2007

AN APPROACH TO INVERSE MODELING THROUGH THE INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

Kirthi Bedida
University of Kentucky, kbed2@uky.edu

ABSTRACT OF THESIS

AN APPROACH TO INVERSE MODELING THROUGH THE INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

A hybrid model integrating predictive capabilities of Artificial Neural Network (ANN) and optimization feature of Genetic Algorithm (GA) is developed for the purpose of inverse modeling. The proposed approach is applied to Superplastic forming of materials to predict the material properties which characterize the performance of a material. The study is carried out on two problems. For the first problem, ANN is trained to predict the strain rate sensitivity index *m* given the temperature and the strain rate. The performance of different gradient search methods used in training the ANN model is demonstrated. Similar approach is used for the second problem. The objective of which is to predict the input parameters, i.e. strain rate and temperature corresponding to a given flow stress value. An attempt to address one of the major drawbacks of ANN, which is the black box behavior of the model, is made by collecting information about the weights and biases used in training and formulating a mathematical expression. The results from the two problems are compared to the experimental data and validated. The results indicated proximity to the experimental data.

KEYWORDS: Artificial Neural Networks, Genetic Algorithms, Hybrid Modeling, Gradient Search, Superplastic Forming

Kirthi Bedida

____

11/27/07

____

AN APPROACH TO INVERSE MODELING THROUGH THE INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

By

Kirthi Bedida

Dr. Marwan Khraisheh

Director of Thesis

Dr. Fazleena Badurdeen

Co-Director of Thesis

L. Scott Stephens

Director of Graduate Studies

Date: November 27, 2007

RULES FOR THE USE OF THESIS

Unpublished thesis submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgements.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name Date

THESIS

Kirthi Bedida

The Graduate School

University of Kentucky

2007

AN APPROACH TO INVERSE MODELING THROUGH THE INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in Mechanical Engineering
in the College of Engineering at the University of Kentucky

By

Kirthi Bedida
Lexington, Kentucky

Director: Dr. Marwan Khraisheh, Professor of Mechanical Engineering
Co-Director: Dr. Fazleena Badurdeen, Assistant Professor of Mechanical Engineering
University of Kentucky Lexington, Kentucky

2007

Dedicated To My Parents

Acknowledgements

I would like to express my gratitude to Dr. Marwan Khraisheh for his extended support throughout the course of my Masters. His constant encouragement, support, and invaluable suggestions have made this work successful. I express my sincere gratefulness to Dr. Fazleena Badurdeen who has been supportive and encouraging all through my M.S. It is because of her enthusiasm for research that I am able to complete my work successfully and on time. It is her motivation that has helped me make presentations and publications.

My sincere thanks go to Dr. Ibrahim Jawahir for taking time out and serving on my thesis committee. I would like to thank Dr. Fadi K. Abu-Farha for his useful inputs and timely help. I would also like to thank all my research team members for their support.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my life. I am also very thankful to my brother Rohit for his continuous encouragement and love without which none of this was possible. Last but not the least; I would also like to thank my friends, especially Puja Thota for her valuable discussions and cooperation.

Table of Contents

Acknowledgements iii

List of Figures vii

List of Tables xi

List of Files xii

Chapter 1: Overview 1

1.1. Motivation 3

1.1.1. Time and Resources 3

1.1.2. Controlling the Process 3

1.2. Objectives 4

1.3. Approach 4

1.4. Thesis Organization 8

Chapter 2: Literature Review 9

2.1. Introduction 9

2.2. Superplastic Blow Forming Method 12

2.3. Modeling Using Numerical Methods 13

2.4. Artificial Intelligence in Manufacturing 15

2.4.1. Artificial Neural Networks (ANN) 15

2.4.1.1. Finite Element Modeling and Artificial Neural Network 18

2.4.1.2. Artificial Neural Networks and Superplastic Forming 20

2.4.2. Genetic Algorithms 22

2.4.3. Integrating Genetic Algorithms and Artificial Neural Network 28

2.5. Summary 32

Chapter 3: Methodology 33

3.1. Uniaxial Tensile Tests 34

3.2. Strain - rate Jump Tests 37

3.3. Artificial Neural Network Modeling 38

3.3.1. Data Collection 40

3.3.2. Data Normalization 41

3.3.3. Network Topology 41

3.3.4. Training 42

3.3.5. Testing and Validation 48

3.4. Mathematical Formulation using the Weights from the ANN Models 48

3.5. Genetic Algorithms (GA) 49

3.5.1. Chromosome Representation 50

3.5.2. Fitness Function 51

3.5.3. Selection Strategy 52

3.5.4. Crossover Strategy 52

3.5.5. Mutation Strategy 53

3.6. Summary 54

Chapter 4: Application of the Proposed Approach to Model the Behavior of Superplastic Materials 55

4.1. Problem 1: Input Parameter Prediction for Strain - rate Sensitivity Index 55

4.1.1. Strain - rate Jump Tests 55

4.1.2. Artificial Neural Network Modeling 55

4.1.3. Mathematical Formulation 62

4.1.4. Optimization Using Genetic Algorithms 65

4.2. Problem 2: Input Parameters Prediction for Flow Stress 72

4.2.1. Uniaxial Tensile Tests 72

4.2.2. Artificial Neural Network Modeling 72

4.2.3. Mathematical Formulation 79

4.2.4. Optimization Using Genetic Algorithms 81

4.3. Summary 87

Chapter 5: Results and Discussions 88

5.1. Problem 1 88

5.2. Problem 2 103

5.3. Summary 111

Chapter 6: Conclusions and Future Recommendations 112

6.1. Conclusions 112

6.2. Unique Features 113

6.3. Future Work 114

APPENDIX I 115

APPENDIX II 116

References 121

VITA 129

List of Figures

Figure 1: Flow Chart Representation of the approach 7

Figure 2: SPF application in automotive industry [Superform-aluminum] 10

Figure 3 : SPF applications in Aerospace Industry [Superform-aluminum]..................... 10

Figure 4 : SPF in medicine and dentistry [Curtis et al, 2005]... 10

Figure 5 : Logarithmic stress-strain rate curve and the sensitivity index (*m*) for a superplastic material [Abu-Farha, F.]. 11

Figure 6 : Superplastic Blow forming process [Abu-Farha, F., 2007]............................. 13

Figure 7 : A Typical Neural Network Structure [Beliganur, N. K. 2007] 16

Figure 8 : A Biological Neuron [Stanford, Neural Networks]... 17

Figure 9 : A Conventional Procedure for Genetic Algorithm [Salem, 2002] 25

Figure 10 : A hybrid ANN-GA approach to the determination of initial process parameters [Mok et al., 2001] 30

Figure 11 : Schematic representation of the ANN-GA approach 33

Figure 12 : Load Frame equipped with a chamber [Abu-Farha, F., 2007] 35

Figure 13 : Stress/ Strain rate curves at different temperatures [Abu-Farha, F., 2007].... 36

Figure 14 : Fracture strain Vs Strain rate at different temperatures [Abu-Farha, F., 2007] 37

Figure 15 : Average strain rate sensitivity index m Vs Strain rate [Abu-Farha, F., 2007]38

Figure 16 : A Neural Network Architecture.. 42

Figure 17 : Chromosome Representation... 51

Figure 18: 3D surface plot of the training data for ANN model...................................... 56

Figure 19 : ANN architecture for Application 1 58

Figure 20 : Comparison of the actual *m* values to the ANN (GS: Step) predicted values59

Figure 21 : Comparison of the actual *m* values to the ANN (GS: Momentum) predicted values... 59

Figure 22 : Comparison of the actual *m* values to the ANN (GS: Conjugate) predicted values... 60

Figure 23 : Comparison of the actual *m* values to the ANN (GS: Delta-Bar-Delta) predicted values ... 60

Figure 24 : Comparison of the actual *m* values to the ANN (GS: Levenberg-Marquardt) predicted values ... 61

Figure 25 : Comparison of the actual *m* values to the ANN (GS: Quick Propagation) predicted values ... 61

Figure 26 : Chromosome Representation ... 66

Figure 27: Strategy 1 GA results for $m = 0.14$ ... 69

Figure 28 : Strategy 1 GA results for $m = 0.2$ ... 69

Figure 29 : Strategy 1 GA results for $m = 0.195$ ... 70

Figure 30 : Strategy 1 GA results for $m = 0.304$ ... 70

Figure 31 : Strategy 1 GA results for $m = 0.54$ ... 71

Figure 32: Strategy 1 GA results for $m = 0.7$ ... 71

Figure 33 : Uniaxial Tensile tests data used for training the ANN ... 73

Figure 34 : ANN architecture for application 2 ... 74

Figure 35: Comparison of Experimental σ values to ANN (GS: Conjugate gradient) predicted values ... 76

Figure 36: Comparison of Experimental σ values to ANN (GS: Momentum) predicted values ... 76

Figure 37 : Comparison of Experimental σ values to ANN (GS: Levenberg- Marquardt) predicted values ... 77

Figure 38 : Comparison of Experimental σ values to ANN (GS: Quick Propagation) predicted values ... 77

Figure 39 : Comparison of Experimental σ values to ANN (GS: Step) predicted values. 78

Figure 40 : Comparison of Experimental σ values to ANN (GS: DBD) predicted value. 78

Figure 41 : GA results for σ = 1.6 MPA ... 84

Figure 42 : GA results for σ = 4.75 MPa ... 85

Figure 43 : GA results for σ = 13 MPa ... 85

Figure 44 : GA results for σ = 48 MPa ... 86

Figure 45 : GA results for σ = 81 MPa ... 86

Figure 46 : Screenshot of an ANN breadboard 88

Figure 47 : Performance of Levenberg-Marquardt gradient search method 90

Figure 48: Screenshot of the user interface for the GA [67] 91

Figure 49 : Comparison of GA predicted values for *m* = 0.14 and the experimental values 93

Figure 50 : Comparison of GA predicted values for *m* = 0.195 and the experimental values 93

Figure 51 : Comparison of GA predicted values for *m* = 0.2 and the experimental value 94

Figure 52 : Comparison of GA predicted values for *m* = 0.304 and the experimental value 94

Figure 53 : Comparison of GA predicted values for *m* = 0.54 and the experimental value 95

Figure 54 : Comparison of GA predicted values for *m* = 0.7 and the experimental value 95

Figure 55 : Results from the ANN-GA model corresponding to strain rates 98

Figure 56 : Experimental results for strain rates (0.0003-0.0006) and (0.001-0.002) 98

Figure 57 : Results from the ANN-GA model corresponding to temperatures (330-380) and (380-430) 99

Figure 58 : Experimental results for temperatures (330-380) and (380-430) 99

Figure 59: Results from the ANN-GA model for strain rates (0.001-0.002) and (0.0003-0.0006) 100

Figure 60 : Experimental results for strain rates (0.001-0.002) and (0.0003-0.0006) 100

Figure 61 : Results from the ANN-GA model for temperatures (330-380C) and (380-430C) 101

Figure 62 : Experimental results for temperatures (330-380C) and (380-430C) 101

Figure 63: Screenshot of the ANN model used in predicting the flow stress 103

Figure 64 : Performance of Conjugate gradient search method 105

Figure 65: Comparison of GA predicted values for σ = 1.6 MPa and the experimental values 107

Figure 66 : Comparison of GA predicted values for σ = 4.75 MPa and the experimental values 108

Figure 67 : Comparison of GA predicted values for σ = 13 MPa and the experimental values.. 108

Figure 68 : Comparison of GA predicted values for σ = 48 MPa and the experimental values.. 109

Figure 69 : Comparison of GA predicted values for σ = 81 MPa and the experimental values.. 109

Figure 70 : Results from the ANN-GA model for strain rates (0.001-0.005) and (0.005-0.01).. 110

Figure 71 : Experimental Results for strain rates (0.001-0.005) and (0.005-0.01)......... 111

List of Tables

Table 1 : Training data for ANN 57

Table 2 : Mean Square Errors 62

Table 3 : GA parameters for experimentation 68

Table 4: Data used for training ANN 73

Table 5 : Mean Square Errors 75

Table 6 : GA parameters used for simulations 83

Table 7 : Experimental parameters for ANN training 89

Table 8: The results obtained using different GS methods 89

Table 9 : Strategy 1 92

Table 10: GA predicted strain rates and temperatures 96

Table 11 : Experimental parameters for ANN training 104

Table 12 : The results obtained using different GS methods 104

Table 13: Parameters used in strategy 1 106

Table 14: GA predicted strain rates and temperatures 106

List of Files

1. Kirthi Bedida’s Thesis.pdf : ~ 2MB (file size)

Chapter 1: Overview

Increasing global population, speedy conversion of the critical habitat for other uses, and rapid pollution etc. have lead to ecological imbalance and thus environmental problems which respect no borders and threaten both life and prosperity. Researchers in the field of manufacturing tried to address this issue through innovations. Environmental concerns and economical expectations of customers are the major driving forces for the automobile sector to come up with clean technologies and explore all possible means to meet these requirements. Conventional materials are continuously replaced with more efficient and much durable light-weight materials such as aluminum and magnesium. This calls for extensive research to explore the mechanical properties and behavior of these metals.

Magnesium is one of the lightest metals on earth and is widely in application due to its light-weight and also due to its good mechanical and electrical properties. Research indicates the use of magnesium components in automobiles to have an environmental advantage as well. BMW claims to have been able to reduce engine weight by 24 % and also reduce engine noise transmission by using magnesium [Curtis et al, 2005]. Owing to its ease in machining and processing, Mg has been in use not just in automobile and aerospace industries but also in medical applications. However, there are some issues with magnesium which does not lend it be used easily for sheet metal applications, this is mainly due to inferior ductility at room temperature.

But it was observed that Mg alloys behaved differently when worked upon at high operating conditions such as high temperatures and high strain rates; this process is termed Superplastic forming (SPF). This phenomenon has gained an explosion of interest in the recent years and a lot of research is being carried out in this field. Materials which exhibit these properties are termed Superplastic materials. SPF offers many advantages when compared to conventional forming techniques. A few to list would be enhanced design freedom, low cost tooling etc. Superplastic alloys can undergo large uniform strains before they fail. For deformation in uni-axial tension, elongations to failure in excess of 200% are usually indicative of Superplasticity, although several materials can attain extensions greater than 1000%. The highest elongations reported are 4850% and 7750% in a Pb-Sn eutectic alloy; 5500% and greater than 8000% for an aluminum bronze [Superform-aluminum]. Superplastically formed materials are in wide use in automobile, aerospace, medical and architectural applications. Latest versions of Morgan sports cars are featuring one piece wings made from Superplastically formed aluminum. A few other examples are wing access panels in the Airbus A310 and A320, bathroom sinks in the Boeing 737, turbo-fan-engine cooling-duct components, external window frames in the space shuttle, front covers of slot machines, and architectural siding for buildings etc.

Though the superplastic phenomenon has received wide spread acceptance, it has not been applied in large scale as anticipated due to certain drawbacks such as unavailability of models which could accurately capture the behavior of superplastic materials during deformation and thus help predict the failure of the material beforehand.

Different techniques to model the behavior of superplastic materials are in use, such as analytical modeling, Finite element modeling, Artificial Neural Networks, Hybrid modeling etc. The use of these approaches has helped in overcoming most of the problems associated with SPF. The present research focuses on inverse modeling, i.e. predicting the process parameters which would lead to a specific condition of the metal.

This is attempted with the aid of two meta- heuristics, Artificial Neural Networks (ANN) and Genetic Algorithms (GA). The use of such tools reduces the time, resources and effort involved in modeling a process without forgoing the final result.

1.1. Motivation

With SPF offering so many advantages over conventional forming methods it appears worthy to spend resources on addressing the drawbacks of the same. Using meta-heuristics to model the SPF offers considerable advantages in comparison to other modeling techniques.

1.1.1. Time and Resources

The application of ANN-GA approach in modeling the SPF reduces the time and resources involved in modeling the process by simulating the process itself. Especially with ANN modeling there is no need to assume an underlying data distribution such as is usually done in statistical modeling and also they work well for complex non-linear relations.

1.1.2. Controlling the Process

In the present approach ANN and GA are integrated to inverse model the input parameters that must be maintained to achieve a given value for the output parameter. For example, this involves predicting the operating conditions such as the strain rate and temperature that must be used to attain a specific value of strain rate sensitivity index (*m*), one of the important mechanical properties of superplastic materials. High strain rate sensitivity leads to a high resistance to necking and enables higher tensile elongations in superplastic materials. The deformation rate during superplastic forming needs to be controlled so that the forming process can be optimized to reduce the forming times. Therefore, knowing the process parameters affecting the m value beforehand would aid in significantly controlling the process. This will also eliminate the need to follow a trial and error method during the actual forming process to achieve desired output.

1.2. Objectives

The objectives of the present research are to model the behavior of superplastic materials and to build an approach for inverse modeling which would help in predicting the process parameters necessary to attain a given output. This is achieved with the use of meta-heuristic; ANN and GA behave analogous to the human nervous system, and in fact are considered as generalized mathematical models of nervous system [Stanford, Neural Networks].

1.3. Approach

The above mentioned objectives are achieved through a procedure which involves:

1. The first step involves collecting data for training the ANN models. The data required to train the models is obtained from literature for the first case study and from the experiments conducted using INSTRON 5582 universal testing machine at different operating conditions for the second case study.
2. The data collected is normalized before feeding it to the model, the normalization co-efficient is computed based on the minimum and maximum

values found across all of the data sets. This is done to enable the neural networks produce accurate forecasts.

3. The Artificial Neural Network (ANN) models are trained with the pre-processed (Normalized) data. The models are trained with six different gradient search methods. The stop criterion for the training is the number of epochs. The gradient search method yielding the lowest mean square error is considered and further experimentation is carried out using the same.
4. The ANN models is cross validated using the experimental data which is different from the one used for training the networks. Cross validation is used as a stop criterion when the network gets over trained.
5. Collecting the information pertaining to the training of the networks which includes the weights and biases the model uses to map the relationship between the input and the output parameters.
6. The neural network models are validated by feeding a totally unseen set of data and predicting the output for the same and these results are compared to the already existing results from the literature and the experimental results.
7. Formulating the relationship between the input process parameters and the output with the help of the weights and biases gathered at the end of the training.
8. The mathematical relationship from the previous step is used as the objective function for the Genetic Algorithm (GA).
9. The various constraints on the objective function and the process parameters are defined and also the information pertaining to the limits of the process

parameters and other experimental parameters is fed to the GA. Multiple trials are made.

10. The genes in each chromosome generated at the end of the trials are the process parameters achieved through inverse modeling.
11. The results obtained from these trials are compared to the experimental data and the hybrid modeling approach is validated.

This approach has been represented schematically in figure 1.

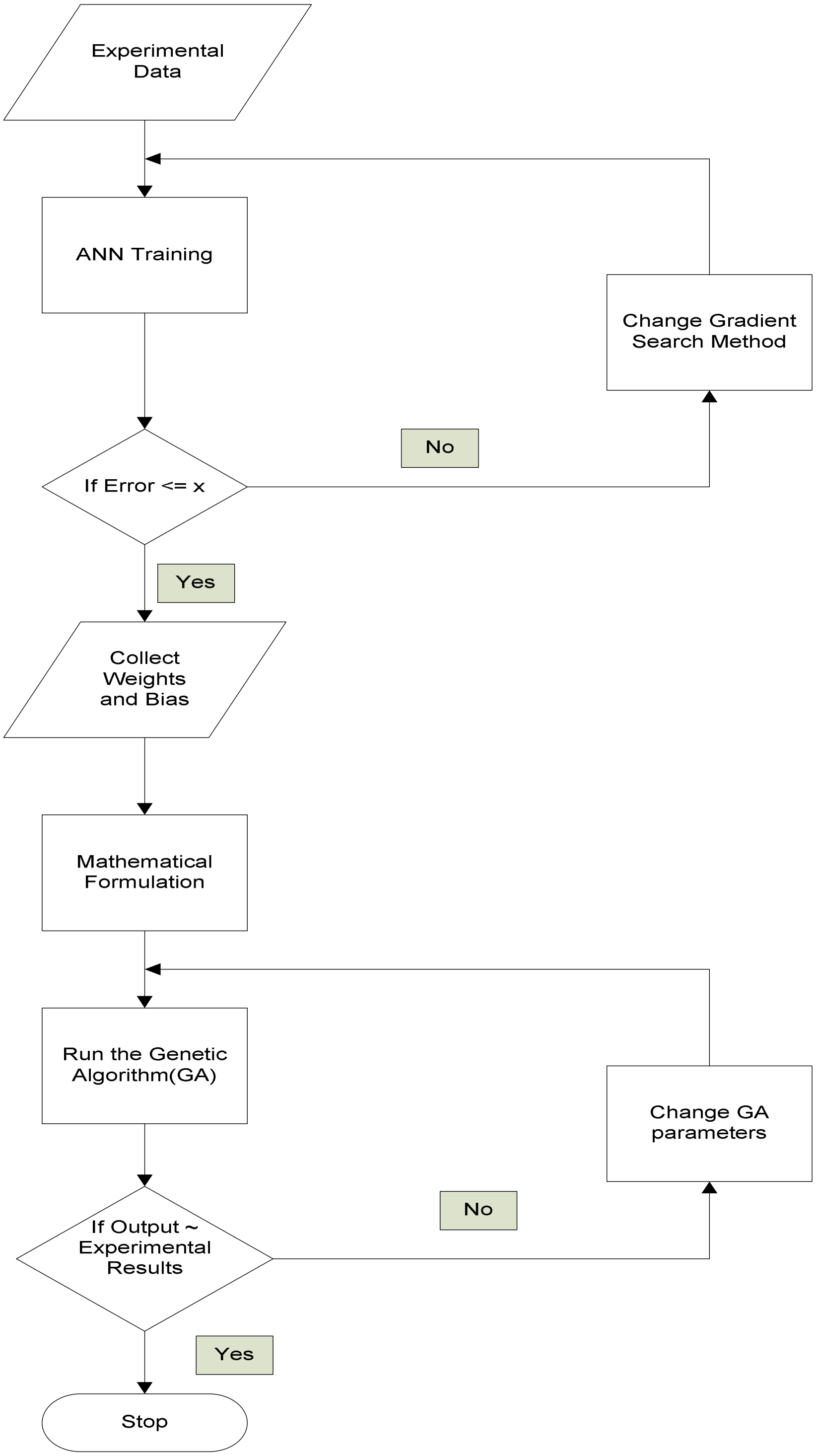


Figure 1: Flow Chart Representation of the approach

1.4. Thesis Organization

Chapter 2 gives an introduction to the concept of SPF and about the various modeling techniques in use to model SPF. Different meta-heuristic methods such as, ANN and GA in particular are also discussed. This is followed by an introduction to inverse modeling, different approaches in existence for inverse modeling and the significance of the present approach and literature related to above topics and recent developments are also discussed.

A detailed discussion of the methodology used in the present research to inverse model using ANN and GA is presented in chapter 3.

Chapter 4 includes the application of the proposed approach on data characterizing the superplastic behavior of Mg AZ31 alloy, and the data for this is obtained from the uniaxial tensile tests and strain rate jump tests.

Chapter 5 discusses the experimental results obtained from applying the approach to SPF

Conclusions and the anticipated future work are put forth in the last chapter.

