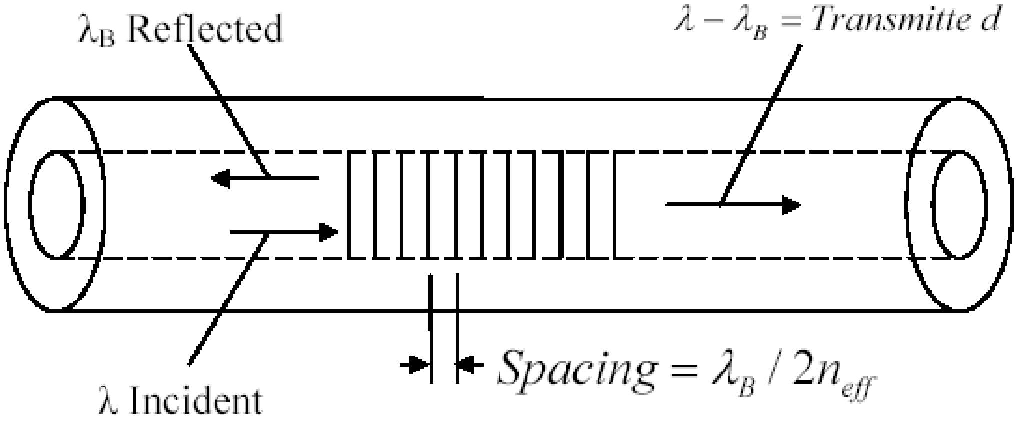
λ_B Reflected
$\lambda - \lambda_B = Transmitte\ d$
λ Incident
$Spacing = \lambda_B / 2n_{eff}$

Figure 3.1: Bragg rasonance for the reflection of incident mode occurs at the wavelength for which grating period along the fiber axis is equal to one-half of the modal wavelength within the fiber core. The back scattering from each crest in the periodic perturbation will be in phase and the scattering intensity will accumulate as the incident wave is coupled to a backward propagating wave [7].

Monitoring of the wavelength shift can be used an indicator of applied strain (see figure 3.2). However, when subjected to non-uniform strain field along the grating, the reflected wavelengths from different locations within the grating are different and reflected spectrum is a function of applied strain distribution [8].

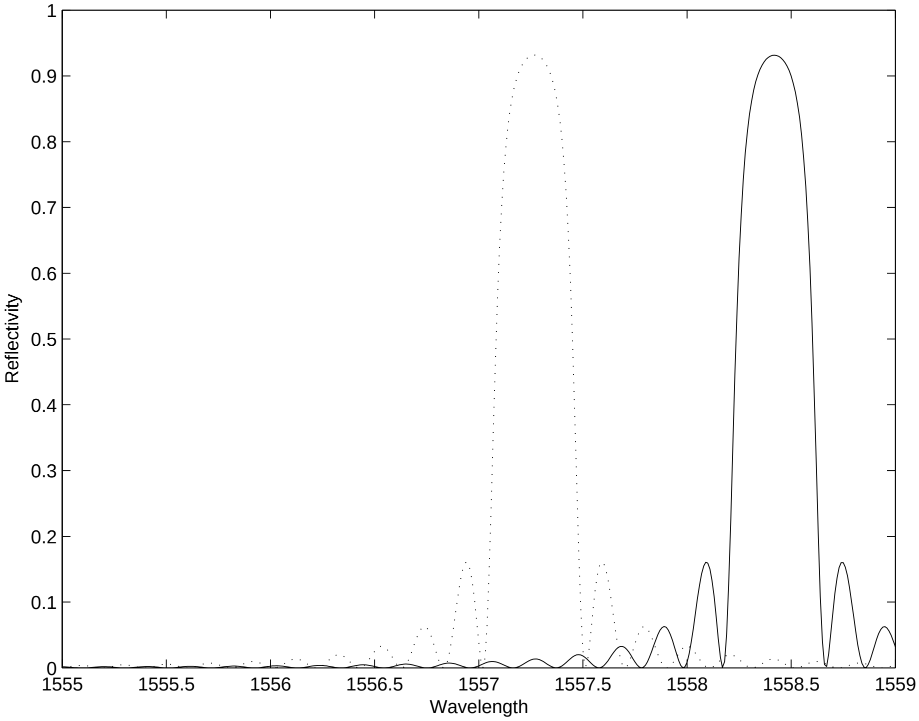

Figure 3.2: Wavelength shift in reflected spectrum of fiber Bragg grating. Reflected spectrum of unloaded grating (zero strain) is plotted by dashed line. Solid line plot gives reflected spectrum of grating as result of applied constant strain.

So the current problem involves inverting the spectral response of a Bragg grating to determine the applied strain distribution. The purpose of this work is to determine the applied strain profile from the reflection spectrum of fiber Bragg grating sensor, which is successfully demonstrated using genetic algorithm in conjunction with Transfer matrix formalism, in the later sections of this thesis.

Other features of fiber Bragg grating sensors which make them a favourable choice as strain sensors are:

- demonstrate excellent strain senstivity (wavelength/strain senstivity of the order of 1.2pm/μ strain);
- immunity to electromagnetic interference, radio-frequency interference and radiation;
- lightweight, compact and can be easily embedded or mounted to create smart materials for operation in harsh environments;
- sensed information is encoded directly into wavelength which is an absolute parameter. Therefore output does not depend on the total light intensity or losses in the connecting fibers and couplers or recalibration or reinitialization of the system [9];
- can be multiplexed. Inherent low transmission loss enables many sensors on a single optical fiber. Using wavelength division multiplexing (WDM) each sensor can be interrogated independently and can be used for distributed sensing over large structures;
- ease of installation and use. Because the gratings are multiplexed over a single fiber, many sensors can be accessed with single connection to optical source and detector. Whereas, conventional electronic strain-guage sensors require each sensor to have its lead wires attached and connected to the data acquisition system.
- low cost as a result of high volume manufacturing for their use as WDM add/drop filters, gain flattening filters, wavelength stabilizers for pump lasers, etc., in telecommunications.

3.2 Inverse Problem

As seen above, once the fiber Bragg grating is subjected to a constant strain field, the shift in wavelength at maximum reflectivity, called the Bragg wavelength, is directly proportional to the applied strain [8]. However, for strain sensing applications involving non-uniform strain fields, the reflected spectrum changes as a function of strain distribution along the length of grating. Therefore the wavelength shift, formerly used as an indicator of applied strain, can no longer determine useful strain information. Thus the calculation of the applied strain distribution from the spectral response of the optical fiber Bragg grating sensor becomes an inverse problem.

Some of the first inversion approaches applied to fiber Bragg grating spectra are reviewed in Huang *et al* [8]. The first inversion method is based on the measured intensity spectrum and requires only the reflection amplitude from the complex reflected spectrum. However, this method is only valid for monotonically varing strain fields. Also, this method averages out all the sharp changes in strain variations. In spite of this, the intensity based spectrum (ISB) approach is still very practical due to its simplicity. The second method is based on the measured phase spectrum and is also valid only for monotonically varying strain fields. Also this technique requires an interference detection scheme and an extremely stable thermomechanical environment to obtain the phase spectrum which makes the measurement process costly and more difficult than with ISB. The third approach based on Fourier transfrom (FT) can determine any arbitrary unknown strain profile but requires both intensity and phase information from the measured spectrum. Thus the FT approach is associated with a costly setup (fast tuning laser) and critical measurement conditions. Furthermore the FT method is only suitable for low reflectivity grating (weak gratings).

An exact solution of the inverse problem was found by Song & Shin who solved the Gelfand, Levitan and Marchenko (GLM) coupled integral equations in scattering theory. Their derivation is an exact solution but these equations can only be solved by numerical methods and only when reflection coefficients can be expressed as a rational function [11]. This limitation was eliminated by a method developed by Peral *et al* which is based on an iterative solution to the GLM equations. Their algorithm provided satisfying results even for high reflectivity gratings and was developed primarily for filter applications in fiber-optic communications [12]. Though this technique has been used for other applications, the solution is approximate due to the finite number of iterations computed which further implies that only a limited number of reflections are considered within the medium. This drawback becomes particularly noticeable for strong gratings (high-reflectivity) and for gratings with discontinuities in the coupling function. So, to improve the algorithmic efficiency of the above method and also to reduce the approximations involved in it, Feced *et al* proposed an exact differential inverse scattering algorithm [13]. Their differential layer-peeling algorithm inverts the spectral response by taking into consideration all the multiple reflections inside the grating and is used for design of complex fiber Bragg gratings. Though this method shows a potential for specific problems in grating synthesis, it does not permit one to weigh certain desired parts of a spectra more than others. This weighing mechanism in an algorithm becomes important for sensing applications in noisy environments and is an essential feature of an algorithm to be successfuly adopted as a sensor interpretation tool.

Recently, new inversion methods based on genetic algorithms (GAs) have been developed for the synthesis (reconstruction) of fiber Bragg grating parameters. Skaar and Risvik developed a genetic algorithm for the design of fiber Bragg gratings as filters for telecommunication applications [14]. This method provides an option to weigh certain parts of filter spectra, as desired in communication applications as compared to above mentioned techniques. Skaar and Risvik, in their GA, used binary representations of the grating coupling coefficients combined with Runge-Kutta analysis method to produce the reflected spectrum as close as possible to desired (target) spectrum. Cormier *et al* reconstructed three design variables - length L, grating period Λ, and average modulation in refractive index δ_{neff}, of a fiber Bragg grating from the measured reflected spectrum intensity information (no phase information required) [15]. They developed a GA which used real encoding of the design variables in conjunction with the T-matrix formalism to minimize the difference between desired and measured spectra. The T-matrix formulation is based on approximating the period distribution as a piecewise contant function over a grating section length, after dividing the grating into many smaller sections. Although the method demonstrates an excellent performance for designing filters it does not permit period variations along the grating due to the parameter formulation, which is a limitation when it comes to strain sensing applications.

Later, Casagrande *et al* described a GA approach to solve an inverse problem for the fiber Bragg grating sensor [16]. Their method fits the grating properties (and therefore the strain distribution) to defined polynomial functions, and calculates the spectral response by solving the resulting Riccati-type differential equation. Their method optimizes the polynomial coefficients to find the solution minimizing the difference between the calculated reflected spectrum and the measured reflected spectrum. Their experiments and simulations matched well for the applied quadratic strain field considered. However, the method of Casagrande *et al* requires that the strain distribution be well described by a polynomial approximation, otherwise an extremely large number of coefficients would be required. Therefore such an approach is not acceptable for damage detection systems where strain field can vary rapidly or even become discontinuous.

3.3 Bragg Grating Simulation Using T-matrix Method

There are a variety of techniques to simulate Bragg gratings to compute their reflection and transmission spectra [6]. The most direct method involves numerical integration of the coupled mode equations using the Runge-Kutta algorithm. A second, less common method requires that the grating is discretized into a stack of discrete, complex reflectors. The entire spectrum is obtained by using a recursive expression and is similar to Rouard's method of thin film optics which has been applied to corrugated waveguide filters [6]. However in this thesis, a fast and accurate technique known as the Transfer-matrix method is used for computing the reflected spectra of the Bragg grating.

The T-matrix method divides the grating of length L into M smaller sections so that each section can be treated as approximately uniform. This T-matrix method used to model non-uniform gratings is based on approximating the grating as piecewise uniform and identifying a 2×2 response matrix for each uniform section of the grating and then multiplying all of these together to obtain a single 2×2 matrix which models the entire profile of the grating [17].

