

## **Genetic Algorithm Approach for Placement Optimization of FBG Sensors for a Diagnostic System**

**Mr. Deba Kumar Mahanta**

*Department of Electronics and Communication Engineering, Assam Don Bosco University, Assam, India*

---

**Abstract:** *The efficiency of a diagnosis system depends on the relevance of the information it can be retrieve from the diagnosed plant. The diagnostic system detects, localizes and assesses the damage which may be in the form of delimitation, crack, buckling etc. Optical sensors like Fiber Bragg Grating sensor may be used to determine the strain, temperature and other diagnostic parameters. In real world structural applications the numbers of sensors available are limited which thus requires the need for an efficient sensor optimization algorithm. This paper describes an approach in which a genetic algorithm is used to determine the optimum sensor positions for a diagnostic system. The fundamental issue regarding any diagnostic systems is which data should be processed and how the sensors should be placed. GAs being general purpose optimization algorithms is extremely suitable for such sensor placement problems. In this work the GA is tested using 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. For initial testing of the GA, we consider that there exists a neural network based identification system for a plate with 20 potential sensors locations.*

**Keywords:** *Diagnostic system, Fiber Bragg grating sensors, Genetic algorithm.*

---

### **I. Introduction**

Fiber Bragg grating sensors obtain the strain data all along the optical fiber. This opens new possibilities for a diagnostic system. Formerly, only point sensors, as a strain gages or fiber Bragg grating, were available, and information about the response to loads was restricted only to those points on which the sensors were bonded. Unless a sensor was located near the damage initiation point, details about the failure initiation and growth were lost. With a distributed system, the information is given as an array of data with the position in the optical fiber and the strain or temperature data at that point. The efficiency of a diagnosis system depends on the information it can retrieve from the diagnosed plant. Obviously, if the information is insufficient, the diagnosis system is not able to perform its task. However, an increased number of sensors alone do not guarantee that the diagnosis system will have a better performance. The relevance of the information brought by an additional sensor must also be taken into account. In some cases, knowing the values of a given variable brings no information from the diagnosis point of view. Thus, the efficiency of a sensor system can be measured by the diagnosability degree it provides. However, the same diagnosability level may be obtained for different sensor configurations. Moreover, in some cases, the highest level of diagnosability might not be necessary. Therefore, economical issues may come into place. When designing a sensor system, one must search for those combinations of sensors that can provide a specified diagnosability level at lowest possible cost. Taking the cost of sensors into account, it is uneconomical to install sensors on every part of a distributed system. For a limited number of sensors, two questions naturally arise: which type of sensor placement optimization performance index should be adopted, and which algorithm can be used? A placement optimization performance index based on damage detection and a genetic algorithm are presented to answer these questions.

A number of different optimisation techniques have been developed over the last decades. These include heuristic approaches, classical and combinatorial optimisation. Many early optimisation methods, so called ad hoc methods, are based on rough and ready ideas without much use of theoretical background. Stepwise techniques can add or remove one or more sensor at the time in order to find the best combination. A number of unconstrained (Newton methods) and constrained (linear and nonlinear programming) deterministic optimisation methods can be used for optimal sensor location. This includes countless methods based on the availability of gradients and Hessians. Simple deterministic techniques are sufficient for local search; constrained optimisation has a great degree of complexity. Recent years have seen the development of combinatorial optimisation methods based on biological and physical analogies. These include Genetic Algorithms (GAs), Simulated Annealing (SA), Tabu Search (TS) and a number of hybrids of these techniques. GAs is a search techniques simulating natural evolution. Possible candidate solutions are coded as chromosomes. Random operations based on natural selection are used to evolve the initial population.

Methods have been developed to place sensors in an optimal fashion to address the identification and control of dynamic structures by Udawadia and Garba [1] and Lim [2]. Kammer [3] proposed an effective independence algorithm based on the contribution of each sensor location to the linear independence of the

identified modes. The initial candidate set of sensor locations was quickly reduced to the number of available sensors. Hemez and Farhat [4] extended the effective independence method in an algorithm where sensor placement was achieved in terms of the strain energy contribution of the structure. Miller [5] computed a Gaussian quadrature formula using the functional gain as a weight function, and thought that the nodes of the quadrature formula gave the optimal locations for sensors.

## II. Genetic algorithms

In this optimization method, information about a problem, such as variable parameters, is coded into a genetic string known as a chromosome (individual). Each of these chromosomes has an associated fitness value, which is usually determined by the objective function to be maximized or minimized. Each chromosome contains substrings known as genes, which contribute in different ways to the fitness of the chromosome. The genetic algorithm proceeds by taking a population, which is comprised of different chromosomes with highest fitness, and increasing the average fitness of each successive generation. Only the fittest chromosomes pass to successive generations. Now we introduce some basic concepts and operators of GAs before the description of the suggested algorithm.