Chapter 2: Literature Review

2.1. Introduction

Superplastic forming (SPF) is considered as a net shape forming method which offers many advantages such as weight reduction and cost reduction. In addition to these Superplastic materials exhibit properties such as extended elongation, these materials can undergo elongation which is almost 200% excessive of the conventional materials. And, also can form very complex shapes which are not easily achievable using conventional forming methods, the die cost is significantly reduced in this approach and also this is a single step process, Superplastic materials have proved to be environmentally benign too. For a material to behave Superplastically, it needs to have fine and stable grain structure, should exhibit controlled deformation rate and the forming temperature needs to be high, at least half the absolute melting temperature of the material. Superplastically formed materials are now in wide spread use in many applications mainly in the automotive, aerospace and medical industry. Significant reduction in weight of the automotive components could be achieved through the replacement of conventional materials with superplastic materials. It has been observed that in the past couple of years there is a gradual decrease of 25% to 30% of the weight in the vehicles. Studies indicate that this reduction in weight has led to energy saving and environmental protection [mse-mtu, Superplastic forming]. Figure 2 indicates some superplastic applications in the automotive industry.

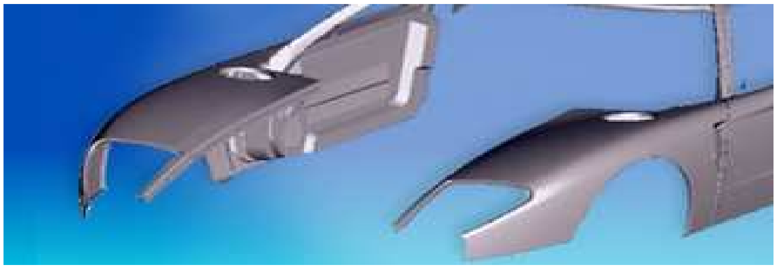


Figure 2: SPF application in automotive industry [Superform-aluminum]

The use of SPF in aerospace industry is mainly driven by omnipresent need for structures of optimum specific strength and stiffness, with minimum cost. Figure 3 indicates some such applications.

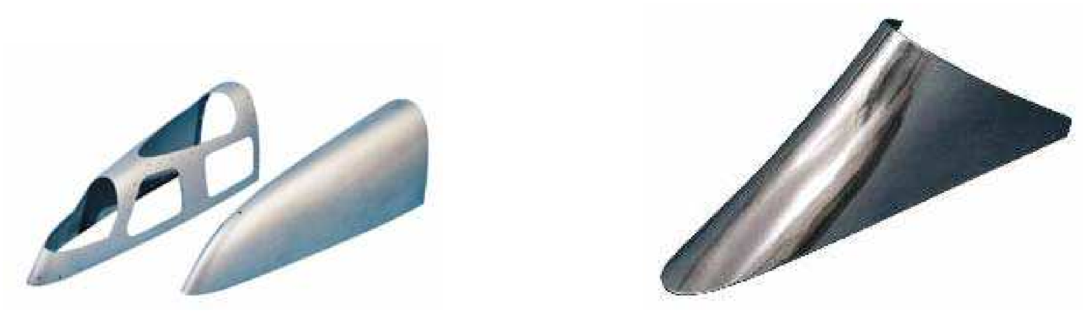


Figure 3 : SPF applications in Aerospace Industry [Superform-aluminum]

Reliable and net shape prostheses could be achieved in medical and dentistry through the application of superplastic forming. The following Figure are some examples of such applications

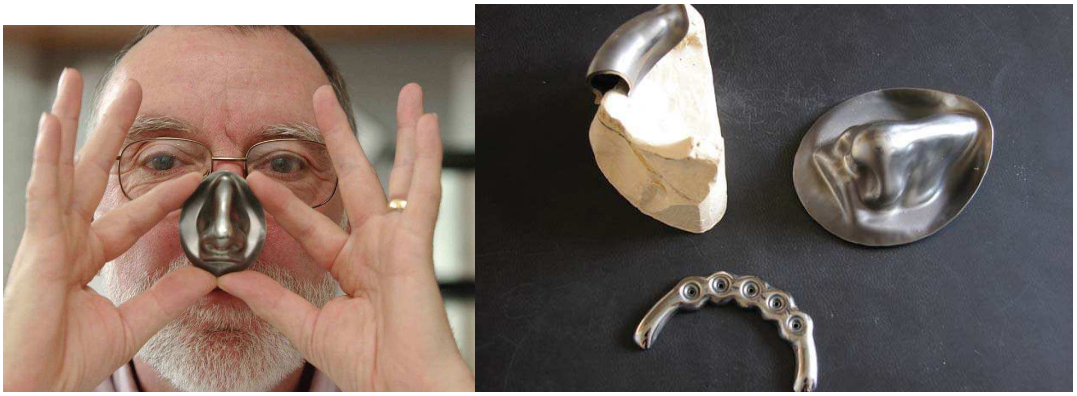


Figure 4 : SPF in medicine and dentistry [Curtis et al, 2005]

Superplasticity can be characterized by low flow stress and high sensitivity of the flow stress to the strain rate; this can be simply expressed by the following relation:

$$\sigma = C\ \dot{\varepsilon}^{m} \qquad (1)$$

Where σ is the flow stress, $\dot{\varepsilon}$ the effective strain rate, C is the strength coefficient and *m* is the strain rate sensitivity index. The value of m ranges from 0.3 to 0.7 for superplastic materials. Figure 5 indicates a typical stress-strain rate curve for a superplastic material.

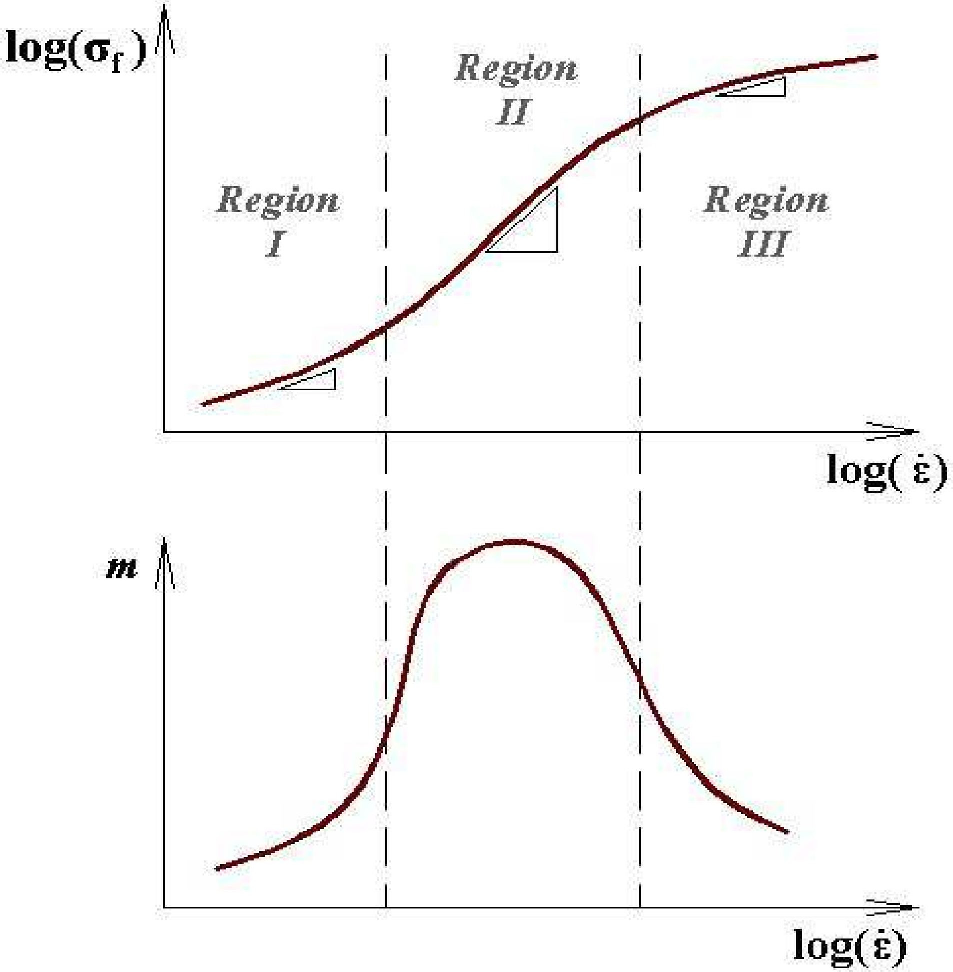


Figure 5 : Logarithmic stress-strain rate curve and the sensitivity index (*m*) for a superplastic material [Abu-Farha, 2007].

The logarithmic curve can be divided into three regions, superplasticity is indicated only be region II of the curve where the material exhibits high strain rate sensitivity index (*m*). The high value of *m* confers a high resistance to neck development and thus results in high tensile elongations. Though the deformation process in this region is not very well understood because of the absence of any mechanism that could explain the deformation in this region, it is believed that grain boundary sliding accompanied by diffusion or dislocation glide and climb is the dominant mechanism.

2.2. Superplastic Blow Forming Method

Figure 6 indicates a schematic representation of the superplastic blow forming process. Blow forming is extensively used to form complex shapes. This is mainly a three step process, first a metal sheet is clamped and gas pressure is applied to it, generally argon gas is used for this purpose as it also helps maintain a protective environment. Initially the sheet is not in contact with the die and the deformation is concentrated at the pole, greatest strain occur at this region and once the material gets in contact with the die surface the material is locked due to friction and thus any further deformation is prevented. The remaining regions deform thus expanding to take the shape of the die. Usually the corners of the dies are more prone to failures as they are the last to fill and more strain occurs in this region.

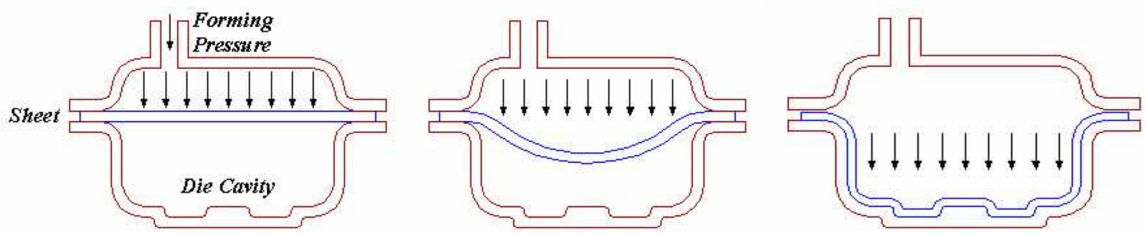


Figure 6 : Superplastic Blow forming process [Abu-Farha, F., 2007]

2.3. Modeling Using Numerical Methods

Though superplastic forming offers many advantages over conventional forming methods, there are certain drawbacks attached to the process and the materials which hinder the wide spread use of these in many applications. A few such disadvantages to list would be:

1. Slow forming process
2. Higher material cost
3. Trial and error in design
4. Lack of accurate models to model the deformation and failure.

Considerable research is being done to develop more accurate models to model the behavior of these materials. Over the years many process models based on analytical and simplified numerical methods have been developed to provide a better understanding of the process mechanics [Chandra, 1998]. Use of Numerical Analysis techniques in the present day research has reduced the use of trial and error procedures considerably. Tikkaya, et al. [2000] has presented a review on the applications of Finite Element Simulation in the sheet metal forming industry. Finite Element Simulations are used to model and analyze a wide range of problems that are too complicated to solve by classical analytical methods. Khraisheh, et al. [2004] used Finite Element Simulations to model and optimize the superplastic blow forming process. Tan and Liew [2004] used Finite element Simulation method to analyze three dimensional SPF.

But Finite element modeling (FEM) approach has some drawbacks such as when a finite element software runs on a PC it uses almost all the computer resources and takes long time to finish one analysis of a certain working condition. Also the trial and error methods are still used widely in numerical simulation process to endure an appropriate working condition. Due to this and the property of FEM, the virtual simulation process will occupy a large quantity of time and computer resource [Sadghei, 1997]. While describing the behavior of material during deformation, numerical simulations with the FEM are reliable only when the law to describe the material is described correctly. But during the hot deformation process, there are many factors that influence the flow stress of the metal. The effects of these factors on the flow stress are complex and the relationship between the flow stress and these factors is no-linear [Dehsmuk,2004].

Lately Artificial Intelligence tools such as ANN and GA are widely used in modeling different manufacturing processes. Considerable research has already been done in the field of superplastic forming using these methods. The following sections provide an overview of these AI tools and previous work involving these tools.

2.4. Artificial Intelligence in Manufacturing

A great deal of research has been done and is still in progress involve the application of Artificial intelligent tools such as ANN, GA and Fuzzy logics in manufacturing. This research focuses on applying ANN and GA to inverse modeling of SPF. The following sections are devoted to a discussion of literature on ANN and GA.

2.4.1. Artificial Neural Networks (ANN)

Neural Networks have gained an explosion of interest in the recent years due to their capability to learn by example and be able to predict. Artificial Neural Network (ANN) is an information processing system which functions analogous to a human nervous system which is composed of biological neurons, as shown in Figure 7. ANNs have been developed as generalizations of mathematical models of human cognition or neural biology [Kohnen, 1997]. These networks are composed of a large number of processing elements called the neurons which work co-ordinate with each other to solve specific problems. Each processing element multiplies and input with a set of weights, and nonlinearly transforms it to a desired output. A conventional neural network has multiple layers of processing elements, the first or the input layer is where the input parameters are fed to the network, and these input parameters are processed in the hidden layers and the last layer is the output layer. Figure 8 is a schematic representation of a typical neural network structure.

In actual neurons the dendrite receives electrical signals from the axons of other neurons whereas in the ANN these electrical signals are represented as numerical values.

At the synapses between the dendrite and axons, electrical signals are modulated in various amounts. This is modeled in the ANN by multiplying each input value by a value called the weight. An actual neuron fires an output signal only when the total strength of the input signals exceeds a certain threshold. We model this phenomenon in an ANN by calculating the weighted sum of the inputs to represent the total strength of the input signals, and applying a function on the sum to determine its output.

The emergence of Artificial Intelligence (AI) dates back to the 1950s. Neural Networks in particular are gaining more and more visibility and are in wide use in the field of manufacturing mainly due to the several advantages they offer [Fausete, 1994]. A Neural network could be used for solving problems where pre hand analysis is not readily available; they are able to describe accurately even nonlinear relationships, and using an ANN does not require a mathematical model.

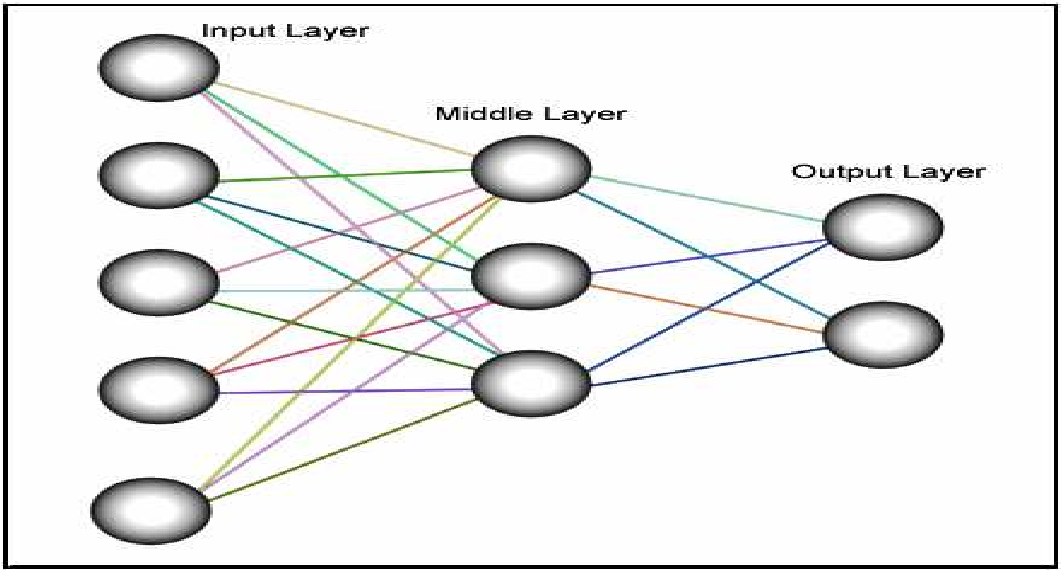


Figure 7 : A Typical Neural Network Structure [Beliganur, 2007]

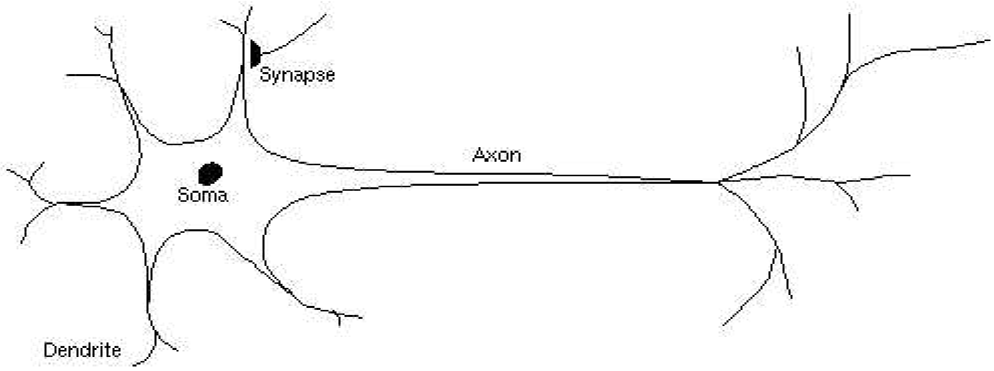


Figure 8 : A Biological Neuron [Stanford, Neural Networks]

Neural Networks are massively parallel interconnected networks of simple elements and their hierarchical organizations are intended to interact with objects of the real world in the same way as biological nervous systems do [Kohonen, 1988]. These can be classified into two categories, namely unsupervised learning and supervised learning depending on the amount of guidance given to the networks during the training process. The former type of network classifies the input sets without any knowledge about the data, the latter works on the basis of the difference between output generated and the desired values [Beliganur, 2007]. Neural Networks are used for modeling and control, optimization, and prediction, monitoring and diagnosis, pattern recognition etc.

Neural networks have been extensively used for applications such as pattern and feature recognition. Some of the very first researchers who used ANN approach for feature recognition in manufacturing were Hwang and Henderson [1992]. Osakada [1991] has utilized the ability of pattern recognition of neural networks to determine the number of forming steps to achieve the desired shapes of rotationally symmetric products. A three layer back-propagation neural network (BPN) was used for feature recognition from a solid B-rep solid model in order to automate the process planning of manufactured products in [44]; the results indicated that the networks were consistent in recognizing the features.

A back-propagation algorithm was used to capture the system behavior during a grinding process in [Satyanarayana et al., 1994], inputs such as feed rate, depth of cut and wheel bond type are associated to outputs such as surface finish, force and power.

2.4.1.1. Finite Element Modeling and Artificial Neural Network

FEM and ANN modeling are applied in conjunction by many researchers. Results from the finite element simulations at different working condition parameters such as material thickness, punch speed, friction coefficient between punch, die and sheet metal and blank holder force were used to train a three layer back propagation neural network in [Lin et al., 2006]. The learning algorithm used here was Levenberg - Marquardt. The network was trained to predict the Limiting Dome Height (LDH). The outcome from the neural network was validated using the Finite element simulations again.

Similar approach is found in [Clocke et al., 2006], where FEM programs and ANNs are used to predict the ductile fracture in a cold forming process to enable efficient process design. FEM was used to model the entire process to get the entire forming history starting from the first principle stress, equivalent stress and the equivalent strain starting with the first deformation to the first crack occurrence. This database consisting of the information pertaining to failure during a forming process was used to train the neural network model. It was observed that the ANN model could predict the failure for new forming histories.

Singh and Kumar [2005] have used a feed-forward back propagation neural network model to predict the thickness along the cup wall in hydro-mechanical deep drawing. The experimental results from the hydro mechanical deep drawing process were used to train the ANN model and the trained model was tested using a new set of data for the prediction of thickness strains. A finite element simulation of the same process is also attempted. The results from the experiments, ANN and the FEM were compared. A similar approach is employed by Lefik and Schrefler [2003] to model the behavior of physically non-linear body and by Casotto, et al., [2005] to predict the final geometry of forged rings after cooling and also by many other researchers.

The concept of using ANN in constitutive modeling was proposed by Ghaboussi et al. Kong and Hodgson [1999] used a combination of constitutive models and ANN models to predict the hot strength of steels. This approach was adapted to address the limitations of these models when used independently. The author in fact considers ANN models as alternatives to constitutive modeling due to their ability to predict accurately and correlate nonlinear relationships between inputs and outputs. Javadi and Tan [2003] in have used a novel approach to predict the stress strain relationship in a material. The ANN model in this case was trained using raw data from experiments which represents the mechanical response of the material to which the load is applied. This trained network is then incorporated into a finite element program to substitute the conventional constitutive model. The results obtained from such an approach were validated with results from the conventional constitutive models.

Pathak et al. [2005] modeled the sheet metal bending process using ANN. The approach was used to predict the responses of the process such as the stresses, strains, springback, loads etc. The inputs for the ANN were sheet thickness and the die. The networks were trained for 44 cases analyzed using FEM and then the ANN model was then tested for five new patterns. The approach proved to economize the computation time used in modeling such process using conventional methods.

2.4.1.2. Artificial Neural Networks and Superplastic Forming

Bariani et al. [2001] used neural networks to model the rheological behavior of nickel-based superalloys under varying hot deformation conditions. Data from constant strain rate compression experiments and continuously varying strain rate compression experiments was used to train the back propagation neural network model. These experiments were carried out using Gleeble 3800 ^{TM} system and on cylindrical specimens. The results when validated using experimental data indicated that a properly trained neural network model can be used as an alternative approach to constitutive equations when material is deformed under varying conditions.

Chen et al. [2002] utilized a single hidden layer fuzzy-neural network model trained using a back-propagation algorithm to predict the microstructure evolution and the constitutive relation of 15 vol% SiCp/LY 12 under various superplastic deformation conditions. The process parameters of superplastic deformation were considered as the input variables and the grain size, volume fraction of cavities and true stress were taken as the output variables. The results when compared with the experimental data proved the neural network was capable of prediction.

Multiple investigations using artificial neural network analysis have been illustrated by Hsiang et al., 2005. A database containing various aspects of magnesium alloy hot extrusion process was constructed. ANN is used to determine the shapes of dies of various extrusion ratios, predict the optimized process parameters for the process and finally predict the tensile strength and maximum extrusion load of the finished product. This study has set the tensile strength and extrusion load of the finished product as the quality characteristics using the Taguchi method, in order to obtain optimal process parameters. Then the weights of the important parameters are changed by conducting the analysis of variance (ANOVA) to analyze the influence of parameters on the extrusion process.

Hsiang, Kuo and Yang [2006] have used a back propagation neural network trained based on steepest gradient search method to analyze the relationship between the temperature and the tensile strengths of a rectangular tube at various extrusion speeds and extrusion ratios. The results from the ANN were confirmed by comparing them to the results from the experiments conducted at different extrusion speeds and ratios. Observing the microstructure of helped in acquiring the relationship between the sizes of the crystalline grain of the magnesium alloy at different working conditions such as temperature.

An ANN constitutive model was developed by Jamal et al., 2007 to model the high strain rate deformation in Al 7075-T6. The research proved that ANN modeling for this application took less development time over other traditional mathematical approaches and also indicated that the existing constitutive model could be easily extended with experimental data using an ANN.

Okuyucu et al. [2007] developed an ANN model to analyze and simulate the correlation between the friction stir welding parameters of aluminum and the mechanical properties. The ANN model was trained and tested using the data from the welding process and the tensile tests.

One major drawback of ANN modeling is the inefficient extrapolation capability. This limitation of an ANN was addressed by Bruschi and Nego [2005] by including the mechanistic knowledge of an analytical equation in ANN modeling. The model chosen as the analytical component here is the Norton-Hoff model, widely used in FEM calculations. It is an empirical constitutive equation representing the instantaneous flow stress as a function of the instantaneous values of the process parameters:

$$\sigma = k\varepsilon^{n}\,\dot{\varepsilon}^{m}\,exp\,(\beta/T) \qquad (2)$$

Where k, n, m and β are material coefficients. The results proved that this approach helped in addressing the extrapolation limitation of ANN.