The Bragg grating is characterized by the following quantities: the design Bragg wavelength λ_D, the effective mode index of refraction of the fiber n_{eff}, and the slowly varying complex coupling coefficient κ. The forward propagating wave R and backward propagating wave S are dependent on the spatial parameter z (see figure 3.3) and are mutually related by coupled mode equations as [10]:

$$\frac{dR(z)}{dz} = i\hat{\sigma}R(z) + i\kappa S(z)$$

$$\frac{dS(z)}{dz} = -i\hat{\sigma}S(z) - i\kappa R(z) \tag{3.2}$$

In Eq.(3.2), $R(z)$ and $S(z)$ are the amplitudes of the forward and backward propagating modes. These amplitudes are also a function of the propagating wavelength, λ. $\hat{\sigma}$ is the general “dc ”self-coupling coefficient defined by

$$\hat{\sigma} = \delta + \sigma - \frac{1}{2}\frac{d\phi}{dz} \tag{3.3}$$

where δ is the phase shift per unit length compared to design Bragg wavelength $\lambda_D = 2n_{eff}\Lambda$ and is given as

$$\delta = 2\pi n_{eff}(\frac{1}{\lambda} - \frac{1}{\lambda_D}) \tag{3.4}$$

For single mode Bragg reflection gratings,

$$\sigma = \frac{2\pi}{\lambda}\overline{\delta n}_{eff} \tag{3.5}$$

And κ is the “AC ”coupling coefficient defined by

$$\kappa = \frac{\pi}{\lambda}\nu\overline{\delta n}_{eff} \tag{3.6}$$

To apply the T-matrix method we divide the grating of length L into M uniform sections and define R_i and S_i to be the field amplitudes after transversing section i. The grating may be considered as a four port device with four inputs/outputs: the forward propagating mode amplitudes $R(-L_1/2)$ & $R(L_1/2)$ and the backward propagating mode amplitudes $S(-L_1/2)$ & $S(L_1/2)$ as shown in figure 3.3. The transfer matrix T_1 represents the grating amplitude and phase response of the first segment. The T-matrix in the following equation transforms the two fields on the RHS of equation into the fields on the LHS as [6],

$$\begin{bmatrix} R(-L_1/2) \\ S(-L_1/2) \end{bmatrix} = [\,T_1\,] \begin{bmatrix} R(L_1/2) \\ S(L_1/2) \end{bmatrix} \tag{3.7}$$

The boundary conditions applied to equation (3.7) directly yield the reflectivity and transmissivity of the grating. The applied boundary conditions depend on whether the grating involves counterdirectional or codirectional coupling. For Bragg gratings, coupling occurs between modes travelling in opposite directions. For a Bragg grating we begin with the RHS amplitude vector for the first segment and transform the problem to one with the

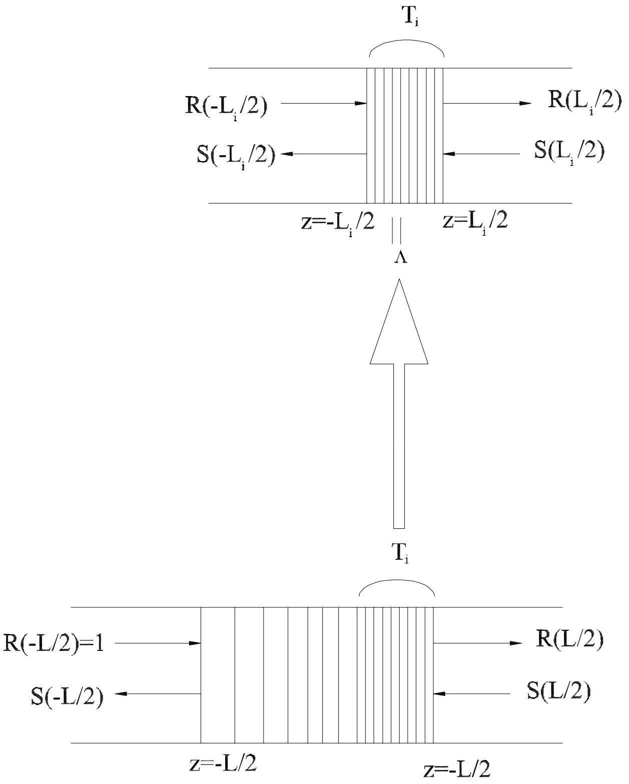

Figure 3.3: Bragg grating Transfer matrix model.

boundary conditions $R(L_1/2) = 1$ & $S(L_1/2) = 0$ and integrate backwards from $z = L/2$ to $z = -L/2$ [10]. Applying the boundary conditions and writing the matrix elements, equation (3.7) becomes,

$$\begin{bmatrix} 1 \\ S(-L_1/2) \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} R(L_1/2) \\ 0 \end{bmatrix} \tag{3.8}$$

The reflected and transmitted amplitudes are then determined from equation (3.8) as

$$\begin{aligned} S(-L_1/2) &= T_{21}/T_{11} \\ R(L_1/2) &= 1/T_{11} \end{aligned} \tag{3.9}$$

These outputs of first section become the input field amplitudes for section 2 and are transformed again by another matrix T_2 and so on, until all the grating sections are modeled such that $L = \sum_{i=1}^{M} L_i$. Each grating segment is assumed to be uniform but to have slightly different values of κ, Λ and n_{eff} from its neighbouring segments, therefore each segment is treated as a uniform grating with its own T-matrix. After calculating the optical response matrices for all the sections, the output amplitudes for the entire Bragg grating is determined using

$$\begin{bmatrix} R(-L/2) \\ S(-L/2) \end{bmatrix} = [\,T\,] \begin{bmatrix} R(L/2) \\ S(L/2) \end{bmatrix} \tag{3.10}$$

where the transfer matrix of the entire grating T is

$$T = [\,T_M\,] \ldots [\,T_3\,]\,[\,T_2\,]\,[\,T_1\,] \tag{3.11}$$

Now the reflectivity, ρ, of the entire grating is then determined using equation(3.12)

$$\rho = \frac{S(-L/2)}{R(-L/2)} = \frac{T_{21}}{T_{11}} \tag{3.12}$$

Also the transmissivity of the entire grating can be found as

$$\tau = (1 - \rho) = \frac{R(L/2)}{R(-L/2)} = \frac{1}{T_{11}} \tag{3.13}$$

From the closed form solutions for the coupled first-order differential equations with constant grating properties in equation (3.2), the transfer matrix elements for the i^{th} section are [8]

$$\begin{aligned} T_{11} &= \cosh[\gamma(\lambda)L_i] - i\frac{\hat{\sigma}(\lambda)}{\gamma(\lambda)}\sinh[\gamma(\lambda)L_i] \\ T_{12} &= -i\frac{\pi\nu\overline{\delta n_{eff}}}{\gamma(\lambda)\lambda}\sinh[\gamma(\lambda)L_i] \\ T_{21} &= i\frac{\pi\nu\overline{\delta n_{eff}}}{\gamma(\lambda)\lambda}\sinh[\gamma(\lambda)L_i] \\ T_{22} &= \cosh[\gamma(\lambda)L_i] + i\frac{\hat{\sigma}(\lambda)}{\gamma(\lambda)}\sinh[\gamma(\lambda)L_i] \end{aligned} \tag{3.14}$$

where $\hat{\sigma}$ and $\gamma(\lambda)$ are coupling coeffcients and have local values in the particular i^{th} section and are given as

$$\begin{aligned} \hat{\sigma}(\lambda) &= \frac{2\pi}{\lambda}(n_{eff} + \overline{\delta n_{eff}}) - \frac{\pi}{\Lambda_i} \\ \gamma(\lambda) &= \sqrt{\left(\nu\,\pi\,\overline{\delta n_{eff}}/\lambda\right)^2 - \hat{\sigma}^2(\lambda)} \end{aligned}$$

When the grating is divided into smaller sections, the length of each section must contain multiple periods i.e., $L_i = N_i\Lambda$ where N_i (i=1, 2, ..., M) are integers. The integer number of periods ensures the coupling between sections by perserving the phase of the periodic index modulation because the coupling coefficient depend on the index modulation. Though a section length can be chosen as short as only one pitch, the number of sections needed for piecewise-uniform calculation is determined by the required accuracy [19].

The T-matrix approach is a very fast method to calculate the response of Bragg gratings as compared to numerically solving the coupled mode equations in direct methods involving simulation of Bragg gratings. The transfer matrix method is used once the non-uniform strain distribution is assumed piecewise constant. Also the error due to the approximations involved in T-matrix formalism is insignificant even for highly non-linear strain fields and therefore significantly less than those due to approximation of some strain distributions into polynomial or other forms as shown by Peters *et al* [18].

Chapter 4

Reconstruction of Bragg Grating Sensor Strain Distribution

The objective of this chapter is to develop a genetic algorithm (GA) for the reconstruction of strain distribution profiles from optical fiber Bragg grating sensors. The genetic algorithm is developed to solve the inverse problem of reconstructing a strain distribution profile using only the output intensity spectrum of the grating. The spectral response is calculated via a T-matrix formulation approach. The developed algorithm simulates the fiber Bragg grating strain sensor to calculate the period distribution along the grating which is then related to the axial strain distribution. The following sections describe in detail the implementation of a real coded genetic algorithm in conjunction with the transfer matrix formalism.

4.1 Analysis Method

The optical fiber Bragg grating is a periodic modulation in the index of refraction along a given length of the optical core, fabricated by UV irradiation of the fiber. Due to coupling between forward and backward propagating modes, some wavelengths of light are reflected rather than transmitted by the fiber at the location of the grating. The reflected spectral response for such a grating is shown in figure 3.2. The wavelength at which maximum reflection occurs, λ_B, is called the Bragg wavelength and is given by [10]

$$\lambda_B = 2n_{eff}\Lambda \tag{4.1}$$

where n_{eff} is the effective mode index of refraction of the optical fiber and Λ is the period of the grating.

The periodic modulation in refractive index of the optical fiber core describing the Bragg grating can be given by the function [10],

$$\delta n_{eff}(z) = \overline{\delta n}_{eff}(z)\{1 + \nu \cos[\frac{2\pi}{\Lambda_0} z + \phi(z)]\} \tag{4.2}$$

where z is the coordinate in the direction of light propagation along the length of fiber grating, $\overline{\delta n}_{eff}$ is the mean mode effective index of refraction change, Λ_0 is the period of the unloaded grating and ν is the fringe visibility of the index change. The function $\phi(z)$ describes the grating chirp (i.e. change in grating period along the grating length) and in this work is used to model the effects of applied strain. The chirping function $\phi(z)$ can be related to the applied axial strain, ϵ_z, along the optical fiber as [18],

$$\phi(z) = \frac{-2\pi}{\Lambda_o} \frac{(1 - p_e)\epsilon_z(z)}{[1 + (1 - p_e)\epsilon_z(z)]} z \tag{4.3}$$

where p_e is the effective strain-optic coefficient of the optical fiber.

The period change along the grating can also be related to the applied strain, expressed as,

$$\Lambda(z) = \Lambda_o[1 + (1 - p_e)\epsilon_z(z)] \tag{4.4}$$

Therefore, the applied strain distribution $\epsilon_z(z)$ can also be determined by calculating the period distribution along the Bragg grating. In general, the strain distribution can be an arbitrary function of z. The solution of the Bragg grating spectral response due to the chirping function of equation (4.3) is computationally intensive. However, as explained in section 3.3 for a constant strain field, the optical response can be expressed as [10]:

$$\begin{bmatrix} P_i \\ N_{i-1} \end{bmatrix} = [\,T_i\,] \begin{bmatrix} P_{i+1} \\ N_i \end{bmatrix} \tag{4.5}$$

where P_i and N_i are the forward and backward propagating waves traveling through the i^{th} grating section, as shown in figure 4.1.

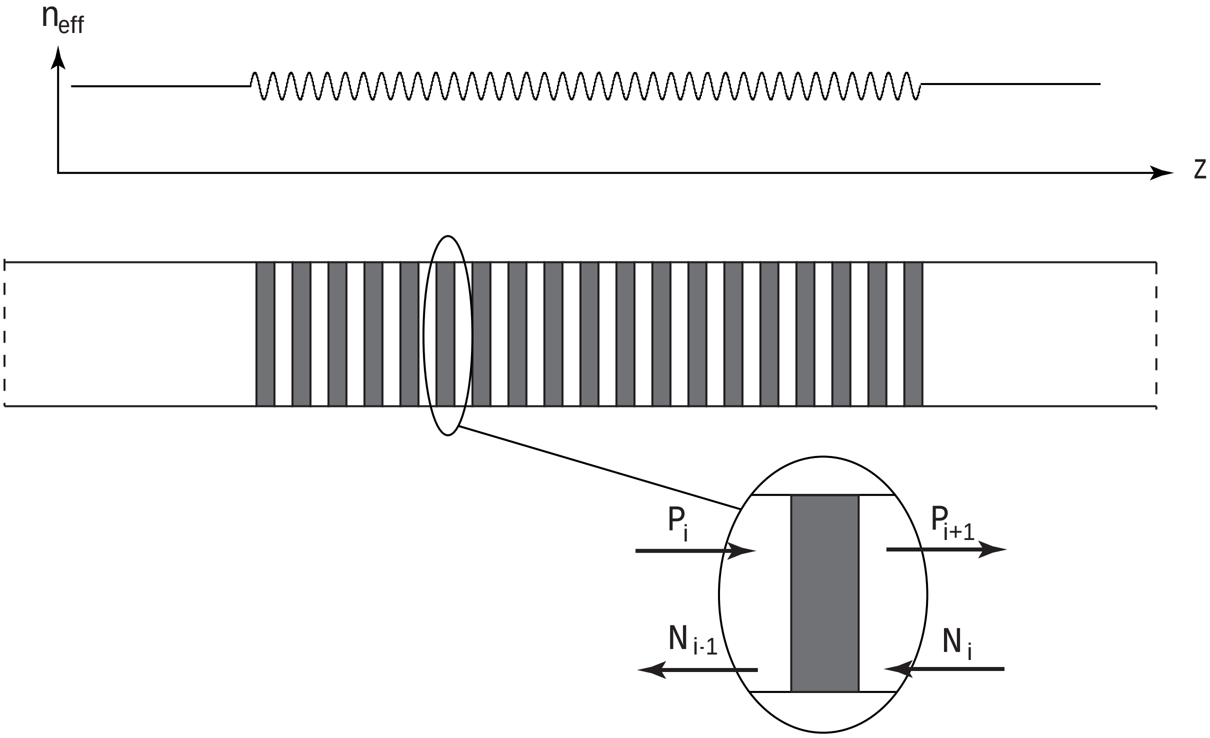

Figure 4.1: Parameters for T-matrix formulation of Bragg grating spectral response. Also shown is effective mode index of refraction of optical fiber.

