CHAPTER 2

LITERATURE REVIEW

2.1 Introductory remarks

Artificial Neural Network (ANN) is a subfield of the artificial intelligence technology that has gained strong popularity in a rather large array of engineering applications where conventional analytical methods are difficult to pursue or show inferior performance. Specifically ANNs have shown a good potential to successfully model complex input/output relationships where the presence of non-linearity and inconsistent/noisy data adversely affects other approaches, ANN model is robust and fault tolerant. ANN can also work with qualitative, uncertain and incomplete information, making it highly promising for inverse problems in structural engineering.

An ANN consists of a large number of interconnected computational elements called “neurons,” organized in a number of layers. The connection between each pair of neurons is called a link and is associated with a “weight” that is a numerical estimate of the connection strength. Each neuron in a layer receives and processes weighted inputs from neurons in the previous layer and transmits its output to neurons in the following layer. The weighted summation of inputs to a neuron is converted to an output according to a transfer function (typically a sigmoid function).

There is a wide range of architectures for ANNs among which the most widely used architecture is the feed forward architecture. In this type of network, there are two distinctive modes, namely, training and query modes. In the training mode, a training set consisting of input/output patterns is presented to the networks and the weights are found through an iterative process. The back-propagation learning algorithm (Rumelhart and McClelland, 1986) is often used to find the weights such that the difference between the expected outputs and the ones calculated by the network is sufficiently small. After the
network is trained, it is presented with the query data (input variables only) to determine how accurately the network has simulated the input/output relationship. In the present work, it is proposed to apply ANNs and hybrid networks for the structural design of R.C.C. elements. Accordingly, a detailed discussion about the ANNs is presented in the following sections.

2.1.1 The biological neural network

The structure and the functioning of the brain have been studied by many neurophysiologists. Even now the exact functional process of the human brain is not known. Only an overview of the functioning of the human brain is available at present. Basically, the brain functions with a very dense network of neurons. The biochemistry of the neurons is also not fully known. The brain contains as many as $10^{11}$ neurons connected to each other by as many as $10^{15}$ interconnections among them (Snell, 1992). Fig. 2.1 shows a typical biological neuron.

The biological neuron consists mainly of the following parts.

- The Cell Body
- The Axon
- The Dendrite

![A biological neuron](image-url)
The dendrite is responsible for carrying the signals from various other neurons to the neuron which it is a part of. These dendrites are spread in a branched form and carry signals in very complex manner in the form of complicated electro-chemical signals. On the other hand, an axon carries the signals from the cell body to various other neurons. The dendrites and the axon meet at a point which is called as the synapses. Although it is often described as the meeting point, there is no physical contact between the axon and the dendrites at the synapses. The functioning of the neuron depends on certain very special types of chemicals which are created in the cell body itself. In the electro-chemical process the electrical process is carried out by the charged ions which move across the potential difference of 70 millivolts. The chemical process is carried out by the chemicals called as neurotransmitters which are released by the axon and are absorbed by the dendrites. At the synapses, the dendrites do not absorb all the chemicals released by the axon but at different places there is a provision for such absorption in parts. The chemicals used by the neurons are always replaced by the new chemicals from the cell body. Although other types of cells in the human body are regenerated, the neural cells are not generated and are maintained for the lifetime of a human being. The dendrites collect the signals at the synapse and they carry it to the cell body for further action. The dendrites do not get the signals at the synapses only. They may also get it directly from the other cell body or even from the axon itself. When all the dendrites carry the signals to the cell body it is essentially averaged up there. Here, after a sufficient time if the average is more than the threshold, then a signal is generated by the cell body and the same is sent down the axon.

As hundreds of thousands of neurons work simultaneously in a form of well organized but extremely complicated network, the human brain can store a very large amount of data and can recall it very quickly. This also gives our brain an amazing
computing power, memory, emotions, thinking power etc. The biological neural network is also supported by many other factors which are very difficult to simulate using the currently available hardware and the software support. The natural net has also time dependent signals. The neurotransmitters are actually released by the axon. This process is controlled by the axon itself in an unknown way. The blood and the oxygen support to the biological neuron also affect their performance. The exact way of the processing of the input signals by the cell body is also not yet known. As this total process is not known very clearly, it is not possible to develop an artificial neuron which could be claimed as an exact model of a biological neuron. Hence, an artificial neuron is developed as a very approximately simulated model which can perform a very basic mathematical function.

2.1.2 The artificial neural network

The artificial neuron is a very approximately simulated model of the biological neuron. The artificial neuron can carry out only a very simple mathematical function and/or can compare the two values Fig. 2.2 describes an artificial neuron.

![Fig.2.2: An artificial neuron](image-url)
An artificial neuron has a typical function associated to it which is often called as *Threshold function* or *Squashing function*. The input and output of the neuron is typically in an analog form and all computations are carried out using these analog signals only. A typical artificial neuron gets an input either from other neurons or directly from the environment (i.e. input nodes). The paths connecting the input nodes to the neurons and the connections between the various neurons are associated with a certain variable weight which represents a multiplying factor for the incoming signal representing the synaptic strength of the connection. These weights are initially set to some random values and are later adjusted in the process of training of the net. The artificial neuron then sums this input which is actually a weighted sum of all the input signals. The input so obtained is used to calculate a node value according to the Squashing function of the neuron. This node value is compared with the threshold value of the neuron and if the node value is higher, then the neuron goes to the "Higher State" (excitation state) and a signal is passed on to the next layer of neurons. The three common types of non-linear nodal functions generally used are shown in Fig.2.3.

![Nodal functions](image)

**Fig.2.3: Nodal functions**
The artificial neural network which is also called as a connectionist model, parallel distributed processing model, a neuromorphic system and so on, is developed by using the approximate neuron as described above. The artificial neural network attempts to simulate a very small part of the biological natural neural network for better computational results. There is no definite rule for configuring the net. The net is described by the nodal characteristics (i.e. by the squashing function used), number of layers (i.e. either a single layer or a multilayered net) or the learning rule. There are different models developed by different researchers. The classification of the ANNs is shown in Fig.2.4 (Lippmann, 1987).

The ANNs are classified based on whether it is binary or continuous valued. They are further subdivided based on the training procedure i.e. supervised or unsupervised.

---

**NEURAL NET CLASSIFIERS**

- **BINARY INPUT**
  - Supervised: Hopfield Net
  - Unsupervised: Hamming Net

- **CONTINUOUS VALUED INPUT**
  - Supervised: Carpenter Grossberg Classifier
  - Unsupervised: Perceptron Multilayer Perceptron

- **SUPERVISED:** Kohonen Self-Organising Feature Maps

---

**Fig.2.4: Classification of networks**
The supervised learning requires the pairing of each input vector with a target vector representing the desired output. An input vector is applied, the output of the network is computed and compared with the corresponding target value. The error is fed back and the weights are updated according to a training algorithm which tends to minimize the error. The supervised learning has been criticized as it is not biologically plausible. Unsupervised learning is more plausible model of learning in biological system. It requires no target vector for the outputs. The training algorithm modifies the connection weights to produce the output vectors that are close to the desired output vectors. The training process extracts the statistical properties of the training set and groups similar vectors into classes. Only supervised learning has been considered in the present work. Accordingly the details of feed forward network are presented in the next section.

2.1.3 The feed forward network

Many configurations of artificial neurons can be used to develop networks to suit particular requirements. The feed forward network is used for mapping complicated relationships between the discrete input and the output vectors. The network is presented with a set of sample input and the corresponding output vector, which defines the desired unknown relationship. The feed forward network, after its successful learning, correlates these vector spaces to arrive at a generalized relationship to define the unknown function. The feed forward network is configured by arranging artificial neurons, in different layers. Fig 2.5 shows a typical feed forward network.
As shown in the figure, the first layer is called the input layer. The last layer is called as the output layer. The network may have one or more hidden layers situated as shown in the figure. The input to the network is provided as the activation for input nodes. This input is further treated within the network by sending it to other neurons of the next layer. The network is fully interconnected for this purpose as shown in Fig.2.5. The output of the network is obtained from the neurons in the output layer. The activation of the output neurons is taken as the output of the net. All connection links are associated with a scalar weight which represents the synaptic strength of connection. Thus, when the neuron in the higher layer receives the input, it automatically receives the weighted sum of the output of all neurons in the lower layer. The neuron processes the incoming weighted sum using a nodal function as

\[ Y_i = \psi \left( \sum X_i \cdot W_{ij} - \theta_i \right) \]  

(2.1)

Where, \( \psi \) is a non-linear function and \( X_i \cdot W_{ij} \) are outputs of \( i^{th} \) neurons and weights from the \( i^{th} \) neuron to the \( j^{th} \) neuron, \( \theta_i \) is the threshold value of the \( i^{th} \) neuron.
The neurons in a feed forward network used for the present investigation make use of a sigmoidal nodal function which is given as.

\[ Y_j = \frac{1}{1 + e^{(X_j - \theta_j)}} \]  

(2.2)

Where, \( X_j \) is the weighted sum coming to \( j^{th} \) neuron as its input.

Thus, the output from the network is obtained as a result of the sigmoidal nodal function of each output node. It can be seen from Eqn.2.2 that the output of any node in the network has a domain \( 0 \leq Y \leq 1 \). As a result, the network processes the information in a normalized form. The input and the output of neurons, therefore, are often normalized to facilitate the working of the network using a sigmoidal nodal function.

The complete network development process of the feed forward network consists of:

- Selection of the network parameters such as the number and size of different layers of the network, nodal functions, etc.
- Selection of the parameters for the information flow within the network i.e. the input/output pattern, normalization factors, etc.
- Selection of the learning rule and other learning parameters to enhance the learning capability and speed.
- Initialization of the network and learning till the network learns the desired relationship between the input and the output vectors.

2.1.4 Configuration of the network

To start with the development of a network, a set of training pairs i.e. the desired input and corresponding output vector is prepared carefully so as to reflect all the
concepts that the network is supposed to learn. In the absence of a suitable well-established method, the number of layers and the number of nodes in each layer for each network have to be decided by trial and error process initially. These parameters are to be modified suitably so as to improve the performance of the network. The modifications shall be based on the analytical and logical conclusions drawn from the performance of each configuration and the study of the training data. The main aim of development and training of networks is to obtain the required output vector for the corresponding input vector presented to the network during the training. The size of the network, i.e. the number of layers and nodes in each layer can be either increased or decreased to enhance the performance of the network. At the same time, an optimum configuration of the network is desired for computer memory and computational time requirements, as networks in the present work have been realized as software simulations. When the results obtained from the network are all within a predefined tolerable limit, it is assumed to have converged. In the present investigation, the convergence of network has been problem oriented and the training of the network has been continued till the error is reduced up to an acceptable limit.

The network configuration state consists of:

1. Selecting input vector,
2. Selecting output vector,
3. Selecting threshold function,
4. Configuring hidden layers(s),
5. Normalizing i/o parameters,
6. Presenting training pairs,
7. Evaluating network performance and,
8. Organizing training set.
2.1.5 Selecting threshold function

The selection of the nodal function for the neurons is important because it affects the learning speed of the network. The selection of threshold function depends mainly on the intended use of the network and method of learning. As the back propagation learning algorithm has been used to train feed forward networks in the present investigation which are intended for a prediction problem, the sigmoidal non-linear nodal function has been used. The three types of nodal functions which are commonly used are shown in the Fig. 2.3. As it can be seen from the figure, the hard linear non-linearity has only two states i.e., 0 and +1, or -1 and +1. The threshold logic has a linearly varying part and the sigmoidal function has low output for the low input and has high output for sufficiently high input. The sigmoidal function is a continuous one which is asymptotic between 0 and +1.

The artificial neuron basically computes by classification. A simple neuron or the perceptron having only two states can perform the function of an OR Gate (Lippmann 1987), i.e. it can decide whether the input belongs to class A or class B. The input to a perceptron connected to n input nodes can be written as

\[ x = \{x_0, x_1, ..., x_{n-1}\}^t \]  \hspace{1cm} (2.3)

The weights of the corresponding connection links are defined as:

\[ w = \{w_0, w_1, ..., w_{n-1}\}^t \]  \hspace{1cm} (2.4)

The weighted sum i.e. \( \sum_{i=0}^{n-1} w_i x_i \) can be replaced with identical vector dot product i.e., \( w, x \)
When the output of the classifier is zero, we have

\[ \sum_{i=0}^{n-1} w_i x_i - \theta = 0 \]  

(2.5)

For two input nodes, Eqn. 2.5 can be expanded as.

\[ x_1 w_1 + x_2 w_2 - \theta = 0 \]  

(2.6)

rearranging this,

\[ x_2 = - \frac{w_1}{w_2} x_1 + \theta \frac{1}{w_2} \]  

(2.7)

This is the equation of the line separating the pattern space (Beale and Jackson, 1990).

Fig. 2.6: The XOR Problem

However, it may be noted here that such a nodal function i.e. hard limiting function can perform only a linear classification. Only patterns which are linearly separable can be classified using the hard limiter perceptron. A pattern such as the one shown in Fig. 2.6 can not be classified by a perceptron. This problem is popularly known
as XOR problem. It can be seen from the figure that it is not possible to draw a single straight line to separate the class A from the class B. Thus, the solution space for this problem is not linearly separable. However, using a combination of such perceptrons, a multiplayer perceptron can be constructed as shown in Fig. 2.7.

It is obvious from the figure that such a multiplayer perceptron can classify the input pattern such as the XOR problem. However, the weight of the different connection links in such a network cannot be learned with a hard limiter threshold function. This is because, according to the learning rule for perceptrons, only the active connections are strengthened. Therefore, the input pattern must be known at the output nodes. This, in turn, means that only single layer perceptrons can be trained. From Fig. 2.7 it is clear that the input pattern cannot be known at the output node to strengthen the active connections. However, the weights in the second layer can be adjusted according to the scale of activation of the neuron in the first layer. This can be done by selecting a

![Figure 2.7: Multiplayer perceptron to solve XOR problem](image-url)
threshold function which is continuous but closely follows the hard limiter threshold function. This is possible using the sigmoidal threshold function. The output from a sigmoidal threshold function goes from low to a high for a very small variation in input. Further, the sigmoidal nodal function is continuous one. Therefore, sigmoidal function can represent the hard limiter nodal function which allows the weight adjustments in proportion to the activation level of the neurons in the lower level.

The neuron with a hard limiter nodal function can separate the solution space into two regions by placing a plane suitably in the given space. With higher order non-linearity, the number of regions in which a neuron can divide the given space increases and hence higher order nodal functions are more powerful.

However, the sigmoidal non-linearity has been adopted in the present work because of the following reasons:

- It represents the response of the actual natural neuron very closely.
- It has a very simple derivative which is useful in the development of the training algorithm.

The sigmoidal function is given by:

\[ y = f(x) = \frac{1}{1 + e^{-x}} \] (2.8)

The derivative of this function is:

\[ f'(x) = \frac{e^x}{(1+e^x)^2} \]
\[ f'(x) = f(x)(1-f(x)) \]
\[ f'(x) = y(1-y) \] (2.9)

This simple derivative facilitates quick calculations in the learning algorithm for the back propagation algorithm.
The sigmoidal nodal function acts as automatic gain controller. When the input to a node is towards extreme values, it falls on that portion of the nodal function, where the slope is less i.e. nearer to zero. On the other hand, the nodal function allows the higher inputs to pass through the region where the slope of the function is more. Thus, the gain term for weight correction, which is proportional to the slope of the nodal function in back propagation, is automatically controlled. The sigmoidal non-linearity has been used in the present case as the back propagation algorithm is used for the training of the network.

2.1.6 Training of the network

The feed forward neural networks map the desired relationship between the input and the output parameters. The network configuration with the hidden units attempts to separate an arbitrary decision boundaries between the given output classes using the changes in the weight matrices during the training. The learning rule for the multilayer network, which affects these changes, is based on the principle of perceptron learning. The training set for the problems adopted in the present work consists of an input vector and the corresponding output vector. This training set is presented to a randomly initiated network. The error at the output nodes is then reduced in stages. The back propagation (BP) algorithm used for the training of the network is well established and has been tested for a variety of problems by a large number of researchers (Leach et al., 1993; Penumadu et al., 1991; Ghaboussi and Joghatie, 1995). The perceptron training algorithm is briefly discussed below followed by a discussion on the BP algorithm.
2.1.7 The perceptron learning

The first perceptron learning algorithm was proposed by Donald Hebb in 1949 (Wasserman, 1989). From his studies of a real neuron, he suggested that only the active links must be reinforced i.e. whenever two interconnected neurons fire, the connection between them is reinforced. This algorithm can be expressed as.

\[ w_{ij}(t+1) = w_{ij}(t) + NET_i \times NET_j \]  

(2.10)

Where, \( NET_i = \sum w_{ij}o_j \)

\( w_{ij} = \) the weight connecting neuron i to neuron j

\( o_j = \) the output of the neuron j

The perceptron training algorithm was later modified by Hoff (cited in Wasserman, 1989) by minimizing the error between the actual output and the desired output. This algorithm has been developed by using a linear threshold instead of hard limiter threshold function. The Least Mean Square error between the actual and the desired output is minimized for all the patterns.

The effort \( E_p \) for \( p^{th} \) pattern is given by

\[ E_p = \sum (d_i - o_i)^2 \]  

(2.11)

Where, the summation is performed over all the output nodes.