2.4.2. Genetic Algorithms

Genetic Algorithm (GA) is an optimization tool which falls under the category of evolutionary algorithms. Genetic Algorithms are a relatively new class of optimization techniques, which are generating a growing interest in the engineering community. They are well suited for a broad range of problems encountered in science and engineering [Goldberg et al., 1989]. GAs were first developed by John Holland in 1975. These algorithms use ideas and get inspiration from natural evolution and adaptation. The principle behind their working is the maintenance of a population of solutions to the problem that evolve towards the global optimum.

GAs are based on the triangle of genetic reproduction, evaluation, and selection. These are classified into two types depending on the genetic representation namely the binary coded GA and the real coded GA. A conventional binary coded GA uses binary representation for the genes but in a real coded GA the gene transformation is allowed directly on the real valued representation of the design variables. The genetic reproduction takes place by means of the crossover and the mutation operator. The crossover schemes available are classified into two types Genotype and Phenotype. In natural systems one or more chromosomes combining to form a total genetic prescription for the construction and operation of an organism are called the *genotype;* on the other hand, organisms formed by the interaction of the total genetic package with the environment are called the *phenotypes* [Beliganur, 2007].

Figure 9 represents a conventional GA as described by Salem, 2002. The following steps illustrate the working of a GA.

1. Initialization: An initial population of chromosomes is randomly generated.

2. Evaluation: The fitness used as a measure to reflect the degree of accuracy of the individual is calculated for each individual in the population.
3. Selection: The selection probabilities for each individual in the population are defined and the selected individuals enter a mating pool, the chance of

a chromosome to be selected for mating is dependent on its fitness function value. Hereafter the overall quality of the population increases.

4. Crossover: This operator combines the features of two parent individuals to form two children individuals that may have new and possibly better characteristics compared to those of their parents and play a central role in the GA optimization process.
5. Mutation: This is generally performed to prevent premature convergence of the genetic algorithm search.
6. Replacement: The offspring population generated at the end of the mutation replaces the parent population either completely or partially, and this new population is taken further for more experimentation.
7. Termination: This is generally a user defined condition indicating when the GA could terminate. It could be the number of generations, or a specific objective function value etc.

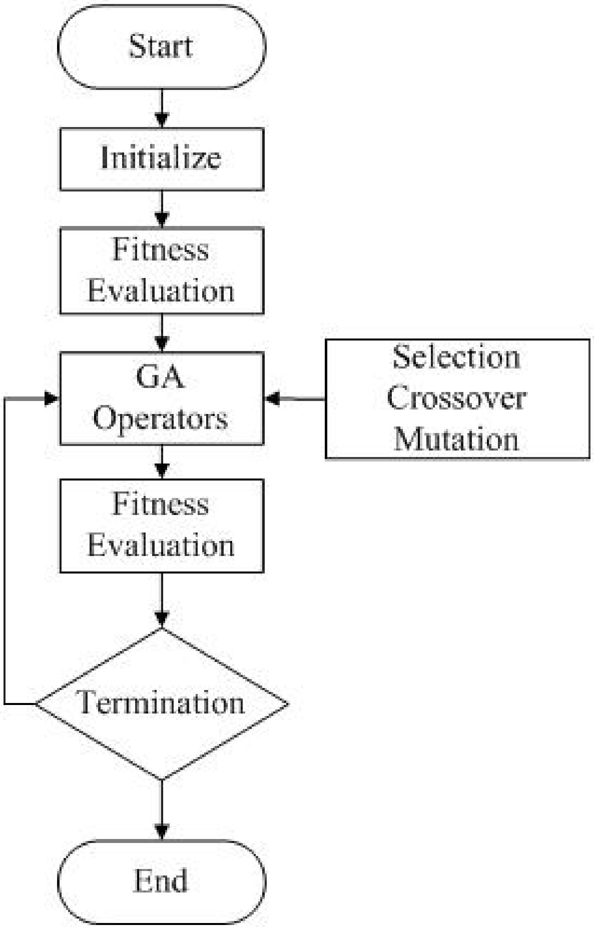


Figure 9 : A Conventional Procedure for Genetic Algorithm [Salem, 2002]

The fact that GAs use only objective function information without the need to incorporate highly domain-specific knowledge points to both the simplicity of the approach from one side and its versatility. This means that once a GA is developed to handle a certain problem, it can easily be modified to handle any type of problems by changing the objective function in the existing algorithm. This is why GAs are classified as general-purpose search strategies [Goldberg, 1989]

Lin, J. and Yang, J. [1999] developed a genetic algorithm based multiple objective optimization technique and applied it to determine the unified viscoplastic constitutive equations for superplastic alloys.

The superplasticity of a material greatly depends on the temperature and occurs only in a narrow range of strain rates with an optimum value which is unique to each material. Thus, the control of deformation rate in a superplastic forming process is extremely important. In order to accurately simulate the superplastic forming process and thoroughly understand the superplastic deformation mechanisms of materials, unified viscoplastic constitutive equations for superplastic alloys need to be established. One of the most difficult tasks encountered in developing the viscoplastic constitutive equations is, how to accurately determine material constants arising in the equations from experimental data and for a set of viscoplastic constitutive equations, the flow stress is related to the accumulated plastic strain, and the plastic strain rate is often a function of stress, hardening parameters, grain size, etc. Apart from this, the equations are not continuous due to the existence of a yield stress parameter. This makes the constitutive models very complicated and strongly non-linear behavior in nature.

It has been found that the GA-based optimization technique is very effective in solving this kind of problems. The authors develop a generic GA to determine material constants for a range of constitutive equations, such as creep damage, cyclic plasticity and superplasticity.

The GA based multiple objective optimization was successfully used for determining material constants arising in viscoplastic constitutive equations, and a software package, named MECHOPT, was developed.

The effectiveness of the technique was demonstrated by determining the material constants within the following equation for the experimental values of Ti-6Al-4V at 927 C.

$$\dot{\varepsilon}_p = \left(\left(|\sigma - X| - R - k\right)/K\right)^n d^{-u} \tag{3}$$

$$\dot{X} = C\dot{\varepsilon}_p - \gamma X|\dot{\varepsilon}_p| \tag{4}$$

$$\dot{R} = b(Q - R)\ |\dot{\varepsilon}_p| \tag{5}$$

$$\dot{d} = \left(\alpha + \beta|\dot{\varepsilon}_p|\right)d^{\gamma_0} \tag{6}$$

$$\sigma = E\left(\varepsilon_T - \varepsilon_p\right) \tag{7}$$

Where ε_T and ε_p are total and plastic strain; *X* and *R* the hardening variables and d the grain size. *K, k, n, u, C,* γ*, b, Q,* α, β, γ_0 are material constants to be determined from experimental data. E is Young's modulus and E = 1000 MPa for Ti-6Al-4V at 927 C.

The optimization technique provided an easy means for simplifying the unified constitutive equations and for the Ti-6Al-4V at 927 C model the original four differential equations was reduced to two and the number of material constants were reduced to seven from eleven.

2.4.3. Integrating Genetic Algorithms and Artificial Neural Network

Once an ANN model is properly trained it can be used in conjunction with other optimization techniques such as GA to optimize process parameters with regard to some metric that determines the final performance. This could be achieved through the search over process parameters for the combination of values which yield the best performance of the objective function. The goal in process design of any manufacturing process is to determine the critical process parameters and optimize them in order to produce the final parts within the desired specifications. Literature illustrates a lot of research where GA and ANN are integrated and applied to solve problems in manufacturing and other fields too.

Holter et al [1998] developed a prototype 'controller' to study the integration of several functions and the utilization of status data to evaluate scheduling and control design alternatives of a single manufacturing machine. The prediction capability was implemented by using ANN, simulation, and GA. The controller attempts to control the planning, scheduling, monitoring, execution, and interfacing of resources. Neural networks are used to predict the behavior of different sequencing policies available in the system and using these policies a simulation is conducted. The results from the simulation are later analyzed and GA is applied to further improve the results. The 'intelligent controller' successfully produces new alternatives which were not thought off by the designers for scheduling and this was used to modify the decision making structure.

Cook et al. [2000] developed a GA-NN system to predict the process parameter values of a particleboard manufacturing process. Particleboard is a composite wood product used in various furniture and building applications. One of the main measures of the quality of the particleboard is the strength of the final board which can be calculated by using the process parameters, i.e. internal bond and modulus of rupture. An ANN model consisting of 26 input parameters and 3 output parameters is used to predict the critical strength parameter based on the process operation parameters and conditions. A GA is then applied to the trained NN model (which acts as the fitness function) to determine which process parameter values would result in the desired product characteristics. These parameter values result in desired levels of the strength parameters for given operation conditions.

Mok et al [2001] developed a hybrid ANN and GA system for the determination of set of initial process parameter settings for injection molding. Initial process parameter setting is a very important step in the molding process as it determines the development time and the quality of molded parts. The system was developed to replace the trial-and-error and operators intuition method, which was followed in the molding industry to determine the initial processes parameters. The NN was employed to model the complex non-linear relationships among the parameters involved in the initial process parameter setting of injection molding, and the GA was applied to determine a set of near optimal process parameter for injection molding.

Figure-10 shows the ANN-GA system which was used for determining the initial process parameters of injection molding.

The system was successfully implemented and without violating the quality criteria, the ANN-GA system was able to recommend the initial process parameters which resulted in molding of high quality molded parts.

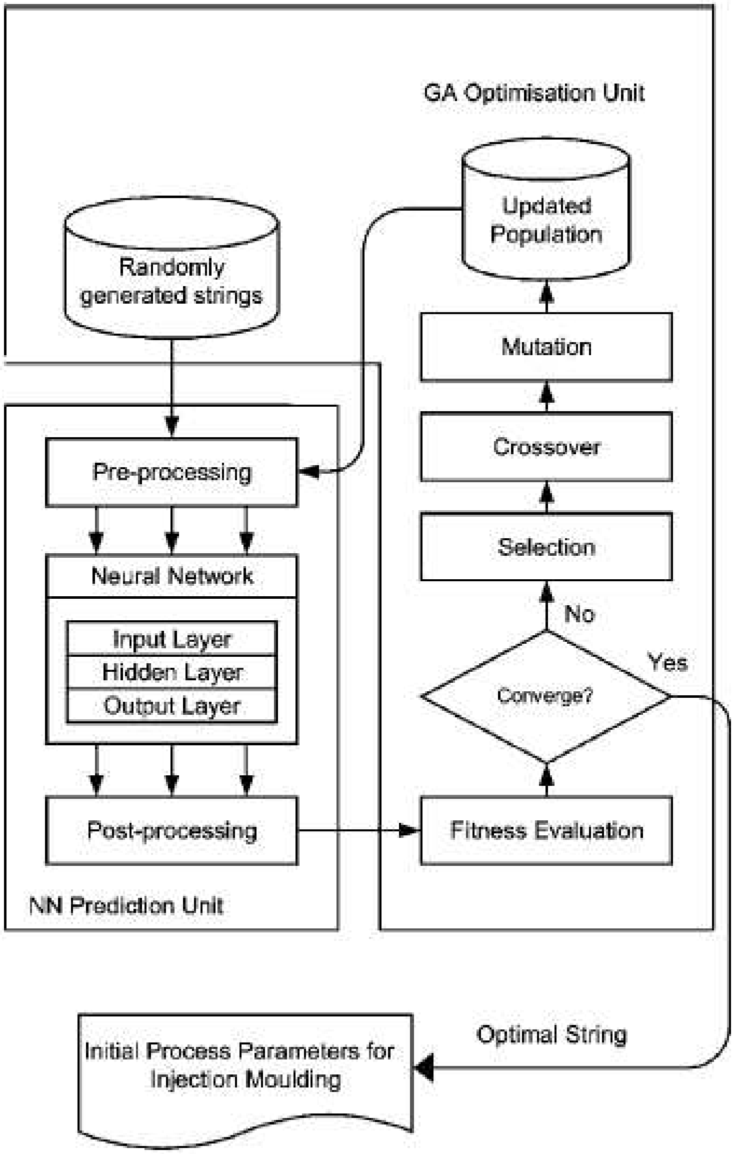


Figure 10 : A hybrid ANN-GA approach to the determination of initial process parameters [Mok et al., 2001]

Kumar et al. [2003] introduced the concept of "artificially intelligent genetic algorithms", in the form of an ANN, to fit the data pertaining to superplastic deformation of several metal alloys. The authors used the combination of ANN with GA to transform a randomly generated initial population of deformation data to a final set of populations that contained solutions approximating to the actual deformation data. In this study, a constitutive equation that describes the superplastic behavior of the material was considered.

$$\dot{\varepsilon} = C_1\left[(\sigma - \sigma_0)\left(1 + \alpha\left(\ln\frac{\sigma - \sigma_0}{\sigma_m} - 1\right)\right) + \alpha\sigma_m\, exp\left(\frac{-1}{\alpha}\right)\right] \quad (8)$$

$$C_1 = C\exp\left(-\frac{Q}{RT}\right)$$

(9)

$\dot{\varepsilon}$ = Strain rate of the deformation, σ = Applied stress, C_1 = material and temperature dependent constant, σ_0 = long range threshold stress, σ_m and α mean and standard deviation of the internal stress distribution arising from the sliding process, C = material constant, Q = activation energy term characteristic of the sliding process, R = Universal gas constant, T= absolute temperature.

The experimental data in the form of stress-strain rate responses of a material at different temperatures were analyzed using an ANN controlled genetic algorithm. The genetic algorithm used in this research was capable of performing parallel computation, thus could be run on a multi node cluster, therefore reducing the overall computation time.

The fitness function used for GA evaluation is,

$$E = \sqrt{\sum \frac{\left[\dot{\varepsilon}_{pre} - \dot{\varepsilon}_{obs}\right]^2}{N}}$$

(10)

A feed forward ANN which was trained using supervised learning strategy was used to control the mating rate of the genetic algorithm. The ANN had 40 input nodes, two layers of 90 and 35 hidden layers and 5 output nodes. A fraction of the output of this ANN was utilized for the GA mating process. A similar ANN was also used to control the mutation and migration rates of the GA. The model was successfully implemented to predict the deformation parameters that govern the constitutive equation of different superplastic materials by validating the results against experimental data.

2.5. Summary

This chapter includes a discussion on the previous research in the field of Superplastic forming, Modeling techniques involved in Superplastic forming, Artificial Neural Network applications in manufacturing, Genetic Algorithms in manufacturing and the application of ANN-GA integration models to different applications and to Superplastic forming.

Chapter 3: Methodology

The Flow diagram in figure 11 is a schematic representation of the approach discussed in this chapter. All the steps indicated in the flow diagram are explained in detail in the following sections.

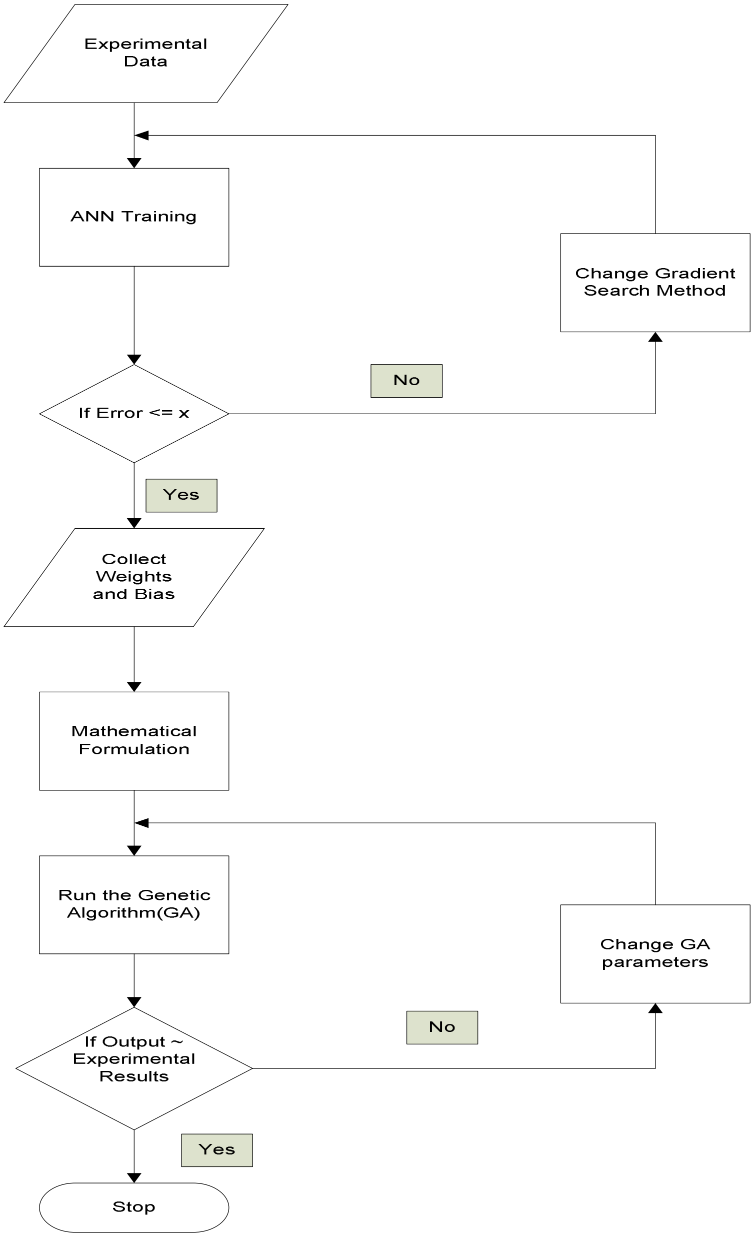


Figure 11 : Schematic representation of the ANN-GA approach

A study involving the comprehensive investigation of the elevated temperature superplastic behavior in the AZ31-H24 magnesium alloy has been carried out by F.K. Abu-Farha and M.K. Khraisheh. Various mechanical aspects and microstructural changes during superplastic deformation have been studied. The mechanical aspects considered in this study are the flow stress, fracture strain and the strain rate sensitivity index *m*. These aspects of superplastic deformation are investigated through sets of uniaxial tensile tests and strain rate jump tests. The details of these tests are covered in the following sections.

3.1. Uniaxial Tensile Tests

Uniaxial tensile tests were carried out on a 5582 INSTRON universal load frame equipped with an electrical resistance furnace that provides a temperature up to 610˚C. Figure 12 is a snapshot of the set up. A 5 KN capacity load cell was used for load measurements. These tests were carried out at constant strain rates, varying between 2 x 10^{-5} and 10^{-2} s^{-1}. Each band of the strain rates is covered at temperatures ranging between 325 and 450˚C at 25˚ C increments. The strains were measured through the direct displacement of the crosshead beam. 3.22mm thick -H24 magnesium alloy sheets were used for the test specimens, 19 x 6.35 mm gauge section specimens were machined at 0˚ with respect to the rolling direction of the sheet [Abu-Farha, Fadi., 2007].

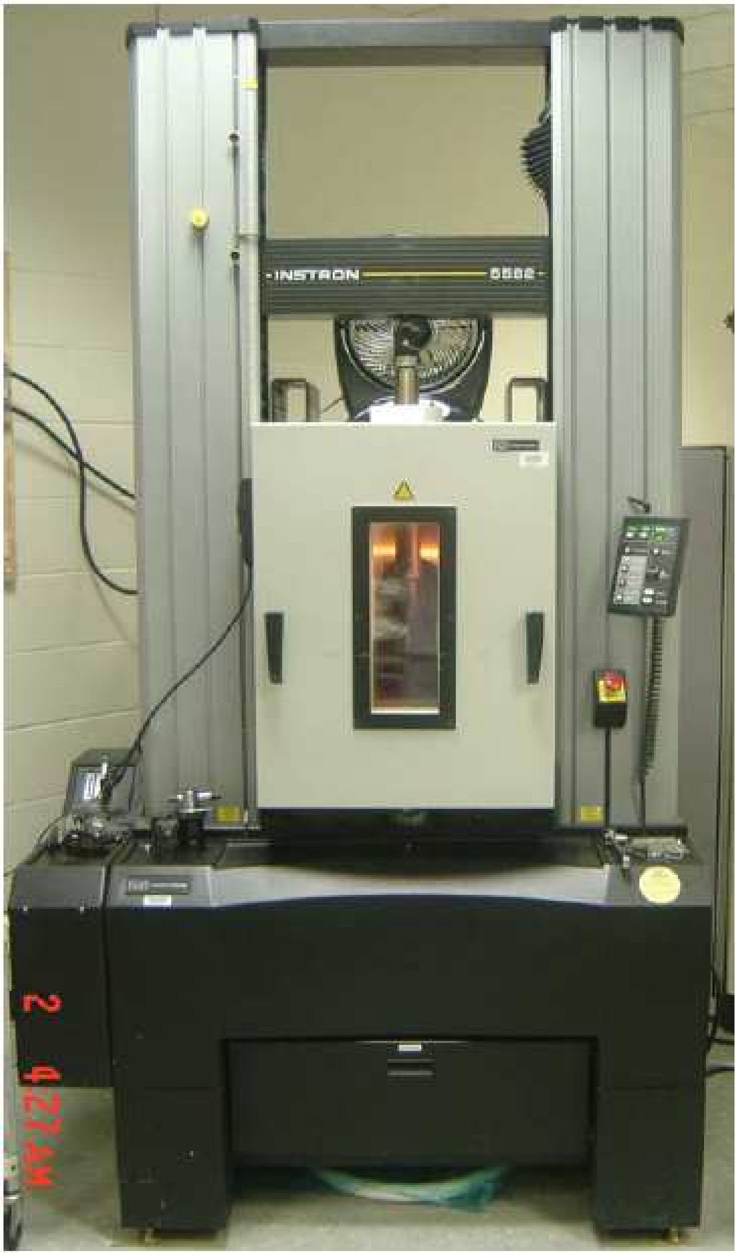


Figure 12 : Load Frame equipped with a chamber [Abu-Farha, F., 2007]

A pre-assigned holding time was allowed before stretching the test specimen at a constant strain rate value up to failure at each temperature. The same was repeated for different strain rates ranging from 2 x 10^{-5} to 10^{-2} s^{-1}. By combining all these results the effects of forming temperature and strain rate on the superplastic behavior could be assessed. Figure 13 represents the flow stress/strain rate curves at various forming temperatures [Abu-Farha, F., 2007].

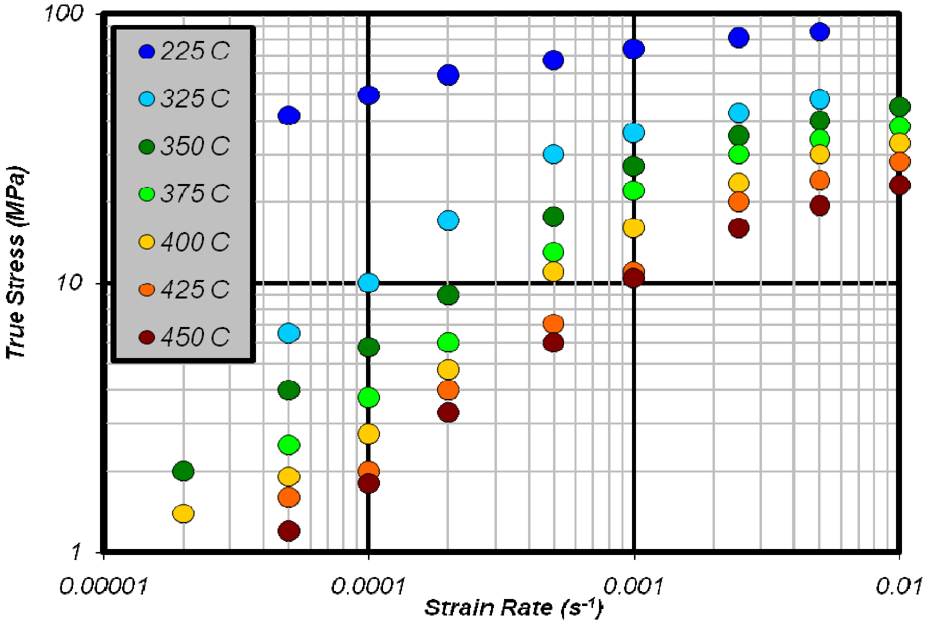


Figure 13 : Stress/ Strain rate curves at different temperatures [Abu-Farha, F., 2007]

The second variable influencing the superplastic behavior of the alloy is the maximum elongation prior to failure, the fracture strain. The fracture strain is plotted against the strain rate for different temperatures in Figure 14.