The approximation involved in the T-matrix formulation is to divide the grating with non-uniform period distribution into series of smaller segments over which the period is assumed constant. In strain sensing applications, this assumption models the non-constant strain field as a piecewise constant strain field. The response of each grating segment is then modeled using equation 3.7 and applying appropriate boundary conditions as discussed in section 3.3. This process is repeated untill an entire profile is modeled and the optical response of the entire grating is then written as a product of the response matrix for each grating segment. As in this case for M segments in a grating, the T-matrix for whole grating is:

$$T = T_M \cdot T_{M-1} \cdot \ldots \cdot T_1. \tag{4.6}$$

In this work, once the piecewise period distribution has been reconstructed using a GA, the strain distribution is calculated through equation 4.4.

4.2 Genetic Algorithm Modeling Approach

The genetic algorithm is an excellent choice for calculating the Bragg grating sensor strain distribution because GAs require only the fitness information of a particular solution and not the gradient information with respect to the fitness function. In the present work, the genetic algorithm requires only the output reflected spectrum of the Bragg grating strain sensor and yet efficiently reconstructs non-linear and discontinuous strain distributions. As described in section 2.4, the essential steps for implementation of a GA are [4]:

- genetic representation (encoding) of potential solutions to the problem;
- initialization of a random population of solutions;
- assignment of a fitness function;
- iterative application of genetic operators : selection, crossover and mutation;
- optimization of genetic parameters such as population size, crossover probability, mutation probability.

The following sections describe the application of the above steps to the optical fiber Bragg grating sensor problem.

4.2.1 Intialization & Encoding

The first step in the implementation of a GA is to encode potential solutions into the form of a chromosome string. The inverse problem to be modeled involves the determination of the period distribution along a Bragg grating from the reflected spectrum (or transmitted spectrum) of the Bragg grating sensor. The solution of this inverse problem makes it possible to calculate the applied axial strain distribution along the grating.

The initial population is made up of randomly generated parameters within specified boundaries. These parameters (or variables) are the genes of the potential individual solution. A random population is generated of η individuals representing possible optimal solutions. In this work real valued encoding is used as it permits a greater degree of accuracy and better convergence than binary representations, as shown in experimental results by Michalewicz [4]. In real valued encoding, each solution is encoded as real valued search vector.

Each gene corresponds to a parameter (or variable) of the function to be optimized. In our analysis approach we model the applied strain distribution as period change along the grating, so in this work potential solutions (or individuals) are represented by the vector of periods for each segment of the grating. The period values for each grating segment are the genes of individual chromosomes (possible solutions). The k^{th} individual, I^k , in a population pool of η potential solutions, is represented by,

$$I^k = \{\Lambda_1^k, \Lambda_2^k, \ldots, \Lambda_M^k\} \tag{4.7}$$

where M is the number of grating segments as mentioned in the T-matrix formalism. The minimum number of segments required to model the applied strain field does not have to be known in advance but can be determined from the GA convergence to the measured reflected spectrum. The dimension of the chromosome string is constant for all the members of the population and in this work is equal to the number of grating segments chosen to model the applied strain distribution. The M^{th} position of the above encoding represents the value of the M^{th} gene in the chromosome string (individual), I^k. For the initial population, individuals are generated with period vectors uniformily distributed around the period corresponding to the centroid of the measured reflected spectrum, λ_c.

$$\Lambda = \lambda_c/2n_{eff} \tag{4.8}$$

The main objective of real-valued encoding is to make the representation closer to the solution space of the problem. The encoding technique given in equation 4.7 used here will be shown to be efficient to represent the highly non-linear and discontinuous strain fields. This feature thus enables the developed GA to be used for strain sensors located near the location of cracks, a disbond, or any discontinuity within a structure. Although the GA itself is as equally numerically efficient for determining highly non-uniform, discontinuous strain fields as for smooth, weakly varying strain fields, an increased number of segments may be required to model the spectral response well.

4.2.2 Selection

The selection of individuals for reproduction is done through the evaluation of a fitness function for each individual. In each generation certain individual members are chosen for the creation of new individuals, forming a reproduction pool. The probability of each individual entering the reproduction pool is determined by its fitness value.

The measured reflected spectrum from the Bragg grating sensor, $\overline{R}(\lambda)$, is discretized into a series of H points. In this work, the fitness function (also called the objective function), F, is defined as the squared difference between the measured reflectivities, $\overline{R}_i$, and the computed ones, R_i^k, over a subset of wavelengths corresponding to a set of H points chosen from the measured reflected spectrum.

$$F(I^k) = \sum_{i \subset \{1,...,H\}} |\overline{R}_i - R_i^k|^2 \tag{4.9}$$

In this work F is to be minimized and the individuals with lower fitness values are more likely to be chosen for reproducing new individuals (offspring).

Many selection techniques exist, however the one used here was developed by Davis [20]. The individual with the lowest fitness value in the current generation receives a relative weight, RW, of η^a, where a is a weight factor ($1.0 \leq a \leq 1.5$). The second best individual is assigned a relative weight of $(\eta - 1)^a$ and so on, until the worst individual, the one with highest fitness value, receives a RW of 1. The probability of selection for reproduction, of each individual is then calculated as:

$$PR^j = \frac{RW^j}{(1/\eta)\sum_{k=1}^{\eta} RW^k} \tag{4.10}$$

where PR^j is the probability of reproduction for individual j (where j=1,2,. . . ,η) and η is total number of individuals in a population (which remains constant during all the generations of GA). The individuals with higher RWs have better chances of selection for reproduction. Once selected, the individuals are entered into an intermediate pool. The frequency of each individual's selection is determined using stochastic remainder selection [2]. For example, an individual with a probability of 2.2 will be selected twice for reproduction where as an individual with probability 1.7 will be selected only once. The number of individuals obtained are not same for every iteration. However to keep the population size constant, the remaining pool members are selected by roulette wheel selection.

4.2.3 Uniform Whole Arithmetical Crossover

After the selection of individuals for recombination to form an intermediate pool, the individuals are then randomly selected from the intermediate pool to act as parents for reproducing new individuals (offspring). The probability that each individual is chosen as a parent to enter into the mating pool is defined as the probability of crossover, p_c. Therefore not all the individuals in an intermediate pool enter the mating pool. The main purpose of crossover is to produce new individuals that result in low fitness values, thus optimizing the characteristics of next generation. For real-valued encoding, there are many crossover operators available but in this work Uniform Whole Arithmetical crossover proposed by Wright is used [21]. In this technique two randomly selected parents P_1 and P_2 produce three offspring (new individuals) O_1, O_2 and O_3 by linear combination of the parent (individual) vectors as given in equation (4.12),

$$\begin{aligned} P_1 &= \{\Lambda_1^1, \ldots, \Lambda_M^1\} \\ P_2 &= \{\Lambda_1^2, \ldots, \Lambda_M^2\} \end{aligned} \tag{4.11}$$

$$\begin{aligned} O_1 &= cP_1 + (1-c)P_2 \\ O_2 &= (1+c)P_1 - cP_2 \\ O_3 &= -cP_1 + (1+c)P_2 \end{aligned} \tag{4.12}$$

where c is the crossover proportion parameter and remains constant for all the generations. The two offspring with the lowest fitness values are chosen and the third offspring is dropped to keep the population pool size constant. The individuals in the intermediate pool which are not selected for crossover (reproduction) are then passed to the next step directly without any change.

4.2.4 Non-uniform Mutation

The process of mutation introduces diversity into the population, thus avoiding that the GA converges to an unacceptable local minimum rather than the global optimal solution. The mutation operator is applied to the new population pool of η members created after the crossover process. The gene (which is the period value for particular segment in this case, as mentioned in section 4.2.1) in each individual member has an equal chance of being mutated according to the user controlled mutation rate p_m. In this work, the non-uniform mutation operator proposed by Michalewicz is used with a small modification to suit the performance of the GA in this problem. The value of those genes chosen for mutation are changed according to the mutation operator defined by,

$$\Lambda_i^k = \begin{cases} \Lambda_i^k + \Delta(\mathrm{Max}[\Lambda_i] - \Lambda_i^k) & S = 0 \\ \Lambda_i^k - \Delta(\Lambda_i^k - \mathrm{Min}[\Lambda_i]) & S = 1 \end{cases} \tag{4.13}$$

where S is a random binary number 0 or 1, which determines whether the value of the gene is increased or decreased respectively. The corresponding increase or decrease in the value of the gene is controlled by the value of the amplitude function $\Delta(y)$ described below. The values $\mathrm{Max}[\Lambda_i]$ and $\mathrm{Min}[\Lambda_i]$ are the upper and lower bounds on the variable Λ_i for all current members of the population. The amplitude function of mutation, $\Delta(y)$, decreases as the number of generations increases and is given by equation 4.14,

$$\Delta(y) = [yr(1 - t/N)]^B \tag{4.14}$$

where r is a random number between 0 and 1, B is a weighting factor such that $2 < B < 5$ and gives the extent of non-unifomity of the mutation process, t is number of the current generation, and N is the total number of generations.

4.2.5 Elitism

Finally, elitism is used to avoid the best individual with lowest fitness value from being lost in the next generation, due to the randomness involved in the application of a GA. Elitism compares the best solution of the current generation with the best solution of the previous generation and keeps the current solution if the current generation solution has a lower fitness value in comparision. Otherwise, the best individual from the previous generation replaces the current generation solution.

The above procedures of selection, crossover, mutation are repeated until the GA converges as decided by the desired lowest value of fitness function, which is determined by the user. For this work, the GA was implemented using MATLAB (see Appendix).

4.3 Results

The genetic algorithm for the reconstruction of Bragg grating sensor strain distribution, was tested on simulated spectral data for three cases: (1) Linear strain distribution, (2) Discontinuous strain distribution, (3) Simulated experimental spectrum.

The parameters of the Bragg grating used for all the simulations in this work are given in table 4.1.

Table 4.1: Parameters of Bragg Grating Sensor

Parameter	Value
Length (L)	4 mm
Mode effective index of refraction (n_{eff})	1.46
Mean index variation ($\overline{\delta n_{eff}}$)	2.5 $\times 10^{-4}$
Initial period (Λ_o)	533.22 nm
Effective strain-optic constant (p_e)	0.26
Fringe visibility (ν)	1

For the first two cases, the simulated data considered to be "measured"data were generated using the Transfer matrix piecewise constant strain field assumption. This was done to distinguish between the performance of the GA and the performance of the T-matrix approximation. For all the cases, the GA was iterated beyond convergence and the optimal solution had a very low fitness value (of the order of 0.015).

4.3.1 Linear Strain Distribution

The reflected spectrum of the Bragg grating sensor subjected to linear strain distribution, $\epsilon_z = \{10^{-10}/m\}z$ is generated using T-matrix formulation and is plotted in figure 4.2. This generated reflectivity data is considered as the "measured"data for case 1. The reflected spectrum of the best individual (one with the lowest fitness value) obtained after a GA run of 300 generations is also plotted in figure 4.2. The parameters of the GA used to reconstruct the applied strain from the reflected spectrum are given in table 4.2 below.

A chosen subset of 31 points from the reflected spectrum of Bragg grating sensor, obtained through T-matrix simulation of grating response, is used as the input for the genetic algorithm to calculate the reflected spectrum of optimal solution. These points are

Table 4.2: Parameters of Genetic Algorithm for Case 1

Parameter	Value
Weight factor (a)	1.2
Mutation weight factor(B)	2
Crossover proportion (c)	0.15
Probability of reproduction (p_c)	0.45
Probability of mutation (p_m)	0.15
Number of generations (N)	300
Population size (η)	100
Number of grating segments (M)	8
Total number of spectral points	401
Number of chosen spectral points (H)	31

highlighted in figure 4.2. From the plots in figure 4.2, we observe that the reflected spactra match fairly well, however they differ at wavelength points away from principal peak.