The perceptrons invited attraction of many researchers until Minsky and Papert (Beale, 1990) demonstrated that the perceptrons could only do linear classification. This area was under near eclipse until a learning algorithm for multi layer perceptron was proposed. The use of multi layer perceptron could classify the non-linear regions as well. The training of such a multi layer network, however, is not possible unless the threshold function is a continuous one. The gradient obtained at a point using this continuous nodal function is made use of in the back propagation algorithm for updating the weights.
2.1.8 The Back Propagation (BP) algorithm

The back propagation algorithm is a generalized form of the least mean square training algorithm for perceptron learning. It uses the gradient search method to minimize the error function equal to the mean square error between the desired and the actual outputs. The nodal function used in the network is sigmoidal non-linearity. The derivation of Back propagation algorithm can be found elsewhere. (Rumelhart and McClelland).

Any continuous non-linear nodal function can be used for the implementation of the BP algorithm.

The algorithm is as follows:

- **Step 1**
  Initialize the weights and thresholds to some random values.

- **Step 2**
  Present an input vector \(x_0, x_1, \ldots, x_n\) and specify the desired outputs \(d_0, d_1, \ldots, d_n\).

- **Step 3**
  Calculate the actual output. At each node, calculate the weighted sum of the inputs and use the sigmoid non-linearity:

\[
Y_j = \frac{1}{1 + e^{(x_j, \theta_j)}}
\]  

(2.12)

Where, \(X_j\) is the weighted sum of inputs coming to \(j^{th}\) node.

\(Y_j\) is the output of the \(j^{th}\) node

\(\theta_j\) is the threshold for the \(j^{th}\) node.

- **Step 4**
  Adapt weight using recursive algorithm at the output nodes and working back, adjust the weights as

\[
w_{ij}(t+1) = w_{ij}(t) + \eta \delta_i x_i
\]  

(2.13)
Wherc, \( w_{ij} \) is the weight from the \( i^{th} \) node to the \( j^{th} \) node,

\[ \delta \text{ is the error at the } j^{th} \text{ node,} \]

\[ \eta \text{ is the gain term constant.} \]

If \( j \) is an internal hidden layer node then

\[ \delta_j = x_j (1-x_j) \sum (\delta_k-w_{kj}) \]  

(2.14)

Where, \( \delta \) is the error correction at the the \( j^{th} \) node in the hidden layer and the summation is performed over all the nodes in the layer above the node \( j \).

- Step 5

Repeat by going to step 2 till the network learns the phenomena to the satisfaction.

In the above algorithm, the weight update is proportional to the current gradient.

The gain term used during the weight update (Eqn.2.13) acts as a step in the steepest descent direction. The error, as it can be seen from the algorithm, is calculated at the output nodes. This error is calculated as the squared difference between the desired and the actual output. However, it can be seen that the actual error at the nodes in the hidden layers cannot be known. Therefore, the error at the output nodes is carried to the hidden layer nodes through the connection weights between the corresponding nodes as depicted in the Eqn. 2.14. The error at all the output nodes is carried to each intermediate layer node in the same fashion.

2.1.9 Drawbacks of back propagation

Although the back propagation algorithm has been widely used for training networks for a variety of applications, it is associated with some drawbacks. The possible drawbacks of the back propagation algorithm are:
a. Network paralysis

The network paralysis is caused by weights having very large values. During the training, if the weights are adjusted to very large values, the neuron operates at the large value i.e. the region of the sigmoidal non-linearity where its derivative is very small. As the weight changes are proportional to the derivative, the weight changes affected by the BP algorithm are negligible. Therefore, the network virtually stops learning. To overcome the network paralysis, the network is generally initialized with a new set of random initial weight and threshold vectors.

b. Local minima

This is caused when the weight vector gets trapped in local minima. Since the back propagation algorithm uses steepest descent method, it takes the weight vector downward on the error surface by adjusting the weights. If the error surface is full of hills and valleys, the weight vector may get trapped in local minima that are nearest to the starting point. This problem may be overcome by adding small random numbers to the weight matrix.

The above drawbacks of BPN can be alleviated by hybridizing the same with GA. This is the spirit of the present work.

2.2 Structural Engineering and Artificial Neural Networks

Structural design is an iterative process. The initial design is the first step in design process. After the problem definition, a designer makes an overall guess about the possible optimum solution, consistent with designers experience, knowledge, constraints and requirements. The analysis of the structure is then carried out for the initial design and using the results of the analysis, are design of the structure is carried out if any of the constraints is not satisfied. The efficiency of the design process depends heavily on this
initial guess. A good initial design, therefore, substantially reduces the number of subsequent analysis-design cycles. This phase is extremely difficult to computerize, because it needs human intuition. As the ANNs can learn from available designs and apply their knowledge in subsequent designs, they offer an attractive method of computerizing the initial design process. Therefore, there is a need to develop a simple and compact model which can predict these responses. Genetic algorithms and back propagation neural networks are becoming popular for modeling the complicated structural behavior in this decade. Neural networks have been already used for modeling complicated multi parameter material behavior (Ghaboussi et al., 1991; Mukherjee et al. 1995; Rao and Mukherjee 1996). The basic advantage of artificial neural network lies in the fact that it is a model-free estimator. The neural networks do not require any external manifestation of parametric relationship. Therefore, complicated relationships between various parameters are mapped automatically by the network. However, the use of simple back propagation neural networks require huge number of training cycles for making it to learn the complicated relationship. Use of genetic algorithms on the other hand may be extremely beneficial in reducing the training difficulties. Though, a few applications of genetic algorithms are reported in structural engineering, a lot of potential still exists for their application (Rajasckaran and Pai, 2003). Hence, it is proposed to use genetic algorithm based back propagation neural networks for developing a compact model for relating the structural design parameters.

2.2.1 Application of ANN in structural engineering

Over the past few years, neural computing has attracted researchers from many areas of structural engineering. Neural computing offers an attractive package of computational flexibility in terms of increased processing speed, machine learning,
reduced knowledge engineering, easy implementation, capabilities for postulating complicated material and human behavior, inherent parallel processing capacity, etc. Therefore, researchers have found this computing technique very useful in fields such as structural optimization, preliminary design, approximate analysis, earthquake characterization, damage detection, construction management, material modeling, inverse problems, capturing human behavior and so forth. A brief review of various structural engineering applications of ANNs is presented below. In this chapter an attempt is made to classify the research in to the different areas of structural engineering and presented accordingly.

2.2.2 Neural networks in structural design

Van Lauchene and Sun (1990) developed a neural network for the initial design of reinforced concrete beam sections. The network predicted the depth of a beam from given bending moment, the reinforcing steel strength, concrete compressive strength, reinforcing ratio and regular section's width to depth ratio. The network used for this problem consisted of five input nodes, two hidden layers with six nodes each and one output node. The network has been trained with twenty one training patterns on a personal computer in 2.5 hours. A constant threshold has been used for all nodes. The network has been tested for ten new patterns. The error in predictions ranged between 0.8 to 42 percent.

Liu and Gan (1991) developed a preliminary structural design expert system called SPRED-I based on neural networks. The architecture of the system is an integration of prediction, evaluation and intelligent control. The architecture uses ANN as blocks for the respective operations. The network training methodology uses Daridon-Fletcher-Powell (DFP) method instead of the standard back propagation...
algorithm SPRED-I has been developed for the preliminary design of space grid structures. The results obtained from the system have been compared with those of finite element analysis.

Ian Flood and Kartam (1994) discussed in two papers providing a discourse on the understanding, usage and potential for application of artificial neural networks within civil engineering. A graphical interpretation of the way in which neural networks operate was first presented. The second paper demonstrated the versatility of neural networks as a problem-solving tool and shown their applicability to different problems in civil engineering.

Mukherjee and Deshpande (1995) used neural networks for synthesis of preliminary design of a simple reinforced concrete beam. A back propagation algorithm was used to train a neural network for the minimum cost design of a simply supported concrete beam. The minimum cost design for training was obtained through the use of a mathematical optimizer. Once trained the network closely followed the optimizer and generated designs within 5% of that of the optimizer. The time taken by the neural network is only a small fraction of that of the optimiser.

Mukherjee and Deshpande (1995) presented the initial design of reinforced concrete multi-span beams. This problem is extremely difficult, if not impossible, to handle using the mathematical optimization procedures. In this case the network was trained with the minimum cost design found from alternatives generated manually. Practical aspects e.g. convenience of construction, feasibility of laying reinforcements, aesthetics etc., were introduced. The network, when presented with the geometry, load and material data, efficiently arrives at an initial design that closely approximates the design created by an expert. The neural network was able to emulate the design of an expert within 5-7%. Various steps in the development of ANN model such as selection of
training examples, arriving at the network configuration, effect of selection of nodal properties and input or output patterns on performance of the net, fault tolerance of the network, effectiveness of activation in hidden layers neurons, handling new examples, extent of generalization, cost estimation, elimination of number crunching routines in synthesis of initial design model along with learning procedure have been addressed in detail. Sankarasubramanian and Rajasekaran (1996) developed a nonlinear hypo-elastic constitutive relation for the analysis of plane and axi-symmetric reinforced concrete structures using artificial neural networks.

Papadrakakis et al. (1996) examined the application of neural networks to the reliability analysis of complex structural systems in connection with Monte Carlo Simulation (MCS). The failure of the system was associated with the plastic collapse. A back propagation algorithm was implemented for training the NN utilising available information generated from selected elasto-plastic analyses. The use of MCS with importance sampling further improved the prediction of the probability of failure with neural networks.

Anderson et al. (1997) have applied ANN technique for steel frames. It is usually the minor-axis beam-to-column connections that govern restraint to the columns against buckling. There is, however, no generally accepted method to predict the behavior of such connections. To clarify the response, a series of tests have been performed, in which significant parameters have been systematically varied. The results have been used to train an artificial neural network to predict bi-linear moment-rotation characteristics for minor-axis connections. The paper describes the test programme, the choice of ANN and the result for each connection, based on learning from twenty other connections. The results are found to provide approximations to the experimental response that are satisfactory for use in structural engineering design.
Cheng Yeh (1998) presented an augmented neural network (ANN) and examined its efficiency and accuracy for structural engineering applications. Experimental results demonstrated that the network's logarithm and exponent neurons provided a markedly enhanced network architecture capable of improving the network's performance for structural engineering applications.

Krishna and Gangadharan (1999) have carried out the analysis of infilled frame by using neural networks. In this paper, they have used the data available from previously conducted experiments to develop training set for neural network training for the analysis of single bay portal frame.

Hungs and Jan (1999) used a newly developed cerebellar model articulation controller (CMAC), one of the supervised neural network learning models and Macro Structure CMAC (MS—CMAC) neural network learning model to aid steel structure design. The topology of the novel learning model was constructed by a number of time inversion CMACs as a tree structure. It was shown that the MS—CMAC not only can learn structural design problems within a reasonable central processing unit time but also can estimate more accurate coefficients.

Jenkins (1999) has developed a neural network for the structural re-analysis. It was reported that structural design requires a more or less continuous re-analysis of the structure. He reported that network parameters like rate of learning and slope coefficient of sigmoid function have considerable effect on the final performance of the network and need of further work is realized in the establishment of these parameters.

Namhee Kim Hong et al. (2002) have presented the concept of artificial neural network to develop preliminary design system for cable-stayed bridges. The prototype of preliminary structural design system has been implemented as a part of integrated design
Muhammad Hadi (2003) discussed the applications of neural networks in concrete structures. Optimum design of simply supported concrete beams, fibrous concrete beams have been carried out using neural nets.

Wei Tang et al. (2003) explored the use of artificial neural networks in predicting the confinement efficiency of concentrically loaded reinforced concrete columns with rectilinear transverse steel. The close correlation between experimental and calculated values has shown that neural network-based modeling was a practical method for predicting the confinement efficiency of reinforced concrete columns with transverse steel because it provided instantaneous result once it was properly trained and tested.

Oreta and Kawashima (2003) applied artificial neural networks to predict the confined compressive strength and corresponding strain of circular concrete columns. The study showed the importance of validating the ANN models in simulating physical processes especially when data is limited.

Suresh et al. (2004) considered the flexural vibration in a cantilever beam having a transverse surface crack by computing modal frequencies to train a neural network to identify both the crack location and depth. The modular neural network method with a radial basis function network was found to perform better than the multi-layer perceptron network. The modular neural network architecture can be used as a non-destructive procedure for health monitoring of structures.

Mishra and Akhil upadhyay (2004) developed a neural network model for design of a rectangular column subjected to combined axial compression and uniaxial bending. The architecture used is 2-7-7-1, with learning rate of 0.75. They used the neural network model for predicting the percentage area of steel. The maximum error found in between
the model prediction and actual percentage area of steel is just 7%. The model took 10,000 cycles for learning the column design.

Kortesis and Panagiotopoulos (2005) proposed and studied neural network model for the treatment of structural analysis problems. Both the cases of bilateral and unilateral constraints were considered and Hopfield-like neural models were proposed. Numerical applications illustrated the theory and shown clearly the advantages of the neural network approach.

Kuzniar and Waszczyszyn (2006) dealt with an application of neural networks for computation of fundamental natural periods of buildings with load-bearing walls. The application of the proposed neural networks was used to identify the natural periods of the buildings with quite satisfactory accuracy for engineering practice. It was shown that this technique was also useful in damage detection and health monitoring of structures.

Cai and Cheng Xiao (2006) proposed an efficient and accurate algorithm to solve the calibration problem of cable safety factors of suspension bridges. The accuracy and efficiency of this method with reference to an example long-span suspension bridge were studied.

2.2.3 Neural networks in prediction of material behavior

The behavior of a material is best understood by carrying out experiments. Conventionally, the experimentally observed behavior of a material is modeled analytically using simple algebraic expressions. The analytical expression should predict the material behavior which agrees closely with the experimental observations. However, it may not always be possible to capture every material behavior by means of a simple expression. The development of such expressions can be extremely difficult and time consuming. Moreover, the behavior of modern materials is becoming more and more
complicated and they demand a more detailed study. The feed forward neural networks can be extremely helpful in capturing the experimentally observed material behavior directly which precludes the necessity of developing analytical expressions. The neural networks generalize on own. Therefore, they are also effective in predicting the behavior of a new material before the material is produced in the laboratory. This may reduce the cost of expensive experiments.

The computational simulation of composite ply micro mechanics using artificial neural networks has been reported by Brown et al. (1991). Three different stage results have been used for prediction of composite hydraulic, thermal and mechanical properties when provided with basic information concerning the environment, constituent materials and component ratios used in the creation of the composite. Excellent results have been reported for the first two stages. However, the error in the predictions of the network for the third stage has been rather high.

Ghaboussi et al. (1991) demonstrated the application of ANN for modeling the stress-strain behavior of concrete successfully. As a first example of material modeling with neural networks, biaxial behavior of plain concrete has been represented using ANNs. The experimental results reported by Kupfer et al. (1969) have been used for the preparation of the training set for the network. A feed forward network has been trained for this problem using back propagation algorithm. However, the paper did not provide sufficient information about the acceptable error limit and the learning parameters used (Penumadu et al., 1991).

Kim et al. (1993) presented neural network-based system identification techniques to predict the compressive strength of concrete based on concrete mix proportions. The compressive strengths estimated by the neural networks were verified by laboratory
testing results. It was demonstrated that the neural network techniques were effective in estimating the compressive strength of concrete based on the mix proportions.

Wittmann and Martonola (1993) applied neural networks for predicting the mechanical properties of concrete. The input parameters were the w/c ratio and the super plasticizer contents. The results displayed a characteristic smoothness of the interpolated description, which does not reflect the typical and often-encountered fuzzy behavior of the concrete material.

Eldin and Senouci (1994) used an ANN for analyzing the strength of the rubberized concrete (concrete with a filler of fragments of used tyres). The input vector consisted of rubber fragments type (shape), their size and percentage (contents) and the concrete age (curing). The output parameter was the tensile strength of rubberized concrete specimens as a fraction of the strength of the plain concrete.

Kasperkiewicz et al. (1995) applied artificial neural network of the fuzzy-ARTMAP type for predicting strength properties of high-performance concrete mixes. A significant enough correlation between the actual strength values and the values predicted by the neural network was observed. It was demonstrated that the approach can be used in multicriterial search for optimal concrete mixes.

Mukherjee, Schmauder and Ruhle (1995) presented artificial neural networks to model the mechanical behavior of metal matrix composites (MMC). The hardening behavior of metal matrix composites has been modeled using artificial neural networks. The process is highly non-linear and it has been difficult to develop a macro-mechanical model, theoretical or empirical, from the experiment or analytical studies in the micro-mechanics of these materials. The artificial neural network performed better than the empirical relationships developed by experts for the problem. In addition, the ANN was
able to predict the strengthening of MMCs for the loading directions and matrix hardening.

Rao (1995) demonstrated artificial neural networks in macro-mechanics of weak-fiber matrix composites. The available material models in this area are too simplistic to be of practical significance. Finite element analysis has been carried out to generate the training examples for the network. The results of the finite element analysis were synthesized into a neural network obviating the use of the expensive finite element analysis to predict the macro-mechanical behavior. An artificial neural network has been used to postulate the constitutive law for $\text{Al}_2\text{O}_3$ (matrix) / SiC (whisker) and SiC / Si composites.

Goh (1995) have applied neural networks for finding ultimate shear strength of deep reinforced concrete beams. The network was trained with six input neurons, (the geometrical characteristics for the beam depth, width and length, the percentage of the horizontal and vertical web reinforcement and the value of the compressive strength of the concrete) to predict the ultimate shear force in concrete beams.

Fletcher and Coveney (1995) applied ANNs to predict the thickening times of the cement slurries used in oil wells.

The research on the failure of the columns has a long history (Bruce, 1983). The departure from the assumptions of the elastic plastic theory makes the task of incorporating all the features of real life columns into a single formula is very difficult. A number of investigators have proposed semi empirical formulae for the critical buckling load of slender columns. Semi empirical formulae, adapted for design specifications often follow a lower bound to experimental observations to include a variety of column types. Therefore, a significant portion of the actual column strength remains unutilized, when such a lower bound is adapted in the design of axially loaded
compression members. Mukherjee, Deshpande and Anmala (1996) reported an ANN tool for prediction of buckling load of columns. The ANN was trained from experimentally observed failure loads avoiding the assumptions associated with the mathematical models.