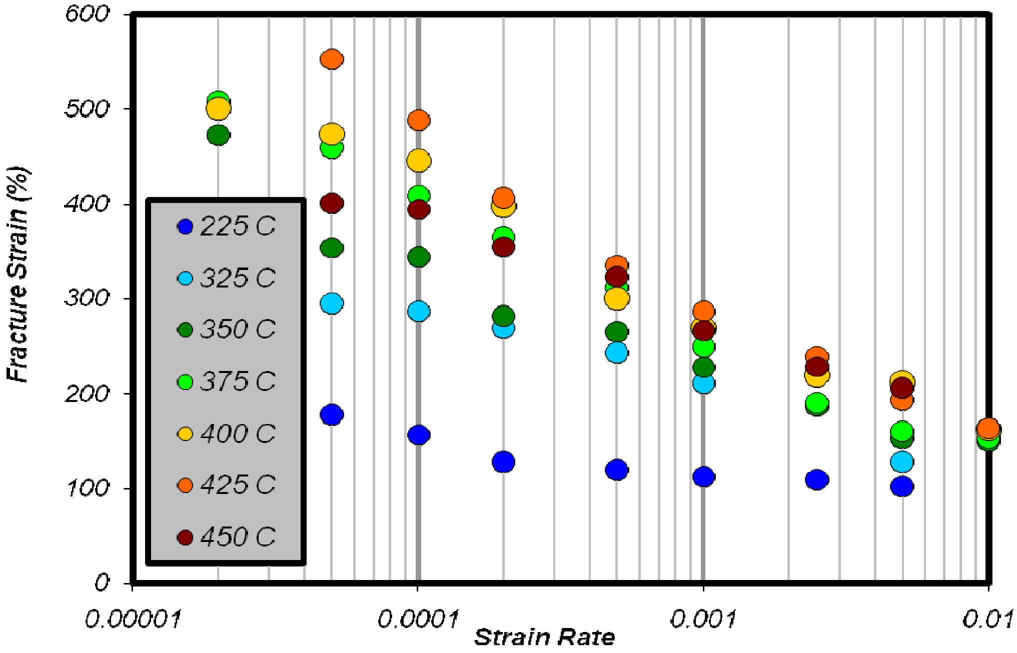


Figure 14 : Fracture strain Vs Strain rate at different temperatures [Abu-Farha, F., 2007]

3.2. Strain - rate Jump Tests

Strain rate sensitivity represents the capacity of the material to resist necking and influences the overall deformation and stability during superplastic deformation [Dehsmuk, P., 2004]. Therefore, in order to capture the deformation characteristics of the materials, the strain rate sensitivity index of the materials needs to be determined.

The strain rate jump tests are considered a good procedure to accurately evaluate *m* [Abu-Farha, F., 2007]. Strain rate jump tests were conducted over a band of strain rates ranging between 1x10^{-5} and 2.5x10^{-2} s^{-1} covering temperatures between 325˚C and 450˚C. To investigate the effect of plastic strain, and enhance the accuracy of the evaluation, jumps between every two successive strain rates were carried out at four plastic strain values, upward jumps were at 0.2 and 0.4 and downward jumps were at 0.3 and 0.5. The average of the four *m* values was set corresponding to the mean of the two strain rates between which the jump took place. The results from the entire jump tests at different temperatures were combined together and the average *m* value for each combination of temperature and strain rate were obtained from the plots. Figure 15 shows the graph plotted for the average strain rate sensitivity index *m* versus strain rate.

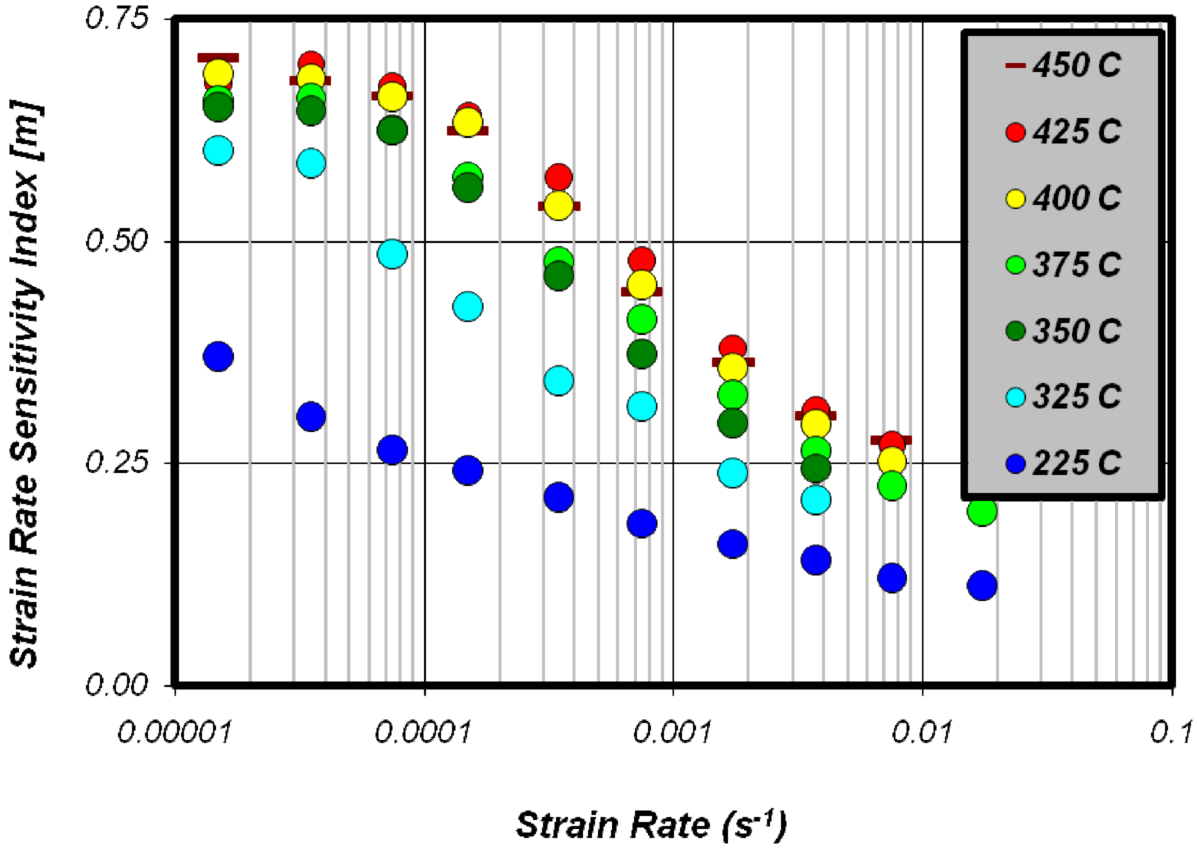


Figure 15 : Average strain rate sensitivity index m Vs Strain rate [Abu-Farha, F., 2007]

3.3. Artificial Neural Network Modeling

NeuroSolutions software is used for ANN modeling in this research. The software is designed with many capabilities. A user can use the software to model through 4 different approaches. NBuilder, here the user can build his own models, define the number of hidden layers , Processing elements etc. thus letting one implement their own ANN models. NSExcel, NeuroSolutions for Excel is an Excel Add-in that integrates NeuroSolutions to MS Excel providing a very powerful environment for manipulating your data, generating reports, and running batches of experiments. The Custom solution wizard (CSW), This is a program that will take any neural network created with NeuroSolutions and automatically generate and compile a Dynamic Link Library (DLL) for that network, which you can then embed into your own application. NExpert, this has models ready to use and once the data is fed to the model, the software decides on the number of hidden layers, processing elements, training paradigm etc. NExpert is used for ANN modeling in the present research. The software also provides unrestricted topologies, 6 different learning paradigms and also has a feature to integrate genetic optimization to these networks.

Modeling any process using ANN involves the following steps:

- Data Collection
- Data Normalization
- Training
- Cross-Validation
- Testing/Production

This section will present a detailed description about the modeling of ANN to assess the superplastic behavior of materials.

3.3.1. Data Collection

Once a problem is defined, the next step is to select the input variables and the desired responses to model the problem. The variables and conditions that appear relevant to the problem need to be analyzed. While collecting the data, one should take care to collect data which covers a wide spectrum of the case, which means the data collected for modeling needs to also cover conditions which might not be present in the training phase but might encounter later. Since ANNs learn from the data, the data must be valid for the results to be meaningful [Neuro Solutions help]. Collecting a large amount of data is not the only requirement but also to have data which is truly representative of the problem and data that will capture the fundamental principles of the problem is quintessential.

The input and the desired response data used for modeling needs to be stored in either ASCII, column-formatted ASCII or binary and bitmap (bmp image) format. Column-formatted ASCII is the most commonly used, since it is directly exportable from commercial spreadsheet programs [Neuro Solutions help].

The variables used as inputs for the ANN models in the present research are strain rate and temperature. The desired responses are the mechanical aspects of flow stress, fracture strain and strain rate sensitivity index *m* which describe the behavior of superplastic materials. As mentioned earlier in the last section the data for training the ANN models to predict the flow stress is obtained from uniaxial tensile tests and the strain rate sensitivity index *m* is obtained from the strain rate jump tests.

3.3.2. Data Normalization

If the data for individual inputs has significant numerical differences, then the data needs to be normalized. Using raw and non-normalized data might influence the rate of convergence of the networks and thus lead to inaccurate predictions. Multiple sets of data are divided by a common variable in order to negate that variable's effect on the data, thus allowing underlying characteristics of the data sets to be compared. Literature indicated that the ANNs performed better if the data used for the training was normalized.

Using NeuroSolutions software for the network design eliminates the need to decide the number of hidden layers and the PEs in each layer. The experimental data used in the present research was spread over a wide range and therefore data normalization had to be performed. All the process variables including the input process parameters and their corresponding output responses were normalized by dividing each parameter by the highest value of the parameter in the data set, thus reducing the range of all the parameters to the range (0, 1).

3.3.3. Network Topology

Deciding the number of hidden layers and the number of processing elements in each layer of the ANN depends on the complexity of the problem being considered. And this decision is based on a trial and error approach.

A Multi-layer feed forward ANN is used in the present research. Both the ANN models for strain rate sensitivity index *m* and the flow stress have a single hidden layer with four processing elements (PE) in the hidden layer for the first case and three for the latter case. Figure 16 indicates a schematic representation of a conventional feed forward ANN. The input layer in a network receives the input from the experimental data and each node of this layer represents and independent parameter which is directly related to the process parameters. The hidden layer processes all the input signals by applying weights to them and the output layer delivers to the output response of the network.

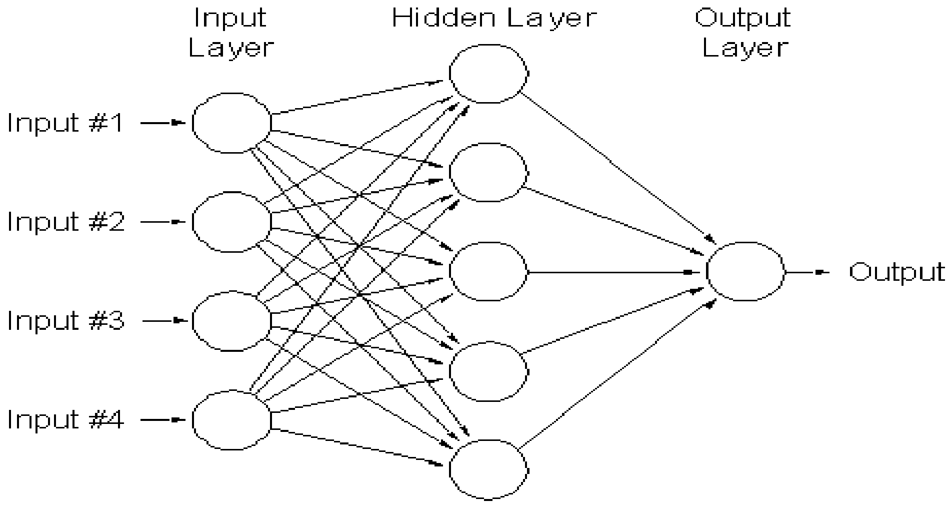


Figure 16 : A Neural Network Architecture

3.3.4. Training

The network learning in the present application is supervised, because the network learns based on the error (i.e. difference between input and network output) and corrects output to minimize error. For good ANN performance, the cost function given by the mean squared difference between the desired output and the actual output must be minimized. Weights are assigned to each of the linkages between the inputs and hidden layer PEs. These weights are optimized during the training phase to achieve the lowest MSE. The error at the end of each epoch is propagated back to the network to train the weights. The

NeuroSolutions offers a number of gradient search methods to train the ANN models. The following discussion briefly presents different gradient search methods used in this research.

3.3.4.1. Levenberg Marquardt

The Levenberg-Marquardt algorithm is a general, non-linear downhill minimization algorithm used when derivatives of the objective function are known. The objective function in this case is the error. This particular search method performed better on data sets when the relation existing between the input and output parameters is very complex. It also performs well in comparison with the other search methods when the number of data sets is less. The only disadvantage with this gradient search method is that it tries to over fit the data. Thus it can be misleading to the user because at times though the mean square error at the end of the training phase is less when compared to other search methods, results obtained in the production or testing phase may not be appropriate.

3.3.4.2. Momentum Gradient

Momentum learning is a robust method to speed up learning and is recommended as the default search rule for all the networks with non-linearity. With the momentum learning rule, weights are changed proportional to how much they were updated in the last iteration. The gradient of the error function is computed for each new combination of weights. However, instead of just following the negative gradient direction a weighted average of the current gradient and the previous correction direction is computed at each step. Theoretically, this approach provides the search process with a kind of inertia.

The weights update equation used by this search method is:

$$\Delta w_i(n+1) = (\gamma \nabla w_i) + (\rho \Delta w_i(n)) \tag{11}$$

$Where,$

$w = Weights$

$n = Iteration$

$\gamma = Step\ size$

$\rho = Momentum\ factor\ ranging\ from\ 0.5\ to\ 0.9$

3.3.4.3. Conjugate Gradient

NeuroSolutions uses the Scaled Conjugate Gradient implementation for this learning technique. Second order learning methods (such as Newton's method) use not only the slope of the performance surface but also the curvature to adjust the weights. As an example of the power of second order methods, it is known that linear systems always have a quadratic performance surface. Though computationally expensive second order methods can reach the bottom of a quadratic performance surface in one step. Conjugate gradient is an approximate second order method that is an excellent trade-off between computational complexity and increased learning speed [Stanford, Neural Networks]. In general, a conjugate gradient training epoch in NeuroSolutions will take twice as long as a standard gradient descent training epoch. The conjugate gradient method, however, will typically train in much fewer epochs and also move to a lower final MSE. Another significant advantage of scaled conjugate gradient learning is that it is parameterless, i.e.

there is no need to set learning rates or momentum terms. It automatically determines the “best” step size at each iteration.

3.3.4.4. Quick Propagation

Despite the name, quick propagation Fahlman, [1988], Patterson, [1996] is not necessarily faster than back propagation, although it may prove significantly faster for some applications. Quick propagation also sometimes seems more inclined to instability and to getting stuck in local minima, than back propagation; these tendencies may determine whether quick propagation is more appropriate for a particular problem. Quick propagation is a batch update algorithm: whereas back propagation adjusts the network weights after each example, quick propagation works out the average gradient of the error surface across all cases before updating the weights once at the end of the epoch. For this reason, there is no shuffle option available with quick propagation, since it would clearly serve no useful function. Quick propagation works by making the (typically ill-founded) assumption that the error surface is locally quadratic, with the axes of the hyper-ellipsoid error surface aligned with the weights. If this is true, then the minimum of the error surface can be found after only a couple of epochs. Of course, the assumption is not generally valid, but if it is even close to true, the algorithm can converge on the minimum very rapidly. Based on this assumption, quick propagation works as follows:

On the first epoch, the weights are adjusted using the same rule as back propagation, based upon the local gradient and the learning rate.

On subsequent epochs, the quadratic assumption is used to attempt to move directly to the minimum.

The basic quick propagation formula suffers from a number of numerical problems. First, if the error surface is not concave, the algorithm can actually go the wrong way. If the gradient changes little or not at all, the change can be extremely large, or even infinite! Finally, if a zero gradient is encountered, a weight will stop changing permanently.

3.3.4.5. Delta Bar Delta

According to Jacobs et al. Delta-bar-Delta is an alternative to back propagation, which is sometimes more efficient, although it can be more inclined to stick in local minima than back propagation. Similar to quick propagation, Delta-bar-Delta is a batch algorithm: the average error gradient across all the training cases is calculated on each epoch, and then the weights are updated once at the end of the epoch. However, it differs in that Delta-bar-Delta tends to be quite stable. Delta-bar-Delta is inspired by the observation that the error surface may have a different gradient along each weight direction, and that consequently each weight should have its own learning rate (i.e. step size).

In Delta-bar-Delta, the individual learning rates for each weight are altered on each epoch to satisfy two important heuristics:

If the derivative has the same sign for several iterations, the learning rate is increased (the error surface has a low curvature, and so is likely to continue sloping the same way for some distance);

If the sign of the derivative alternates for several iterations, the learning rate is rapidly decreased (otherwise the algorithm may oscillate across points of high curvature).

To satisfy these heuristics, Delta-bar-Delta has an initial learning rate used for all weights on the first epoch, an increment factor added to learning rates when the derivative does not change sign, and a decay rate multiplied by the learning rates when the derivative does change sign. Using linear growth and exponential decay of learning rates contributes to stability.

The algorithm described above could still be prone to poor behavior on noisy error surfaces, where the derivative changes sign rapidly even within an overall downward trend. Consequently, the increase or decrease of learning rate is actually based on a smoothed version of the derivative [Statsoft, ANN].

The stop criteria for learning are very important. The stop criterion based on the error of the cross validation set is very popular. Using a cross-validation set also helps in identifying over learning of the ANN model. Other methods limit the total number of iterations (hence the training time), stopping the training regardless of the networks performance. Another method stops training when the error reaches a given value. Since the error is a relative quantity, and the length of time needed for the simulation to get there is unknown and hence this may not be the best stop criterion. Another alternative is to stop on incremental error. This method stops the training at the point of diminishing returns, when iteration is only able to decrease the error by a negligible amount. However, the training can be prematurely stopped with this criterion because performance surfaces may have plateaus where the error changes very little from iteration to iteration [48]. The stop criterion adopted in the present research is the number of iterations, which was set to 1000.

3.3.5. Testing and Validation

Once the network is trained enough to model the data, the next step which follows is the production. Here the model is fed with a set of input parameters but to predict output. In this phase the neural network model uses the knowledge gained during the training phase (stored in the form of weights and biases) to predict the desired responses. Thus, in this research an unseen set of data for strain rate and the temperature is used to predict the strain rate sensitivity index and the flow stress, respectively in the two models.

The predicted responses from the neural network model are compared with the experimental results to validate the performance of the neural network model.

As mentioned earlier the objective of the present research is to inverse model the superplastic behavior of materials. Neural network models are used to simulate the behavior in the forward direction, i.e. given a particular strain rate and temperature the model is trained to predict the corresponding strain rate sensitivity index *m* and the flow stress. Now the next goal is to construct an approach to enable predicting the input process parameters corresponding to a given *m* value or flow stress. The formulation of a mathematical relationship between the output and input process parameters is explained in detail in the following section.

3.4. Mathematical Formulation using the Weights from the ANN Models

ANN models can be trained to high precision levels to predict or to substitute a process or a constitutive relationship. However these models cannot generate a mathematical relationship existing between the inputs and the corresponding outputs.

Mathematical formulation of a model can be considered as an approach to addressing the black box issue of the ANN.

The information or knowledge gained during the training phase of ANN modeling is stored in the form of weights. At the end of the training these weights are saved and the file is retrieved to obtain all the possible information on the weights and biases generated during the training process. This information is gathered and a mathematical relation which gives the output (either the strain rate sensitivity index *m* or the flow stress) as a function of the two process parameters, the temperature and the strain rate is formulated. The mathematical formulation is again cross validated using the results from the experiments. This relationship is then used as the objective function for the GA, details of which are presented in the next section.

3.5. Genetic Algorithms (GA)

A detailed description about GA and related concepts has been presented in the last chapter. This section is problem specific and gives details about the various strategies used in the present application. The GA code used for this research is from the Genetic Algorithms Laboratory (KanGAL), IIT Kanpur, India. It's a generic code written in C language. It is specifically written for single objective optimization problems. The code provides the flexibility to change the objective function and other GA parameters as well as the input parameters to make it problem specific. The selection strategy used, Chromosome representation, the objective function, the crossover strategy, the mutation strategy and the termination criterion have been explained in detail in this section. The following is a general structure of a GA [Mitsuo Gen, et al.]:

```
begin
$t \leftarrow 0;$
$initialize\ P(t);$
$evaluate\ p(t);$
while $(not\ termination\ condition)$ do
 begin
$recombine\ P9t)\ to\ yield\ C\ (t);$
$evaluate\ C(t);$
$select\ P(t+1) from\ P(t) and\ C(t);$
$t \leftarrow t+1;$
end
end
```

3.5.1. Chromosome Representation

The working of the GA starts with a set of initial solutions called the population. Every individual solution in the population is called a chromosome; the chromosome comprises of a string of parameters called the genes. These genes could be binary representations such 0 and 1 or could be real numbers.

In the present research a two gene chromosome is used. The first gene represents the strain rate and the second the temperature and they are represented using real numbers. The upper bounds and the lower bounds for both genes are determined beforehand. Depending on user specifications, the chromosomes are encoded to have values for genes within these bounds. Figure 17 is a schematic representation of the chromosome representation used for the present application.

To begin the GA, we define an initial population of N_{pop} chromosomes. A matrix represents the population with each row in the matrix being a $1 \times N_{\text{var}}$ array (chromosome) of continuous values. Given an initial population of N_{pop} chromosomes, the full matrix of $N_{pop} \times N_{\text{var}}$ random values is generated by using a random number generator. (N_{var}=2 in this case).

	Strain rate	Temperature
Chromosome 1	0.0035	325
Chromosome 2	0.0005	400

Figure 17 : Chromosome Representation

3.5.2. Fitness Function

The fitness function is the objective function for the problem. This objective function is used to evaluate the fitness or goodness of the chromosomes in solving the given problem. The very purpose of using a GA is to either minimize or maximize this objective function.

The objective function used in the present research is the error which is the difference between the Experimental values and output responses obtained from the GA. And the objective function is minimized in this case. The following is an illustration of the same.

$F = desired\ m$

$F_i = m\ from\ chromosome\ i$

$F_i = A(Strain\ rate) + B(Temperature) + C$

$E_i = |F - F_i|$

The Fitness function used is generated using the weights and the biases got at the end of the ANN training.