Although the differences in the reflected spectrum obtained from the optimal GA solution as compared to the measured spectra from T-matrix formulation do not at first appear significant, these differences become more evident when the obtained strain distribution for the reflected spectra are compared. Figure 4.3 plots the applied linear strain distribution, the piecewise constant function used in the T-matrix approximation and the piecewise constant strain values calculated through the GA. From figure 4.3 we observe that the strain reconstruction obtained through the GA is not good at all . From both figures 4.2 & 4.3, we note that the GA can converge to a non-unique optimal solution for which the spectral data is reasonably good, however the obtained strain distribution is not good.

Further analysis revealed that although solutions with very low fitness values exist, the obtained strain distribution does not neccesarily match well with the applied linear strain distribution. The reason of this deviation is the selection of a limited subset of wavelengths chosen for comparison in equation 4.9, for the calculation of fitness values of individuals. After observing that the two spectra of figure 4.2 match well at the selected points, it was decided to improve the simulations by increasing the size of the wavelength subset. It should be noted that increasing the size of wavelengths chosen does not noticably increase the computational cost of the algorithm.

Based on these observations, we performed a GA using a broader wavelength set

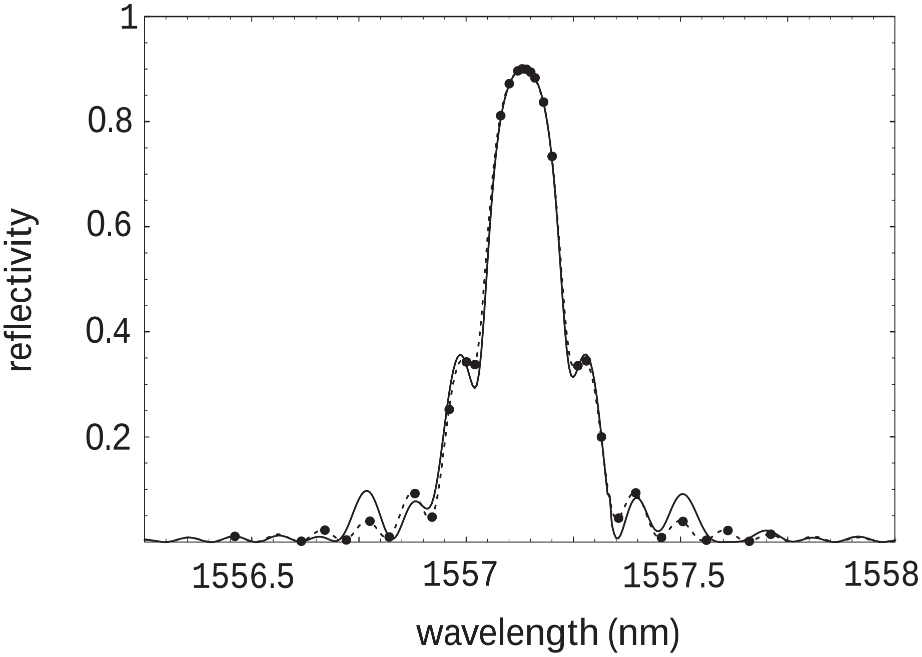

Figure 4.2: Reflected spectrum of Bragg grating subjected to the linear strain field $\epsilon_z = \{10^{-10}/m\}\ z$, obtained through T-matrix calculation of grating response (dashed line). Also plotted is the reflected spectrum of optimal solution from GA (solid line). Properties of grating are defined in table 4.1. Points used to calculate fitness function are highlighted.

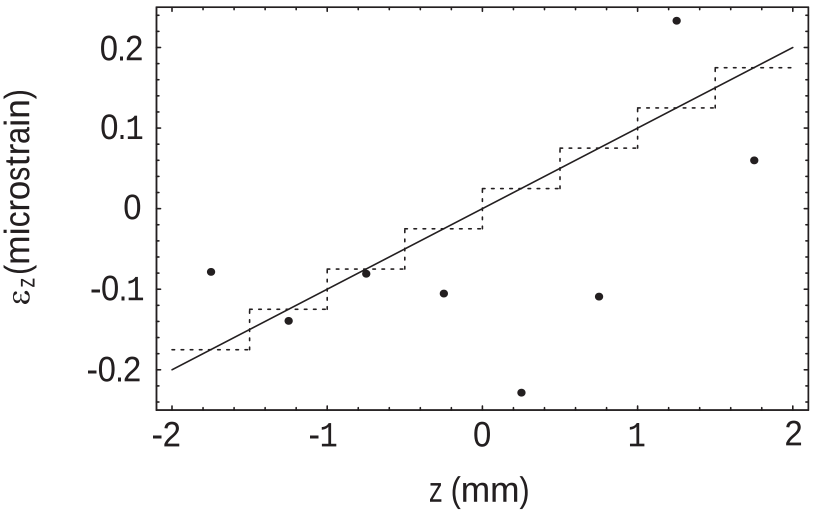

Figure 4.3: Axial strain distribution, ϵ_z: solid line plots applied linear strain distribution; dashed line piecewise constant approximation used as input for measured spectrum calculation; points represent strain in each segment obtained from GA optimal solution.

of 251 points. The same spectral data was used as input to the GA for the reconstruction of the strain distribution. In this case, only population size and number of generations for the GA were increased to prevent premature convergence of algorithm. The modified parameters of the GA used for this simulation are given in table 4.3.

Table 4.3: Modified parameters of Genetic Algorithm for Case 1 (as compared to table 4.2)

Parameter	Value
Number of generations (N)	350
Population size (η)	150

For the GA with modified parameters, the plots of the input reflected spectrum and the reflected spectrum of the optimal solution obtained through the GA match very closely as shown in figure 4.4. Not only do the reflected spectra match well in this case, but the reconstruction of the strain distribution is also excellent. The plots of obtained strain distribution and applied linear strain distribution are shown in figure 4.5.

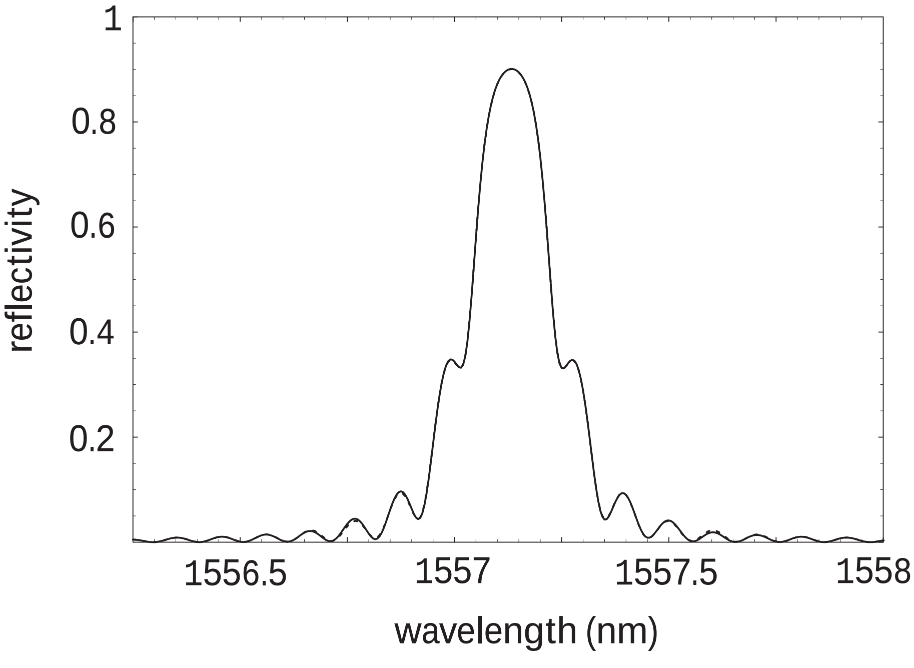

Figure 4.4: Reflected spectrum of optimal solution from GA with modified parameters of table 4.3 (solid line). Also plotted is input reflected spectrum of Bragg grating subjected to linear strain field (dashed line).

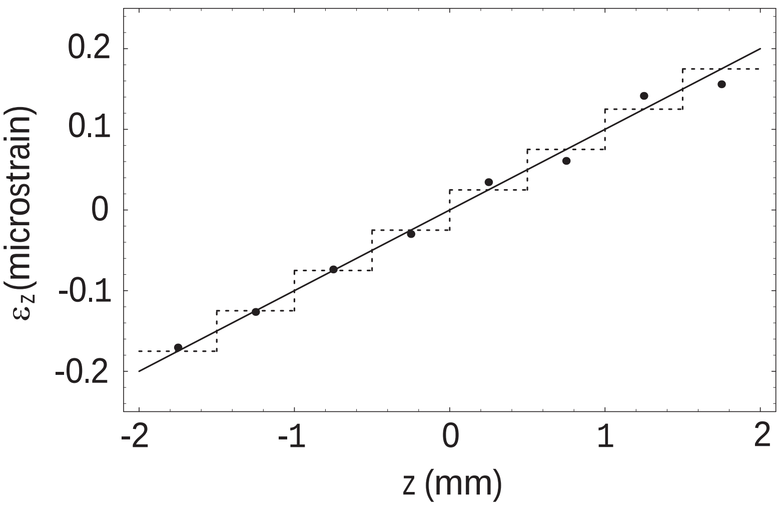

Figure 4.5: Axial strain distribution obtained from modified GA, ϵ_z: solid line plots applied linear strain distribution; dashed line piecewise constant approximation used as input for measured spectrum calculation; points represent strain in each segment obtained from GA optimal solution.

These results demonstrate the calculation of a linear strain field using the GA from the reflected spectrum data from the Bragg grating sensor. These calculations were performed without any assumed knowledge about the applied strain field and thus identify the linear strain distributions well.

4.3.2 Discontinuous Strain Distribution

We are even more interested to test the algorithm for identification of discontinuous strain fields, so an arbitrary strain field, as plotted in figure 4.6, is applied. The reflected spectrum of the Bragg grating subjected to this discontinuous strain field is obtained through T-matrix simulation of the grating, and is considered as the "measured"input spectrum for this case. The reflected spectrum is plotted as a dashed line in figure 4.7. The parameters of the GA used for this simulation are given in table 4.4.

The reflected spectrum of the optimal solution obtained through the GA is also plotted in figure 4.7. We note that both spectra are as closely matched for this case as for the case of figure 4.4. The applied and calculated strain distribution plots are given in figure 4.8. The reconstruction of the strain distribution in this case is not as good as that in case 1 (figure 4.5), however it is good enough to demonstrate the ability of algorithm for

Table 4.4: Parameters of Genetic Algorithm for Case 2

Parameter	Value
Weight factor (a)	1.2
Mutation weight factor(B)	2
Crossover proportion (c)	0.15
Probability of reproduction (p_c)	0.45
Probability of mutation (p_m)	0.15
Number of generations (N)	225
Population size (η)	150
Number of grating segments (M)	8
Total number of spectral points	401
Number of chosen spectral points (H)	251

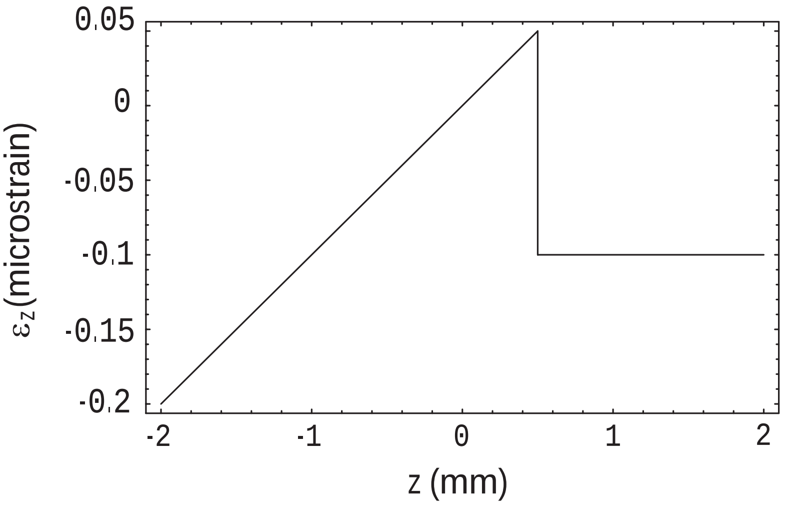

Figure 4.6: Discontinuous strain distribution, ϵ_z, applied to Bragg grating.

reconstruction of a discontinuous strain field without a-priori information about the applied strain field. Such a reconstruction has not been demonstrated before.