Patodi et al. (1997) correlated non destructive testing parameters to the strength of the structure. As there is no direct relation between rebound number and concrete strength or pulse velocity, the development of an ANN simulator seems to be the natural choice for such problems because pre defined mathematical relationship among the variables is not required in an artificial neural network.

The behavior of concrete structures that are exposed to extreme thermo-mechanical loading is an issue of great practical importance. The mechanical behavior of concrete at high temperature is extremely complex. In addition, the constituent materials, influence the response significantly, it is very difficult to include all the contributing factors in to a mathematical model. Mukherjee and Nag Biswas (1997) presented a new approach to the problem by applying artificial neural networks. Implementing a feed forward network and back propagation algorithm, the stress-strain relationship of the material was captured. The behavior of concrete under uniaxial loading was modelled.

Hegazy et al. (1998) used neural networks as a means to develop efficient predictive models of the structural behavior of concrete slabs. Four neural networks were developed to model the load-deflection behavior of concrete slabs, the final crack-pattern formation and both the reinforcing-steel and concrete strain distributions at failure. The developed tool was useful for teaching purposes and for reasonable prediction of the behavior of concrete slabs without additional experimental testing.

Patodi and Purani (1998) used feed forward neural networks for predicting the flexural behavior of fibre reinforced concrete beams. The flexural behavior of two
different types of steel fibre reinforced concrete beam problems was modeled using neural networks. The results obtained for both the problems were found to be in excellent agreement with the actual experimental values.

Woo ho et al. (1999) applied artificial neural network as a tool to minimize the uncertainties and errors of proportioning concrete mixes. The required compressive strengths and also the actual compressive strengths with variations obtainable from the final compressive strength test were used to train and test the network. The results show that neural networks have a strong potential as a tool for concrete mix proportioning.

Basma et al. (1999) presented a neural network model for the predicting the degree of hydration of cement. The results indicated that the neural networks are very efficient in predicting degree of hydration with great accuracy using minimal processing of data.

Tam and Fang (1999) used artificial neural networks to establish the relationship between the quantities/costs of concrete and formwork required for the structural elements of high-rise commercial buildings. The neural network models were proven to be accurate in predicting the costs of using high-performance concrete in wall-frame structures for high-rise building construction.

Patodi and Satodia (1999) applied back propagation algorithm to predict the behavior of fibre reinforced concrete deep beams using a menu driven simulator developed in Fortran90. A generalized delta rule was used to train the networks based on the existing experimental results for two types of deep beam problems i.e., FRC deep beams with and without reinforcement. In case of FRC deep beam without reinforcement, four inputs (length of beam, shear span, span/depth ratio and percentage fibre content by weight) were related to five outputs(first cracking load, failure load, maximum average shear stress, maximum experimental moment at failure and theoretical maximum...
moment) using one hidden layer with seven nodes. The network was given 6,80,000 cycles of training. The maximum error in predicted values for failure load, cracking load, maximum average shear stress, maximum moment at failure and theoretical maximum moment was found as 3.83%, 9.60%, 3.74%, 0.41% and 0.14% respectively. In the case of FRC deep beam with reinforcement, a topology of 7-9-4 was selected to relate seven inputs (X/D ratio, L/D ratio, longitudinal steel area, area of horizontal web reinforcement, area of vertical reinforcement, fibre aspect ratio and fibre volume fraction) to four outputs (experimental and theoretical cracking and ultimate loads). The network was tested with twenty patterns which indicated a maximum error of 4.83% in predicted output.

Kasperkiewicz (2000) discussed the applications of ANNs in concrete technology, where a main obstacle is lack of reliable data base. The application of ANNs for predicting properties in the simulation of concrete-like composite materials in design and in optimization were also reported.

Ni Hong-Guang et al. (2000) proposed a method to predict 28-day compressive strength of concrete by using multi-layer feed-forward neural networks. A model was built to implement the complex non-linear relationship between the inputs and the output. The neural network model with eleven input neurons, a single hidden layer with seven neurons and output layer with a single neuron was used.

Nehdi et al. (2001) investigated the use of ANN to predict the performance of cellular concrete mixtures. It was shown that production yield, foamed density, unfoamed density and compressive strength of cellular concrete mixtures were predicted much more accurately using the ANN method compared to existing parametric methods.

Rami Haj–Ali et al. (2001) proposed simulated micromechanical models using artificial neural networks, to generate micromechanical models for non-linear and damage behavior of heterogeneous materials. Artificial neural networks were trained with results
moment) using one hidden layer with seven nodes. The network was given 6,80,000 cycles of training. The maximum error in predicted values for failure load, cracking load, maximum average shear stress, maximum moment at failure and theoretical maximum moment was found as 3.83%, 9.60%, 3.74%, 0.41% and 0.14% respectively. In the case of FRC deep beam with reinforcement, a topology of 7-9-4 was selected to relate seven inputs (X/D ratio, L/D ratio, longitudinal steel area, area of horizontal web reinforcement, area of vertical reinforcement, fibre aspect ratio and fibre volume fraction) to four outputs (experimental and theoretical cracking and ultimate loads). The network was tested with twenty patterns which indicated a maximum error of 4.83% in predicted output.

Kasperkiewicz (2000) discussed the applications of ANNs in concrete technology, where a main obstacle is lack of reliable data base. The application of ANNs for predicting properties in the simulation of concrete-like composite materials in design and in optimization were also reported.

Ni Hong-Guang et al. (2000) proposed a method to predict 28-day compressive strength of concrete by using multi-layer feed-forward neural networks. A model was built to implement the complex non-linear relationship between the inputs and the output. The neural network model with eleven input neurons, a single hidden layer with seven neurons and output layer with a single neuron was used.

Nehdi et al. (2001) investigated the use of ANN to predict the performance of cellular concrete mixtures. It was shown that production yield, foamed density, unfoamed density and compressive strength of cellular concrete mixtures were predicted much more accurately using the ANN method compared to existing parametric methods.

Rami Haj-Ali et al. (2001) proposed simulated micromechanical models using artificial neural networks, to generate micromechanical models for non-linear and damage behavior of heterogeneous materials. Artificial neural networks were trained with results
from detailed non-linear finite element analysis of a repeating unit cell, with and without induced damage, e.g., voids or cracks between the fibre and matrix phases. The finite element simulations are used to form the effective stress-strain response for a unit cell with different geometry and damage parameters. It is shown that models of this type exhibit many interesting features, including different tension and compression responses that are usually difficult to model by conventional micromechanical approaches. This type of models can be easily applied in a displacement-based finite element non-linear analysis of composite structures.

Sanad and Saka (2001) explored the use of artificial neural networks in predicting the ultimate shear strength of reinforced-concrete deep beams. It was reported that neural networks provide an efficient alternative method in predicting the shear strength capacity of reinforced-concrete deep beams where several equations exist, none of which produce an accurate result.

Abdullateef et al. (2002) developed the functional relationship between the fire resistance of a concrete filled steel column and the parameters which cause the fire resistance using an artificial neural network model. The model predicted values were compared with actual test results. The results indicated that the model can predict the fire resistance of the composite element and characteristics of its material composition and the fire environment with average operational error of 8.5%.

Tang et al. (2003) have demonstrated the application of neural networks to predict the complicated non-linear behavior of reinforced concrete columns with rectilinear transverse steel. Fifty five experimental test results were collected from the literature. A comparative study between the neural network model and four parametric models was also carried out. It was found that the neural network model could reasonably capture the
underlying behavior of confined reinforced concrete columns. It was concluded that when compared with parametric models, the neural network approach provides better results.

Cho (2003) applied neural networks for dispersive characteristic measurement of multi-layer cement mortar slabs using neural networks. The training sets were prepared using the training pairs obtained from the actual experimental results and computed velocities. It was observed that the network predictions were in close agreement with the experimental results.

Ragip Ince (2004) used artificial neural network based modeling as an alternative way to model the material behavior which generally involves the development of a mathematical model derived from observations and experimental data. In this paper the Two-Parameter Model (TPM) in the fracture of cementitious materials was modeled with a back-propagation ANN. The results of an ANN-based TPM look viable and very promising.

Flood et al. (2004) developed a new approach for simulating the thermal behavior of buildings using neural networks. The thermal behavior of each coarse-grain element was captured using artificial neural networks. The approach was shown to sustain highly accurate long-term simulation runs, if the coarse-grain modeling elements were implemented as ANNs.

Cladera and Mary (2004) developed artificial neural network model to predict the shear strength of reinforced and high strength concrete beams with stirrups failing on diagonal tension failure and based on its results, a parametric study was carried out to study the influence of each parameter affecting the shear strength of beams with web reinforcement. Non-linear relationship between the amount of stirrups and the shear strength was also reported. New design expressions were developed, taking into account the observed behavior for the design of high-strength and normal-strength reinforced
concrete beams with shear reinforcement. The new expressions correlated much better with the empirical tests than EC-2 or ACI procedures.

Sedat Akkurt et al. (2004) developed a fuzzy logic prediction model for the 28-day compressive strength of cement mortar under standard curing conditions. Data collected from a cement plant were used in the model construction and testing. The prediction of fifty sets of the 28-day cement strength data by the developed fuzzy model was quite satisfactory. The model was compared with the ANN model for its error levels and ease of application.

Kim et al. (2004) applied neural networks for estimating the compressive strength of concrete and shown the effectiveness of the neural network-based technique in estimating the compressive strength of concrete based on concrete mix proportion parameters before placing the concrete. The input for the network were water-cement ratio, fine aggregate percentage, unit water content, unit cement content, unit fine aggregate content, unit coarse aggregate, admixture and slump. The output of the network was compressive strength of the concrete. The maximum errors between the estimated and tested results were 3.9% in the specified strengths and 3.2% in the required average strengths.

Labossière (2005) applied neural networks to predict failure of anisotropic materials under any loading condition. An example of failure envelope for a typical fibre-reinforced material was illustrated.

Sakla and Ashour (2005) attempted to predict the tensile capacity of single adhesive anchors using artificial neural networks. The predictions of ANN show that the tensile capacity of adhesive anchors was linearly proportional to the embedment depth.

Seleemah (2005) applied the artificial neural network technique as an analytical alternative to existing methods for predicting the shear capacity of concrete beams
without transverse reinforcement. Both ANN and eight different codes and researcher's predictions of the shear capacity of the specimens of the database were compared. The ANN predictions were shown to be much superior to those of any of the current available relationships.

Cheng Yeh (2006) described a method of modeling slump of fly ash and slag concrete using artificial neural networks. The model built was examined with response trace plots to explore the slump behavior of fly ash and slag concrete. It is concluded that response trace plots can be used to explore the complex non-linear relationship between concrete components and concrete slump.

Mo and Han (2006) used neural networks in concrete modeling. The network was presented with the experimental data and learned the relationships between stresses and strains. The behavior of prestressed concrete frames was modeled with a back propagation neural network.

Jung and Ghaboussi (2006) developed a neural network constitutive model which adjusts itself to describe given stress and strain relationship. They developed a rate-dependent NN constitutive model formulation and its implementation in finite element analysis. The NN model was verified for a standard solid visco elasticity model. The model was then applied to analysis of time-dependent behavior of concrete. It is reported that the model had potential of capturing any rate-dependent material models, provided enough data sets were given.

Cheng Yeh (2006) investigated the potential of using design of experiments and neural networks to determine the effect of fly ash replacements for different amounts. It was reported that high correlations between the compressive strength and the component composition of concrete was developed using the generalization capabilities of the neural networks.
Gupta et al. (2006) used ANN as an attempt to obtain more accurate concrete strength prediction based on parameters like concrete mix design, size and shape of specimen, curing technique and period, environmental conditions, etc. The predicted strengths found by employing ANN were compared with the actual values.

2.2.4 Neural networks in damage detection

The tools of structural mechanics are efficient in predicting the response of the structure due to excitation. These tools however, poor in reconstructing the structure from known loads and response. As a result, it is difficult to develop a detection tool using structural mechanics. As a result, existing tools for detection of damage are limited in many ways.

- Most of them model damage as a reduction in member stiffness, while damage occurs at zones of high stresses
- Most of them work on the basis of deviation in dynamic response (Dynamic fingerprinting) while the deviation in dynamic response may be caused by many other factors.
- The noise tolerance of the tools must be guaranteed.
- Most of them are based on higher order information such as strain or curvature mode shapes. Although these can be measured in laboratory they are very difficult to measure in the field.

Artificial neural networks map the known input (response and load of the structure) to the output (damage state). Moreover, they are able to generalize the input-output relationship. Therefore, the detection tools can be developed directly from the measured response without having to describe the mechanics behind. This is the main attraction of ANN's in damage detection. Some networks also have the ability to work with incomplete information and with input noise. A few applications of ANN in damage detection are presented below.
Wu et al. (1992) have reported an experiment, by modelling a three storied frame. The building was modeled as a shear building with girders infinitely rigid and the columns flexible. This three degree of freedom structure was subjected to an earthquake base acceleration and the transient response was computed in time domain. The Fourier spectra of the computed relative acceleration time histories of the top floor were used for training the network. The Fourier spectra between 0 to 20 Hz were discretised at an interval of 0.1 Hz and the resulting two hundred spectral values were given as input to the network. The output layer contains one node per member and the activation values at these nodes represent the damage state in the member. The authors concluded that the section of suitable location of the measurement transducers or accelerometers is of primary importance in correctly capturing the vibration characteristics of the structure.

Yeh et al. (1993) demonstrated the use of ANN for diagnosing damage of prestressed concrete piles. The net predicted the cause of damage from observed features as input. Both input and output were in binary form. A network with eighteen input nodes and twelve output nodes has been trained using one hundred and twenty examples. The network also used a hidden layer with eighteen nodes. A study of network learning by changing various network parameters was also presented.

The use of neural networks for damage detection in structures based on feature sensitive neural networks was reported by Szewchzyk and Hajela (1994). The detection of damage in structural systems was formulated as an inverse problem. The damage was modeled by reducing the stiffness of structural element. The damaged member is detected from the deviation observed in static displacements under prescribed loads. A modified counter propagation network was developed for this inverse mapping. The object of the network is to identify the components $K_{ij}$ in the stiffness matrix given the experimentally observed response data that are dependent on the stiffness matrix $K$. 
Elkordy et al. (1993) attempted detection of damage in the bracing members of a multi-storied steel frame. Experiments were conducted on a model by vibrating it on a shake-table. The damages were introduced by replacing the bracing members with a member of smaller cross-sectional area. Both the strain and displacement mode shapes were used. The deviation in response from the undamaged to the damaged structure was used for training a back-propagation neural network. The network was trained with the experimentally observed response of the structure with a predetermined extent of damage.

Pandey and Barai (1995) presented an application of multilayer perceptron in the damage detection of steel bridge structures. The training patterns were generated for multiple damaged zones in a structure. The engineering importance of the whole exercise was demonstrated from the fact that the measured input at only a few locations in the structure is needed in the identification process using the ANN.

Gilbert and Yamazaki (1995) proposed a back propagation neural network for a quick estimation of the damage due to an earthquake from the indices recorded ground motion. A sensitivity study was carried out to identify the most influential input parameters. It was also indicated that instability in the net occurred when the input crossed the range of training. The network performed satisfactorily within the range of training.

Chen and Kim (1995) used a back propagation neural network as an information processing technique for a three-dimensional steel truss-type bridge structure, instrumented with accelerometers and strain gauges. The feasibility of identifying and locating structural damage using the Matlab neural networks tool box was studied. Vibration signals were measured from a series of experiments performed on the instrumented scale model bridge structure. Four networks were involved in their study

(1) One identified damage using accelerations data; (2) One located damage in the model
bridge using strain data; (3) One determined both the damage and the location using the simulated sensor signals by finite-element analysis and the accompanying accelerometer signals; (4) One was trained with acceleration data produced by finite-element model. Damage was introduced through different cuts in the flanges and the webs of the girders. An impact hammer was used for excitation. The study of Chen and Kim led to the following conclusions: (1) The recognition of damage severity was less accurate than that of damage detection and location; and (2) sensor signals used as inputs to the networks were feasible for damage for damage identification and location in this model bridge structure instrumented with accelerometers and strain gauges.

Durga Prasad et al. (1996) have applied ANN for damage assessment of structures. They have used eighteen structural failures out of twenty four structural failures reported by earlier researchers for training the network. The merits of ANN over artificial intelligence and expert systems were presented. The network predicted the damage satisfactorily and captured the non-linear interdependency of the parameters effectively.

Faravelli and Pisano (1997) presented a neural network based method for damage detection in multi-bay planar truss structures. The neural network approach was able to uniquely identify the damaged element in almost all of the investigated cases.

Mukherjee and Kamath (1997) presented Kohonen architecture for the detection of damage in the multi-story frame buildings. They attempted to introduce the damage realistically (i.e. at zones of high stress concentration rather than through out the member). They conducted experiments to observe the response of a damaged frame. The same response was emulated in a mathematical model. The mathematical model was used to observe response of other damage states. The self organizing network could classify the response for the frame. The noise tolerance of the network was also examined.
Amini et al. (1997) presented simulation and experimental studies of identification of damage in civil engineering structures using neural networks. The neural network was trained and examined using the measured structural responses under different earthquake loading conditions. It was shown that the trained neural network was capable of providing sensible outputs when presented with input data that has never been used during its training.