3.5.3. Selection Strategy

Choosing the right selection method is essential to ensure the best chromosomes are chosen. The extent to which the better individuals are generated depends mainly on the selection pressure. The convergence rate of a GA depends on the selection pressure, higher the selection pressure higher are the convergence rates.

Tournament selection is employed for the selection process in the present work. Tournament selection is one of the robust selection mechanisms used in genetic algorithms. The working of this mechanism is very simple. Tournament selection involves randomly choosing two candidates from the current population, comparing their fitness values and choosing the fit one.

3.5.4. Crossover Strategy

The most-fit chromosomes from the mating pool, make up for a mother and a father pair in some random fashion. Each pair produces two offspring that contain traits from each parent. In addition the parents survive to be part of the next generation. There are several approaches for pairing the chromosomes, such as top-down pairing, random pairing, tournament selection, rank weighting, and cost weighting. The one used in the current research is ‘Simulated Binary Crossover (SBX)’.

(SBX) is a real-parameter recombination operator. The operator involves a parameter which dictates the spread of offspring solutions in direct relation to the parent solutions. This simulates the working of the single point crossover which is generally in use in binary coded Gas. The intervals in the parents are preserved in the offspring. Thus the two offspring are symmetric about the parent solutions. This avoids a bias towards a particular parent solution.

3.5.5. Mutation Strategy

Mutation is an important GA operator which makes sure that the search does not get stuck in local optima. GA may converge too quickly to some region of the fitness surface. If this area is a region of global optima, it is good; but this does not happen often. To avoid this problem of overly fast convergence, we force the routine to explore other areas of the cost surface by randomly introducing changes, or mutations, in some of the variables. The mutation rate determines the percentage of genes to be mutated.

In the present research polynomial mutation strategy is used for mutation. A random number generates a random number between the upper and the lower bounds corresponding to each gene (process parameter).

The above mentioned procedure is repeated until the stated number of generations is reached. The final results which are the gene values corresponding to the best solution are validated using the results from the uniaxial tensile tests for the first application and the strain rate jump tests for the second application.

3.6. Summary

A hybrid approach which can be used in inverse design is built. This approach involved the integration of ANN and GA. The chapter discussed in detail the construction of the approach.

Chapter 4: Application of the Proposed Approach to Model the Behavior of Superplastic Materials

Two cases are used to demonstrate the application of the ANN-GA model for the inverse modeling. The first case involved the prediction of the strain rate and the temperature to achieve a given strain rate sensitivity index *m.* The second case involves the prediction of the same two parameters as in the first case to achieve a given flow stress.

4.1. Problem 1: Input Parameter Prediction for Strain - rate Sensitivity Index

4.1.1. Strain - rate Jump Tests

As mentioned in the previous chapter data from the uniaxial tensile tests were used to develop the ANN model. The strain rates range from $1x10^{-5}$ and $2.5x10^{-2}$ $s^{-1}$and the temperatures vary from 225˚C to 450 ˚C in increments of 25 ˚C as indicated in figure 16 and table 1. These strain rates, temperatures and the corresponding m values are used for further experimentation.

4.1.2. Artificial Neural Network Modeling

4.1.2.1. Data Collection

The input parameters for the neural network model are the strain rates and the temperatures obtained from the strain rate jump tests and the desired output is the strain rate sensitivity *m*. 65 Exemplars of data are used for training. The data used to train the ANN model is given in table 1. This data is closely representative of the problem as it is spread over a wide spectrum of the problem. The input data to the ANN needs to be in Column-formatted ASCII.

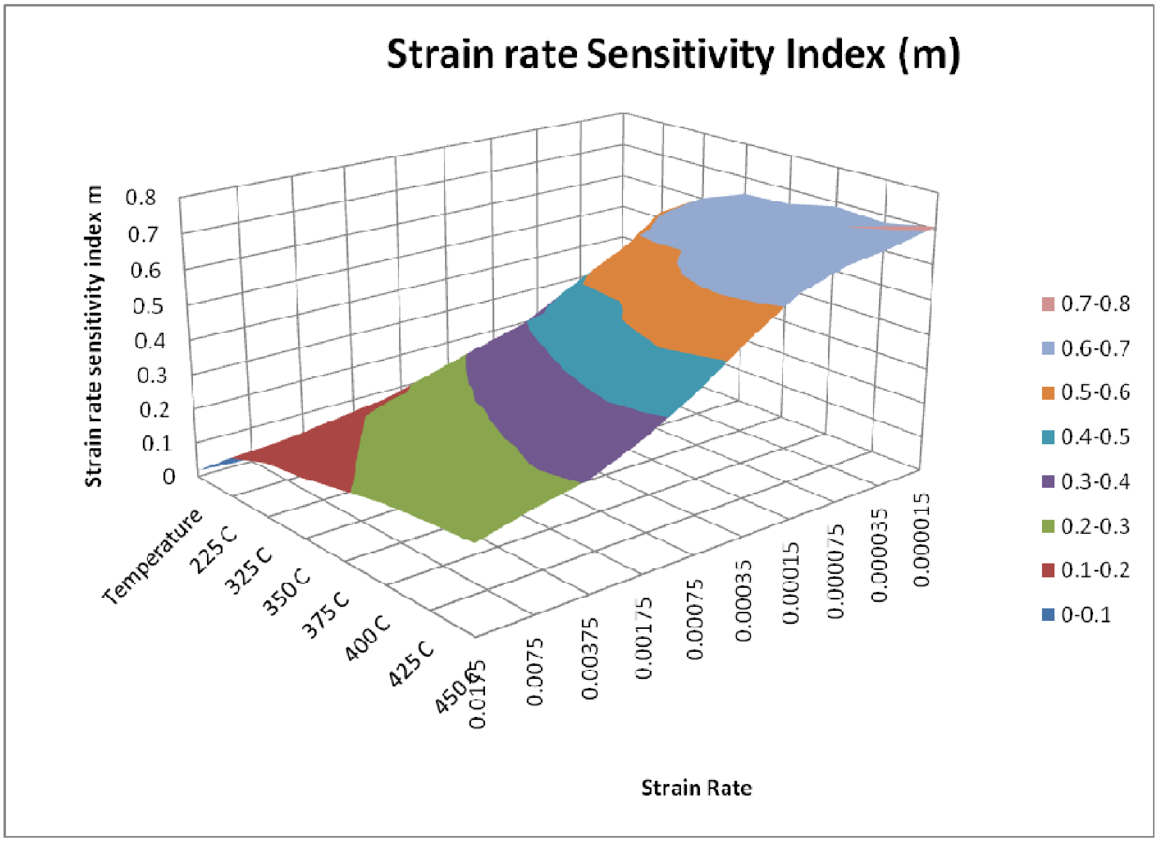


Figure 18: 3D surface plot of the training data for ANN model

Table 1 : Training data for ANN

Strain Rate	225 C	325 C	350 C	375 C	400 C	425 C	450 C
0.0175	0.112	0.155	0.170		0.219	0.232	0.240
0.0075	0.12	0.160		0.224	0.252	0.270	0.277
0.00375		0.208	0.245	0.265	0.293	0.309	
0.00175	0.16	0.239	0.294	0.326	0.356	0.380	0.364
0.00075	0.183	0.314	0.372	0.412	0.450	0.478	0.442
0.00035	0.211	0.342	0.461	0.476		0.571	0.539
0.00015	0.241	0.425	0.560	0.571	0.632	0.641	0.624
0.000075	0.265	0.485	0.625	0.624	0.662	0.673	0.664
0.000035	0.302	0.587	0.646	0.660	0.681		0.680
0.000015	0.369	0.601	0.651	0.657	0.687	0.678	0.706

4.1.2.2. Network Topology

A Multi layer feed forward network is used for this application. It is a 3 layer network with one hidden layer. The number of processing elements in the first layer is two corresponding to the two input parameters, the strain rate and the temperature. The hidden layer has four processing elements. The output layer has one processing element, which is the strain rate sensitivity index *m*. Figure 19 is a schematic representation of the ANN architecture used for the prediction of *m*.

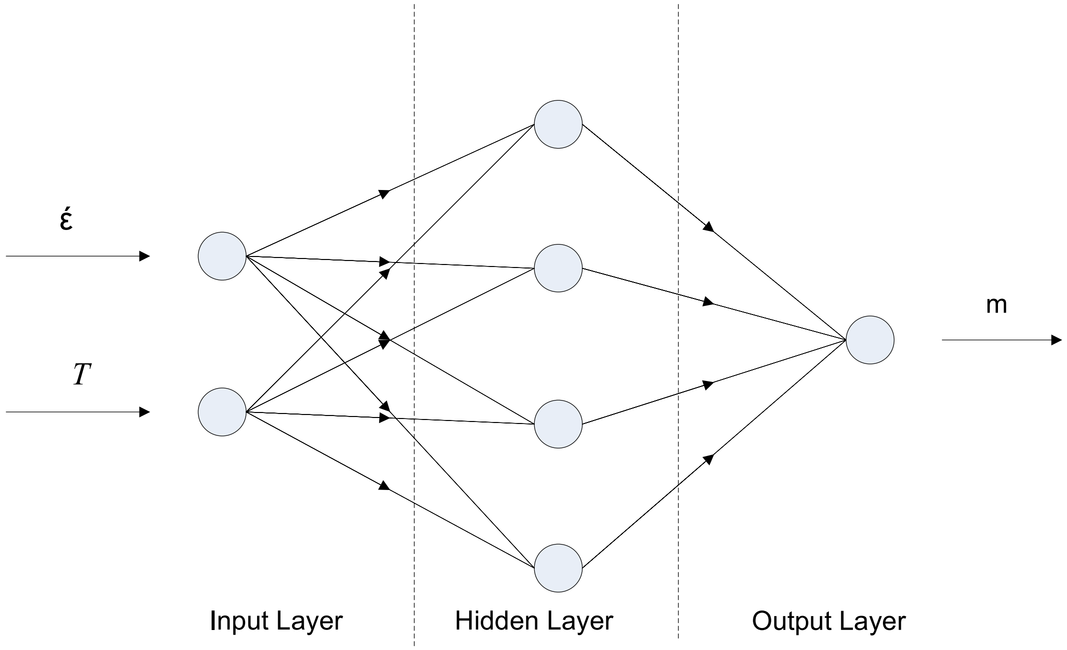


Figure 19 : ANN architecture for Application 1

4.1.2.3. Training

The ANN model used to predict the *m* value is trained using six different gradient search methods. The mean square error obtained at the end of the training is used for comparison and the best gradient search method is taken further for experimentation. Using each gradient search method the ANN model is trained for 1000 epochs.

Cross-validation of the data could not be performed due to the limit on the number of exemplars available. Table 2 includes the mean square error at the end of training. As mentioned earlier the ANN is trained using 6 different gradient search methods, the performance of the ANN using the all these search methods is illustrated in the following 6 plots.

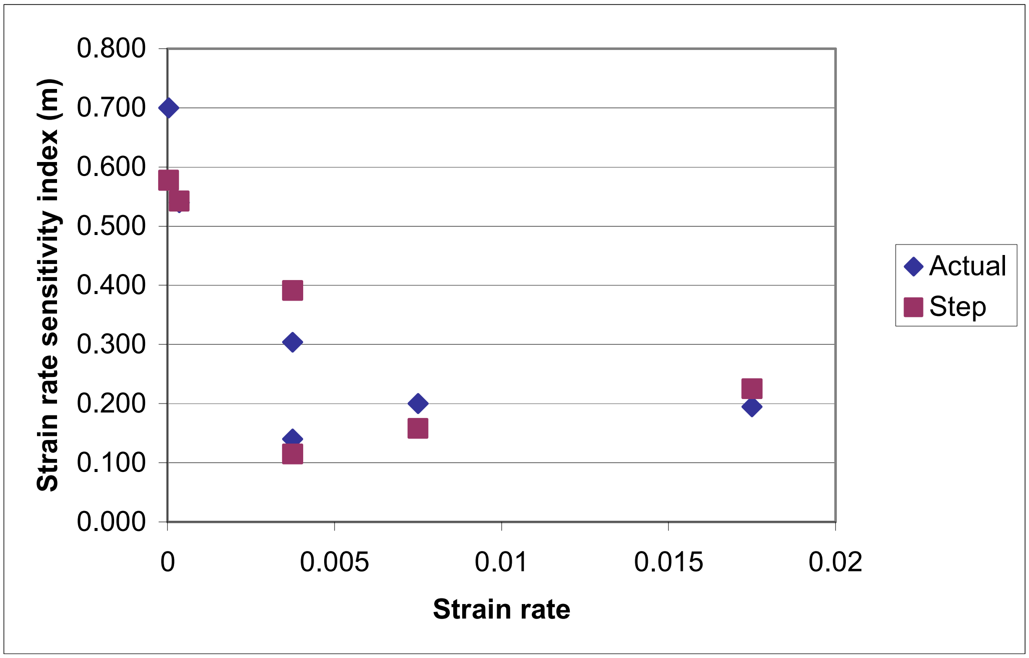


Figure 20 : Comparison of the actual *m* values to the ANN (GS: Step) predicted values

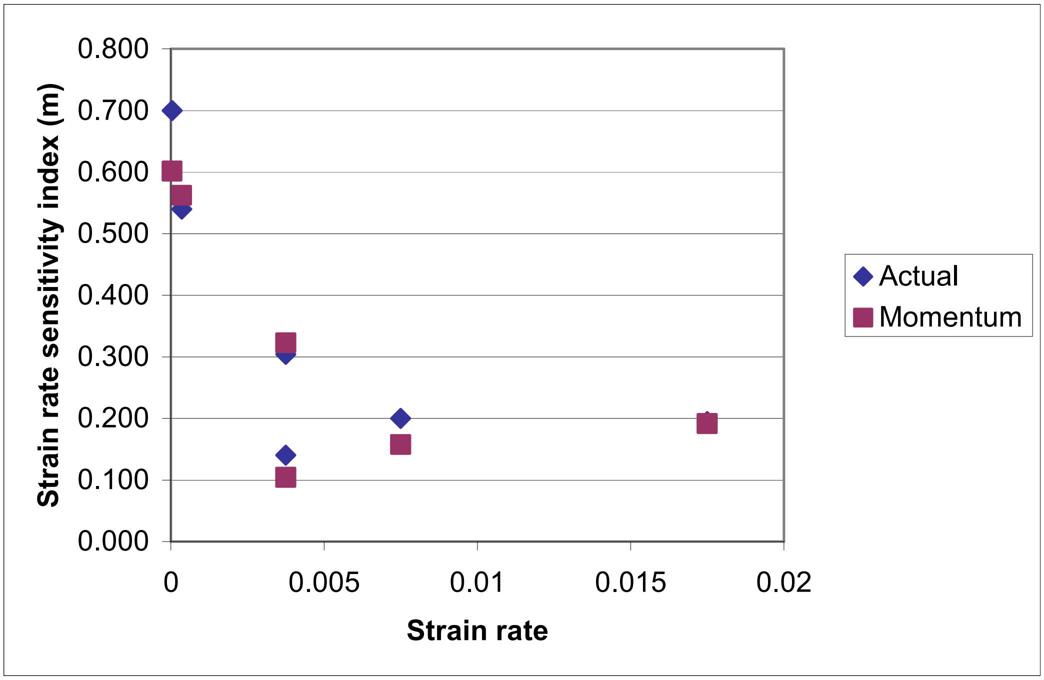


Figure 21 : Comparison of the actual *m* values to the ANN (GS: Momentum) predicted values

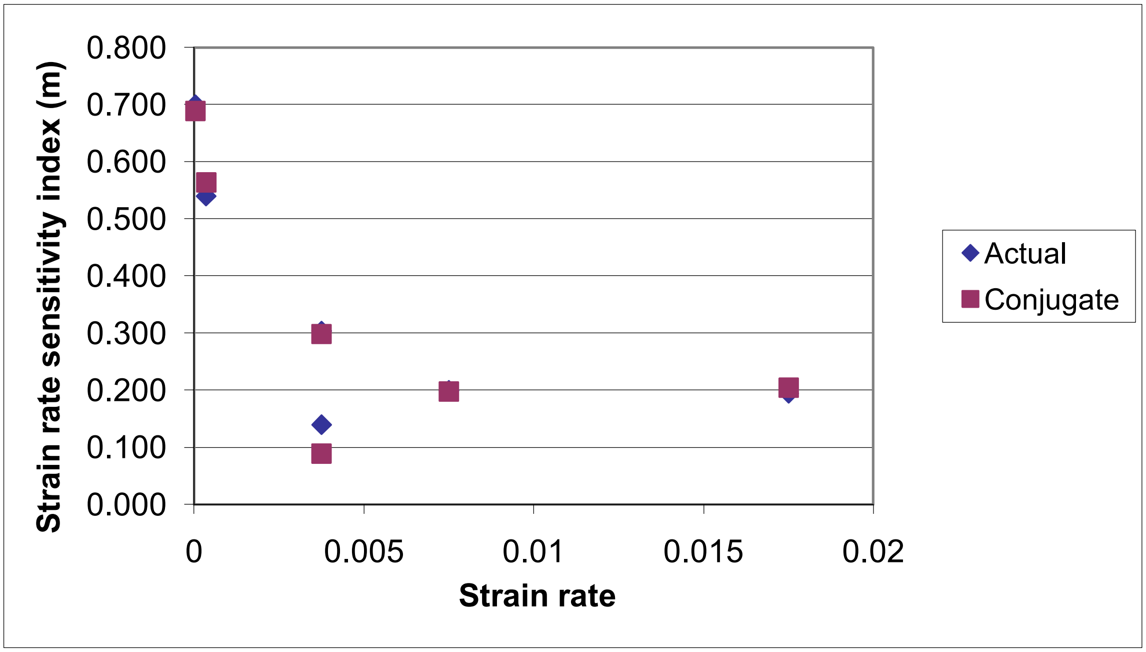


Figure 22 : Comparison of the actual *m* values to the ANN (GS: Conjugate) predicted values

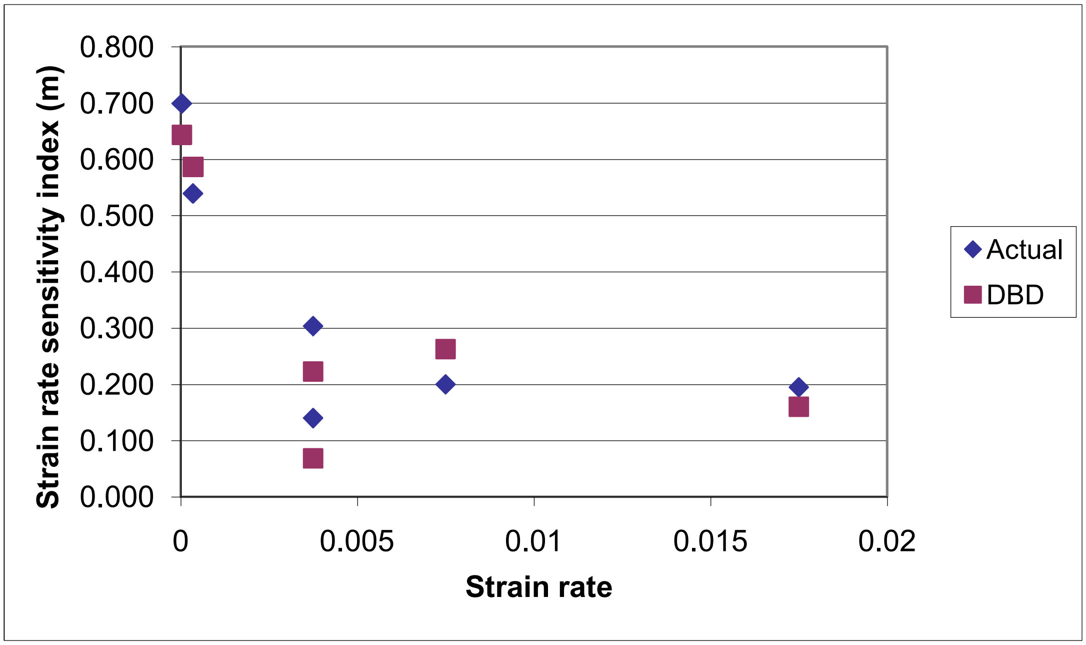


Figure 23 : Comparison of the actual *m* values to the ANN (GS: Delta-Bar-Delta) predicted values

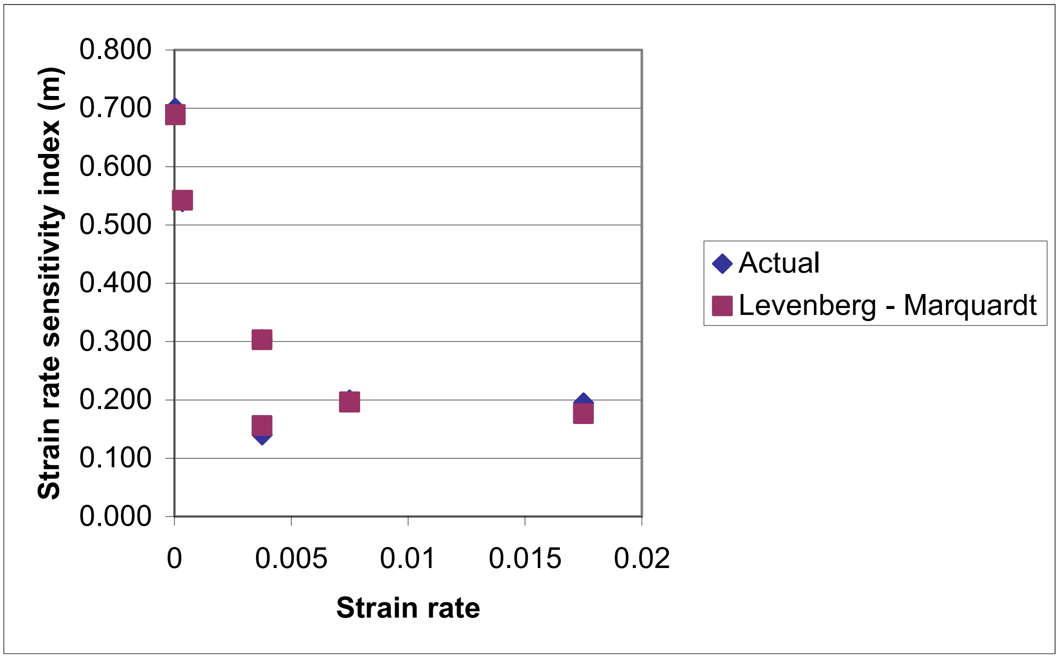


Figure 24 : Comparison of the actual *m* values to the ANN (GS: Levenberg-Marquardt) predicted values

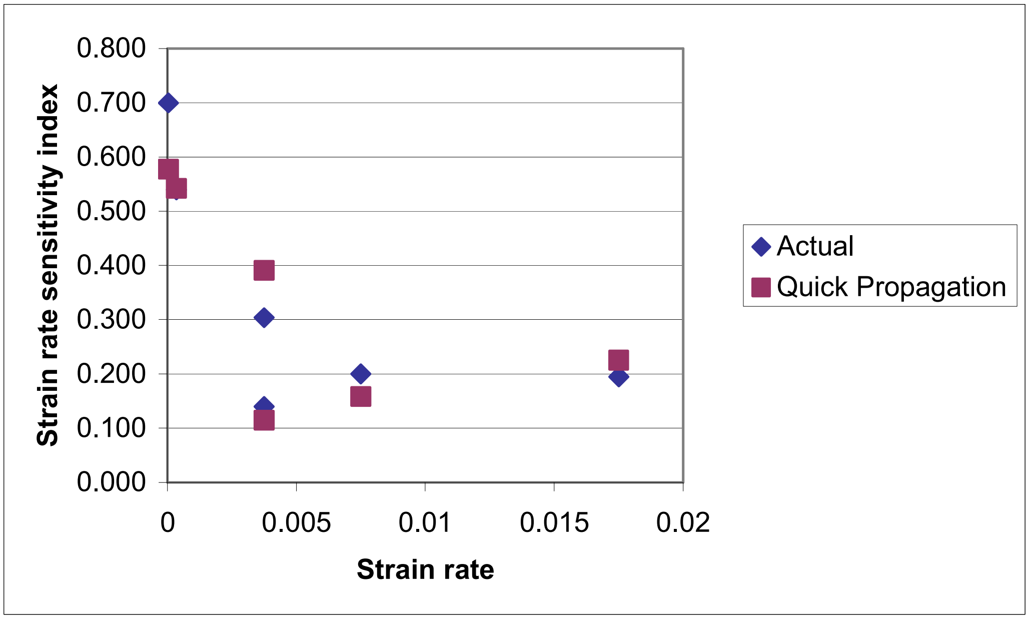


Figure 25 : Comparison of the actual *m* values to the ANN (GS: Quick Propagation) predicted values

Table 2 : Mean Square Errors

Gradient Search Method	Step	Momentum	Conjugate Gradient	Delta-Bar-Delta	Quick Propagation	Levenberg-Marquardt
MSE	0.0188	0.0141	0.0038	0.0066	0.0187	0.0014

It can be observed that Levenberg-Marquardt gradient search (GS) method outperformed all the other gradient search methods. The weights and biases obtained using this GS method is used to develop a mathematical expression.