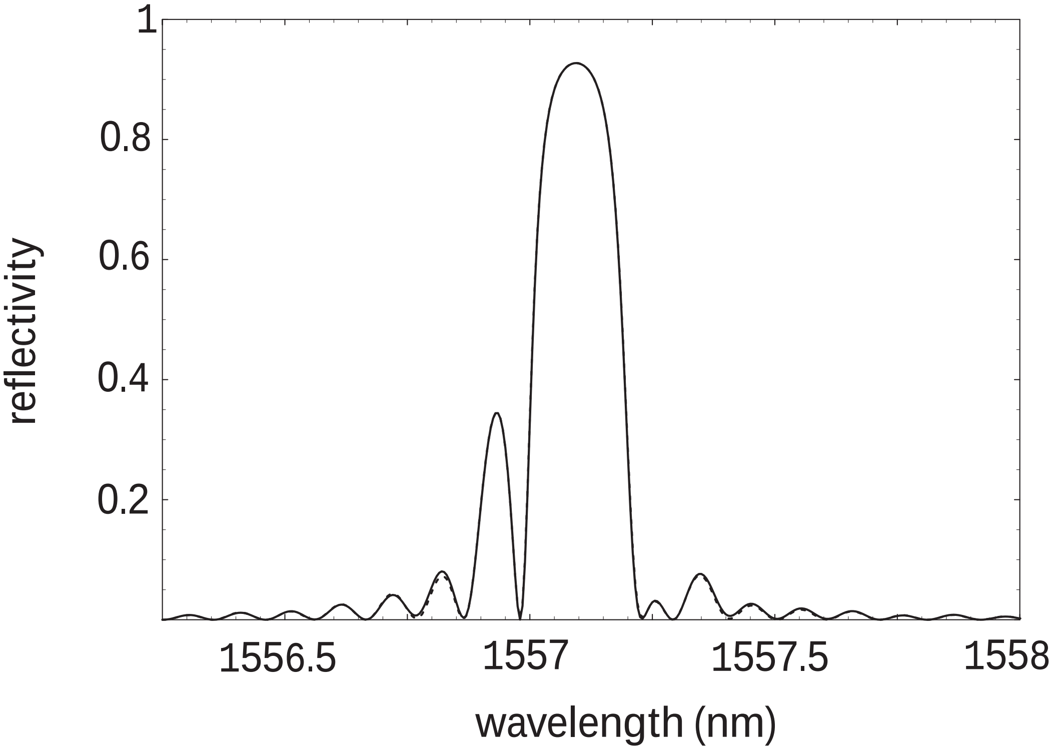

Figure 4.7: Reflected spectrum of Bragg grating subjected to the discontinuous strain field of figure 4.6, obtained through T-matrix calculation of grating response (dashed line). Also plotted is the reflected spectrum of optimal solution from GA (solid line). Properties of grating are defined in table 4.1.

4.3.3 Simulation of Sensor Experimental Data

Finally, the GA was also tested for sensor experimental data obtained by directly solving the coupled mode equations using the Runge-Kutta method. This method simulates the Bragg grating sensor to generate the "measured"spectrum for input to GA. This method gives an exact solution to coupled mode equations and does not involve any approximations as in the T-matrix method due to the piecewise constant strain assumption. Therefore, the result represents a more realistic grating spectrum. Figure 4.9 shows that the "measured"spectrum and the one obtained from the optimal solution of the GA are extremely close. The applied and obtained strain distributions are also plotted in figure 4.10. Similar

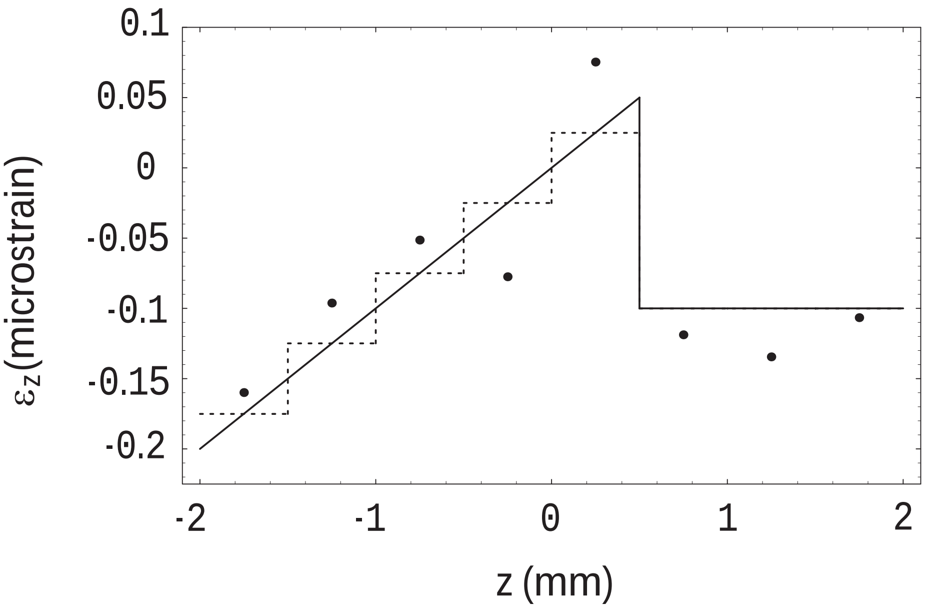

Figure 4.8: Discontinuous strain distribution, ϵ_z: solid line plots applied linear strain distribution; dashed line piecewise constant approximation used as input for measured spectrum; points represent strain in each segment obtained from GA optimal solution.

conclusions to those made about the previous case can be made for figures 4.9 and 4.10. This case tests the performance of the GA for data that cannot be matched exactly with the T-matrix approximation used (i.e. for a limited number of grating segments). Therefore, a similar behavior of the GA is expected for experimental data with noise.

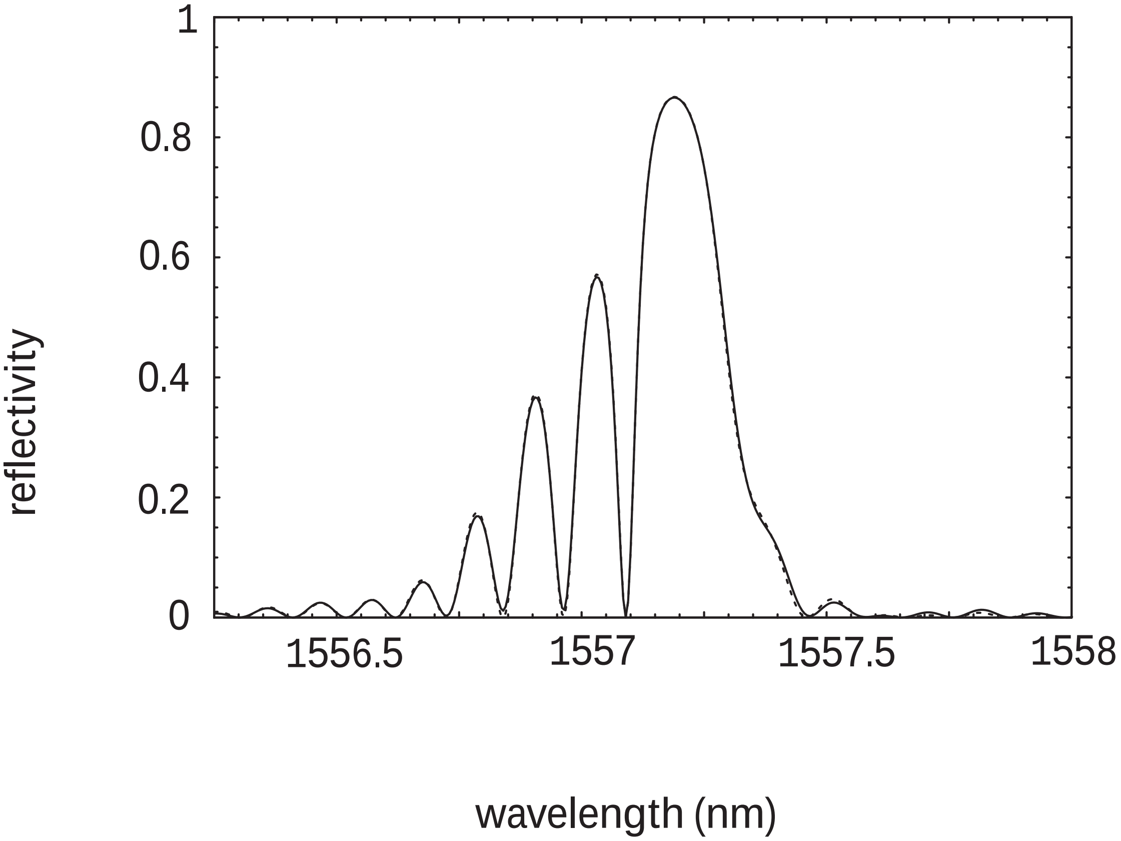

Figure 4.9: Reflected spectrum of Bragg grating subjected to discontinuous strain field of figure 4.10, obtained through numerical solution of Bragg grating coupled mode equations (dashed line). Also plotted is the reflected spectrum of optimal solution from GA (solid line).

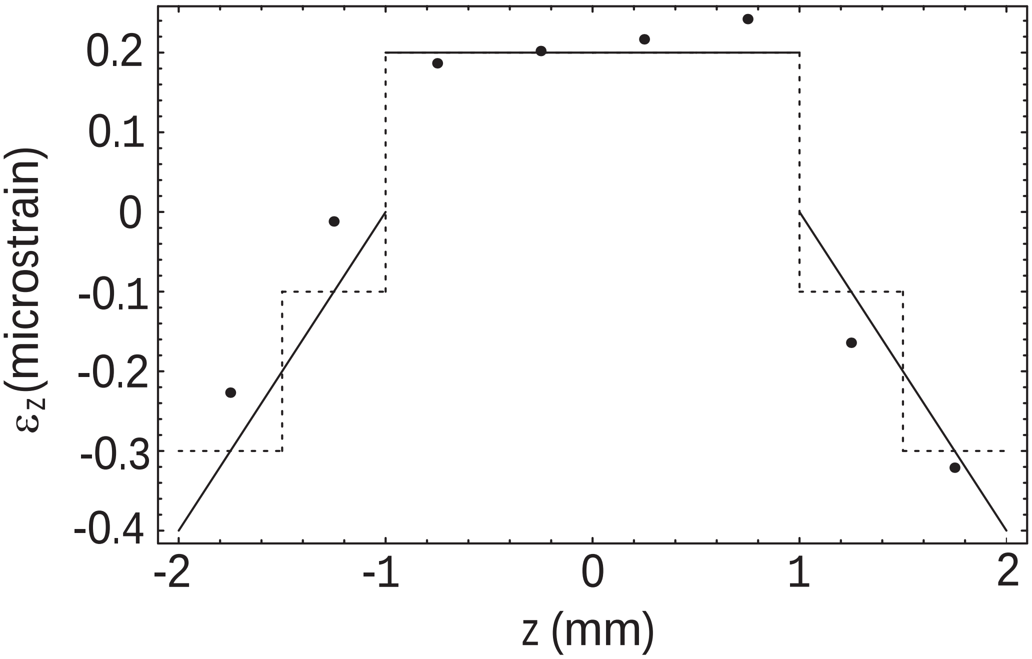

Figure 4.10: Applied strain distribution, ϵ_z: (solid line) applied linear strain distribution; (dashed line) piecewise constant approximation used as input for measured spectrum calculations; (points) strain in each segment obtained from GA optimal solution.

Chapter 5

Genetic Algorithm for Sensor Placement

This section describes an approach in which a genetic algorithm is used to determine the optimum sensor positions for a diagnostic system. Here we assume the existence of an effective damage identification procedure, and consider the problem of where the sensors should be placed for optimum efficiency of the detection system. In real world structural applications the number of sensors available are limited which thus requires the need for an efficient sensor optimization algorithm.

The recent past has seen considerable progress on the problem of determining the number and location of sensors for a particular application in an engineering structure. The sensor distribution optimization problem has been addressed predominantly in three areas: control engineering, system/modal identification and damage detection. However the least effort has been done for sensor placement problem for damage detection. A review of previous work in this area can be found in Worden and Burrows [22].

In this work we assume that an effective damage detection system, for example based on neural networks is available. The diagnostic system detects, localizes and assesses the damage which may be in the form of delamination, crack, buckling etc. The neural network is trained and tested using the experimental data from sensor measurements through signal processing procedures. However the fundamental issue regarding such diagnostic systems is which data should be processed and how the sensors should be placed. GAs being general purpose optimization algorithms are extremely suitable for such sensor placement problems. In this work we do not implement a neural network for damage detection but just assume that one is available so we do not go into the detail of neural network set up and training and testing data used. The GA based approach for the optimization of sensor placement also does not require any such knowledge. It only requires the information about the performance and percentage error of the neural network in localizing and assessing the damage as this data is used to form the fitness function of GA.