Jun Zhao et al. (1998) used counter propagation neural networks to locate structural damage for a beam, frame and support movements of a beam in its axial direction. The investigation considered a variety of diagnostic parameters, including static displacements, natural frequencies, mode shapes and other parameters based on mode shapes. The method was first demonstrated on a plane frame, based on static displacements. It was then applied to continuous beams using dynamic properties of structures. The required data was obtained through computer simulation by finite-element analysis. The results demonstrated that these parameters can be used as diagnostic parameters for artificial neural networks in structural engineering.

Stefano et al. (1999) studied the fundamental learning and generalization capabilities of neural networks to obtain an estimate of the vulnerability of structural systems. Kert and Yee (2000) analyzed the deformed behaviors for culvert structure under a static loading by applying dynamic backpropagation neural networks with energy function as minimization index. The training process was avoided by using stiffness matrix and force vector of the structure instead of using weighting matrix and bias vector in the neural networks calculations. The ability of neural networks was verified by comparing the results with analytical solutions and finite element solutions.

Chang et al. (2000) developed structural damage detection method based on parameter identification using an iterative neural network technique. The trained NN
model was used to predict the structural parameters by feeding in measured dynamic characteristics. The predicted structural parameters were then used in the finite element model to calculate the dynamic characteristics. The results indicated that this approach can identify both the location and the extent of damages in the beam.

Lopes et al. (2001) presented a non-model based technique to detect, locate and characterize structural damage by combining the impedance-based structural health monitoring technique with an artificial neural network.

Li et al. (2001) presented crack damage detection algorithm using a combination of global (changes in natural frequencies) and local (strain mode shapes) vibration-based analysis data as input in artificial neural networks for location and severity prediction of crack damage in beam-like structures. Finite element analysis was used to obtain the dynamic characteristics of intact and damaged cantilever steel beams for the first three natural modes. The necessary features for crack detection were selected and introduced to the ANNs. Finally, the Radial Basis Function ANNs were trained using the data obtained from the numerical damage case to predicate the severity and localization of the crack damage.

Huei Tsai and shiu Hsu (2002) proposed a methodology for damage diagnosis of structures using neural networks. The methodology was applied on a simply supported reinforced concrete beam. The damage scenario of each test beam was also diagnosed by using the well-trained NN according to the displacement time history, which is the history of the responses caused by the impact loading acting at the beam centers. Based on the study and test results, the damage scenarios of the ten sets of test beams were successfully classified.

Wu et al. (2002) presented an approach for damage detection of sub-structures of a multi-degree-of-freedom structure system by using neural networks. By using the
trained decentralized detection neural networks, the difference of the interstory restoring force between the damaged sub-structures and the undamaged sub-structures was calculated. It was shown that the decentralized parametric evaluation method has the potential of being a practical tool for a damage detection methodology applied to structure-unknown smart civil structures.

Ni et al. (2002) applied neural networks for hierarchical identification of structural damage location and extent from measured modal properties. The damage detection capacity of constructed networks was experimentally verified on a steel frame with extent-unknown damage inflicted at its connections and the applicability of the hierarchical identification strategy to cable-supported bridges was discussed.

Demetriou and Hou (2003) presented a performance study of two fault detection/diagnosis approaches, namely the wavelet-based and the neural-network-based on-line schemes. Numerical studies have shown that both approaches may be successfully implemented on-line to detect the times when faults occurred and to locate regions where faults occurred. It was demonstrated that these schemes can be successfully applied for on-line structural health monitoring.

Kao and Hung (2003) presented a neural network-based approach for detecting structural damage. Numerical and experimental examples demonstrated that the proposed method was feasible for detecting structural damage.


Li et al. (2005) presented a neural network-based damage detection method using the modal properties which can effectively consider the modeling errors in the baseline finite element model from which the training patterns are to be generated. The differences
or the ratios of the mode shape components before and after damage are used as the input to the neural networks in this method, since they are found to be less sensitive to the modeling errors than the mode shapes themselves. Two numerical examples on a simple beam and a multi-girder bridge were presented to demonstrate the effectiveness of the developed method. Results of laboratory test on a simply supported bridge model and field test on a bridge with multiple girders confirm the applicability of the proposed method.

Fang et al. (2005) explored the structural damage detection using frequency response functions as input data to the back-propagation neural network. The analysis results on a cantilevered beam show that, in all considered damage cases the neural network can assess damage conditions with very good accuracy.

Hung and Kao (2005) presented a neural network-based approach to detect structural damage. Examples were presented to demonstrate the feasibility of using the proposed method for damage detection of linear structures.

Nakamura et al. (2005) presented neural network-based approach for the detection of changes in the characteristics of structure-unknown systems. The trained network, which was subsequently fed data before the repairs, successfully identified the difference between the damaged storey and the undamaged storey. It was shown that the proposed approach has the potential of being a practical tool for a damage detection methodology applied to smart civil structures.

Antony Jeyaschar and Sumangala (2006) developed an artificial neural network based approach for the assessment of damage in prestressed concrete beams from natural frequency measurements. Comparison was made in between ANN trained only with natural frequency data and ANN trained with a mix of static and dynamic data. It was
demonstrated that an ANN trained with dynamic data assessed the damage with less than 10% error.

Xu and Jag Mohan (2006) presented a new robust two-step algorithm that used the modal energy-based damage index to locate the damage and an artificial neural network technique to determine the magnitude of damage. The proposed algorithm was applied to detect simulated damage in a finite element model of a girder and a similar model of a real bridge named Crowchild Bridge located in Alberta, Canada. The results show that the proposed algorithm was quite effective in identifying the location and magnitude of damage, even in the presence of measurement errors in the input data.

Fai Lam et al. (2006) employed artificial neural network technique as a tool for systematically identifying the damage pattern corresponding to an observed feature. A design method based on a Bayesian probabilistic approach for model selection was proposed. A truss model was employed to demonstrate the proposed methodology.

Kao and Hung (2006) presented a neural network based-approach for detecting structural damage. The proposed approach involved two steps. The first step, system identification, used Neural System Identification Networks (NSINs) to identify the undamaged and damaged states of a structural system. The second step, structural damage detection, used the aforementioned trained NSINs to generate free vibration responses with the same initial condition or impulsive force. Comparing the period and amplitude of the free vibration responses of the damaged and undamaged states allowed the extent of changes to be assessed. An experimental example demonstrated the feasibility of applying the proposed method for detecting structural damage.
2.2.5 Neural networks in other inverse problems

There are several inverse problems in structural engineering at present. For example, a smart structure needs to find out the precise magnitude and location of load from an input of displaced shape of the structure. This is an inverse problem which is difficult to solve with the existing tools of mechanics. Moreover if the load is a moving one, the solution to the problem must be obtained very quickly. This is definitely a challenging task.

ANNs are very effective in such situations. The ANNs work intuitively and they need not solve the inverse problem. Moreover as they are massively parallel processing systems, they can offer a very fast solution to the problem. Therefore, they have the ability to work in an on-line situation. Although there are not many published investigations, this field is certainly going to gain importance in future.

Srivastava et al. (1994) dealt with the vibration control of smart laminated fiber-reinforced plastic composites using back propagation neural networks. To simulate the system dynamics, a finite element model for smart layered fibre-reinforced plastic composites was developed, which could incorporate any number of piezoelectric layers in addition to the usual composite layers. To control the vibrations of smart FRP laminated composites, back propagation neural network was used. The active vibration control performance of beams/plates with piezoelectric sensors and actuator layers was studied using this model. It was observed that the use of neural network can control the displacement reasonably well.

Ghaboussi and Joghatie (1995) developed a neural network to forecast the response of a structure plus actuator from the response history of the structure. Another network estimates the controlling force to reduce the displacements. A feed forward network with back propagation learning was used.
Stavroulakis and Antes (1997) considered an inverse problem in non-linear elastostatics which concerns the identification of unilateral contact cracks by means of boundary measurements for given static loadings using neural networks. The applicability of the method was demonstrated by some numerical examples.

Chikata et al. (1998) studied the scenery evaluation which depends on the evaluator’s experiences or perceptions. Inverse analysis by neural networks of scenery evaluation of concrete retaining walls was examined in this study.

Yoshimura et al. (1998) applied neural networks for structure identification. This paper proposed a new regularization method suitable for the inverse analysis approach. The neural network learned the training patterns efficiently as well as accurately.

Jones et al. (1999) investigated experimentally neural network-based method of determining the location and magnitude of transverse impact events on isotropic plates. The neural network model was able to determine the impact magnitude with an average of 13.8% error.

Mathew et al. (1999) utilized the capability of artificial neural networks in solving complex non-linear problems for the analysis of masonry panels under biaxial bending. An artificial intelligence based technology was used to solve new problems by adapting solutions to similar problems solved in the past, which were stored in the case library. In this paper a hybrid system was described that utilizes the capabilities of both ANNs and the case-based reasoning.

Patodi and Sushanth Singh (1999) used neural networks to evaluate large deflection response of fixed immovable rectangular plates subjected to patch loading. The three inputs for the network are plate aspect ratio, the patch size and pressure coefficient. The eight outputs are the central deflection, bending and membrane stresses in the x and y directions at key locations of the rectangular plate. A neural network with
4-30-25-8 configuration was trained with ninety training sets for different combinations of plate aspect ratio, patch size and lateral pressure values. The error after 44,000 epochs was 0.00198.

Lu et al. (2001) sorted out the relationships between an output variable and an input parameter based on the BPN algorithm. The sensitivity analysis of the BPN was successfully applied to analyze the labor production rate of pipe spool fabrication in a real industrial setting.

Lin and Ghaboussi (2001) presented a new stochastic neural network that was capable of generating multiple earthquake accelerograms from a single-response spectrum. The proposed method produced a stochastic ensemble of earthquake accelerograms from any response spectra or design spectra. An example was presented that used hundred recorded accelerograms to train the neural network and several design spectra and response spectra to test this improved methodology.

Liu and Kapania (2001) developed a method of modeling trapezoidal built-up wing structures by coupling, in an indirect way, an equivalent plate analysis (EPA) with neural networks. Neural networks for the material properties were trained in terms of the design variables of the wing structure.

Jan et al. (2002) predicted the diaphragm wall deflection by using the adaptive supervised neural network. Simulation results indicated that the artificial neural network can reasonably predict the magnitude, as well as the location, of maximum deflection of the diaphragm wall.

Busheer (2002) investigated on the effectiveness of Time-Delay ANNs (TDANNs) in mapping hysteresis behavior of geomaterials under repeated loading reversals. Using sequential one-lag data and dynamic training, TDANNs were found to be viable tools for modeling the hysteresis behavior in loading reversal environment and
could be used to simulate such behavior with high accuracy for an unlimited number of cycles within and beyond the training data domain. A non-linear recursive simulator was also developed containing the TDANN to enable forecasting of complete (σ-ε) curves from the knowledge of only the initial σ-ε condition of the tested material.

Chen and Chang (2002) proposed a neuro-fuzzy recognition approach (NFRA) and introduced a random sampling plan to help select positions for image taking during steel bridge coating inspections.

Reda Taha et al. (2003) examined the ability of ANNs in predicting creep. ANN model was applied to the prediction of creep of structural masonry. The ANN developed was able to predict the creep performance with good accuracy compared with that of conventional models.

Martin et al. (2004) developed artificial neural receptor system for structural health monitoring. An artificial neural network analysis was used to extract excitation locations and individual sensor strains from the array.

Lin et al. (2006) used the neural networks in active control. Experimental results for a full-scale steel frame structure with a smart active control system under shaking-table testing were presented. Experimental results demonstrated that the control system can be applied to buildings after the whole testing process.

Mujica and Joseph (2006) presented two approaches for structural damage identification, each based on a different philosophy. The virtual distortion method (VDM) was a model-updating method of damage assessment; Case-based reasoning (CBR) was a soft computing method utilizing wavelet transformation for signal processing and neural networks for training a base of damage cases to use for retrieving a similar relevant case. A numerical example of a beam was presented including a demonstration of the complexity of the inverse problem.
2.2.6 Neural networks in fast analysis

The modern tools of mechanics (e.g. FEA) can analyze a structure with a great deal of accuracy. They may, however, take a large amount of time and effort to perform the analysis. The time requirement multiplies when non-linearity is involved. It has been mentioned earlier that design is an iterative procedure. Therefore, usually the analysis is repeated several times to obtain the final result. The increased processing offered by the ANNs can be utilized to obtain an estimate of the response before carrying out a detailed analysis. This is helpful in reducing the number of iterations.

Rehak et al. (1989) published the first research paper in this area to demonstrate the development of ANN for the dynamic analysis. They attempted to make a neural network learn the relationship between response in terms of acceleration, velocity and displacement at time step i (corresponding to time $t = \Delta t$) and the corresponding response at the time step $i+1$. The input for the network was chosen as response at the previous step and the forcing function. The network was expected to predict the response at next step. The initial attempts to simulate a two degree of freedom system could not be successful. Therefore, a single degree of freedom system was investigated. The convergence of the network was studied for a single degree of freedom system.

Hajela and Berke (1991) reported the use of ANN for mapping the load-displacement relationship for use in determining the desired displacement or response in static structural analysis for a five bar truss problem. This load-displacement relation was mapped for use in an optimization algorithm to obtain a quick analysis. The displacement at predicted joints was considered as output for the network. The experiments with different networks configured for the five bar truss problem were reported. The use of ANN for an optimization procedure was suggested using a sequence in the architecture to include the neural network analysis routine. The extension of this
idea for application of ANN to a ten bar truss problem was also reported. The numerical experimentation was performed to watch the performance of different network configurations. Networks with 10-11-8 configuration (i.e. 10 input nodes, 11 hidden layers and 8 output nodes), 10-11-2 configuration and 10-6-6-2 configuration were studied. Different nodes of the truss configuration were considered for each configuration to obtain the output from the network. Hajela and Berke concluded that to realize the increased learning rate in network, proper input enhancements must be presented to the network.

Hajela and Berke (1992) have presented an overview of a variety of network configurations for structural engineering applications. A network with ten input nodes, two hidden layers with six nodes each and two output nodes was developed to predict the displacement at predetermined joint of the plane truss.

Chen and Shah (1992) studied the changes in frequencies and displacements of a pier in an actual bridge structure using a back propagation neural network. The loads on the pier were caused by normal vehicle traffic. They have used accelerations as inputs to the network. The target outputs were the frequencies, displacements and displacement mode shapes obtained from signal processing. The network was trained with the data obtained in two stages. (1) Before the explosion demolished three near by buildings and (2) after the explosion. The network successfully detected the changes in the frequencies and displacements of the pier due to explosion.

Rix (1994) studied the presence and absence of defects in foundation piles using transient dynamic response test method to obtain accelerations of the piles and to train a three layer back propagation neural network. In laboratory tests, the network correctly identified the presence or absence of the defects in 98.4% of the test cases. No attempt was made to identify the location and nature of the defect.
Murtaza and Fisher (1994) presented an approach for decision making about construction modularization using neural networks. The performance of the trained neural network system was compared with the recommendations provided by human experts. The results of statistical tests performed to validate the system were also presented.

Williams and Gucunski (1995) applied neural-network models to perform the inversion of the spectral-analysis-of-surface-waves (SASW) test results. Three, four and five-layer back-propagation models were employed. All of the neural-network models produced results that were reasonably close to the actual output. The results indicated that back-propagation neural networks were useful for performing the inversion procedure of SASW tests.

Cattan and Mohammadi (1997) described the application of neural network systems in developing the relation between subjective ratings and bridge parameters as well as that between subjective and analytical ratings. It was shown that neural networks were trained and used successfully in estimating a rating based on bridge parameters.

Manevitz et al. (1997) applied neural networks to the problem of mesh placement for the finite element method. In this paper the self-organizing algorithm of Kohonen was adapted to solve the problem of automatically assigning (in a near-optimal way) coordinates from a two-dimensional domain to a given topologic grid (or mesh) of nodes in order to apply the finite element method effectively when solving a partial differential equation with boundary conditions over that domain.

Rajasekaran et al. (1998) used artificial neural network in civil engineering. In this work, an artificial neural network was itself trained for the difuzzification process. The training was carried by back propagation algorithm. It was found that normalization of input and output vectors to 0-1 gives improved performance of the networks.
Sinha and McKim (2000) developed a methodology for predicting the level of organizational effectiveness in a construction firm. A multilayer back-propagation neural network based on the statistical analysis of training data was developed and trained. Findings show that by applying a combination of the statistical analysis and artificial neural network to a realistic data set, high prediction accuracy is possible.

Li (2000) presented Global Flexibility Simulation approach and Element Stiffness Simulation (ESS) to simulate finite element analysis with neural networks. Principal procedures of ESS approach, which include network construction, constrain application, network solving and network learning, were presented according to conventional finite element analysis procedures. Simulation results were then compared with finite element method results. The effect of analysis scale was also discussed.

Lu et al. (2000) discussed the derivation of a probabilistic neural network classification model and its application in the construction industry. The probability inference neural network (PINN) model was tested on real historical productivity data at a local construction company and compared to the classic feed forward back-propagation neural network model. This showed marked improvement in performance and accuracy.

Hyo Kim et al. (2000) developed a design procedure for a structure's monitoring system using sensitivity analysis and a neural network. Truss and frame examples were used to show the validity and applicability of the monitoring system design procedure.

Consolazio (2000) used artificial neural networks as a domain knowledge-encoding mechanism, together with a preconditioned conjugate gradient iterative equation-solving algorithm to derive iterative equation solver for bridge analysis. In the algorithm, neural networks were used to seed the initial solution vector and to precondition the matrix system using customizable and trainable neural networks. A case study was presented on flat-slab highway bridge analysis. Analytical load-displacement
data was generated using finite-element analyses and subsequently used to train neural networks. Acting collectively, the neural networks predicted approximate displacement patterns for flat-slab bridges under arbitrary loading conditions.