4.1.3. Mathematical Formulation

The weights and biases are used to formulate a mathematical relationship between the input parameters, the strain rate and the temperature and the output strain rate sensitivity index. The breadboard obtained using the Levenberg-Marquardt training paradigm are included in the Appendix. The following relations are used to obtain the mathematical expression.

$$Ndata_1 = (\dot{\varepsilon} * A_1) + O_1 \tag{12}$$

$$Ndata_2 = (T * A_2) + O_2 \tag{13}$$

$Where, \ \dot{\varepsilon} = Strain\ rate$

$T = Temperature$

$A_1, A_2 = Amplitudes,$

$O_1, O_2 = Offsets$

$$Tan\,h_i = (Ndata_1 * w_{1i}) + (Ndata_2 * w_{2i}) + B_i \tag{14}$$

$where\ i = 1, 2, 3, 4$

$w_{1i}, w_{2i} = weights\ corresponding\ to\ each\ input\ parameter$

$B_i = Bias$

$$N_{output} = \sum_{i=1}^{4} (W_i * tanh_i) + B_{output} \tag{15}$$

$Where,$

$W_i = Weights\ at\ the\ output\ layer,$

$B_{output} = Bias$

$$Result = (N_{output} - Offset_{output})/Amplitude_{output} \tag{16}$$

$$Relative\ Error = (Actual\ Output - Result)/(Actual\ Output) \tag{17}$$

The above mentioned procedure is demonstrated using the data from the breadboard saved at the end of the training for Levenberg-Marqaurdt GS method.

Step 1: Normalizing Data

Input Number	Input	Amplitude	Offset	Ndata
1.	0.00035	103	-0.902	-0.866
2.	400	0.008	-2.7	0.5

Refer to equations (12) and (13) for Ndata

Step 2: Calculations for hidden layer

Weights	N1	N2	N3	N4
1.	-1.07	20.9	1.96	1.66
2.	-2.15	-0.583	2.51	2.34

The number of processing elements in the hidden layer = 4

Ndata*W	-0.149	-18.4	-0.441	-0.267
Bias	-0.147	18.6	0.405	0.322
Tanh	-0.297	0.181	-0.0369	0.055

Refer to Equation (14) for tanh calculation

Step 3: Calculation of the output

Weights	-8.8	-0.607	16.2	-24.8
Weights*Tanh	2.61	-0.11	-0.598	-1.36

Bias	0.00532
Noutput	0.544

Step 4: Normalizing the output

For Output

Amplitude	Offset
3.03	-1.24
Result	0.589

Actual Output	0.5422
Relative error	0.0816

From equations (15) and (16)

The mathematical relationship was formulated using the above mentioned procedure. The final relationship gives the strain rate sensitivity index *m* as a function of the temperature and strain rate, as shown in equation (18). The coefficients of the equation are determined through the simplification of the weights and biases.

$$m = -431\dot{\varepsilon} + 0.005T - 1.3 \tag{18}$$

where,

$m = strain\,rate\;sensitivity\;index$

$\dot{\varepsilon} = strain\,rate$

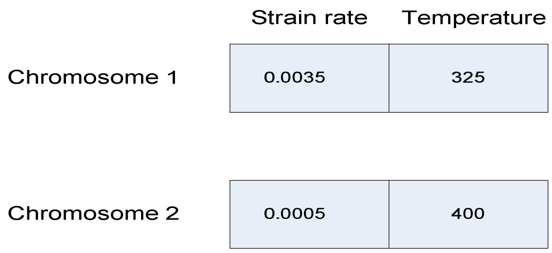
$T = Temperature$

4.1.4. Optimization Using Genetic Algorithms

The genetic algorithm code from the Kanpur Genetic Algorithm Laboratory (KanGAL), IIT Kanpur, India is used in this research The code is written in C language. The code is customized to meet the present requirements by changing a few subroutines corresponding to the objective function, the chromosome representation, and by defining the GA parameters apt for our application.

4.1.4.1. Chromosome Representation

The genetic representation used for this application is real coded. So the genetic transformations are carried out on the real parameter representations which are the design variables. The design variables in this case are the strain rate and the temperature. There the chromosome is a 1x2 matrix, and the genes corresponding to the $\dot{\varepsilon}$ and T. Figure 26 is a schematic representation of the chromosome used for this application.



	Strain rate	Temperature
Chromosome 1	0.0035	325
Chromosome 2	0.0005	400

Figure 26 : Chromosome Representation

4.1.4.2. Fitness Function

The fitness function or the objective function for the GA is the relative error which is given as the difference between the GA generated strain rate sensitivity index m_{GA} and the experimental m value. The mathematical representation of the objective is as follows

$$Error = {(m - m_{GA})}/{m} \tag{19}$$

where,

$m_{GA} = GA\ generated\ strain\ rate\ sensitivity\ index$

$m = Experimental\ strain\ rate\ sensitivity\ index$

The objective of the problem is to generate the strain rates and temperatures which would minimize the error. The fitness function value is the basis for ranking the chromosomes to be selected for further processing.

4.1.4.3. Genetic Algorithm Parameters Used for Simulations

Four different strategies are used for experimentation. The parameters which are varied in the four categories are the number of trials and the crossover and mutation probabilities. The purpose of varying the number of trials is to check for the repeatability of the GA. The results for the 10 trials or strategy 1 have been included in the discussion in this thesis. Results corresponding to the other strategies are included in the Appendix.

Table 3 : GA parameters for experimentation

	Strategy 1	Strategy 2	Strategy 3	Strategy 4
Number of generations	100	100	100	100
Population Size	10	20	10	10
Lower & upper limits for $\dot{\varepsilon}$	(0.0175, 0.000015)	(0.0175, 0.000015)	(0.0175, 0.000015)	(0.0175, 0.000015)
Lower & upper limits for T	(225, 450)	(225, 450)	(225, 450)	(225, 450)
Niching Parameter*	0.1	0.1	0.1	0.1
Number of Trials	10	20	50	10
Crossover Probability (Cp)	0.45	0.45	0.45	0.5
Mutation Probability (Mp)	0.15	0.15	0.15	0.2
Seeding Factor**	0.0001	0.0001	0.0001	Varying

*Niching allows restricted tournament selection. A Niching factor of 0.1 is recommended.

**Seeding factor helps by speeding up the GA in finding a solution.

With the above tabulated values as the operating parameters for the GA, The ANN predicted *m* value is fed to the genetic algorithm. The GA gives the strain rate and temperature values corresponding to a minimized error. This is carried out for six different m values using the four different strategies mentioned in table 3. The results obtained using strategy 1 is considered for further analysis. The results from the GA corresponding to each m value using strategy 1 are represented in the following 6 plots.

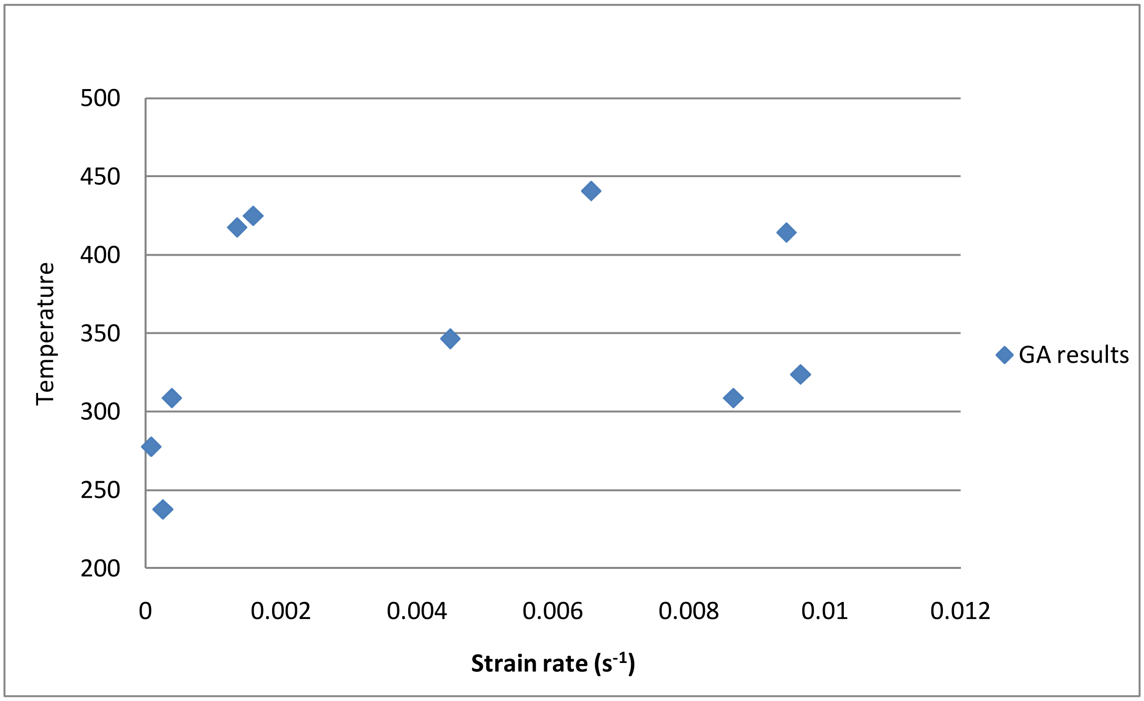


Figure 27: Strategy 1 GA results for *m* = 0.14

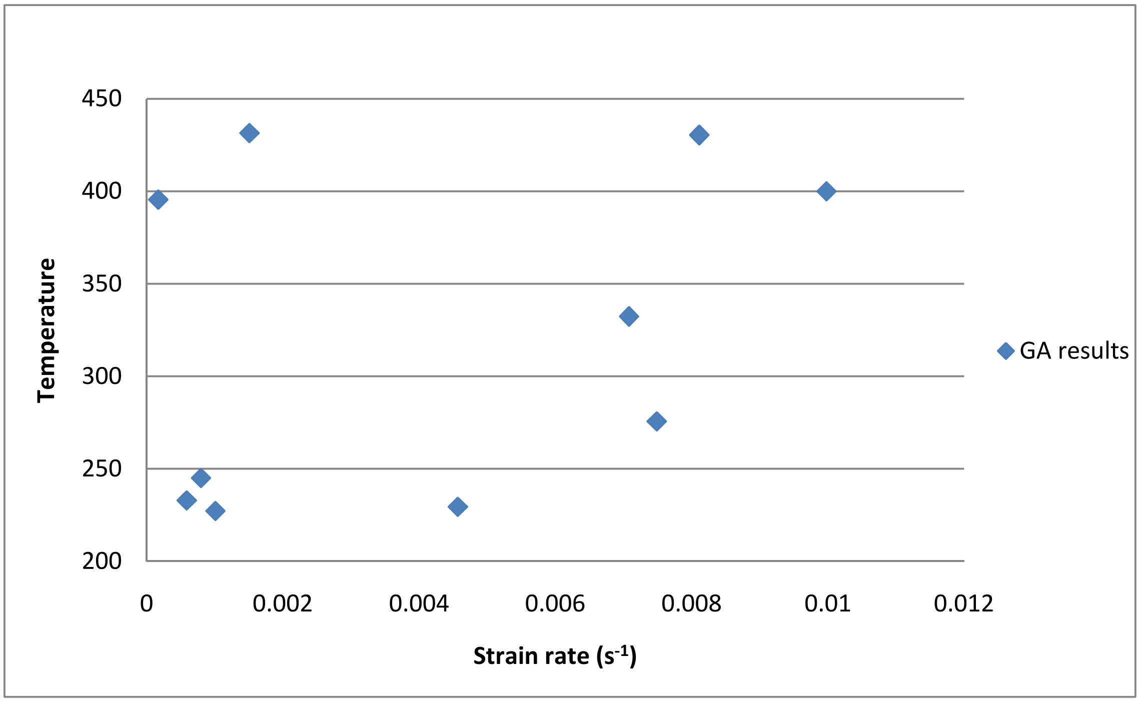


Figure 28 : Strategy 1 GA results for *m* = 0.2

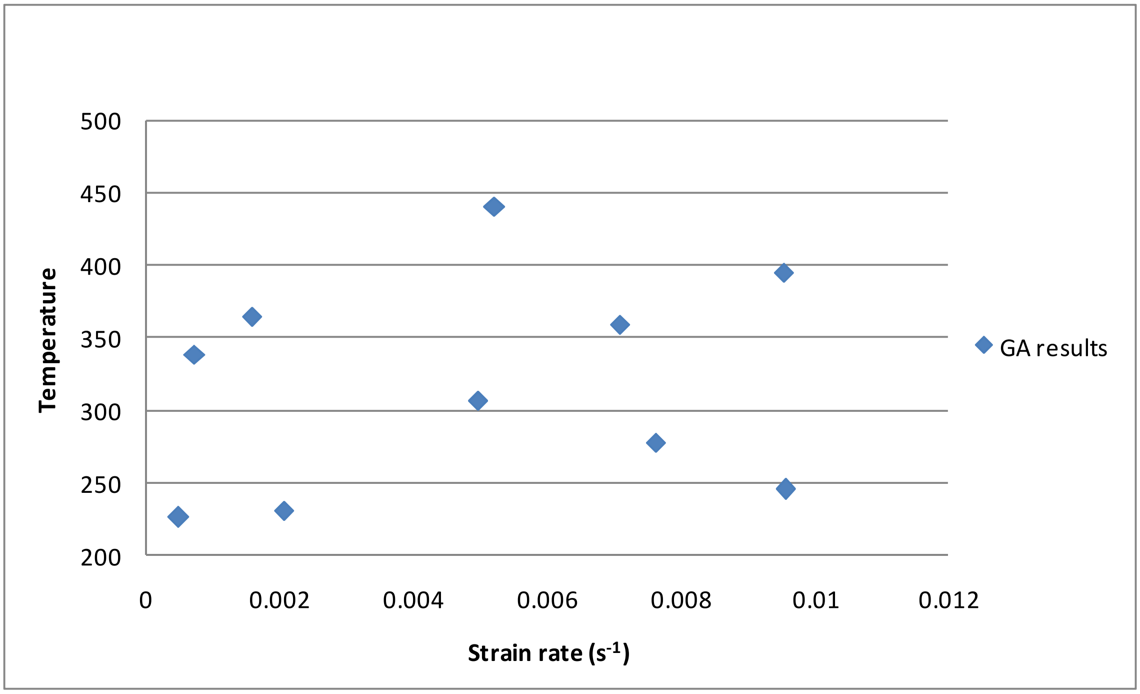


Figure 29 : Strategy 1 GA results for *m* = 0.195

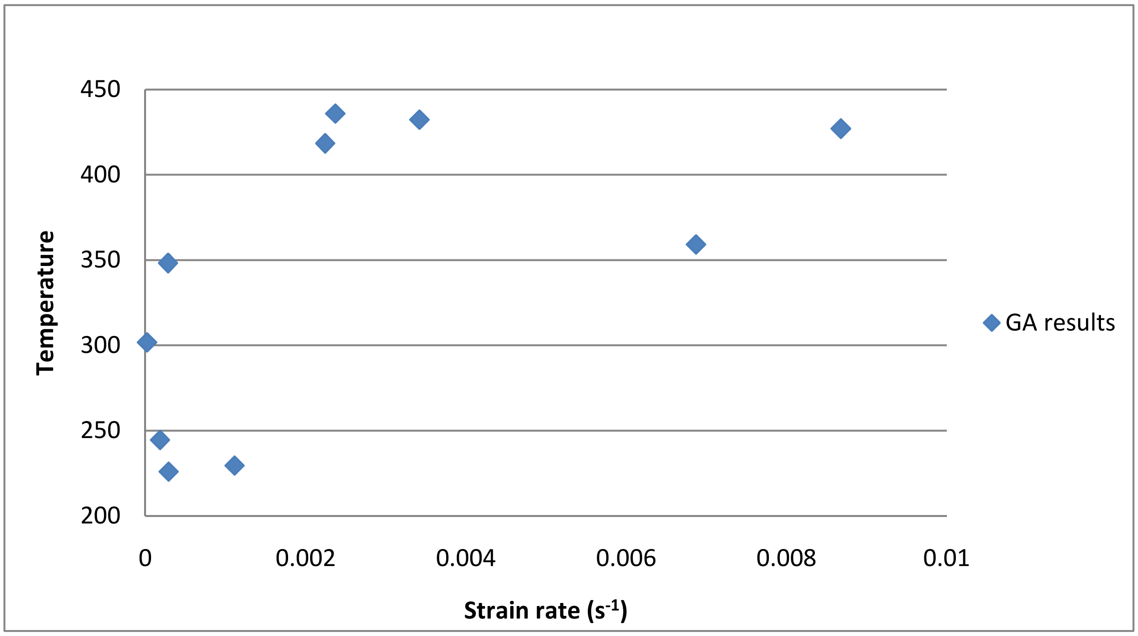


Figure 30 : Strategy 1 GA results for *m* = 0.304

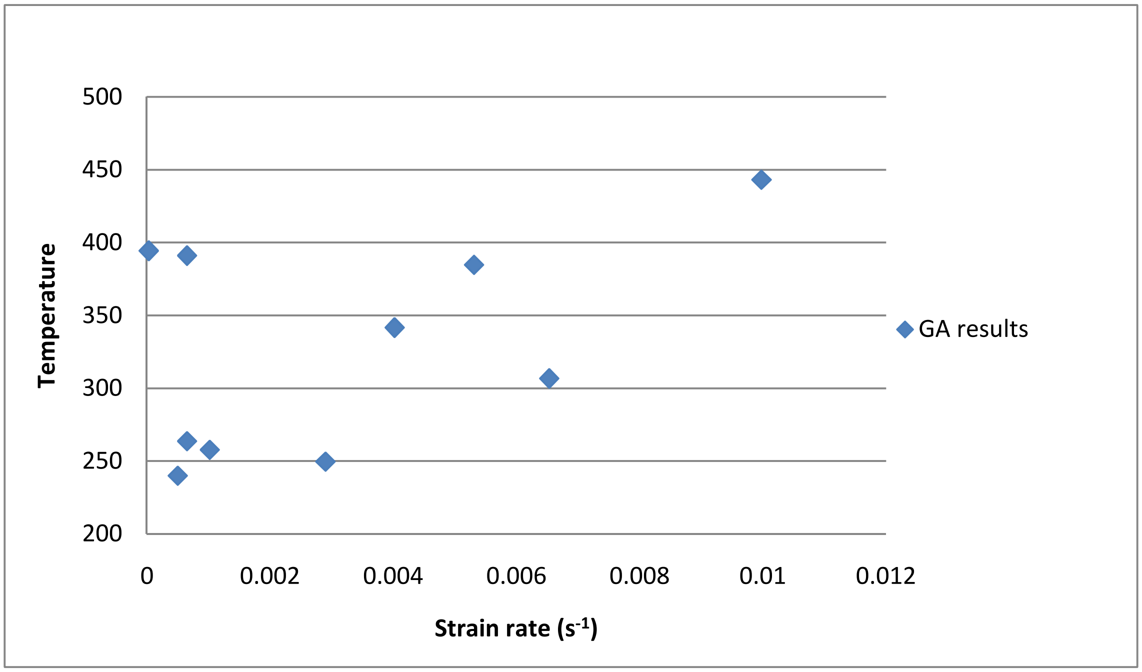


Figure 31 : Strategy 1 GA results for *m* = 0.54

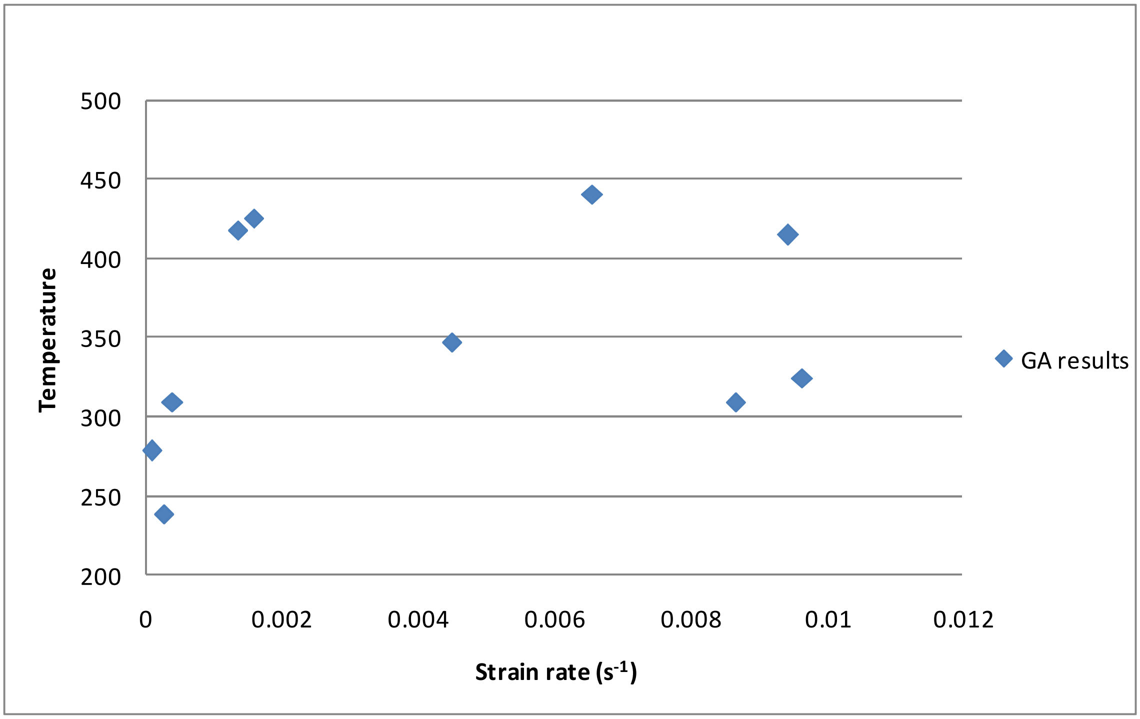


Figure 32: Strategy 1 GA results for *m* = 0.7

4.2. Problem 2: Input Parameters Prediction for Flow Stress

4.2.1. Uniaxial Tensile Tests

As mentioned in chapter 3 data from the strain rate jump tests is used to model the ANN model. The strain rates range from 1x10^{-5} and 2.5x10^{-2} $s^{-1}$and the temperatures from 225˚C to 450 ˚C in increments of 25 ˚C. These strain rates, temperatures and the corresponding Flow stress values are used for further experimentation.