## III. Coding

An essential characteristic of a GA is the coding of the variables that describe the problem. A binary coding method can be used. The method is to transform the variables to a binary string of specific length. If there are  $n$  optional degrees of freedom to place sensors, the coding length of a string is  $n$ . If the value of the  $i^{th}$  bit position of a string is 1, this denotes that a sensor is located on the  $i^{th}$  degree of freedom; in contrast, if the value of the  $i^{th}$  bit position is 0, this denotes no sensor on the  $i^{th}$  degree of freedom. For example, *001001100100* denotes that sensors are located on the third, sixth, seventh, and tenth degrees of freedom.

## IV. Initial population

GA starts from an initial population of individuals as initial variable. Each individual in the initial population should meet the constraint. Thus, it is easy for GAs to reach the global optimum solution.

## V. Selection

The selection procedure used by GAs is based on the fitness of each individual. There is a number of reproduction schemes commonly used in GAs. These include proportionate reproduction, ranking selection, tournament selection, steady state selection, and greedy over-selection. Tournament selection can be used.

## VI. Crossover

Crossover is the operator that produces new individuals (offspring) by exchanging some bits of a couple of randomly selected individuals (parents). The general form of the operator is one-point crossover, described as follows. First, a cutting position is chosen at random between the first and the last bit of the parents, and the parents are thus divided into two parts by the cutting. Then, the two offspring are generated by each taking the first part from one parent and the second from the other. For example, the constraint is  $q = 4$ ; assume a number 4 is chosen at random as the cutting position. The one-point crossover operation is shown in figure 1. We can find that even if the parent strings satisfy the constraint, the generated offspring still violate the constraint. Therefore, we will improve the crossover operator in the following section.

## VII. Mutation

Mutation operates on a single individual with a small probability. With this operation, one or more bits are chosen at random from the individual and changed into a different symbol. This is shown in figure 2.

Parent 1	0	1	0	0	1	1	0	0	1	1
Parent 2	1	0	1	0	0	0	1	0	1	0
Offspring 1	0	1	0	0	0	0	1	0	1	0
Offspring 2	1	0	1	0	1	1	0	0	1	0

Figure1. Sketch of one-point crossover

Parent 3	0	1	0	0	1	1	0	0	1	0
Offspring3	0	1	0	0	0	1	0	0	1	0

Figure2. Sketch of basic mutation

Similarly, in order to meet the constraint, the mutation operation should be improved, too. In order to apply a genetic algorithm to the sensor optimization placement problem, we have the following steps.

- (1) Create an initial population randomly, and calculate the fitness values of the strings.
- (2) Select the fittest individuals according to fitness values, and apply the crossover operation and mutation operation.
- (3) Calculate the fitness values of the new strings.
- (4) Repeat (2) and (3) until the termination criterion is met.

### VIII. Implementation of GA:

The potential solution for this sensor optimization 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,

$$I^k = \{3; 7; 10; 14; 18\} \quad (1)$$

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

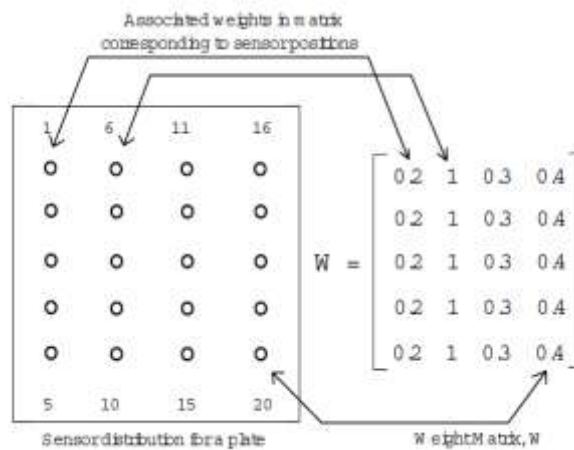


Figure 3: Sensor placement problem

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 are 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 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 (1) is given as,

$$F(I^k) = W3 + W7 + W10 + W14 + W18 = 2.9 \quad (2)$$

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 along with the fitness function to discard infeasible sensor configurations.