Iranmanesh, Kaveh (2000) presented a neurocomputing strategy which combines data processing capabilities of neural networks and numerical structural optimization. Two artificial neural networks were trained, one for the constraints and the other for the gradients of the constraints and structural optimization was accomplished by using these nets. All required parameters such as weight matrices in the neural networks or the gradient computations were automated in this neuro-optimizer strategy. Numerical examples were included to demonstrate the accuracy of the results.

Rafiq et al. (2001) presented practical guidelines for designing ANN for engineering applications. Practical guidelines on data selection, NN training using a modified and extended version of the Jenkins ‘hyper cube’ was discussed and tested. It was shown that this method can drastically reduce the size of the training data. A brief introduction to NN was given; major aspects of three types of NN, multi-layer perceptron (MLP) radial basis network (RBF) and normalized RBF (NRBF) were discussed. When compared to three types of networks the MLP and NRBF performed equally good. The RBF showed a poorer performance. New methods for selection and normalisation of training data were introduced and a practical example of a reinforced concrete slab design was presented.

Huang and Loh (2001) proposed a neural-network-based method for the modeling and identification of a discrete-time non-linear hysteretic system during strong earthquake motion. It was shown that a multilayer neural network was suitable for the extreme non-linear input-output mapping problems. Numerical simulation of a three-story building and a real structure (a bridge in Taiwan) subjected to several recorded earthquakes were used
to demonstrate the proposed method. The results illustrated that the neural network approach was a reliable and feasible method.

Kim and Lee (2001) applied neuro-controller training algorithm based on cost function to a multi-degree-of-freedom system: and a sensitivity evaluation algorithm replacing the emulator neural network was proposed. Numerical examples showed that the proposed control algorithm was valid in structural control.

Jenkins (2002) applied neural network as a computational device for continuous updating of the analysis with provision for a wide range of structural design modifications including those affecting geometry, topology, nodes, loads, supports and material properties. The practicality of the method was justified by the remarkable increase in processing speeds and memory capacities of modern computers.

Chang and Li Zhou (2002) explored the use of neural networks for emulation of inverse dynamics for a magnetorheological damper (MR). Recurrent NN models were constructed to emulate the inverse dynamics of the MR damper. Numerical results indicated that, using the recurrent NN models, the MR damper force can be commanded to follow closely the desirable optimal control force.

Domer et al. (2003) described the use of neural networks to improve the accuracy of the dynamic relaxation method in order to correspond more closely to data measured from a full-scale laboratory structure. Tests showed that artificial neural networks increased model accuracy when used with the dynamic relaxation method.

Kuzniar and Waszczyszyn (2003) applied neural networks for simulation of dynamic response of prefabricated buildings subjected to paraseismic excitations. It was shown that the application of BPNs gives satisfactory neural simulation of displacement records in time domain for vibrations with large amplitudes without analysis of the building motion equations.
Tesar (2003) developed a macro-and micromechanical simulation model adopting the neural network approach for the numerical analysis of the control of the ultimate fatigue behaviour of slender thin-walled structures. The application on the actual slender bridge was made in order to demonstrate the efficiency of the procedures.

Hwan Jeng and Mo (2004) presented a methodology of generating quick seismic response estimations of a prestressed concrete bridge using artificial neural networks. A simple augmented form of multi layer perceptrons that can be quantitatively determined was proposed. These networks were trained and tested based on the analytical data obtained from the non-linear dynamic finite fiber element analyses of the target prestressed concrete bridge. The augmented multi layer perceptrons were found to be much more efficient than the multi layer perceptrons in modeling the critical bending moments of the piers and girder of the prestressed concrete bridge.

Maru and Nagpal (2004) investigated the feasibility of using the neural network model to simulate the inelastic deflections of reinforced concrete frames. The model was found useful in estimating inelastic deflections of consistent procedure rapidly.

Kim et al. (2004) proposed an automated quality assessment technique for rapidly detecting excessive size variations during the production of stone aggregates. The wavelet-based features were used as inputs to an artificial neural network, which was trained to classify the aggregate sample. Taken together, these components formed a neural network-based classification system that determined the compliance of an aggregate product with a given specification.

Jiang and Adeli (2005) presented a multiparadigm dynamic time-delay fuzzy wavelet neural network (WNN) model for non-parametric identification of structures using the non-linear autoregressive moving average with exogenous inputs. The model was applied to two highrise moment-resisting building structures, taking into account
their geometric non-linearities. Validation results demonstrated that the new methodology provides an efficient and accurate tool for non-linear system identification of high-rise buildings.

Morcos and Lounis (2005) proposed a methodology for predicting the time to onset of corrosion of reinforcing steel in concrete bridge decks while incorporating parameter uncertainty. It was based on the integration of artificial neural network, case-based reasoning (CBR), mechanistic model and Monte Carlo simulation (MCS). This study demonstrated the feasibility, adequate reliability and computational efficiency of the proposed integrated ANN-MCS and CBR-MCS approaches for preliminary project-level and also network-level analyses.

Goh and Kulhawy (2005) demonstrated the use of an integrated neural network-reliability method to assess the risk of serviceability failure through the calculation of the reliability index. By first performing a series of parametric studies using the finite element method and then approximating the non-linear limit state surface (the boundary separating the safe and 'failure' domains) through a neural network model, the reliability index was determined with the aid of a spread sheet.

Song Pei et al. (2005) modeled non-linear hysteretic behavior typically observed in structural joints subject to extreme dynamic loads using neural networks. Numerical simulations were presented to demonstrate the efficiency and engineered feature of this approach. A training example was provided to show that this approach enables neural networks to carry some "meaning" (either physical or phenomenological) while remaining flexible and powerful in system identification.

Madan (2006) proposed a general approach for back-propagation training of multilayer feed-forward neural networks for active control of earthquake-induced vibrations in multidegree-of-freedom structures. The method was implemented in
structural control systems with more than one control action. Case studies were presented to demonstrate the feasibility of implementing the training approach for effective vibration control of structures subjected to earthquake ground motions.

Rao and Datta (2006) developed artificial neural network based control scheme for reduction of the seismic response of a multistory building frame. For developing the control scheme, two sets of neural nets were trained. Results of the study show that the control scheme was highly effective in controlling both displacement and acceleration responses of the frame for the unknown earthquake record.

Regelbrugge and Calalo (2006) presented an application of Probabilistic Neural Networks (PNN) to identify dynamic responses of structures. PNN-based identification of structural responses from structural-member strain measurements was presented to illustrate operation of the network. The network was shown to be capable of classifying dynamics in a spatial-frequency domain very quickly using a small number of active elements.

Wang (2006) presented a numerical approximation of optimal control problems for non-linear distributed Hopfield Neural Network equations with diffusion term. For one spatial dimensional case, a semi-discrete numerical algorithm was constructed to find optimal control variable using finite element discretization, updated conjecture gradient iteration method.

Lin et al. (2006) utilized both Fiber Bragg Grating (FBG) sensors and neural networks to construct a system similar to the human being. To implement the concept of a smart structure, a smart active control system was presented. The system robustness was evaluated under both time delay and disconnection problems. Analytical results demonstrated that the NEURO-FBG system can effectively control the response of the structure and provided a more reliable system than ordinary active control.
Song Pei and Smyth (2006) applied an artificial neural network approach to design streamlined network models to simulate the non-linear dynamic response of single-degree-of-freedom oscillators using the restoring force-state mapping interpretation. The neural networks which used sigmoidal activation functions were shown to be highly robust in modeling a wide variety of commonly observed non-linear structural dynamic response behaviors.

Rajendra et al. (2006) discussed identification and robust control of smart structures using artificial neural networks. A neural network based method was developed to estimate the Markov parameters of a multi-input, multi-output system from experimental test data. The closed loop performance and robustness properties of the conventional and the neural network based controller were compared experimentally.

2.2.7 Neural networks in optimization

The advantage of neural networks in design was highlighted earlier. ANN can provide a good initial design close to optimum. As a result, the number of cycles in optimization can be reduced substantially. This feature can be utilized in any other optimization as well.

The application of neural network to simulate analysis in an optimization process was presented by Rogers and Lamarsh (1991). The development of NETS/PROCESS software package has been reported. The research addressed the following specific questions.

- Can a neural network replace a finite element analysis program in the optimization process and will the design coverage to a reasonably accurate optimum solution?
- What is the best way to select the training pairs?
- What is the appropriate number of training pairs to use to train the neural
network?

- Can a user of the system begin the optimization process from different starting points with the neural network and still converge?

The experimentation with different network parameters was reported in an attempt to address the above questions. The investigators expressed the need to develop general guidelines to develop ANNs for mathematical optimisation.

The global optimum design curves were obtained by using the neural dynamics model for optimization problems developed by Adeli and Park (1995). This is a neural network-based optimization technique particularly suitable for problems with highly non-linear and complicated constraints. The robustness of the model was first verified by application to a linear structural optimization problem. Subsequently the model was applied to large non-linear structural optimization problems subjected to the actual constraints of the American Institute of Steel Construction (AISC) specifications.

Rau et al. (1993) studied the problem of learning and retrieving for a pair of correlated patterns within an extensive number of uncorrelated patterns, for networks where learning may be treated as an optimization process with respect to an arbitrary cost function. They defined and discussed several different retrieval phases, whose existence depends on the competitive interplay of the loading level and the pattern correlation.

Rogers (1994) applied neural networks for simulating structural analysis. Guidelines for designing and training a neural network to simulate a structural analysis program were developed. These guidelines included the selection of training pairs and determining the number of nodes on the hidden layer. A sample problem was presented which showed the reduction in the amount of time taken in optimization process.

Ramaswamy and Rajasekaran (1996) have reported a comparison of ANN and genetic algorithms for the design optimization of industrial roofs. This paper aims to
explain knowledge based expert system for the design of industrial roofs, incorporating the artificial neural networks to arrive at the optimum design of industrial roofs. The ANN eliminates the re-analysis and re-designs since the expert system considers the database sequentially and design by expert system is an optimal one.

Adeli and Karim (1997) developed a general mathematical formulation and computational model for optimization of cold-formed steel beams. The non-linear optimization problem was solved by adapting the robust neural dynamics model of Adeli and Park.

Parvin and Serpen (1999) presented an improvement for an artificial neural network paradigm that was shown significant potential for successful application to a class of optimization problems in structural engineering. The complete procedure of solving an optimization problem with a single-layer, relaxation-type recurrent neural network was introduced. Simulation results confirmed that the discrete Hopfield network locates a locally optimal solution after each relaxation once the weight parameters were specified as defined in the suggested technique.

Cheng Yeh (1999) described a method of optimizing high-performance concrete mix proportioning for a given workability and compressive strength using artificial neural networks and non-linear programming. To demonstrate the utility of the proposed methodology, experimental results from several different mix proportions based on various design requirements were presented.

Kim et al. (2000) proposed an optimal control algorithm using neural networks. The controller neural network was trained by a training rule developed to minimize cost function. Both the linear structure and the non-linear structure were controlled by the proposed neuro-controller. A bilinear hysteretic model was used to simulate non-linear
structural behavior. Examples showed that structural vibrations were controlled successfully.

Kassas et al. (2001) have shown the advantage in application of neural networks in the design of cold-formed member to overcome the complex rules that govern the design both in optimization terms and in terms of practical design. The potential for using neural networks to overcome these design problems was investigated. The trained neural network with data relating section profile, aspect ratio and size to the load carrying capacity was used to estimate the best section requirements in new applications.

Papadrakakis and Lagaros (2002) examined the application of neural networks to reliability-based structural optimization of large-scale structural systems. The failure of the structural system was associated with the plastic collapse. The optimization part was performed with evolution strategies, while the reliability analysis was carried out with the Monte Carlo simulation (MCS) method incorporating the importance sampling technique for the reduction of the sample size.

Muhammad Hadi (2003) discussed the applications of neural networks in concrete structures. The paper covers two applications of neural networks in concrete structures. Backpropagation networks are chosen for the network, which is written using the programming package MAT-LAB. Optimum design of simply supported concrete beams, fibrous concrete beams have been carried out. The overall results are compared and observed for the performance of the networks. Based on the applications it was found that neural networks are comparatively effective for a number of reasons, which include the amount of CPU memory consumed by neural networks is less than that consumed by conventional methods and their ease of use and implementation, neural networks provide both the users and the developers more flexibility to cope with different kinds of problems.
Manolis Papadrakakis and Lagaros (2003) examined the efficiency of soft computing techniques in structural optimization. Algorithms based on evolution strategies combined with neural networks, for solving large-scale, continuous or discrete structural optimization problems were investigated. The combined algorithms were implemented both in deterministic and reliability based structural optimization problems, in an effort to increase the computational efficiency as well as the robustness of the optimization procedure. The use of neural networks was motivated by the time-consuming repeated finite element analyses required during the optimization process. A trained neural network is used to perform either the deterministic constraints check or, in the case of reliability based optimization, both the deterministic and the probabilistic constraints checks. The suitability of the neural network predictions is investigated in a number of structural optimization problems in order to demonstrate the computational advantages of the proposed methodologies.

Lagaros and Papadrakakis (2004) developed a methodology to improve the learning of neural networks used for optimization of structures. The efficiency of the proposed adaptive procedure was examined in structural optimization problems where a trained neural network was used to replace the structural analysis phase and capture the necessary data for the optimizer. The optimizer used in this study was an algorithm based on evolution strategies.

Salajegheh and Heidari (2004) applied wavelet neural network and filter banks for optimum design of structures against earthquake. Optimum design of structures for earthquake was achieved by simulated annealing. The wavelet neural networks were employed as a general approximation tool for the time history dynamic analysis. A number of structures were designed for optimal weight and the results were compared to those corresponding to the exact dynamic analysis.
Chen and Weigand (2005) presented new dynamic optimization technique which combines a neural network model with a universal dynamic matrix control algorithm. It was shown that a state-space-based neural network model which utilized a priori process knowledge was best suited for optimization calculations. Advantages of this technique were illustrated by simulation for two chemical processes.

Sirca Jr and Adeli (2005) applied counter propagation neural networks for total cost optimization of precast prestressed concrete I-beam bridges. Typical network convergence curves presented using three different network starting points demonstrate the excellent convergence and the robustness of the optimization model.

Ramana et al. (2006) developed evolutionary computations using neural network based particle swarm optimization technique to evolve the required data for the network to obtain the safety index. Based on the results obtained, it appeared that the proposed approach of obtaining the safety of a structure was reasonable and the developed simulation program was useful as an initial estimate to assess the structural integrity.

2.2.8 Neural networks in prediction of uncertain loading

ANNs have been employed in the characterization of earthquake ground motion. Earthquake characterization involves interaction of many parameters such as magnitude, epicenter, focus, intensity, duration to name a few. These parameters interact in a complex manner to determine the ground motion during the earthquake. The mechanics behind this is not very clear and semi analytical tools had limited success. ANNs with their capability of intuitive prediction can avoid the unclear mechanics relating the parameter and the ground motion and map the relation. Leach et al. (1993) attempted to predict the yield from the seismogram from the input of energy release, body wave
magnitude and other parameters. Dowla et al. (1990) developed a neural network to differentiate between natural earthquake and underground explosion.

Kawiecki (1993) applied feed-forward backpropagation neural network for fast and accurate estimation of the location and size of a crack in a cantilever beam. The presented network was trained and tested using data generated by a linear, closed-form, one-dimensional theoretical model of the cracked beam.

Seibi and Alawi (1997) explored the potential use of artificial neural networks (ANNs) in the field of fracture mechanics. Parameters that influence the value of the fracture toughness were used to develop the ANN model and the contribution of these variables to the variation of fracture toughness was then found. It also demonstrated that ANN was an excellent analytical tool that, if properly used, can reduce cost, time and enhance structure reliability.

Teh et al. (1997) proposed back-propagation neural network model for estimating static pile capacity from dynamic stress-wave data. It was shown that the neural network model predicted the total capacity reasonably well.

Anderson et al. (1997) applied neural networks for prediction of minor axis connections. In steel frames, it is usually the minor-axis beam-to-column connections that govern restraint to the columns against buckling. The artificial neural network was trained to predict bi-linear moment-rotation characteristics for minor-axis connections.

Conte et al. (1997) presented neural network based approach to model the seismic response of multi-story frame buildings. It was found that appropriately configured neural network models were successfully learned and simulated the linear elastic dynamic behavior of multi-story buildings.
Ao He and Wu (1998) used self-recurrent neural networks in a control system, one as an emulator and the other as a controller. The neural network control algorithm was developed for on-line control of structural seismic response in real time.

Seshasayee and Yang (1999) proposed two neural network architectures for use in structural control applications: a failure detection neural network and a failure accommodation neural network. Examples of two simple structures were used to illustrate the features of the networks. Sensor failures were simulated during control operation and the ability of the networks to detect and accommodate the failures was examined. The numerical results revealed that these networks show promise for automated intelligent fault detection, identification, classification and accommodation and as such may have potential use in real civil structures.

Elaine et al. (2001) presented the application of artificial neural networks to forecast the ultimate resistance of steel beams subjected to patch loads. A single design formula for this structural engineering problem is very difficult to obtain, due to the influence of several independent parameters. On the other hand, creating new experimental data in laboratory is very time consuming and expensive. This work demonstrates that new data can be obtained from a neural network system composed of four back propagation networks. The proposed neural network system presented a maximum error value lower than 15%, while the existing formulas errors were over 20%. These results confirmed the possibility of using this methodology to generate new trustworthy data. These data, coupled with experiments found in the literature, can surely help the development of a more consistent and accurate design formula.

Xu et al. (2003) proposed a decentralized, non-parametric identification and control algorithm with neural networks based on the concept of decentralized information structures for the purpose of suppressing the vibration of a documented six-cable-stayed
Ao He and Wu (1998) used self-recurrent neural networks in a control system, one as an emulator and the other as a controller. The neural network control algorithm was developed for on-line control of structural seismic response in real time.