In this work the GA was tested using a fitness function based on an assigned weight matrix and is developed while setting up the problem to simulate the presence of an effective damage detection system. In this work for simplicity and intial testing of the GA, we considered that there exists a neural network based identification system for a plate with 20 potential sensor locations, as shown in figure 5.1. Though, after developing the GA, this algorithm could be applied to sensor placement problem for any desired engineering structure.

5.1 Implementation of GA

The steps involved in the implementation of GAs have been discussed in detail in the previous chapters of this thesis. However for this sensor optimization problem, the potential solutions and some of the GA operators are different due to the discrete nature of the problem. The potential solution for this problem is a vector of integers such that each integer specifies the position of sensor. For example the k^{th} individual of the population pool could be given as,

$$I^k = \{3, 7, 10, 14, 18\} \tag{5.1}$$

The above individual represents the sensor distribution such that sensors are present at locations 3, 7, 10, 14 & 18 on the plate out of twenty possible sensor locations. Note that the vector for each member of the population is ordered such that multiple vectors do not correspond to the same solution, e.g. {3, 7, 10, 14, 18} and {3, 14, 10, 7, 18}. As new members of the population are created through crossover and mutation, their vectors are

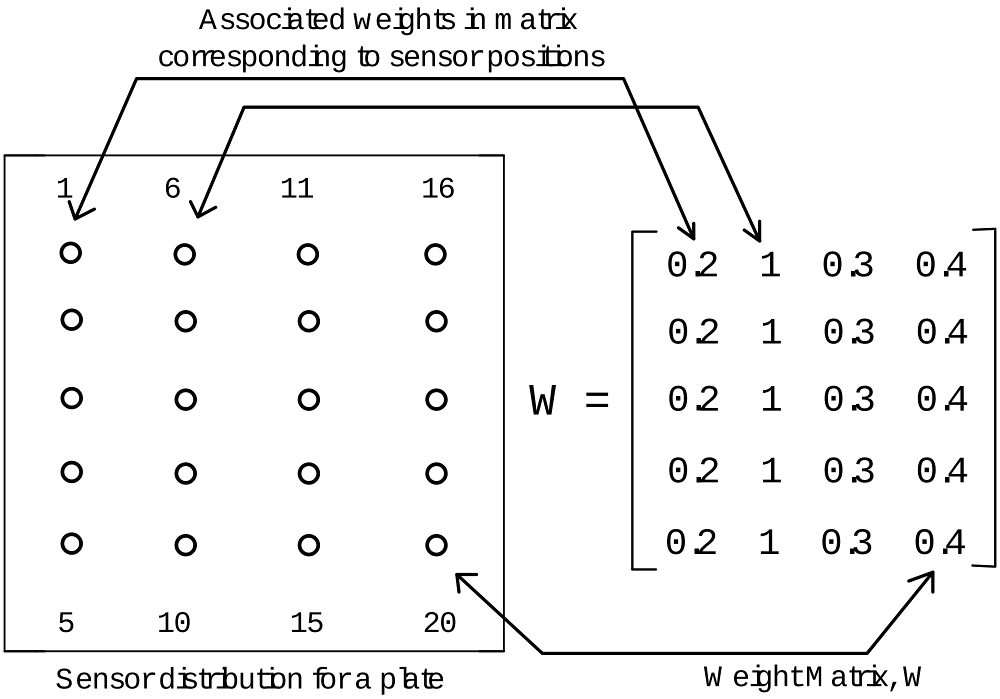

Figure 5.1: Sensor placement problem

ordered, before they are included in the population for the optimization procedure. The dimension of the individual can be set according to the subset desired (five in this case) during analysis for sensor optimization problem. So a gene in this representation (encoding) technique, refers to the position of sensor.

It was assumed in this work that a maximum number of sensors is given as a constraint for the optimization problem. Therefore, the optimal solution to the problem allowing five possible sensors could actually have only four or three sensor locations for example. Therefore an alternative approach was considered to allow these other solutions, allowing the vector I^k to vary in dimension from 1 to 5. This approach was not chosen however, for two reasons:

- it significantly complicated the operator of crossover within the GA;
- it is more efficient to perform the GA procedure over each search space (e.g. vectors of dimension 5, vectors of dimension 4 etc.) individually and then compare the optimal result of each one, than to perform the GA search over the entire search space.

An arbitrary weight function is used as the fitness function for this problem. A weight matrix was assigned to the sensor positions (locations) such that the position number of each sensor has a weight corresponding to the element number of the weight matrix as its position number, as shown in figure 5.1. The fitness function is the sum of respective weights of all the sensors for a particular individual. In this problem, the higher the fitness value better the individual. For example, the fitness function for the individual (potential solution) given in equation (5.1) is given as,

$$F(I^k) = W_3 + W_7 + W_{10} + W_{14} + W_{18} = 2.9 \tag{5.2}$$

The procedures of selection, crossover, mutation and elitism are applied similarly as in the case of Bragg grating sensor strain distribution problem. The modifications in this problem are that a new operator called the inversion operator is applied, discrete crossover is used instead of uniform arithmetical crossover and a penalty function is employed alongwith the fitness function to discard infeasible sensor configurations. Due to the randomness involved during intialization of the population, we may have individuals which repeat sensor locations (for example, $I = \{4, 4, 10, 12, 16\}$). To avoid such individuals from further participating in the search process we applied static penalty functions which decrease their fitness value by a constant and thus eventually lead to their elimination from future populations [23]. For infeasible solutions the penalized objective function would then be the unpenalized objective function minus a penalty (constant) for this maximization problem, since the search is driven by maximization of F. The value of the penalty (constant, C_1) depends on the deviation from a feasible configuration (which is five unique sensor locations here). For example, an individual $I = \{4, 4, 4, 9, 16\}$ with three unique sensors is penalized more as compared to an individual $I = \{4, 4, 9, 13, 16\}$ which has four unique sensors. A tuning of penalty constants is desirable for effective exploration of possible solutions.

In discrete crossover, for each position i of the first offspring we choose (with fixed probability, 0.5 for this case) the parent whose i^{th} gene will be transmitted to this descendent. The respective position of the second offspring will be completed with the value of corresponding gene from the other parent. For example,

$$P_1 = \{2, 5, 9, 11, 15\}$$
$$P_2 = \{3, 7, 10, 13, 18\}$$

$$O_1 = \{2, 5, 10, 11, 18\}$$
$$O_2 = \{3, 7, 9, 13, 15\}$$

The process of mutation is carried out using the Michalewicz non-uniform mutation operator as in the Bragg grating sensor strain distribution reconstruction problem. However, in this problem, a new operator called the inversion operator is used to improve the performance of the GA. This operator selects two genes on the chromosome string of an individual according to the user defined probability called the probability of inversion, p_{in} [22]. The values of these two genes are then mutually exchanged. This operator improved the convergence of the GA and is suitable for cases where discrete representations are used as for this sensor placement problem. Finally, elitism is adopted similar to the previous problem and all the steps are repeated until the GA converges.

5.2 Example Results

To demonstrate the performance of this GA implementation, the GA was applied to the sensor placement problem shown in figure 5.1. The parameters of the GA used are given in table 5.1. The GA converged after 100 generations and gave the optimal sensor distribution, one with the highest weight value obtained after adding weights associated with all the sensor positions. Convergence plots for the GA are given in figure 5.2. For this problem, the search progresses according to the maximization of the fitness function, as given in equation 5.2. It is shown that the GA converges to a solution (individual, I={6, 7, 8, 9, 10}) for which the fitness function is maximum i.e. $F = 5$ as controlled by chosen weight matrix. Since elitism is used, the convergence plot for the best individual shown in figure 5.2 does not give meaningful information after the optimal solution is present. Therefore, the convergenece plot for the mean fitness values of best 10 individuals is also plotted in figure 5.2 to ensure that the final population is dominated by optimal solutions.

Table 5.1: Parameters of Genetic Algorithm for Sensor Placement Problem

Parameter	Value
Probability of crossover (p_c)	0.45
Probability of mutation (p_m)	0.15
Probability of inversion (p_{in})	0.01
Number of generations (N)	100
Population size (η)	50
Weight factor (a)	1.2
Mutation weight factor(B)	2

The optimal solution was validated for the chosen weight matrix. This demonstrates the efficiency of the GA to determine the optimum sensor placement. Now by modifying the fitness function, this GA can be applied to neural network based damage identification system. For each individual (five sensor distribution), a neural network diagnostic can be trained and the percentage error in predicting the damage can be evaluated over a testing set. The inverse of this probability of misclassification can used as fitness function to drive the sensor placement optimization problem for a particular case. The training data for each network would be generated by restricting the full 20-sensor distribution to the subset specified by the five sensors defining a given individual. The above

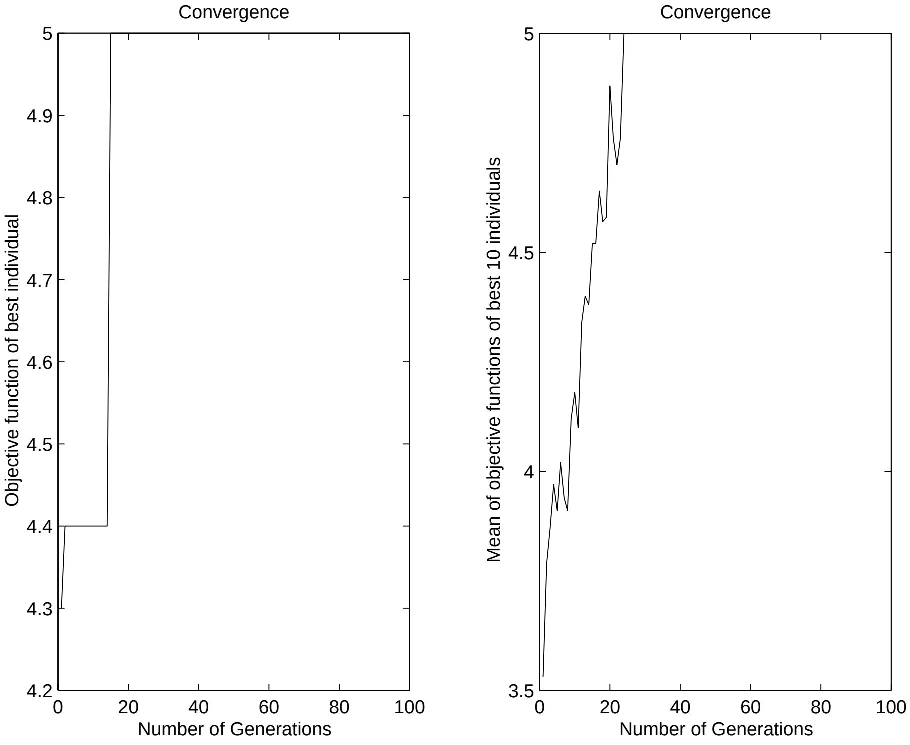

Figure 5.2: Convergence of GA for sensor placement problem

presented results allows the GA to be used for optimal placement of sensors, in an existing damage identification system.

Chapter 6

Conclusions

This thesis presents a genetic algorithm for the inversion of Bragg grating sensor spectral data to reconstruct the applied strain distributions. An efficient GA algorithm in conjunction with the T-matrix formalism is presented for the interrogation of optical fiber Bragg grating strain sensors. The GA method easily analyzes the spectra due to highly non-linear and discontinuous strain fields. The representation technique used in this method efficiently models even highly non-uniform and discontinuous strain distributions as a period distribution along the grating. This feature has been addressed in the results from simulations for the applied strain distributions. The GA algorithm permits the use of Bragg grating sensors for damage identification especially near the location of cracks, disbond or other singular features, and for structures with irregular geometries. Although in this work only reflection measurements were used, the algorithm can also be applied to transmitted spectra from Bragg grating sensors. Also, the developed method does not require any phase information about the reflected spectra of Bragg grating, thus eliminating the need for costly equipment and stable measurement environments which is otherwise required for the measurement of phase distributions.

The minimum number of segments required to model the applied strain distribution does not need to be known a-priori but can be determined from the convergence of the GA solution to the measured spectrum. Other than the increased number of segments required, the GA itself is not numerically less efficient for highly non-uniform strain fields than for smooth, weakly varying strain fields. Initial testing of the algorithm has demonstrated its senstivity to the wavelength subset chosen for comparison. Further work is required to determine the appropriate selection of wavelength points and their relative weight in the fitness function. The resolution of this issue and optimization of the GA parameters is required before the algorithm can be applied to experimental sensor data. The efficieny of the method is demonstrated through the shown reconstructions of Bragg grating sensor simulated data without any assumed knowledge of the form of applied strain fields.