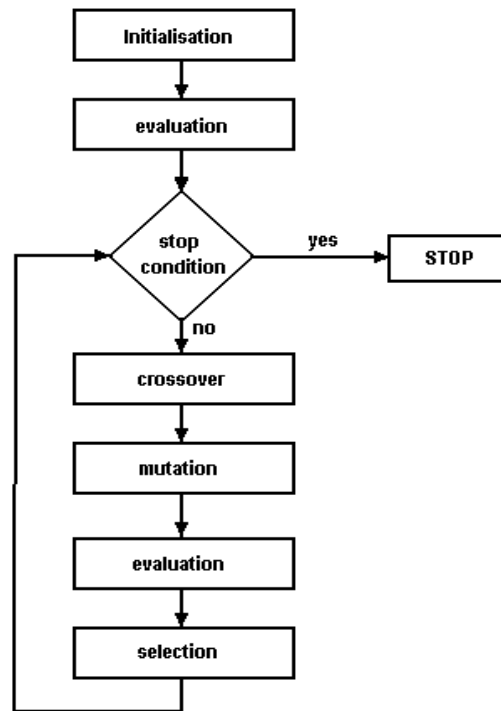


Figure 4. Basic genetic algorithm

Due to the randomness involved during initialization 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 [6]. 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,  $C1$ ) 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,

P1 = {2; 5; 9; 11; 15}  
 P2 = {3; 7; 10; 13; 18}  
 .....  
 O1 = {2; 5; 10; 11; 18}  
 O2 = {3; 7; 9; 13; 15}

The process of mutation is carried out using the Michalewicz non-uniform mutation. 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,  $p_{in}$  [7]. 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.

### IX. Example Results

To demonstrate the performance of this GA implementation, the GA was applied to the sensor placement problem shown in figure 3. The parameters of the GA used are given in table 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 convergence plot for the mean fitness values of best 10 individuals is also plotted in figure 5 to ensure that the final population is dominated by optimal solutions.

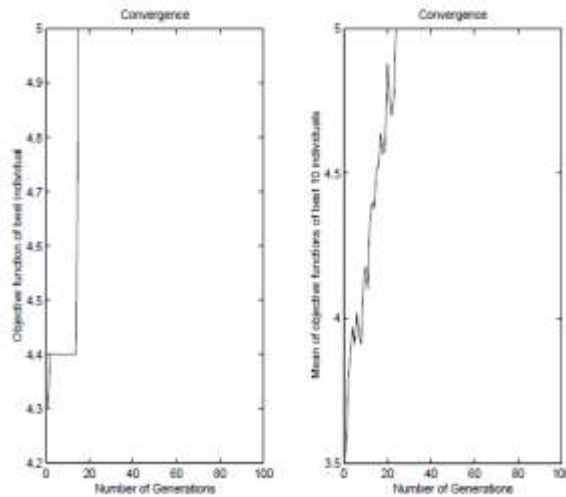


Figure 5: Convergence of GA for sensor placement

Table 1: Parameters of Genetic Algorithm for Sensor Placement Problem

Parameter	Value
Probability of crossover ( $pc$ )	0.45
Probability of mutation ( $pm$ )	0.15
Probability of inversion ( $pin$ )	0.01
Number of generations ( $N$ )	100
Population size ( $\square$ )	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 be 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 presented result allows the GA to be used for optimal placement of sensors, in an existing damage identification system

### X. Conclusion

In this work 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.

### References

- [1] Udawadia F E and Garba J A 1985 Optimal sensor locations for structural identification *Proc. JPL Workgroup on Identification and Control of Flexible Space Structures (San Diego, CA)* pp 247–61
- [2] Lim T W 1991 Sensor placement for on-orbit modal identification *Proc. 32nd Conf. AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials* (Reston, VA: American Institute of Aeronautics and Astronautics)
- [3] Kammer D C 1991 Sensor placement for on orbit modal identification and correlation of large space structures *AIAA J. Guid. Control Dyn.* 14 251–9
- [4] Hemez F M and Farhat C 1994 An energy based optimum sensor placement criterion and its application to structure damage detection *Proc. 12th Conf. on Int. Modal Analysis* (Honolulu, HI: Society of Experimental Mechanics) pp 1568–75
- [5] Miller R E 1998 Optimal sensor placement via Gaussian quadrature *Appl. Math. Comput.* 97 71–97
- [6] A. E. Smith, D. W. Coit, "Constraint Handling Techniques - Penalty Functions", *Handbook of Evolutionary Computation*, Chapter C5.2, 1997.
- [7] K. Worden, A. P. Burrows, "Optimal sensor placement for fault detection ", *Engineering Structures*, Vol. 23, pp. 885-901, 2001.