Seshasayee and Yang (1999) proposed two neural network architectures for use in structural control applications: a failure detection neural network and a failure accommodation neural network. Examples of two simple structures were used to illustrate the features of the networks. Sensor failures were simulated during control operation and the ability of the networks to detect and accommodate the failures was examined. The numerical results revealed that these networks show promise for automated intelligent fault detection, identification, classification and accommodation and as such may have potential use in real civil structures.

Elaine et al. (2001) presented the application of artificial neural networks to forecast the ultimate resistance of steel beams subjected to patch loads. A single design formula for this structural engineering problem is very difficult to obtain, due to the influence of several independent parameters. On the other hand, creating new experimental data in laboratory is very time consuming and expensive. This work demonstrates that new data can be obtained from a neural network system composed of four back propagation networks. The proposed neural network system presented a maximum error value lower than 15%, while the existing formulas errors were over 20%. These results confirmed the possibility of using this methodology to generate new trustworthy data. These data, coupled with experiments found in the literature, can surely help the development of a more consistent and accurate design formula.

Xu et al. (2003) proposed a decentralized, non-parametric identification and control algorithm with neural networks based on the concept of decentralized information structures for the purpose of suppressing the vibration of a documented six-cable-stayed
bridge model induced by earthquake excitations. The effectiveness of the decentralized identification and control algorithm based neural network was evaluated through numerical simulations and the adaptability of the decentralized neuro-controlers for different kinds of earthquake excitations and for a damaged cable-stayed bridge model was demonstrated through numerical simulations.

Vanlaere et al. (2004) have applied neural networks for predicting the failure load of a construction by considering the geometrical imperfections of the construction. They obtained lower bound of test results with the help of neural networks. The network used was a feed forward neural network with two input neurons, one output neuron and a hidden layer with ten neurons. The training set contains eighty examples; half of them are situated above the required bound and the other half under the bound. The network was trained successfully with 30,000 cycles.

Sherief Sakla (2004) has studied the potential of using artificial neural networks to predict the load-carrying capacity of pin-ended eccentrically loaded single-angle struts. Results of reported experimental studies on eccentrically loaded steel single-angle struts were used to train and validate the developed ANN model. The performance of the ANN model was compared to that of the AISC specifications. It is shown that neural networks outperform the current AISC specification and provide an efficient alternative method in predicting the load-carrying capacity of eccentrically loaded single-angle struts.

Bandari and Pradeep kumar (2006) applied artificial neural networks as a tool for load rating. The prediction using ANN on the field data was more realistic in comparison to other analytical and mathematical tools as a large number of variables were responsible for the load carrying capacity of such complex structures. A methodology was presented for the determination of the safe load carrying capacity corresponding to some predefined acceptance criteria using neural network.
2.3 Genetic Algorithms

In the present work it is proposed to use a hybridization of back propagation networks and Genetic Algorithms (GA). Accordingly a brief discussion about the basic concepts of GAs is presented below.

2.3.1 General

Decision-making features occur in all fields of human activities such as scientific and technological and affect every sphere of our life. Engineering design, which entails sizing dimensioning and detailed element planning is also not exempt from its influence.

For example an aircraft wing can be made from aluminum or steel and once material and shape are chosen, there are many methods of devising the required internal structure. In civil engineering also, designing a roof to cover large area devoid of intermediate columns require optimal designing.

The aim is to make objective function a maximum or minimum, that is, it is required to find an element $X_0$ in $A$ if it exists such that

$$F (X) \approx F (X_0) \text{ for minimization}$$

$$F (X) < F (X_0) \text{ for maximization} \quad (2.15)$$

The following major questions arise in this process

- Does an optimal solution exist?
- Is it unique?
- What is the procedure?
- How sensitive the optimal solution is?
- How the solution behaves for small changes in parameters?

Since 1940, several optimization problems have not been tackled by classical procedure including:
1. Linear programming
2. Transportation
3. Assignment

![Classification of optimization techniques](image)

4. Non-linear programming
5. Dynamic programming
6. Inventory
7. Queuing
8. Replacement
9. Scheduling
Fig. 2.9: Classes of search techniques

The classification of optimization techniques is shown in Fig. 2.8. Basically, we have been following traditional search technique for solving non-linear equations. Fig. 2.9 shows the classes of both traditional and nontraditional search techniques. Normally, any engineering problem will have a large number of solutions out of which some are feasible and some are infeasible. The designer's task is to get the best solution out of the feasible solutions. The complete set of feasible solutions constitutes feasible design space and the
progress towards the optimal design involves some kind of search within the space. The search is of two kinds, namely deterministic and stochastic.

In the case of deterministic search, algorithm methods such as steepest gradient methods are employed (using gradient concept), whereas in stochastic approach, random variables are introduced. Whether the search is deterministic or stochastic, it is possible to improve the reliability of the results where reliability means getting the result near optimum. A transition rule must be used to improve the reliability. Algorithms vary according to the transition rule used to improve the result.

Nontraditional search and optimization methods have become popular in engineering optimization problems in recent past. These algorithms include:

1. Simulated annealing (Kirkpatrick et al., 1983)
2. Ant colony optimization (Dorigo and Caro, 1999)
3. Random cost (Kost and Baumann, 1999)
4. Evolution strategy (Kost, 1995)
5. Genetic algorithms (Holland, 1975)

Simulated annealing mimics the cooling phenomenon of molten metals to constitute a search procedure. Genetic algorithm and evolutionary strategies mimic the principle of natural genetics and natural selection to construct search and optimization procedures. The collective behavior that emerges from a group of social insects such as ants, bees, wasps and termites has been dubbed as Swarm intelligence. The foraging of ants has led to a novel algorithm called Ant colony optimization for rerouting network traffic in busy telecommunication systems. This method was originally developed by Deneubourg and extended by Dorigo (1999) of Brussels. Random cost method is a stochastic algorithm which moves as enthusiastically uphill as down-hill. The method has no severe problems
in escaping from a dead end and is able to find the optima. In the following sections fundamentals of genetic algorithm are discussed.

2.3.2 Basic concepts

Genetic algorithms are good at taking larger, potentially huge, search spaces and navigating them looking for optimal combinations of things and solutions which we might not find in a lifetime.

Genetic algorithms are very different from most of the traditional optimization methods. Genetic algorithms need design space to be converted into genetic space. So, genetic algorithms work with a coding of variables. The advantage of working with a coding of variable space is that coding discretizes the search space even though the function may be continuous. A more striking difference between genetic algorithms and most of the traditional optimization methods is that GA uses a population of points at one time in contrast to the single point approach by traditional optimization methods. This means that GA processes a number of designs at the same time. As we have seen earlier, to improve the search direction in traditional optimization methods, transition rules are used and they are deterministic in nature but GA uses randomized operators. Random operators improve the search space in an adaptive manner.

Three most important aspects of using GA are:

1. Definition of objective function
2. Definition and implementation of genetic representation
3. Definition and implementation of genetic operators.

Once these three have been defined, the GA should work fairly well beyond doubt. We can, by different variations, improve the performance, find multiple optima (species if they exist) or parallelise the algorithms.
2.3.2.1 Biological background

All living organisms consist of cells. In each cell, there is a set of chromosomes which are strings of DNA and serve as a model for the whole organism. A chromosome consists of genes on blocks of DNA as shown in Fig. 2.10. Each gene encodes a particular pattern. Basically, it can be said that each gene encodes a trait, e.g. colour of eyes. Possible settings of traits (bluish brown eyes) are called alleles. Each gene has its own position in the chromosome search space. This position is called locus. Complete set of genetic material is called genome and a particular set of genes in genome is called genotype. The genotype is based on organism's phenotype (development after birth), its physical and mental characteristics such as eye colour, intelligence and so on.

![Fig. 2.10: Genome consisting of chromosomes](image)

2.3.3 Creation of offsprings

During the creation of offspring, recombination occurs (due to cross over) and in that process genes from parents form a whole new chromosome in some way. The new created offspring can then be mutated. Mutation means that the element of DNA is modified. These changes are mainly caused by errors in copying genes from parents. The fitness of an organism is measured by means of success of organism in life.
2.3.3.1 Search space

The space for all possible feasible solutions is called search space. Each solution can be marked by its value of the fitness of the problem. 'Looking for a solution' means looking for extrema (either maximum or minimum) in search space. The search space can be known by the time solving a problem and we generate other points as the process of finding the solution continues (see Fig. 2.11).

![Fig. 2.11: Examples of search space](image)

The problem is that, search space is complicated and one does not know where to look for the solution or where to start from and this is where genetic algorithm is useful. GAs are inspired by Darwinian theory of the survival of the fittest. Algorithm is started with a set of solution (represented by chromosomes) called populations. Solutions for one population are taken and used to form a new population. This is motivated by a hope that new population will be better than the old one. Solutions, which are selected to form new population (offspring) are select according to their fitness. The more suitable they are, the more chances they have to reproduce. This is repeated until some conditions (number of populations) for improvement of best solution are satisfied.
2.3.4 Working principle

To illustrate the working principle of GA, first consider unconstrained optimization problem. How GA can be used to solve a constrained optimization problem should be discussed later. Let us consider the following maximization problem.

\[
\text{maximize } f(X) \quad (2.16)
\]

\[X_i^{(t)} \leq X_i \leq X_i^{(t+1)} \text{ for } i = 1, 2, \ldots, N\]

If we want to minimize \( f(X) \), for \( f(X) > 0 \), then we can write the objective function as

\[
\text{Maximize } \frac{1}{1 + f(x)} \quad (2.17)
\]

if \( f(X) < 0 \) instead of minimizing \( f(X) \), maximize \( |f(X)| \). Hence, both maximization and minimization problems can be handled by GA.

2.3.5 Encoding

There are many ways of representing individual genes. Holland (1975) worked mainly with string bits but we can use arrays, trees, lists or any other object. Here, only bit strings are considered.

2.3.5.1 Binary encoding

Binary coding is the most commonly used in GA as shown in Table 2.1.

<table>
<thead>
<tr>
<th>Table 2.1: Chromosomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome A 101101100011</td>
</tr>
<tr>
<td>Chromosome B 010011001100</td>
</tr>
</tbody>
</table>
Binary encoding gives many possible chromosomes even with small number of Halictont. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after genetic operator corrections.

2.3.6 Fitness function

As pointed out earlier GAs mimic the Darwinian theory of survival of the fittest and principle of nature to make a search process. Therefore, GAs are usually suitable for solving maximization problems. Minimization problems are usually transformed into maximization problems by some suitable transformation. In general, fitness function $F(X)$ is first derived from the objective function and used in successive genetic operation.

Certain genetic operators require that fitness function be non-negative, although certain operators do not have this requirement. Consider the following transformation.

$$F(X) = f(x) \quad \text{for maximization problem}$$

$$F(X) = rac{1}{f(X)} \quad \text{for minimization problem, if } f(X) \neq 0$$

$$F(X) = \frac{1}{1+f(x)}, \quad \text{if } f(x) = 0$$ \hspace{1cm} (2.18)

A number of such transformations are possible. The fitness function value of the string is known as string's fitness.

2.3.7 Reproduction

Reproduction is usually the first operator applied on population. Chromosomes are selected from the population to be parents to cross over and produce offspring. According to Darwin's evolution theory of survival of the fittest, the best ones should survive and create new offspring. That is why reproduction operator is sometimes known as the selection operator. There exists a number of reproduction operators in GA literature but the essential idea in all of them is that the above average strings are picked
from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner. The various methods of selecting chromosomes for parents to cross over are:

1. Roulette-wheel selection
2. Boltzmann selection
3. Tournament selection
4. Rank selection
5. Steady-state selection

2.3.7.1 Roulette-wheel selection

The commonly used reproduction operator is the proportionate reproductive operator where a string is selected from the mating pool with a probability proportional to the fitness. Thus, *ith* string in the population is selected with a probability proportional to \( F_i \), where \( F_i \) is the fitness value for that string. Since the population size is usually kept fixed in a simple GA, the sum of the probabilities of each string being selected for the mating pool must be one. The probability of the *ith* selected string is

\[
P_i = \frac{F_i}{\sum_{j=1}^{n} F_j}
\]

(2.19)

where 'n' is the population size

2.3.7.2 Boltzmann selection

Simulated annealing is a method of functional minimization or maximization. This method simulates the process of slow cooling of molten metal to achieve the minimum function value in a minimization problem. The cooling phenomenon is simulated by
controlling a temperature like parameter introduced with the concept of Boltzmann probability distribution so that a system in thermal equilibrium at a temperature $T$ has its energy distributed probabilistically according to

$$P(E) = \exp\left(-\frac{E}{kT}\right)$$  \hspace{1cm} (2.20)

where '$k$' is Boltzmann constant. This expression suggests that a system at a high temperature has almost uniform probability of being at any energy state, but at a low temperature it has a small probability of being at a high energy state. Therefore, by controlling the temperature $T$ and assuming search process follows Boltzmann probability distribution, the convergence of the algorithm is controlled. More details about this can be found in Deb (1995).

2.3.7.3 Tournament selection

GA uses a strategy to select the individuals from population and insert them into a mating pool. Individuals from the mating pool are used to generate new offspring, which are the basis for the next generation. As the individuals in the mating pool are the ones whose genes will be inherited by the next generation, it is desirable that the mating pool consists of good individuals. A selection strategy in GA is simply a process that favours the selection of better individuals in the population for the mating pool.

There are two important issues in the evolution process of genetic search, population diversity and selective pressure, as given by Whitley (1989).

Population diversity means that the genes from the already discovered good individuals are exploited while promising the new areas of the search space continue to be explored.
Selective pressure is the degree to which the better individuals are favoured.

The higher the selective pressure the more, the better individuals are favoured. The selective pressure drives GA to improve population fitness over succeeding generations. The convergence rate of GA is largely determined by the selective pressure and population diversity. In general, higher selective pressure results in higher convergence rates. However, if the selective pressure is too high, there is an increased chance of GA prematurely converging to local optimal solution because the population diversity of the search space to be exploited is lost.

2.3.7.4 Rank selection

The Roulette-wheel will have problem when the fitness values differ very much. For example, the best chromosome fitness is 90%, its circumference occupies 90% of Roulette-wheel and then other chromosomes will have very few chances to be selected. Rank selection first ranks the population and taken every chromosome, receives fitness from the ranking. The worst will have fitness 1, the next 2,... and the best will have fitness \( N \) (\( N \) is the number of chromosomes in the population). The Roulette-wheel selection is applied to the modified wheel as shown in Figs. 2.12 and 2.13.

Fig.2.12: Roulette-wheel according to fitness.
Fig. 2.12 is according to fitness and Fig. 2.13 is according to rank. The method can lead to slow convergence because the best chromosome does not differ so much from the other.

![Fig. 2.13: Roulette-wheel according to rank.](image)

2.3.7.5 Steady-state selection

This is not a particular method of selecting the parents. The main idea of the selection is that bigger part of chromosome should survive to next generation. Here, GA works in the following way. In every generation selected, a few (good individuals with high fitness for maximization problem) chromosomes, for creating new off springs. Then, some (bad with low fitness chromosomes are removed and new offspring is placed in that place. The rest of populations survive a new generation.

2.3.8 Genetic Algorithm based Back Propagation Networks (GA/BPN)

Neural networks solve problems by self-learning and self-organization. The back propagations network (BPN) is probably the most well known and widely used among the currently available neural network systems. Back propagation network has been applied
to classification problems, speech synthesis from text, adaptive robotics control, system modeling and various problems in engineering as discussed earlier section 2.3.

The learning algorithm behind BPN (Rumelhart et al., 1986) is a kind of gradient descent technique with backward error (gradient) propagation. However, ‘while the network can recognize patterns similar to those they have learnt, they do not have the ability to recognize new patterns’ (Fu, 1994). Also, while the network must be sufficiently trained to extract a sufficient set of general features applicable to both seen and unseen instances, overturning the network may lead to undesired effects.

Backpropagation (BP) searches on the error surface by means of the gradient descent technique in order to minimize the error criterion.

\[
E = \frac{1}{2} \sum (T_j - O_j)^2
\]  

(2.21)

Where \( T_j \) is the target output and \( O_j \) is the output calculated by the network. It is therefore likely to get stuck in a local minimum.

On the other hand, there exist genetic algorithms (GAs) which are adaptive search and optimization algorithms that mimic the principles of natural genetics. Genetic algorithms are quite different from traditional search and optimization techniques used in engineering design problems but at the same time exhibit simplicity, ease of operation, minimal requirements and global perspective.

Conventionally, a BPN determines its weights based on a gradient search technique and therefore runs the risk of encountering the local minimum problem. GAs on the other hand, though not guaranteed to find global optimum solution to problems, have been found to be good at finding “acceptably good” solutions to problems “acceptably quickly” (less number of iterations).
The idea to hybridize the two approaches, namely GA and BPN follows naturally. Whitley and coworkers (Whitley and Bogart, 1989, 1990; Whitley and Hasmon, 1989; Whitley and Starkwerther, 1990) used GAs to guide back propagation network in finding the necessary connections instead of full connections in order to enhance the speed of training. Though Kitano (Kitano, 1990) proposed some empirical evidence to show that GA/BPN mating does not provide any advantage over a randomly initialized multiple application of Quickprop (a fast variant of BP) alone, at least for shallow networks and easy fitness functions, successful reports have been reported with a hybrid approach. Harp, Samad and Guha (1989) proposed a combined genetic back propagation learning algorithm and encoded BP parameters in the individuals together with the network structure. In the present investigation, it is proposed to use this hybridization of GAs and neural networks.