In second application, the implementation of the GA search procedure for the optimization of sensor locations for a particular structural application is presented. The performance of the GA is evaluated using a test weight function and assuming the existence of an effective diagnostic system. The combination of the developed GA with already established damage identification system to find the optimal sensor distribution is demonstrated. Given the existence of a diagnostic system, the developed GA with a modified fitness function can be implemented for damage detection of a desired structural application. A generic method is proposed for optimizing the efficiency of the existing diagnostic system.

Bibliography

[1] J. H. Holland, “Adaption in Natural and Artificial Systems”, *The University of Michigan Press*, Ann Arbor, MI, 1975.

[2] David E. Goldberg, “Genetic Algrithms in Search Optimization and Machine Learning”, *Addison Wesley Publishing Company, Inc.*, MA, 1989.

[3] D. Whitley, “The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of reproductive Trials is Best”, *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Los Altos, CA, 1989.

[4] Zbigniew Michalewicz, “Genetic Algorithms + Date Structures = Evolution Programs”, *Springer Verlag*, Berlin, 1992.

[5] T. Bäck, H. Schwefel, “An Overview of Evolutionary Algorithms for Parameter Optimization”, *Evolutionary Computation*, Vol. 1, 1993, pp. 1-23.

[6] R. Kashyap, “Fiber Bragg gratings”, *Academic Press*, 1999.

[7] K. O. Hill, G. Meltz, “Fiber Bragg Grating Technology Fundamentals and Overview”, *Journal of LIghtwave Technology*, Vol. 15, No. 8, pp. 1263-1276, 1997.

[8] S. Huang, M. Ohn, M. LeBlanc, R. M. Measures, “Continuous arbitrary strain profile measurements with fiber Bragg gratings”, *Smart Materials and Structures*, Vol. 7, pp. 248-256, 1998.

[9] W. W. Morey, G. Meltz, W. H. Glenn, “Fiber Bragg grating sensors”, *Proc. SPIE Fiber Optic and Laser Sensors*, Vol. 1169, pp. 98-107, 1989.

[10] T. Erdogan, "Fiber grating spectra", *Journal of Lightwave Technology*, Vol. 15, pp. 1277-1294, 1977.

[11] G. H. Song, S. Y. Shin, "Design of corrugated waveguide filters by Gelfand-Levitan-Marchenko inverse scattering method", *J. Opt. Soc. Amer.*, Vol. 2, pp. 1905-1915, 1995.

[12] E. Peral, J. Capmany, J. Marti, "Iterative solution to the Gelfand-Levitan-Marchenko coupled equations and applications to synthesis of fiber gratings", *IEEE Journal of Quantum Electronics*, Vol. 32, pp. 2078-2084, 1996.

[13] R. Feced, M. Zervas, M. Muriel, "An Efficient Inverse Scattering Algorithm for the Design of Nonuniform Fiber Bragg Gratings", *IEEE Journal of Quantum Electronics*, Vol. 35, pp. 1105-1115, 1999.

[14] J. Skaar, K. M. Risvik, "A Genetic Algorithm for the Inverse Problem in Synthesis of Fiber Gratings", *Journal of Lightwave Technology*, Vol. 16, pp. 1928-1932, Oct. 1998.

[15] G. Cormier, R. Boudreau, S. Thériault, "Real-coded Genetic algorithm for Bragg grating parameter synthesis", *Journal of the Optical Society of America B*, Vol. 18, pp. 1771-1776, Dec. 2001.

[16] F. Casagrande, P. Crespi, A. M. Grassi, A. Lulli, R. P. Kenny, M. P. Whelan, "From the reflected spectrum to the properties of a fiber Bragg grating: a genetic algorithm approach with application to distributed strain sensing", *Applied Optics*, Vol. 41, pp. 5238-5244, Sept. 2002.

[17] M. Yamada, K. Sakuda, "Analysis of almost-periodic distributed feedback slab waveguides via a fundamental matrix approach", *Applied Optics*, Vol. 26, pp. 3474-3478, 1987.

[18] K. Peters, M. Studer, J. Botsis, A. Locco, H. Limberger, R. Salathé, "Embedded Optical Fiber Bragg Grating Sensor in a Nonuniform Strain Field: Measurements and Simulations", *Experimental Mechanics*, Vol. 41, March, 2001, pp. 19-28.

[19] S. Huang, M. LeBlanc, M. M. Ohn, R. M. Measures, "Bragg intragrating structural sensing ", *Applied Optics*, Vol. 34, pp. 5003-5009, 1995.

[20] L. Davis, "Handbook of Genetic Algorithms", *Van Nostrand Reinhold*, New York, 1991.

[21] A. H. Wright, "Genetic algorithms for real parameter optimization", *Foundations of Genetic Algorithms*, pp. 205-218, 1991.

[22] K. Worden, A. P. Burrows, "Optimal sensor placement for fault detection ", *Engineering Structures*, Vol. 23, pp. 885-901, 2001.

[23] A. E. Smith, D. W. Coit, "Constraint Handling Techniques - Penalty Functions", *Handbook of Evolutionary Computation*, Chapter C5.2, 1997.

Appendix A

MATLAB code for GA implementation

```
%*******************************************************************
% GA code for reconstruction of applied strain distributions from
% reflected spectrum of fiber Bragg grating sensor
%*******************************************************************
neff=1.46;              % effective refractive index
fv=1;                   % fringe visibility
l=4*(10^6);             % length of grating
delz=5*(10^5);
delnbar=2.5*(10^(-4));
individuals=150         % Number of individuals
lambda2=1559;
lambda1=1555;           % wavelength range
step=.01;
segments=(l/delz);
points=((lambda2-lambda1)/step)+1;
ref=zeros(individuals,points);
a=1.2;
pi=3.14159265358979;
k=1:points;
wave=lambda1+(k*(step));
pop=rand(individuals,segments)+533;        % INITIALIZATION
T=input('Enter number of generations=');   % Number of generations
```

```
mvector12      % calls another file of "measured" reflectivity vector
%(code for mvector12 given below)

%p=90:345;     % Wavelength points chosen (31 points)
%p=[92 123 134 144 155 164 176 184 192 200 204 216 220 224 226 228...
%230 232 236 240 252 256 263 271 279 291 301 312 322 332 342];
p=50:349;               % Wavelength subset (251 points)
for GEN=1:T
Generation=GEN

for j=1:individuals     % number of individuals

    for k=1:points      % number of wavelength points on spectra
        lambda=wave(k);

        f1=[1 0;0 1];

        for i=1:segments

sig=(((2*pi*neff)/lambda)-(pi/pop(j,i))+((2*pi*delnbar)/lambda));
gammab=((((pi*fv*delnbar)/lambda)^2-((sig)^2)))^.5;
f11=(cosh(gammab*delz)-sqrt(-1)*(sig/gammab)*sinh(gammab*delz));
f22=conj(f11);
f21=(((sqrt(-1)*pi*fv*delnbar)/(gammab*lambda))*sinh(gammab*delz));
f12=conj(f21);
f1=f1*[f11 f12;f21 f22];
    end

P=f1*[1;0];
po=P(1,1);
no=P(2,1);
ref(j,k)=(abs(no/po))^2;

    end
end

for j=1:individuals

    refc=ref(j,[p]);                % Reflectivity computed

    OBJFUN(j)=sum((refm-refc).^2); % OBJECTIVE FUNCTION

end
```

```
OBJFUN;
CI=sort(OBJFUN);

for j=1:individuals % Finding the rank of each individual in that pool
for h=1:individuals
        if OBJFUN(j)==CI(h)
            seq(j)=h;   % Gives rank of an individual in that pool
        else
        end
    end
end

Best(1,GEN)=min(OBJFUN);
K=CI;
M(1,GEN)=mean(K([1:10]));
% Mean of objective functions of 10 best individuals
Worst(1,GEN)=max(OBJFUN);
for j=1:individuals
    if seq(j)==min(seq)
        note(1,GEN)=j;
    else
    end
end
if GEN~=1
if Best(1,GEN-1)<Best(1,GEN)                  % ELITISM
    Best(1,GEN)=Best(1,GEN-1);
    pop([note(1,GEN)],:)=pop([note(1,GEN-1)],:);
else
end
else
end


for j=1:individuals
    RW(j)=((individuals)-(j-1))^a;     % Assigning relative weights
end
SEQRW=RW([seq]);      % Relative weights of individuals in sequence

for j=1:individuals
    PR(j)=(SEQRW(j)/((1/individuals)*sum(SEQRW)));
end

PR;                                % Probabilities of individuals
int=floor(PR);         % SELECTION ( stochastic remainder selection)
```

```
S=sum(int);
RP=[];

for j=1:individuals
    if int(j)~=0
        s=j;
        t=int(j);
%Generate members by repeating according to respective probabilities
        RP=[RP; repmat(pop(s,:),t,1)] ;

    else
    end
end
RP;
UO=individuals-S;    % unoccupied slots
TS=sum(OBJFUN);      % total fitness
CS=cumsum(OBJFUN);   % running total
RR=rand(1,UO)*TS;
% Random number generation between 0 and total fitness
e=[];

for h=1:UO           % Roulette wheel procedure on remaining pool
    for j=1:individuals
        if RR(h)<=CS(j)
            e(h)=j;
        break
        else
        end

    end
end

e;
RRP=pop([e],:);
RP=[RP;RRP];            % Reproduction pool
pc=0.45;                % Probability of crossover
RC=[];
RN=rand(1,individuals); % Generate random numbers for all individuals
xpool=[];
for j=1:individuals
    if RN(j)<pc
        RC=[RC j];
% Gives individual # that undergo crossover(eg.6th, 11th etc)
    else
    end
end
xpool=size(RC,2)
RPC=[];
if RC~=[]
    RPC=RP([RC],:);    % Pool of individuals that undergo crossover
    RP([RC],:)=[];
    aa=0.15;
    o=[];              % Offspring pool


for g=1:2:size(RC,2)   % Uniform Whole Arithmetic crossover (Wright's)
    if g==size(RC,2)
        o=[o; RPC(g,:)];
    else
        o=[o; ((aa*(RPC(g,:)))+((1-aa)*(RPC((g+1),:))));...
((1+aa)*(RPC(g,:)))-((aa*(RPC((g+1),:)))); (((-aa)*RPC(g,:))...
+((1+aa)*RPC((g+1),:)))];
   end
end
o;

TMAT12 % (code is given below under TMAT12)
%CALLS ANOTHER FUNCTION which evaluates offspring pool and modifies it

RP=[oo;RP];
% Gives modified reproduction pool including modified offspring pool
else
    RPC=RP;
end

pm=0.15;                % Proability of mutation
amp=[];
xx=[];
yy=[];
B=2;
c=1;
RG=rand(individuals,segments);
% Generate random numbers for all segments
for j= 1:individuals
    for i=1:segments
        if RG(j,i)<=pm
           xx(c)=j;
           yy(c)=i;
           c=c+1;

        else
        end
    end
end
xx;
% Gives individual number to be mutated, in a reproduction pool
yy ;
% Gives segment number in an individual to be mutated

RPG=[];
UB=[];
LB=[];

for j=1:size(xx,2)             % NON UNIFORM MUTATION
    RPG(j)=RP(xx(j),yy(j));
    UB(j)=max(RP(:,yy(j)));
% Upper bound for particular segment in all the individuals
    LB(j)=min(RP(:,yy(j)));
% Lower bound for particular segment in all the individuals
end
[RPG;UB;LB];
y=[];

for j=1:size(xx,2)
    coin=rand(1);
    toss=round(coin);
    if toss==0                           % i.e Heads
        y=UB(j)-RPG(j);
        amp=(y*coin*(1-(GEN/T)))^B;
        RPG(j)=RPG(j)+amp;
    else                                  % tails
        y=RPG(j)-LB(j);
        amp=(y*coin*(1-(GEN/T)))^B;
        RPG(j)=RPG(j)-amp;
    end
end
RPG;                    % Gives mutated segments

for j=1:size(RPG,2)
RP(xx(j),yy(j))=RPG(j);
% Modify the reproduction pool by replacing the mutated segments
end
```

```
pop=RP          % FINAL REPRODUCTION POOL AFTER T th generation
SUPER=pop(note(1,GEN),:)
end

for j=1:individuals
    subplot(15,10,j)
    plot(wave,ref(j,:));

end

SUPER=pop(note(1,T),:)
% Gives period vector of Best Individual, in a pool

NGEN=1:T;

figure
subplot(1,2,1)
plot(NGEN,Best),title('Convergence')
xlabel('Number of Generations')
ylabel('Objective function of best individual')

subplot(1,2,2)
plot(NGEN,M),title('Convergence')
xlabel('Number of Generations')
ylabel('Mean of objective functions of best 10 individuals')

figure
plot(NGEN,Worst),title('Convergence')
xlabel('Number of Generations')
ylabel('Objective function of worst individual')

figure
plot(wave,ref(note(1,T),:),wave,REF,'r:')
xlabel('Wavelength')
ylabel('Reflectivity')

M
%Mean of objective functions of 10 best individual for each generation
Best                      % Best objective functions for T generations
Worst                    % Worst objective functions for T generations
OBJFUN
% Objective functions of individuals(for pool) after T generations
pop                                   % Population after T generations
```

```
%***********************mvector12***************************************
% Reflected spectrum through T-matrix calculation considered as
% as "measured" spectrum to GA.
%***********************************************************************
neff=1.46
lambdab=1557;  % Bragg wavelength
fv=1;          % Fringe visibility
pe=0.26;       % Photoelastic constant
l=4*(10^6);    % Length of grating
delz=5*(10^5);
delnbar=2.5*(10^(-4));
%p=[92 123 134 144 155 164 176 184 192 200 204 216 220 224 226 228
%230 232 236 240 252 256 263 271 279 291 301 312 322 332 342];
% columns chosen
p=50:349;      % Wavelength points chosen
count=1;
for lambda=1555:.01:1559
    f1=[1 0;0 1];
    for i=1:8
    periodn=[533.1501 533.1699 533.1896 533.2093 533.2290...
            533.2488 533.2685 533.2882];

sig=(((2*pi*neff)/lambda)-(pi/periodn(i))+((2*pi*delnbar)/lambda));
gammab=((((pi*fv*delnbar)/lambda)^2-((sig)^2)))^.5;
f11=(cosh(gammab*delz)-sqrt(-1)*(sig/gammab)*sinh(gammab*delz));
f22=conj(f11);
f21=(((sqrt(-1)*pi*fv*delnbar)/(gammab*lambda))*sinh(gammab*delz));
f12=conj(f21);
f1=f1*[f11 f12;f21 f22];
end

P=f1*[1;0];
po=P(1,1);
no=P(2,1);

REF(count)=(abs(no/po))^2;
c(count)=(lambda/lambdab);
wave(count)=lambda;
count=count+1;
end
refm=[REF(p)];   % Measured reflectivity data
```

```
%***************************TMAT12*************************************
% Function to evaluate the offspring pool
%**********************************************************************
OFUN=[];

for j=1:size(o,1)        % Offspring pool

    for k=1:points       % number of points on spectra
        lambda=wave(k);

        f1=[1 0;0 1];

        for i=1:segments

sig=(((2*pi*neff)/lambda)-(pi/o(j,i))+((2*pi*delnbar)/lambda));
gammab=((((pi*fv*delnbar)/lambda)^2-((sig)^2)))^.5;
f11=(cosh(gammab*delz)-sqrt(-1)*(sig/gammab)*sinh(gammab*delz));
f22=conj(f11);
f21=(((sqrt(-1)*pi*fv*delnbar)/(gammab*lambda))*sinh(gammab*delz));
f12=conj(f21);
f1=f1*[f11 f12;f21 f22];
end

P=f1*[1;0];
po=P(1,1);
no=P(2,1);
ref(j,k)=(abs(no/po))^2;

end
end
for s=1:size(o,1)

    refc=ref(s,[p]);                 % Reflectivity computed

    OFUN(s)=sum((refm-refc).^2);     % OBJECTIVE FUNCTION

end
OFUN;
SEQ=[];
SEQV=[];
SEQN=[];
SEQNEW=[];
ci=sort(OFUN);
for q=1:size(o,1)
```

```
    for r=1:size(o,1)
        if OFUN(q)==ci(r)
            SEQ(q)=r;
        else
        end
    end
end
SEQ;
SEQV=sort(SEQ);
for q=1:size(o,1)
    for r=1:size(o,1)
        if SEQV(q)==SEQ(r)
            SEQN(q)=r;
        else
        end
    end
end
SEQN;
SEQNEW=SEQN([1:size(RPC,1)]);
oo=[];
oo=o([SEQNEW],:);          % Gives modified offspring pool

%********************************************************************%
```