4.2.2. Artificial Neural Network Modeling

4.2.2.1. Data collection

50 Exemplars of data are used for training. The input parameters here are the strain rates and temperatures and the output response is the flow stress. This data collected from the uniaxial tensile tests is spread over a broad range and thus is closely representative of the problem. Similar to the first application, in this case too the data to be fed to the ANN is in column-formatted ASCII. Figure 33 is the graphical representation of the data used in training the ANN.

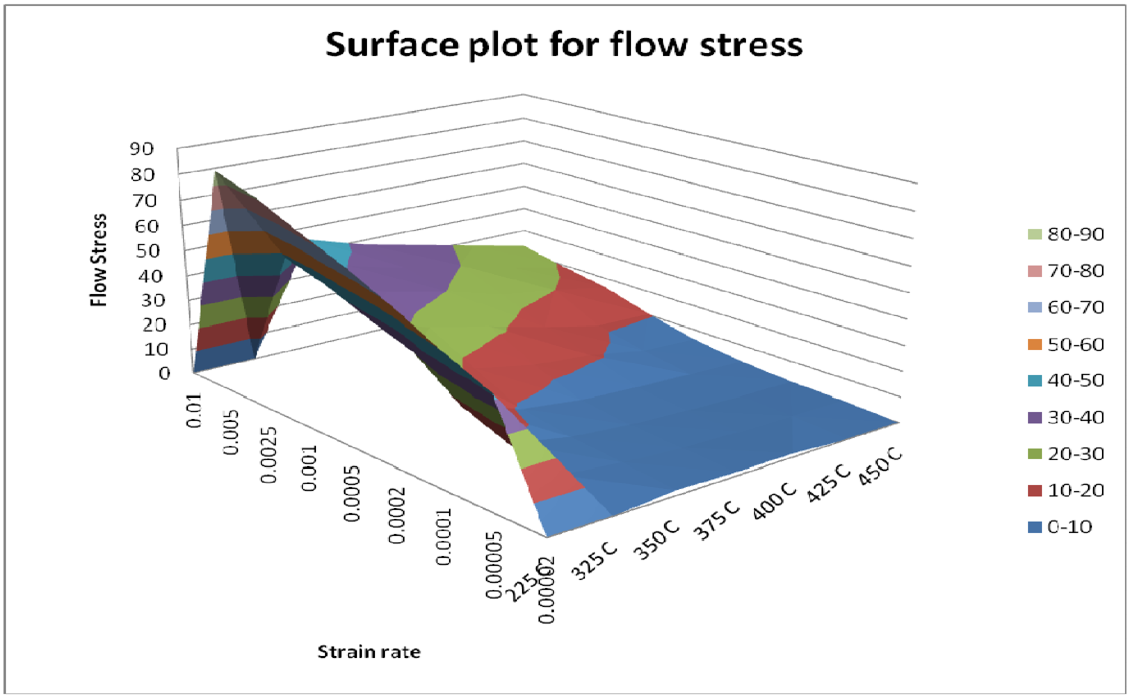


Figure 33 : Uniaxial Tensile tests data used for training the ANN

Table 4: Data used for training ANN

Strain Rate	225 C	325 C	350 C	375 C	400 C	425 C	450 C
0.01			45	38	33	28	23
0.005	86	48	40	34	30	24	19.3
0.0025	81	43	35	30	23.33	20	16
0.001	73.5	36	27	22	16	11	10.4
0.0005	67	30	17.5	13	11	7	6
0.0002	59	17	9	6	4.75	4	3.3
0.0001	50	10	5.75	3.75	2.75	2	1.8
0.00005	42	6.5	4	2.5	1.9	1.6	1.2
0.00002			2	1.4	1.4		

4.2.2.2. Network Topology

The ANN model used in this application is a 3 layered network with one hidden layer. The number of processing elements in the hidden layer is 3. Each processing element in the input layer corresponds to the input parameters, the strain rate and the temperature. The one processing element in the output layer corresponds to flow stress. Figure 34 is a schematic representation of the ANN model used for this application.

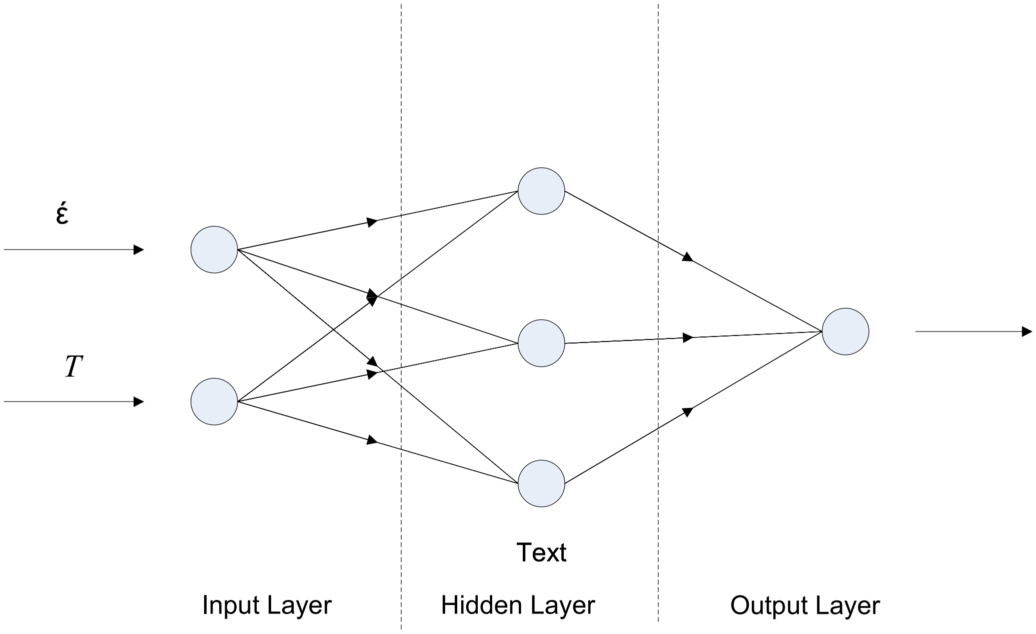


Figure 34 : ANN architecture for application 2

4.2.2.3. Training

The training procedure is similar to the one used in the strain rate sensitivity index prediction. The ANN was trained using the six gradient search methods and for 1000 epochs. The training set included the strain rate and the temperature as the input process parameters and the flow stress was also fed to the model as the desired response.

The weights and the biases at the end of the training were saved for all the gradient search methods. The error which is the deciding factor for the gradient search methods is tabulated in the table 5. The predictions of the ANN model using the six different gradient search methods are graphically represented in the following six plots.

Table 5 : Mean Square Errors

Gradient Search Method	Step	Momentum	Conjugate Gradient	Delta-Bar-Delta	Quick Propagation	Levenberg-Marquardt
MSE	0.00052	0.000219	0.0001	0.000169	0.00054	0.00034

It can be observed that Conjugate gradient search (GS) method outperformed all the other gradient search methods. The weights and biases obtained using this GS method is used to develop a mathematical expression.

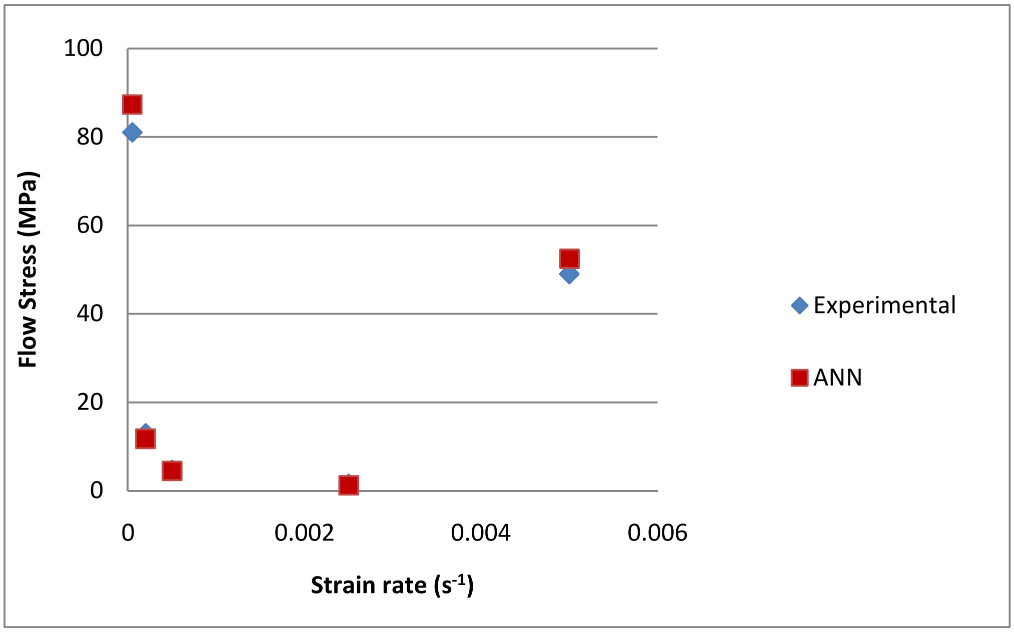


Figure 35: Comparison of Experimental σ values to ANN (GS: Conjugate gradient) predicted values

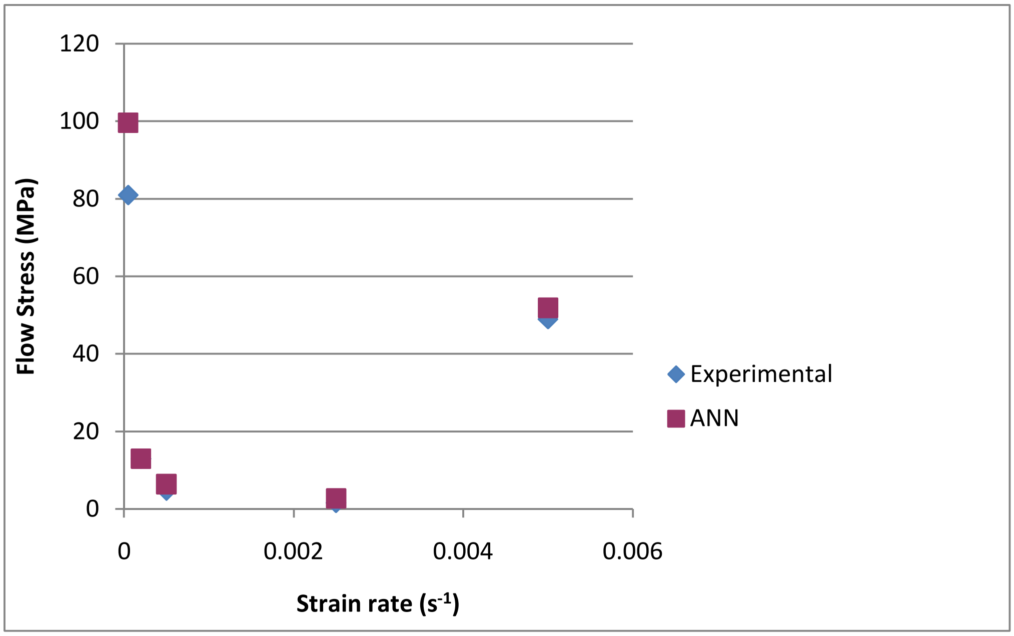


Figure 36: Comparison of Experimental σ values to ANN (GS: Momentum) predicted values

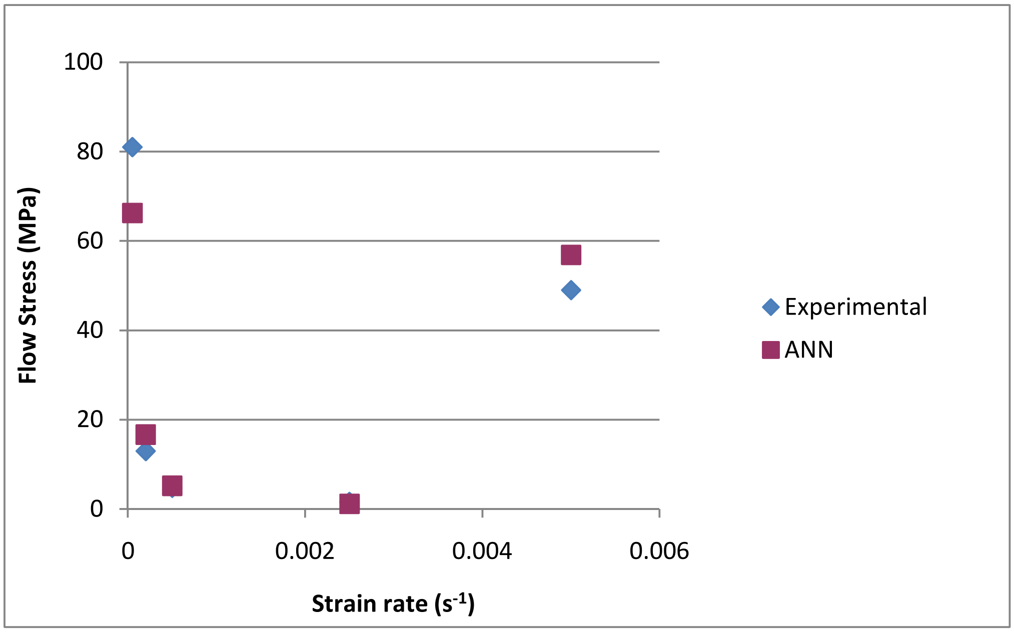


Figure 37 : Comparison of Experimental σ values to ANN (GS: Levenberg- Marquardt) predicted values

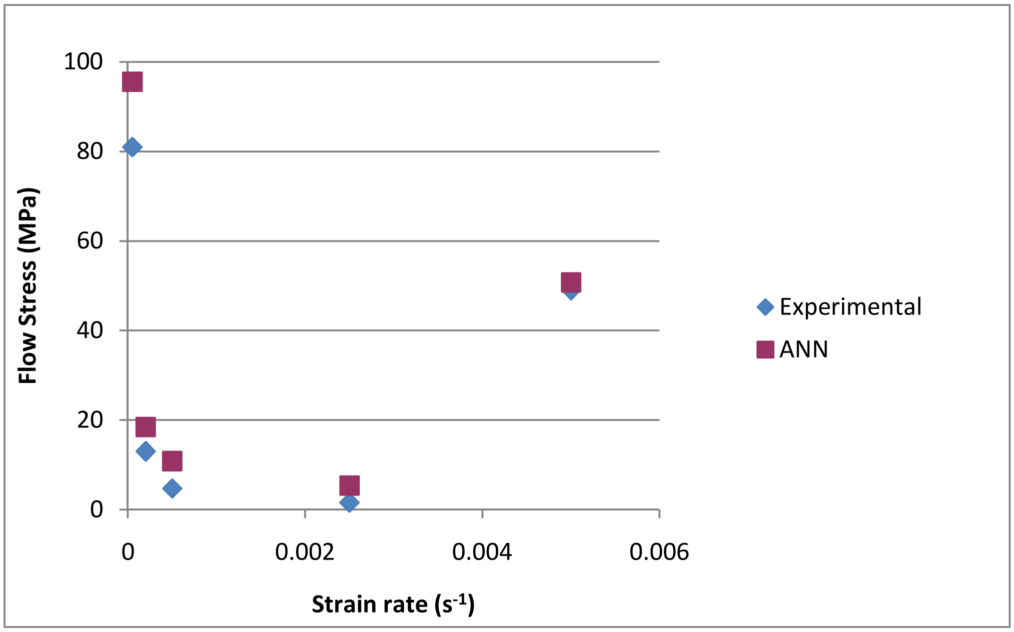


Figure 38 : Comparison of Experimental σ values to ANN (GS: Quick Propagation) predicted values

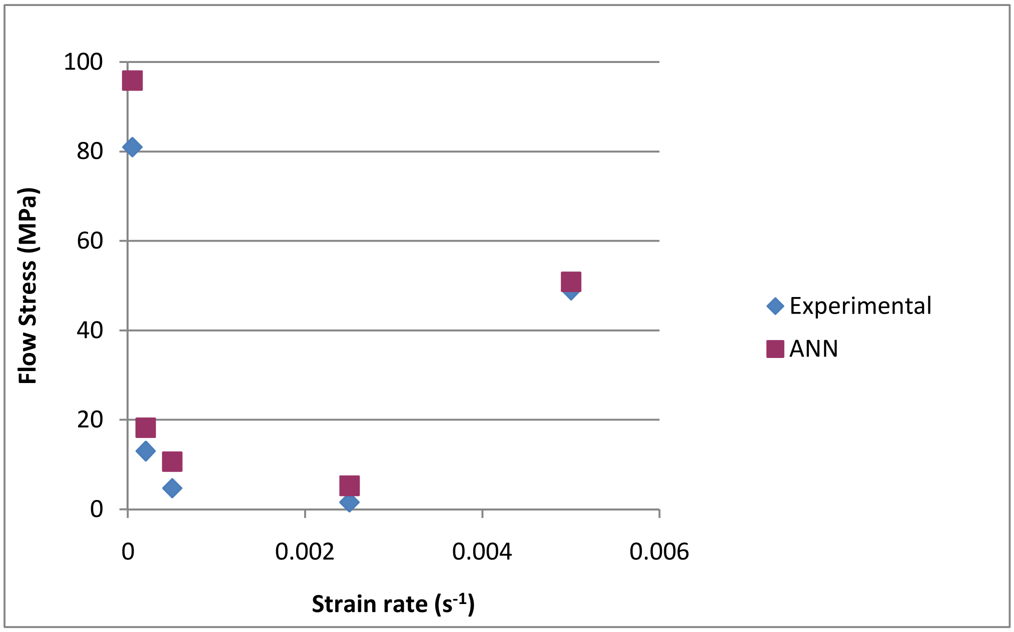


Figure 39 : Comparison of Experimental σ values to ANN (GS: Step) predicted values

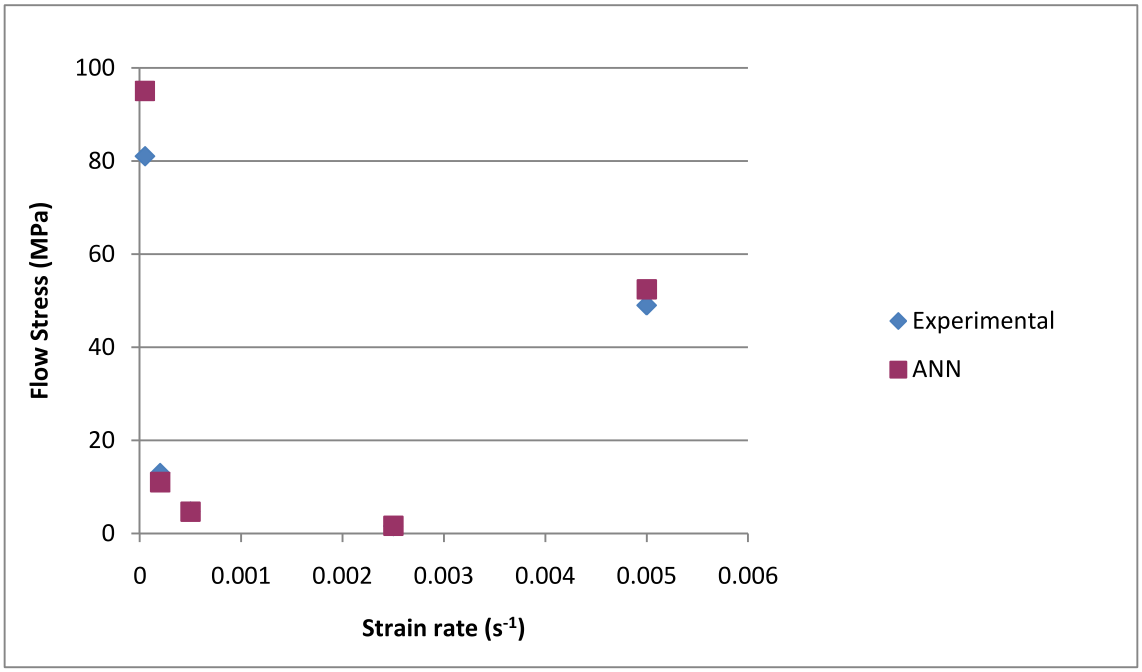


Figure 40 : Comparison of Experimental σ values to ANN (GS: DBD) predicted value

4.2.3. Mathematical Formulation

Similar to the previous application the best performing gradient search method is chosen for further experimentation. The weights and biases obtained at the end of the training are saved. This saved information on the weights and biases is used to formulate a mathematical relationship.

Refer to equations (1), (2), (3), (4), (5) and (6) for the mathematical formulation. In this case the value of i = 1, 2, 3. This is because the number of processing elements in the hidden layer is 3.

The procedure is demonstrated using the data from the breadboard saved at the end of the training for Conjugate GS method.

In this application the mathematical relationship obtained represents the output, which is the flow stress given as a function of the weights and biases and the inputs, temperature and the strain rate.

Step 1: Normalizing Data

Input Number	Input	Amplitude	Offset	Ndata
1.	0.0002	364	-0.918	-0.845
2.	400	0.008	-2.7	0.5

Refer to equations (12) and (13) for Ndata

Step 2: Calculations for hidden layer

Weights	N1	N2	N3
1.	-0.584	-1.38	0.623
2.	0.843	0.109	0.174

The number of processing elements in the hidden layer = 3

Ndata*W	0.915	1.22	-0.439
Bias	0.299	0.209	0.527
Tanh	1.21	1.43	0.0878

Refer to Equation (14) for tanh calculation

Step 3: Calculation of the output

Weights	-1.87	1.31	1.54
Weights*Tanh	-2.27	1.88	0.135

Bias	-0.569
Noutput	-0.825

Step 4: Normalizing the output

For Output

Amplitude	Offset
1.83	-0.925

Result	4.74
Actual Output	4.45
Relative error	0.062

From equations (15) and (16)

The mathematical relationship formulated using the above mentioned procedure is as follows. The final relationship gives the Flow stress as a function of the temperature and strain rate, as shown in equation (20). The coefficients of the equation are determined through the simplification of the weights and biases.

$$\sigma = 4150\dot{\varepsilon} - 0.44T + 178.66 \tag{20}$$

where,

$\sigma = Flow\,Stress$

$\dot{\varepsilon} = strain\,rate$

$T = Temperature$

4.2.4. Optimization Using Genetic Algorithms

The code used for the previous application is used here too. The subroutines corresponding to the objective function and the GA parameters are changed to best fit this problem.

4.2.4.1. Chromosome Representation

The chromosome representation for both the applications is the same. The genes in the string represent the strain rate and the temperature respectively. Refer to figure 24 for the schematic representation of the chromosome.

4.2.4.2. Fitness Function

The fitness function or the objective function here is the relative error computed as a difference of the Experimental flow stress σ and GA predicted flow stress $\sigma_{GA.}$ The mathematical representation of the objective is as follows

$$Error = {(\sigma_{GA} - \sigma)}/{\sigma_{GA}} \tag{21}$$

where,

$\sigma_{GA} = GA\ generated\ Flow\ stress$

$\sigma = ANN\ predicted\ flow\ stress$

4.2.4.3. Genetic Algorithm parameters used in experimentation

Similar to Case 1 four different strategies have been used for the purpose of experimentation. The results obtained using strategy 1 is discussed in this section. The results from the other three strategies are included in the Appendix.