2.3.9 GA based weight determination

Genetic algorithms which use a direct analogy of natural behavior, work with a population of individual strings, each representing a possible solution to the problem considered. Each individual string is assigned a fitness value which is an assessment of how good a solution is, to a problem. The high-fit individuals participate in "reproduction" by cross-breeding with other individuals in the population. This yields new individual strings as offspring which share some features with each parent. These least-fit individuals are kept out from reproduction and so "die out". A whole new population of possible solutions to the problem is generated by selecting the best (high fit) individuals from the current generation. This new generation contains characteristics which are better than their ancestors.
Progressing in this way, after many generations, owing to mixing and exchange of good characteristics, the entire population inherits the best characteristics and therefore turns out to be fit solutions to the problem. If the GA has been designed well, then most promising areas of search space are explored, resulting in the population converging to an optimal solution to the problem.

Before a GA is executed, a suitable coding for the problem needs to be devised. The fitness function, which assigns merit to each of the individuals in the population, has to be formulated. During the run, parents must be selected for reproduction and crossed over to generate offspring. These aspects of the GA for the weight determination of the BPN are described in the following sections.

2.3.9.1 Coding

The parameters which represent a potential solution to the problem, genes, are joined together to form a string of values referred to as a chromosome. Most conventional GAs code these chromosomes into binary alphabet. However, in this work, binary coding has been dispensed with and a real (decimal) coding system adopted.

Assume a BPN whose network configuration is I-m-n (I input neurons, m hidden neurons and n output neurons). The number of weights that are to be determined are (I + n) m. With each weight (gene) being a real number and assuming the number of digits (gene length) in the weight to be d, a string S of decimal values representing the (I + n) m weights and therefore having a string length \(L = (I + n) \times d\) is randomly generated. The string S represents the weight matrices of the input-hidden and hidden-output layers, in a linear form, arranged according to row-major or column-major order as selected by the designer. An initial population of p chromosomes is randomly generated where p is referred to as the population size.
2.3.9.2 Weight extraction

To determine the fitness values for each of the chromosomes, we extract weights from each of chromosomes.

Let \( x_1, x_2, \ldots, x_d, \ldots, x_l \) represent a chromosome and \( x_{kd+1}, x_{kd+2}, \ldots, x_{k+1d} \) represent the \( k \)th gene \( (k \geq 0) \) in the chromosome. The actual weight \( w_k \) is given by

\[
w_k = \begin{cases} 
\frac{x_{kd+1} \cdot 10^{d-2} + x_{kd+2} \cdot 10^{d-3} + \ldots + x_{k+1d}}{10^d}, & \text{if } 5 \leq x_{kd+1} \leq 9 \\
\frac{x_{kd+1} \cdot 10^{d-2} + x_{kd+2} \cdot 10^{d-3} + \ldots + x_{k+1d}}{10^d}, & \text{if } 0 \leq x_{kd+1} \leq 5 
\end{cases}
\]

(2.22)

2.3.9.3 Fitness function

The fitness function must be devised for each problem to be solved. Algorithm \( \text{FITGEN}() \) illustrates the procedure.

Algorithm \( \text{FITGEN}() \)

\{
Let \( (l_i, T_i), i = 1, 2, \ldots, N \) where \( l_i = (l_{i1}, l_{i2}, \ldots, l_{ih}) \) and \( T_i = (T_{i1}, T_{i2}, \ldots, T_{im}) \) represent the input-output pairs of the problem to be solved by BPN with a configuration \( l \cdot m \cdot n \).

For each chromosome \( C_i, i = 1, 2, \ldots, p \) belonging to the current population \( p_i \) whose size is \( p \)

\{ Extract weights \( w_i \) from \( C_i \) with the help of equation \}

Keeping \( w_i \) as a fixed weight setting, train the BPN for the \( N \) input instances;

Calculate error \( E_i \) for each of the input instances using the formula given below.

\[
E_i = \sum_j (T_{ij} - O_{ij})^2
\]

(2.23)
Where $O_i$ is the output vector calculated by BPN:

$$E_i = 1.2 \ldots N$$

$$E = \sqrt{\frac{\sum E_i}{N}}$$

Calculate the fitness value $F_i$ for each of the individual strings of the population as

$$F_i = \frac{1}{E}$$

\}

output $F_i$ for each $C_i$, $i = 1, 2, \ldots, P$;

\}

END FITGEN

2.3.9.4 Reproduction

In this phase, the mating pool is first formed, before the parent chromosomes reproduce to belief of offspring with better fitness. For the given problem, the mating pool is first formed by excluding that chromosome $C_{1}$ with the least fitness $F_{\text{min}}$ and replacing it with a duplicate copy of the chromosome $C_{1}$ reporting the highest fitness $F_{\text{max}}$. That is, the best fit individuals have multiple copies while worst fit individuals die off.

Having formed the mating pool, the parents are selected in pairs at random. The chromosomes of the respective Pairs are recombined using the two-point cross over operator of a standard GA. Recollect that in two-point cross over, the exchange of gene segments by the parent pair requires selection of cross-sites (cut-points). If $P_a$ and $P_b$ are two parent chromosomes, the offsprings $O_a$ and $O_b$ are produced as a result of executing the two-point cross over operator. This is illustrated in Fig. 2.14.
The offsprings which now form the current population again have their fitness calculated as illustrated by algorithm FITGEN ( ).

2.3.9.5 Convergence

For any problem, if the GA is correctly implemented, the population evolves over successive generations with the fitness value increasing towards the global optimum. Convergence is the progression towards increasing uniformity. A population is said to have converged when 95% of the individuals constituting the population share the same fitness value.

The population P1 now undergoes the process of selection, reproduction and cross over. The fitness values for the chromosomes in P1 are computed, the best individuals replicated and reproduction carried out using two-point cross over operator to form the next generation P2 of chromosomes. The process of generation proceeds until at one stage 95% of the chromosomes in the population P1 converge to the same fitness value.
At that stage, the weights extracted from the population $P$, are the final weights to be used by the BPN.

The algorithm for the GA based weight determination can be summarized and is illustrated in Algorithm GA-NN-WT( ).

Algorithm GA-NN-WT( )

\begin{verbatim}
{ $i \leftarrow 0$; \\
Generate the initial population $P$, of real-coded chromosomes $C_i^t$, each representing a weight set for the BPN; \\
\hspace{1em} \text{While the current population } P_i \text{ has not converged} \\
\hspace{1em} \{ \text{Generate fitness values } F_i^t \text{ for each } C_i^t \in P_i \text{ using the} \\
\hspace{1.5em} \text{Algorithm FITGEN( ).} \\
\hspace{1em} \text{Get the mating pool ready by terminating worst fit individuals and duplicating high fit} \\
\hspace{1em} \text{individuals;} \\
\hspace{1em} \text{Using the cross over mechanism, reproduce offspring from the parent} \}
\hspace{1em} \text{chromosomes;} \\
\hspace{1em} \quad i \leftarrow i + 1; \\
\hspace{1em} \text{Call the current population } P_i' \\
\hspace{1em} \text{Calculate fitness values } F_i^t \text{ for each } C_i^t \in P_i' \\
\hspace{1em} \} \\
\text{Extract weights from } P_i' \text{ to be used by the BPN;} \\
\}\end{verbatim}

END GA-NN-WT.
2.4 Genetic Algorithms: Review

Genetic algorithms are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection. Prof. Holland of University of Michigan, Ann Arbor, envisaged the concept of these algorithms in the mid-sixties and published his seminal work (Holland, 1975). Thereafter, a number of students and other researchers have contributed to the development of this field.

To date, most of the GA studies are available through some books by Davis (1991), Goldberg (1989), Holland (1975), Michalewicz (1992) and Deb (1995) and through a number of conference proceedings. The first application towards structural engineering was carried by Goldberg and Sarntani (1986). They applied genetic algorithm to the optimization of a ten-member plane truss. Jenkins (1991) applied genetic algorithm to a trussed beam structure. Deb (1991) and Rajeev and Krishnamoorthy (1992) have also applied GA to structural engineering problems. Apart from structural engineering there are many other fields in which GAs have been applied successfully. It includes biology, computer science, image processing and pattern recognition, physical science, social sciences and neural networks.

Jenkins (1991) has investigated the application of the genetic algorithm in the optimization of structural design. Stochastic processes generate initial population of designs. A principle of natural selection/survival of the fittest is applied to improve the designs. He has considered the analysis and design interfaces and the basic operations of selection, crossover, mutation and parameter scaling. The example problems presented are minimization of a simple algebraic function, optimum design of a trussed-beam roof structure. The results produced by the genetic algorithm are compared with other optimization studies and found that both are in good agreement.
Rajeev and Krishnamoorthy (1992) presented a simple genetic algorithm for optimizing structural system with discrete design variables. They transformed constrained problem in to unconstrained one as GAs are best suited for unconstrained optimizing problems. The concept of optimization using genetic algorithm is presented in detail using a three-bar truss problem. Formulation and comparison of results for a ten bar truss and a twenty-five bar truss are also presented. They concluded that genetic algorithms are best suited for structural optimization and problems where gradient computations are difficult.

Jenkins (1992) applied genetic algorithm for plane frame optimum design environment. The use of the environment is illustrated in a study of a cable-stayed structure. Design optimization was based on the use of a genetic algorithm in which a population of individual designs was changed generation-by-generation, applying principles of natural selection and survival of the fittest. The fitness design was assessed using an objective function in which violations of design constraints are penalized. Facilities are provided for automatic data editing and reanalysis of the structure.

Hajela and Lin (1992) described an implementation of genetic search methods in multicriterion optimal designs of structural systems with a mix of continuous, integer and discrete design variables. Two distinct strategies to simultaneously generate a family of pareto optimal designs are presented in the paper. These strategies stem from a consideration of the natural analogue, wherein distinct species of life forms share the available resources of an environment for sustenance. The efficacy of these solution strategies are examined in the context of representative structural optimization problems.
reflect the preference of designers. A numerical example of coloring a dam structure presented to demonstrate the applicability of the system developed here.

Cuello et al. (1997) presented an optimization model for the design of rectangular reinforced concrete beams subjected to a specified set of constraints. The model minimizes the cost of the beam on strength design procedures, while also considering the costs of concrete, steel and shuttering. As there are an infinite number of possible beam dimensions and reinforcement ratios that yield the same moment of resistance, a simple genetic algorithm was applied as the search engine. The results were compared with those obtained via geometric programming.

Liu et al. (1997) implemented a simple genetic algorithm with two additional techniques, Pareto optimality ranking and fitness sharing, for the deck rehabilitation plan of network level bridges, aiming to minimize the total rehabilitation cost and deterioration degree. This approach was illustrated by a simple example and then applied to a practical bridge system with a large number of bridges.

Topping et al., (1998) have worked on parallel implementations of neural networks and genetic algorithms. It was shown how a parallel finite element analysis may be undertaken in an efficient manner by processing of the finite element model using a genetic algorithm utilizing a neural network predictor.

Leite and Topping (1998) have applied improved genetic operators for optimization of structures. Revised genetic operators and new recombination scheme were presented. The guidelines to handle the constraints using transformation methods to map the genetic algorithms onto engineering optimization problems were also presented.

Harris et al. (1998) developed new methods based on the use of genetic algorithms for solving crystal structures from powder diffraction data. In this paper, the fundamental
concepts underlying genetic algorithms were discussed and the implementation of this approach for structure solution from powder diffraction data was described.

Wei Huang et al. (1998) developed a genetic algorithm for engineering applications that involve sequencing of operations. The proposed algorithm used some new operators that were different from those typically used in genetic algorithms. Some enhancements for improving performance of the algorithm were also described.

Kiong Soh and Yang (1998) presented an approach to the layout and shape-optimization problem of bridge truss structures using genetic algorithms. The objective was to find an optimal layout design that will have minimum weight or material volume, subject to performance constraints related to member stresses, joint displacements and member buckling. Two examples concerning bridge truss structure were investigated to demonstrate the effectiveness of the proposed method in solving these layout-optimization problems.

Hajela et al. (1998) described the use of genetic algorithms in determining the optimal layout and sizing of two-dimensional and three-dimensional grillage structures for stress, displacement and element buckling constraints. Strategies designed to alleviate the computational requirements of a genetic algorithm based search were discussed in the paper.

Koumousis and Arsenis (1998) employed genetic algorithms to perform the optimal detailed design of reinforced concrete members of multi-story buildings. Various parameters of the genetic algorithm were considered and the corresponding results were presented.

Lin and Yang (1998) proposed techniques in sharing- enhanced genetic algorithms to identify regions where designs cluster in a multimodal design space. Two illustrative
multimodal function minimization problems were used as benchmarks to test the proposed techniques.

Rajeev and Krishnamoorthy (1998) presented design optimization of reinforced concrete plane frames using genetic algorithm based methodology. Examples of reinforced concrete plane frames were solved and results were presented.

Botello et al. (1999) studied the performance of two stochastic search methods: Genetic algorithms and simulated annealing, applied to the optimization of pin-jointed steel bar structures. They have shown that it is possible to embed these two schemes into a single parametric family of algorithms and that optimal performance (in a parallel machine) can be obtained by a hybrid scheme. Examples of applications to the optimization of several real steel bar structures were presented.

Cheng Yeh (1999) proposed a hybrid GA that combines the concept of survival of the fittest with the concept of adaptation to speed up the convergence. The fully stressed design optimality criterion was employed to play the role of adaptation. Numerical examples show that even though the displacement constraints are active, (1) both average weight and minimum weight obtained by a hybrid GA are less than those obtained by a pure GA, (2) a hybrid GA is more stable than a pure GA and (3) the speed of convergence of a hybrid GA is superior to that of a pure GA.

Wook Park and Grierson (1999) presented a computational procedure for multicriteria optimal conceptual design of the structural layout of buildings subject to given specifications and requirements. An example conceptual building layout design was presented using the multi objective genetic algorithm and the applicability and efficiency of the developed computational conceptual design procedure was discussed.

Savic et al. (1999) developed software for the optimal design of general and symmetric/balanced laminates (or sandwich panels) with specified mechanical properties.
The approach taken relied on the analysis of a given laminate of known materials and stacking sequence and on an efficient genetic algorithm based optimization procedure to come up with the best design with respect to single or multiple objectives and constraints.

Fuat Erbatur et al. (2000) reported the development of a computer-based systematic approach for discrete optimal design of planar and space structures composed of one-dimensional elements. The main characteristic of the solution methodology was the use of a genetic algorithm as the optimizer. Applications and experience on steel frame and truss structures were discussed. The results of comparative studies of the GA against other various discrete and continuous optimization algorithms for a class of representative structural design problems were reported to show the efficiency of the former.

Li et al. (2000) proposed an integrated approach to the modeling and optimization of structural control systems in tall buildings. An artificial neural network was applied to model the structural dynamic responses of tall buildings subjected to strong earthquakes and a genetic algorithm was used to optimize the design problem of structural control systems which constitutes a mixed-discrete, non-linear and multi-modal optimization problem. The neural network model of the structural dynamic response analysis was included in the genetic algorithm and was used as a module of the structural analysis to estimate the dynamic responses of tall buildings. A numerical example was presented in which the general regression neural network was used to model the structural response analysis.

Hayalioglu (2000) presented a genetic algorithm for the optimum design of geometrically non-linear elastic plastic steel frames with discrete design variables. Design variables were selected from practically available sets of standard steel sections. Relatively large displacement restrictions were considered in the optimum designs. The
analysis covered the geometric non-linearity and elastic plastic effects of the material as well. An incremental load approach with a Newton Raphson type of iteration was used in the analysis of the frames. The designs obtained for non-linear elastic-plastic frames were compared to those where linear elastic behavior was assumed.

Saka et al. (2000) developed a genetic algorithm based method for the optimum design of grillage systems. The algorithm not only selects the optimum sections for the grillage elements from a set of standard universal beam sections, but also finds the optimum spacing required for the grillage system.

Sarma and Adeli (2000) developed a fuzzy augmented Lagrangian GA by extending the augmented Lagrangian genetic algorithm of Adeli and Cheng for optimization of steel structures subjected to the constraints of the AISC allowable stress design specifications taking into account the fuzziness in the constraints. The features and advantages of the new fuzzy GA included acknowledging the imprecision and fuzziness in the code-based design constraints, increased likelihood of obtaining the global optimum solution, improved convergence and reduced total computer processing time.

Pezeshk et al. (2000) presented a genetic algorithm based optimization procedure for the design of two dimensional, geometrical, non-linear steel-framed structures. The approach presented used GAs as a tool to achieve discrete non-linear optimal or near-optimal designs. The differences between optimized designs obtained by linear and geometrically non-linear analyses were discussed.

Vairavamoorthy and Ali (2000) proposed an optimal design methodology for the design of water distribution systems based on genetic algorithms. The method was tested on several networks, including networks used for benchmark testing least-cost design algorithms and was shown to be very efficient and robust.
Woo et al. (2001) presented a structural application of a shape optimization method based on a genetic algorithm. The method produced a sequence of fixed-distance step-wise movements of the boundary nodes of a finite element model to derive optimal shapes from an arbitrary initial design space. The GA was used to find the optimal or near-optimal combination of boundary nodes to be moved for a given step movement. For illustrative purposes, the method was applied to structural shape-optimization. Two examples of structural shape optimization are included showing local and global optimization through material removal and addition.

Li et al. (2001) presented an integrated optimum problem of structures subjected to strong earthquakes and wind excitations, optimizing the number of actuators. The configuration of actuators and the control algorithms simultaneously was studied. Two control algorithms, optimal control and acceleration feedback control were used as the control algorithms. A multi-level optimization model was presented with respect to the solution procedure of the optimum problem. The characteristics of the model were analyzed and the formulation of each sub optimization problem at each level was presented. To solve the multi-level optimization problem, a multi-level genetic algorithm (MLGA) was used. The developed model and MLGA were used to solve two multi-level optimization problems in which the optimization of the number of actuators, the positions of actuators and the control algorithm were considered in different levels.