Appendix B

MATLAB code for GA implementation to Sensor Placement Problem

```
%*********************************************************************
%  GA code for Sensor Placement Problem
%*********************************************************************
individuals=50;
sensors=5;
tsensors=20;           % Possible sensor locations
W=transpose(
[ .2 .2 .2 .2 .2 ; 1 1 1 1 1 ; .3 .3 .3 .3 .3 ; .4 .4 .4 .4 .4]) ;
% Chosen weight matrix
a=1.2;
population=(ceil(rand(individuals,sensors)*tsensors));
T=100;                 % Number of generations
C1=1;
C2=2.5;                % Penalty constants
for i=1:individuals
    pop(i,:)=sort(population(i,:));
end
pop;
```

```
for i=1:individuals
    check(i)=size(unique([pop(i,:)]),2);
end
check;
cc=[];

for v=1:individuals
if check(v)<sensors
    cc=[cc v];
else
end
end
pop(cc,:)=[];
pop;

c=size(cc,2);

while c~=0
    check=[];
    popi=(ceil(rand(c,sensors)*tsensors));
    for i=1:c
    popi(i,:)=sort(popi(i,:));
end

for i=1:c
    check(i)=size(unique([popi(i,:)]),2);
end
d=c;
c=[];
for v=1:d

    if check(v)<sensors
    c=[c v];
else
end
end
popi(c,:)=[];
pop=[pop;popi];
c=size(c,2);
end
pop;                                  %INITIAL population

for GEN=1:T
```

```
    Generation=GEN

if GEN==1
    for i=1:individuals               % Objective function
    OBJFUN(i)=sum(W(pop(i,[1:sensors])));
    end
else
    OBJFUN;
end
CI=fliplr(sort(OBJFUN));
% Finding the rank of each individual(sensor placement) in that pool
for j=1:individuals
    for h=1:individuals
        if OBJFUN(j)==CI(h)
            seq(j)=h;
% Gives rank of an individual in that pool
        else
        end
    end
seq;




for j=1:individuals
    RW(j)=((individuals)-(j-1))^a;       % Assigning relative weights
end
SEQRW=RW([seq]) ;       % Relative weights of individuals in sequence

for j=1:individuals
    PR(j)=(SEQRW(j)/((1/individuals)*sum(SEQRW)));
end

PR  ;                            % Probabilities of Individuals
int=floor(PR);                   % SELECTION
S=sum(int);
RP=[];

for j=1:individuals
    if int(j)~=0
        s=j;
        tt=int(j);
        RP=[RP; repmat(pop(s,:),tt,1)] ;
% Generate members by repeating according to respective probabilities
```

```
    else
    end
end
RP ;

UO=individuals-S ;   % Unoccupied slots
TS=sum(OBJFUN);      % Total fitness
CS=cumsum(OBJFUN);   % Running total
RR=rand(1,UO)*TS;
% Random number generation between 0 and total fitness
e=[];

for h=1:UO           % Roulette wheel procedure on remaining pool
    for j=1:individuals
        if RR(h)<=CS(j)
            e(h)=j;
        break
        else
        end

    end
end

e;
RRP=pop([e],:);
RP=[RP;RRP];
RP  ;                 % INTERMEDIATE POPULATION

pc=0.45;              % Probability of crossover
RC=[];
RN=rand(1,individuals); % Generate random numbers for all individuals
xpool=[];
for j=1:individuals
    if RN(j)<pc
        RC=[RC j];      % Gives individual # that undergo crossover
    else
    end
end
RC;
xpool=size(RC,2);
RPC=[];

RPC=RP([RC],:) ;        % MATING POOL
RP([RC],:)=[];
```

```
o=[];

P=rand(size(RC,2),sensors);

for i=1:2:size(RC,2)      %   DISCRETE CROSSOVER
    if i==size(RC,2)
        o=[o; RPC(i,:)];
    else
        for j=1:sensors
            if P(i,j)>=.5
                o(i,j)=RPC(i,j);
                o(i+1,j)=RPC(i+1,j);
            else
                o(i,j)=RPC(i+1,j);
                o(i+1,j)=RPC(i,j);
            end
        end
    end
end
o  ;              % OFFSPRING POOL
pool=[o;RP];


pm=0.15;          % Proability of mutation
amp=[];
xx=[];
yy=[];
B=5;
w=1;
RG=rand(individuals,sensors);
% Generate random numbers for all segments
for j= 1:individuals
    for i=1:sensors
        if RG(j,i)<=pm
           xx(w)=j;
           yy(w)=i;
           w=w+1;

        else
        end
    end
end
xx;
```

```
% Gives individual number to be mutated, in a reproduction pool
yy ; % Gives bit number in an individual to be mutated in a pool

RPG=[];
UB=[];
LB=[];

for j=1:size(xx,2)             % NON UNIFORM MUTATION
    RPG(j)=pool(xx(j),yy(j));
    UB(j)=max(pool(:,yy(j)));
% Upper bound for particular bit(sensor) in all the individuals
    LB(j)=min(pool(:,yy(j)));
% Lower bound for particular bit(sensor) in all the individuals

end
[RPG;UB;LB];
y=[];

for j=1:size(xx,2)
    coin=rand(1);
    toss=round(coin);
    if toss==0                           % i.e Heads
        y=UB(j)-RPG(j);
        amp=ceil(y*(1-coin^((1-(GEN/T))*B)));
        RPG(j)=RPG(j)+amp;
    else                                  % tails
        y=RPG(j)-LB(j);
        amp=ceil(y*(1-coin^((1-(GEN/T))*B)));
        RPG(j)=RPG(j)-amp;
    end
end
RPG;                    % Gives mutated bits

for j=1:size(RPG,2)
pool(xx(j),yy(j))=RPG(j);
% Modify the reproduction pool by replacing the mutated segments
end
pool;

pinv=0.01;                 % Probability of inversion
RI=[];
RINV=rand(1,individuals);% Generate random numbers for all individuals
invpool=[];
for j=1:individuals
```

```
    if RINV(j)<pinv
        RI=[RI j];        % Gives individual # that undergo crossover
    else
    end
end
RI;
invpool=size(RI,2);

for i=1:invpool                                % INVERSION OPERATOR
    invert=ceil(rand(1,1)*individuals);
    invertb=ceil(rand(1,2)*sensors);
    bits=pool(invert,[invertb]);
    pool(invert,[invertb])=[fliplr(bits)];
end
pool;


for i=1:individuals
    check(i)=size(unique([pool(i,:)]),2);        % Penalty function
    if check(i)==sensors
       OBJFUN(i)=sum(W(pool(i,[1:sensors])));
   end
   if check(i)==sensors-1
        OBJFUN(i)=sum(W(pool(i,[1:sensors])))-C1;
    end
    if check(i)<=sensors-2
        OBJFUN(i)=sum(W(pool(i,[1:sensors])))-C2;
    end
end
pop=pool;
OBJFUN;
K=fliplr(sort(OBJFUN));
% Finding the rank of each individual(sensor placement) in that pool
for j=1:individuals
    for h=1:individuals
        if OBJFUN(j)==K(h)
            SEQ(j)=h;
% Gives rank of an individual in that pool
        else
        end
    end
end
SEQ;
```

```
Best(1,GEN)=max(OBJFUN);
K=fliplr(sort(OBJFUN));
M(1,GEN)=mean(K([1:10]));
% Mean of objective functions of 10 best sensor placements
Worst(1,GEN)=min(OBJFUN);
for j=1:individuals
    if SEQ(j)==min(SEQ)
        note(1,GEN)=j;
    else
    end
end

if GEN~=1
if Best(1,GEN-1)>Best(1,GEN)                   % ELITISM
    Best(1,GEN)=Best(1,GEN-1);
    pop([note(1,GEN)],:)=pop([note(1,GEN-1)],:);
    OBJFUN(note(1,GEN))=OBJFUN(note(1,GEN-1))
else
end
else
end

%for i=1:individuals
   % pop(i,:)=sort(pop(i,:));
    %end

pop;
OBJFUN;
SENSOR=pop(note(1,GEN),:) % Best sensor configuration after each gen.
end

for i=1:individuals
    pop(i,:)=sort(pop(i,:));
end


pop
OBJFUN
SENSOR=pop(note(1,T),:)
% Best sensor configuartion in a pool,obtained after T generations
group=[];
for i=1:individuals
    if OBJFUN(i)==max(OBJFUN)
        group=[group;pop(i,:)];
    else
    end
end
groupsize=size(group,1)

NGEN=1:T;

figure
subplot(1,2,1)
plot(NGEN,Best),title('Convergence')
xlabel('Number of Generations')
ylabel('Objective function of best individual')

subplot(1,2,2)
plot(NGEN,M),title('Convergence of GA for sensor optimization problem')
xlabel('Number of Generations')
ylabel('Mean of objective functions of best 10 individuals')

%**********************************************************************%
```