Iranmanesh and Fahimi (2001) improved the efficiency of the counter-propagation neural net response in structural analysis and optimization. To achieve this, a modification has been made on the learning coefficients, which resulted in a higher performance. The net was trained by two different procedures, random and genetic generation of training pairs. To examine the efficiency of the net, different examples were investigated. The results of genetic trained counter-propagation net and the random
trained one were compared with the exact solution. The effects of various parameters, such as number of training pairs, number of Kohonen units and the number of winning nodes were studied.

Saka (2001) applied genetic algorithm to develop an optimum design method for pitched roof steel frames with haunches for the rafters in the eaves. The algorithm selects the optimum universal beam sections for columns and rafters from the available steel sections table. Furthermore, it determines the optimum depth of haunch at eaves and length of the haunch required to reach the most cost-effective form. Formulation of the design problem is based on the elastic design method. The serviceability and the strength constraints were included in the design problem as defined in BS5950. Furthermore, the overall bucking of columns and rafters in the torsional mode between effective torsional restraints to both flanges was also checked. A pitched roof frame was designed by the algorithm developed to demonstrate its practical application.

Dimou and Koumousis (2001) introduced competition among the populations of a number of genetic algorithms to calibrate the size of the populations of the GAs by altering the resources of the system i.e. the allocated computing time. The evolution of the different populations is controlled on the level of metapopulation, i.e. the union of populations, on the basis of statistics and trends of the evolution of every population. The method was applied to the reliability based optimal design of a simple truss. Numerical results were presented and the robustness of the proposed algorithm was discussed.

Miyamura et al. (2001) applied genetic algorithms for the limit analysis of masonry walls with rectangular openings subjected to vertical and horizontal loading. An equivalent shear truss model whose structural parameters are defined by typical failure patterns due to earthquake damages, such as shear diagonal crack opening, local panel rocking and bed joint sliding, full or partial overturning of an entire wall, is introduced to
in-plane collapse analysis of a single masonry wall. By means of this truss model the limit analysis was formulated as a searching problem for a collapse mode with minimum failure load among a set of elementary collapse modes and their combinations. For this combinatorial optimization problem, the Stochastic Schemata Exploiter (SSE), which is one of the GAs, was applied.

Hadi (2001) presented the application of genetic algorithms for the design of continuous reinforced concrete T and I beams based on Australian design standards, AS: 3600. This optimization problem was implemented by writing a computer program using Matlab. In order to evaluate the developed system, a number of examples were analyzed and designed using the developed code.

Xu and Gong (2001) presented an efficient design approach for the preliminary design of long-span king-post steel truss systems. A genetic algorithm was applied to search for the optimal key design parameters. Finally, design examples based on the study of two arena projects in North America were presented. The results of the study suggested that the proposed design method provided an efficient and practical solution for the design of long-span king-post truss systems.

Yin Zhai et al. (2002) presented an integrated feature extraction approach, which is based on rough set theory and genetic algorithms. Based on this approach, a prototype feature extraction system was established and illustrated in an application for the simplification of product quality evaluation. The prototype system successfully integrates the capability of rough set theory in handling uncertainty with a robust search engine, which was based on a GA. The results show that it can remarkably reduce the cost and time consumed on product quality evaluation without compromising the overall specifications of the acceptance tests.
Caldas and Norford (2002) applied genetic algorithms as a generative and search procedure to look for optimized design solutions in terms of thermal and lighting performance in a building. The GA is first used to generate possible design solutions, which are then evaluated in terms of lighting and thermal behavior using a detailed thermal analysis program. The results from the simulations were subsequently used to further guide the GA search towards finding low-energy solutions to the problem under study. The specific problem addressed in this study was the placing and sizing of windows in an office building.

Lagaros et al. (2002) investigated the efficiency of various evolutionary algorithms (EA), such as genetic algorithms and evolution strategies, when applied to large-scale structural sizing optimization problems. The modified versions of the basic EA are implemented to improve the performance of the optimization procedure. The modified versions of both genetic algorithms and evolution strategies combined with a mathematical programming method to form hybrid methodologies were also tested and compared and proved to be promising. The numerical tests presented demonstrate the computational advantages of these methods, which become more pronounced in large-scale optimization problems.

Yang et al. (2002) presented an inverse procedure for the detection of a three-dimensional crack in plates and shells using a genetic searching algorithm. The integral strain, which can be measured by optical fibers, was utilized as the input for the inverse procedure. Two parameters, each with three components, were used to describe the location and size of the crack. A shell element model was created with a re-mesh operation in the numerical simulation of the integral strains. The squared difference of the integral strains obtained from a plate with an actual crack and from a plate with a trial crack was defined as the objective function. The best crack parameters were found by
minimizing the objective function using a searching technique of the micro genetic algorithm (μGA). Numerical examples were presented to demonstrate the procedures and effectiveness of the method.

Loi and Que (2002) worked on a problem with the application and performance of two evolutionary search techniques to identify the parameters characterizing mode 1 cohesive crack models. Genetic-based heuristic schemes, implemented via a classical genetic algorithm and also as a differential evolution process, were considered. Actual experimental data was used to assess the schemes for both piecewise linear and non-linear softening models. The results of these two adaptive algorithms were also compared with the performance of a previously proposed local optimization approach involving formulation of the inverse problem as a mathematical program with equilibrium constraints.

Singh et al. (2002) presented an approach for optimum design of tuned mass dampers for response control of torsional building systems subjected to bi-directional seismic inputs. Four dampers with fourteen distinct design parameters, installed in pairs along two orthogonal directions, were optimally designed. A genetic algorithm was used to search for the optimum parameter values for the four dampers. Several optimal design criteria, expressed in terms of performance functions that depend on the structural response, were used. Several sets of numerical results for a torsional system excited by random and response spectrum models of seismic inputs were presented to show the effectiveness of the optimum designs in reducing the system response.

Krishnamoorthy et al. (2002) discussed the object-oriented design and implementation of such a core library. They have shown how classes derived from the implemented libraries can be used for the practical size optimization of large space trusses, where several constructability aspects have been incorporated to simulate real-
world design constraints. Strategies were discussed to model the chromosome and to code genetic operators to handle such constraints. The implemented libraries were tested on a number of large previously fabricated space trusses and the results were compared with previously reported values. It is concluded that genetic algorithms implemented using efficient and flexible data structures can serve as a very useful tool in engineering design and optimization.

Chen et al. (2003) presented genetic algorithms based evolutionary strategy for classification problems, which includes two aspects: evolutionary selection of the training samples and input features and evolutionary construction of the neural network classifier. For the first aspect, the GA based $k$-means-type algorithm (GKMT) was proposed, which combines GA and $k$-means-type (KMT) to achieve the optimal selection of the training samples and input features simultaneously. By this algorithm, the "singular" samples will be eliminated according to the classification accuracy and the features that facilitate the classification will be enhanced. On the opposite, the useless features will be suppressed and even eliminated. For the second aspect, the hierarchical evolutionary strategy is proposed for the construction and training of the neural network classifier. This strategy uses the hierarchical chromosome to encode the structure and parameters of the neural network into control genes and parameter genes respectively, designs and trains the network simultaneously. Finally, the experimental study pertained to the fault diagnostics for the rotor-bearing system was given and the results presented show that the proposed evolutionary strategy for the classification problem was feasible and effective.

Lep and Michal (2003) presented the outlines of an application of genetic algorithm based strategies to a class of optimization tasks associated with the design of steel reinforced concrete structures. The principal design objective considered was to minimize the total cost of a structure and also to comply with all strength and
serviceability requirements for a given level of the applied load. They used genetic algorithm to handle such a complex optimization problem with a number of constraints.

As an example, a simple continuous steel reinforced beam was analyzed.

Ramanjaneyulu et al. (2003) have formulated genetic algorithm based optimal design of continuous span prestressed concrete bridge box girders. The design forces due to self-weight, superimposed dead loads and traffic loads were computed based on the influence functions for bending moments at critical sections. These were interfaced with genetic algorithm based optimization algorithm for formulating constraints. The studies were carried out for arriving at optimal cost of three and four span continuous bridge box girders. Studies have also been carried out to assess the influence of arrangement of spans (equal and unequal spans) on the optimal cost of continuous prestressed concrete bridge box girders. It was noted that unequal span arrangement results in about 10% saving in the cost of two-lane bridge of single cell box section.

Sakla and Elbeltagi (2003) developed a procedure for the design of steel roofs subjected to non-uniform loads such as drifted snow. The presented approach uses genetic algorithms to achieve optimal or near-optimal designs. Details of model development were described and example applications were presented to demonstrate the capabilities of the model.

Koh et al. (2003) formulated substructural identification and progressive structural identification methods by adopting the strategy of "divide-and-conquer". The main idea was to divide the structure into sub-structures such that the number of unknown parameters was within manageable size in each stage of identification. A non-classical approach of genetic algorithms was employed as the search tool. Numerical simulation study was presented including for a fairly large system of 50 degrees of freedom, to illustrate the identification accuracy and efficiency. The methods were tested for known-
mass and unknown-mass systems with up to one hundred and two unknown parameters, accounting for the effects of incomplete and noisy measurements.

Jiang et al. (2003) have applied a structural modular neural network, by combining the BP neurons and the radial basis function neurons at the hidden layer, to construct a better input-output mapping both locally and globally. The use of genetic algorithm in searching the best-hidden neurons makes the structural modular neural network less likely to be trapped in local minima than the traditional gradient-based search algorithms.

Lee and Ahn (2003) used a genetic algorithm to perform the discrete optimization of reinforced concrete plane frames subject to combinations of gravity loads and lateral loads. It was shown that the developed genetic algorithm obtained an optimal design for reinforced concrete plane frames.

Camp et al. (2003) developed a design procedure implementing a genetic algorithm for discrete optimization of reinforced concrete frames. The objective was to minimize the material and construction costs of reinforced concrete structural elements subjected to serviceability and strength requirements. Examples were presented demonstrating the efficiency of the procedure for the flexural design of simply supported beams, uniaxial columns and multi-story frames.

Min Liu et al. (2004) considered life cycle cost for multiobjective design optimization of Seismic steel Moment-Resisting Frame (SMRF) structures. The maximum interstory drift ratio was selected as the single seismic performance parameter for a code-compliant SMRF design and is evaluated through a static pushover analysis. Effects of randomness and uncertainty in estimating seismic demand and capacity as well as in describing seismic hazards were considered. Using a multi-objective genetic algorithm, the automated optimization procedure produced a set of alternative designs
that exhibits optimized tradeoff among the three conflicting objectives. Therefore, designers have much freedom to choose the final structural solution with a preferred balance of initial cost and lifetime seismic damage cost while taking into account the degree of design complexity.

Biondini et al. (2004) presented an approach of wide generality for assessing the reliability of reinforced and prestressed concrete structures. All the uncertainties were modeled by using a fuzzy criterion in which the model was not defined through a set of fixed values, but through bands of values, bounded between suitable minimum and maximum extremes. The reliability problem was formulated at the load level, with reference to several serviceability and ultimate limit states. For the critical interval associated to each limit state, the membership function of the safety factor was derived by solving a corresponding anti-optimization problem. The strategic planning of this solution process was governed by a genetic algorithm, which generates the sampling values of the parameters involved in the material and geometrical non-linear structural analyses. The effectiveness of the developed approach and its capability to handle complex structural systems were shown by carrying out a reliability assessment of a prestressed concrete continuous beam and of a cable-stayed bridge.

Papadimitriou (2004) presented theoretical and computational issues arising in the selection of the optimal sensor configuration for parameter estimation in structural dynamics. The information entropy, measuring the uncertainty in the system parameters, was used as the performance measure of a sensor configuration. A useful asymptotic approximation for the information entropy, valid for a large number of measured data was derived. Two algorithms were proposed for constructing effective sensor configurations that are superior, in terms of computational efficiency and accuracy, to the sensor configurations provided by genetic algorithms. The theoretical developments and the
effectiveness of the proposed algorithms were illustrated by designing the optimal configuration for a 10-degree-of-freedom chain-like spring mass model and a 240 degree-of-freedom, three-dimensional truss structure.

Liu and Frangopol (2004) presented optimal bridge maintenance planning. The uncertainties associated with the deterioration process under no maintenance and under maintenance were not taken into consideration. Such uncertainties were confined to the parameters that define the selected computational models and their effects were evaluated by means of Monte Carlo simulations. A multi-objective genetic algorithm based numerical procedure was used to locate the best possible tradeoff maintenance planning solutions with respect to three objective functions, namely, condition index, safety index and cumulative life-cycle maintenance cost.

Luciano Catallo (2004) dealt with the reliability assessment of precast concrete framed structures. The unavoidable uncertainties in geometrical and the mechanical properties that define the structural problem which cannot be considered as deterministic quantities were modeled using a fuzzy theory based approach.

Fedele et al. (2005) studied the identification of the parameters contained in an elastic plastic material model, apt to simulate steady-state ratcheting, with reference to cyclic biaxial tests on cylindrical compact specimens. The material considered is steel employed for high-speed train wheels. In order to generate strain states close to those expected in severe service conditions due to wheel-rail rolling-sliding contact, out-of-phase tension-torsion unsymmetric cycles are applied in laboratory tests. The experiments were simulated by conventional finite elements and Chaboche model with non-linear kinematic hardening. The material parameters were identified through a deterministic, batch (non-sequential) inverse analysis in two stages (genetic and first-order algorithms) in view of the peculiar constraints in the minimization problem.
Cristina et al. (2005) investigated three families of GA’s variations namely multi-objective genetic algorithm, Roulette and Tournament which have been proposed in order to find a solution to the problem of minimizing the production costs of hollow core slabs. In each family the elements differ from each other with respect to the reproduction scheme and the way the population is restored to its original size. Results were presented and analyzed to support a discussion about advantages and disadvantages of the proposed GAs variations in finding the solution to the problem. In addition, results obtained with a conventional optimization method were presented for comparison.

Yun and Kim (2005) applied a genetic algorithm for the optimum design of plane steel frame structures. A genetic algorithm based optimum design algorithm and a program incorporated with the refined plastic hinge analysis method, one of the second-order inelastic analysis methods were presented.

Youl Lee and Chang Wooh (2005) dealt with a method to identify structural damage using the combined finite element method and the advanced uniform micro genetic algorithm. The novelty of this study was to use dynamic loading and its response due to the anomalies in a structure under testing. The results demonstrated the excellencies of the method from the standpoints of computation efficiency as well as its ability to avoid premature convergence.

Tan et al. (2005) proposed an integrated technique to place devices and design controllers based on the use of genetic algorithms. The approach was flexible, allowing the designer to base the placement scheme on performance goals and/or system requirements. The improvements in the effectiveness of the proposed methodology as compared to previously developed techniques were demonstrated through comparative studies.
Cengiz Toklu (2005) formulated an aggregate-blending as a multi-objective optimization problem and solved by using genetic algorithms. It was shown that in this way all existing formulations of an aggregate-blending problem were covered and solved.

Rao et al. (2006) presented the simulation of stress-strain response of Al₂O₃ (matrix)/SiC (whisker) ceramic composite using a hybrid neural network which incorporates the effect of interface shear strength (IFS) in the analysis. For efficient and quick training, the weights for the BPN were obtained by using a genetic algorithm. The GA was modeled with 150 genes and a chromosome string length of 750. The network simulation was based on the stress-strain response obtained from the finite element analysis. The finite element analysis has been carried out only for a limited number of specimens. However, the simulation model is capable of predicting the stress-strain relationship for new interface shear strengths even with this limited information. Thus, the robustness and the generalization capability of the hybrid neural network model were demonstrated.

Elbehairy et al. (2006) introduced an integrated model for bridge deck repairs with detailed life cycle costs of network-level and project-level decisions. Two evolutionary-based optimization techniques that were capable of handling large-size problems, namely genetic algorithms and shuffled frog leaping were then applied on the model to optimize maintenance and repair decisions. Results of both techniques were compared on case study problems with different numbers of bridges. Based on the results, the benefits of the bridge deck management system were illustrated along with various strategies to improve optimization performance.
Perera and Torres (2006) presented nondestructive global damage detection and assessment methodology based on the changes in frequencies and mode shapes of vibration of a structural system using genetic algorithms. The method was applied at an element level using a finite-element model. It was shown that the proposed GA yields suitable damage location and severity detection while introducing numerous advantages compared to classical methods. The influence of noise in the modal data was also been considered.

Balling et al. (2006) presented a genetic algorithm that can simultaneously optimize size, shape and topology of skeletal structures, including both trusses and frames. It was also executed on two standard test problems as well as on a plane frame example. This algorithm presented the designer with more choices and more information than algorithms that converge to a single optimum design.

From the review of the literature presented above, it is noticed that lot of potential exists for the application of artificial neural networks and genetic algorithms in the field of structural engineering. In the field of structural design of R.C.C. elements, few works have been reported. But most of these works have used simple back propagation neural networks. Further, it is observed that vary scanty information is available on the use of genetic algorithms for the design of R.C.C. structural elements. Thus, there exists a gap in the literature for conducting a systematic study for demonstrating the applicability of ANNs and GAs in the field of structural design. It is also observed that most of the researchers tried either simple BP networks or genetic algorithms separately. The potential of utilizing hybrid neural networks which can result in several benefits has been not exploited. Thus, there is need to conduct a systematic study and develop hybrid neural network models for the structural design of R.C.C. elements and compare their
performance with simple back propagation network models. Thus, this literature review conforms the specific objectives of this investigation listed in section 1.2.