Table 6 : GA parameters used for simulations

	Strategy 1	Strategy 2	Strategy 3	Strategy 4
Number of generations	100	100	100	100
Population Size	10	20	10	10
Lower & upper limits for $\dot{\varepsilon}$	(0.01, 0.00002)	(0.01, 0.00002)	(0.01, 0.00002)	(0.01, 0.00002)
Lower & upper limits for T	(225, 450)	(225, 450)	(225, 450)	(225, 450)
Niching Parameter*	0.1	0.1	0.1	0.1
Number of Trials	10	20	50	10
Crossover Probability (Cp)	0.35	0.35	0.35	0.5
Mutation Probability (Mp)	0.15	0.15	0.15	0.2
Seeding Factor**	0.0001	0.0001	0.0001	Varying

*Niching allows restricted tournament selection. A Niching factor of 0.1 is recommended.

**Seeding factor helps by speeding up the GA in finding a solution.

Using these GA parameters for the simulations, the Experimental flow stress value is fed to the GA. The GA gives the strain rate and the temperature values which minimize the error. This has been tried for five different flow stress values and has been carried out with four different strategies as mentioned previously and in table 6. The results obtained using strategy 1 are presented in the following 5 plots.

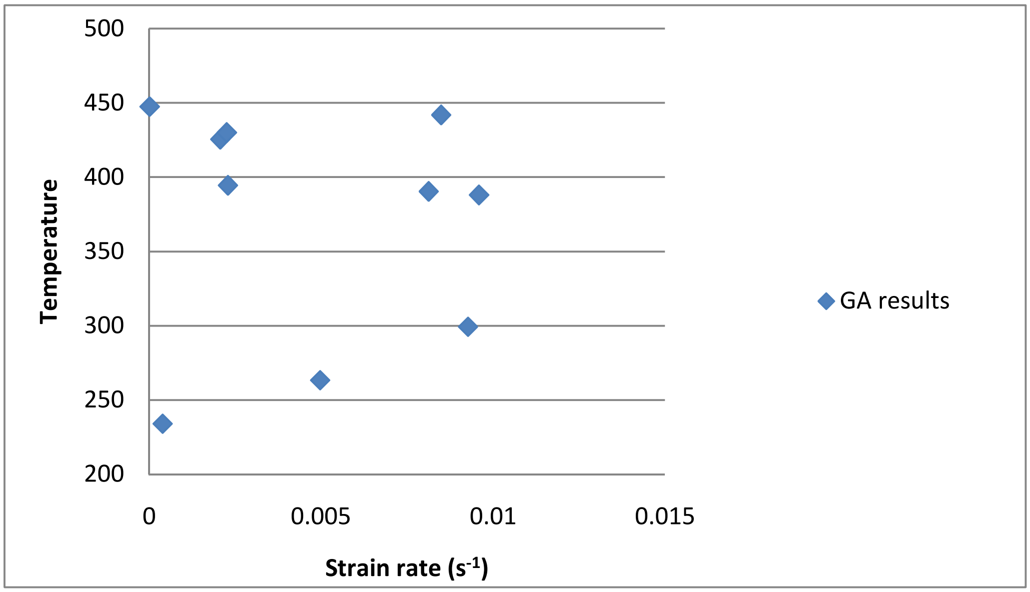


Figure 41 : GA results for σ = 1.6 MPA

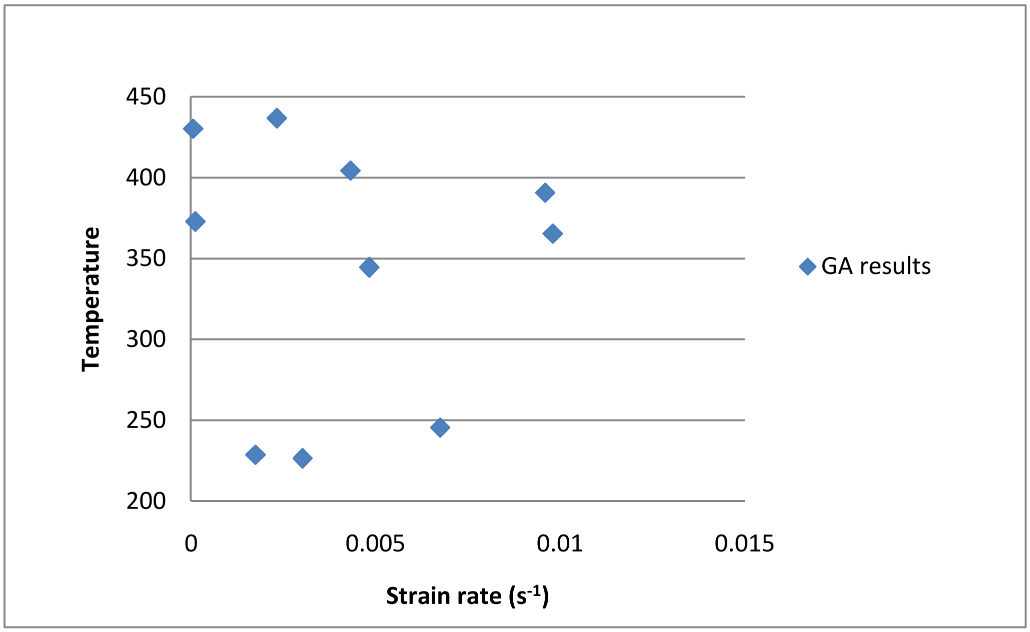


Figure 42 : GA results for σ = 4.75 Mpa

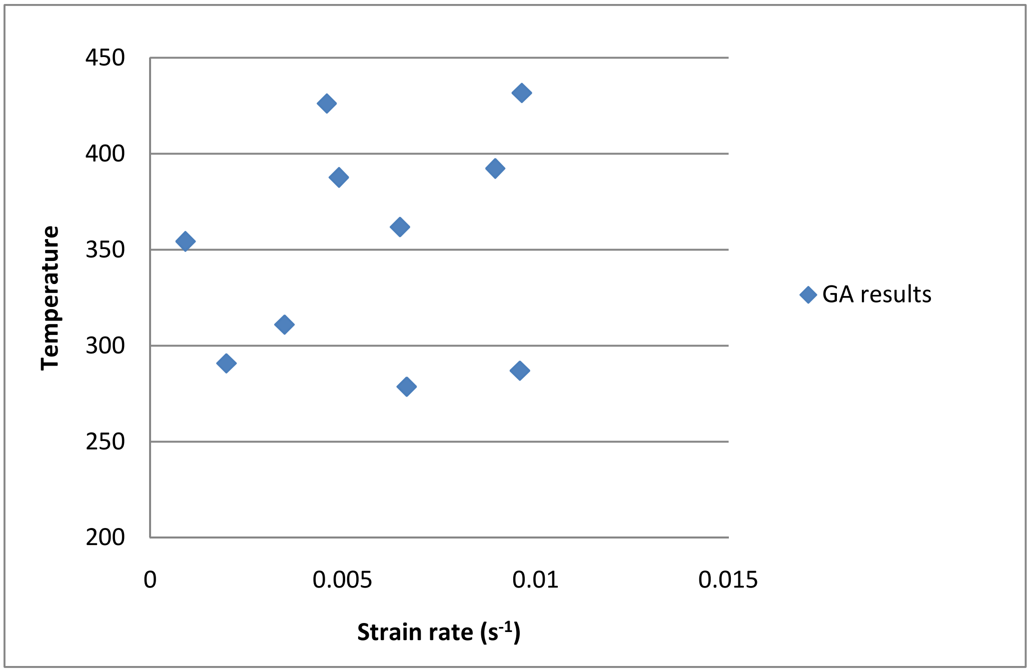


Figure 43 : GA results for σ = 13 MPa

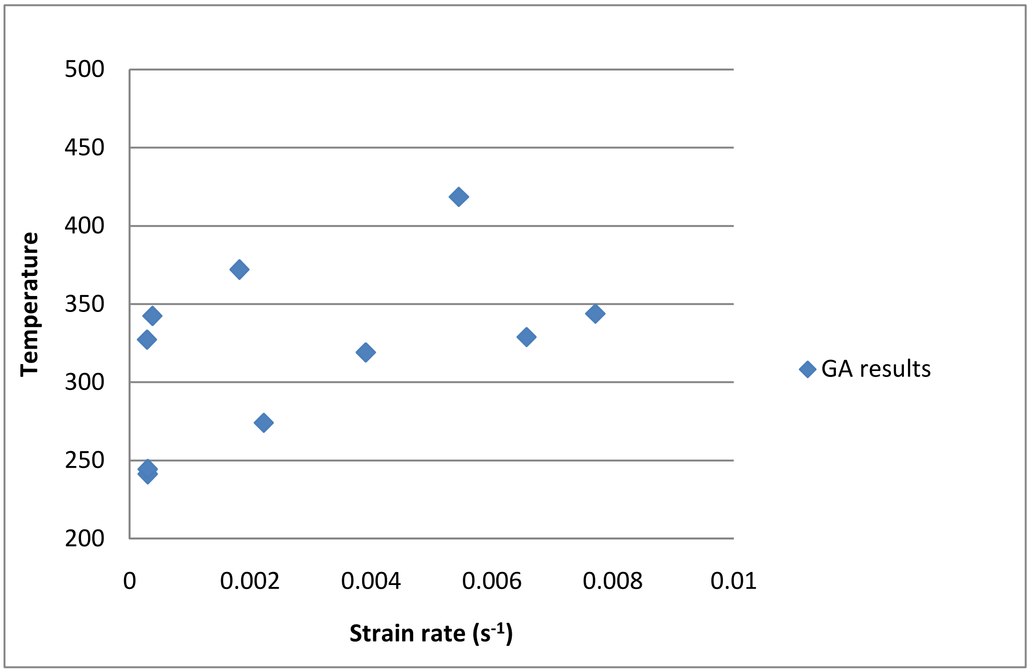


Figure 44 : GA results for $\sigma = 48$ MPa

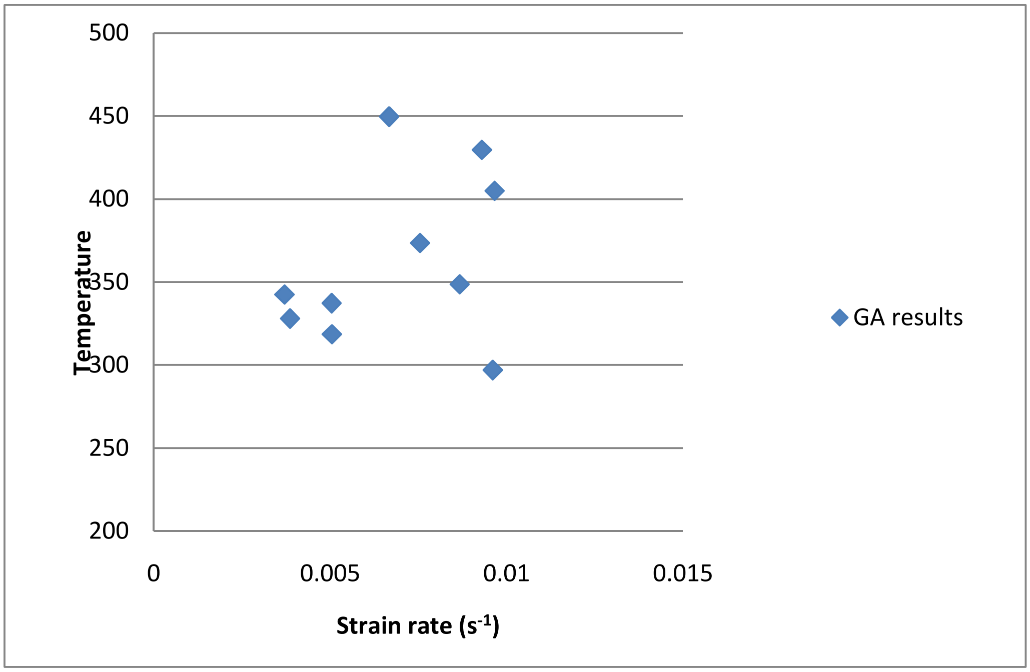


Figure 45 : GA results for $\sigma = 81$ MPa

4.3. Summary

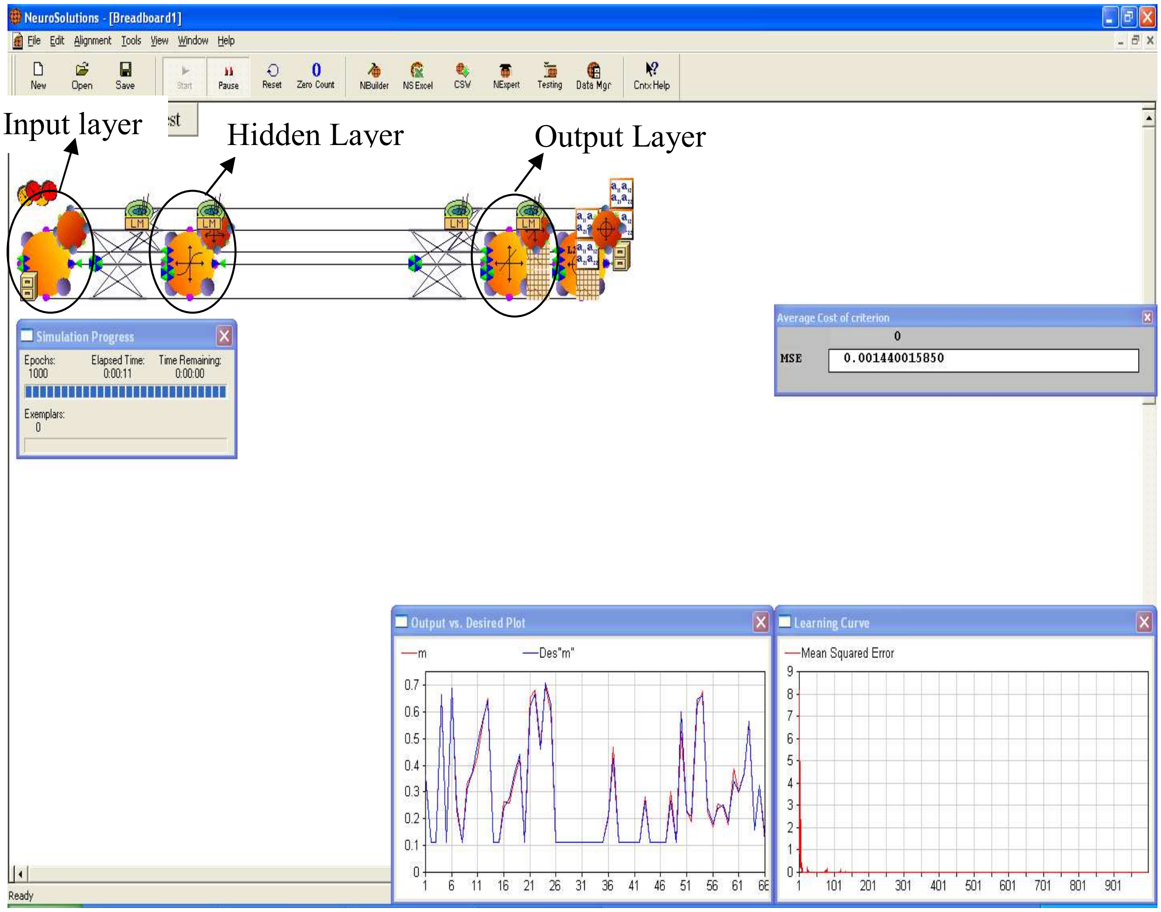
The proposed approach integrating ANN to GA is applied to Superplastic forming in this chapter. In the first application ANN is used to predict the strain rate sensitivity index given the strain rates and the temperatures and is then integrated to GA via a mathematical formulation using MS Excel. This integration is used as an inverse design tool. Given a desired m value the integrated model is prepared to predict the strain rates and temperatures which yield the given m value.

The second problem is a similar application where the ANN-GA integration model is used to obtain the strain rate and the temperature values which would generate a desired flow stress value.

Chapter 5: Results and Discussions

5.1. Problem 1

Six different gradient search (GS) methods have been used for training the ANN model. The best among these was taken further for the experimentation. This selection was made on the basis of the mean square error at the end of the training and also the proximity of the predicted data to the actual experimental data is considered. Figure 46 is a screenshot of the ANN model with Levenberg-Marquardt as the gradient search or the learning paradigm.



Epochs: 1000
Elapsed Time: 0:00:11
Time Remaining: 0:00:00
Exemplars: 0

0
MSE 0.001440015850

Figure 46 : Screenshot of an ANN breadboard

Table 7 : Experimental parameters for ANN training

- GS Method: Levenberg-Marquardt
- No. of input parameters : 2
- No. of Output parameters: 1
- No. of hidden layers : 1
- No. of PEs in hidden layer : 4
- Termination Criteria : Epochs
- No. of Epochs: 1000

The following includes the results obtained using all the six gradient search methods.

Table 8: The results obtained using different GS methods

Actual *m*	Conjugate	Delta-Bar-Delta	Levenberg-Marquardt	Momentum	Quick Propagation	Step
0.700	0.688651	0.643422	0.688842	0.601488	0.57738	0.577354
0.540	0.563476	0.587073	0.542263	0.562282	0.542127	0.542127
0.14	0.088934	0.068377	0.156038	0.104352	0.114558	0.114625
0.304	0.298574	0.223232	0.303339	0.322767	0.391	0.391046
0.200	0.197347	0.262647	0.196503	0.157556	0.15818	0.158138
0.195	0.204395	0.160639	0.17645	0.191361	0.225262	0.225182

Among the six GS methods it was observed that Levenberg-Marquardt training algorithm outperformed all the others. This is clearly evident from figure 47 and also from table 7.

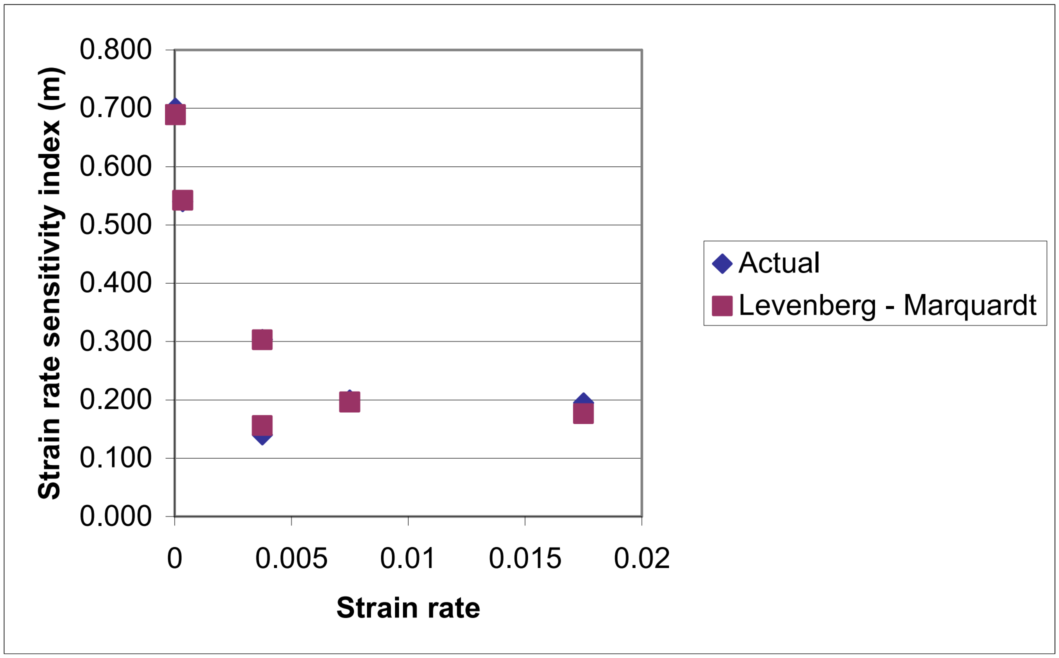
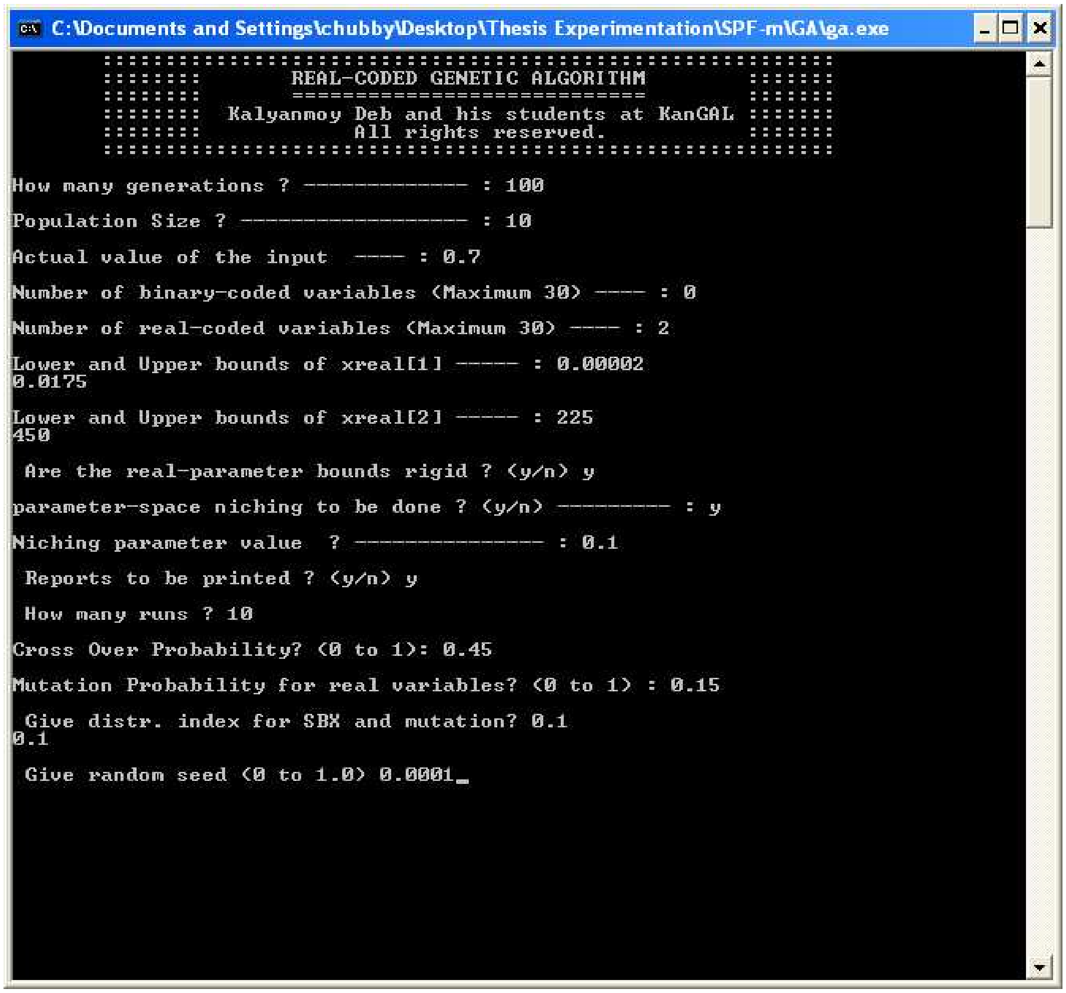


Figure 47 : Performance of Levenberg-Marquardt gradient search method

Thus the Levenberg-Marquardt (LM) gradient search method is used for further experimentation. The weights and the biases saved at the end of the training using LM are retrieved to formulate mathematical relationship which is used as an objective function.

Simulations are made using GA to obtain the best combination of strain rates and the temperatures which yield the required value of *m*. Figure 48 is a screenshot of the user interface for the GA. The subroutines customized for the purpose of the present research are included in the appendix.



C:\Documents and Settings\chubby\Desktop\Thesis Experimentation\SPF-m\GA\ga.exe

```
       :::::::::::::::::::::::::::::::::::::::::::::::::::::::::
       ::::::::        REAL-CODED GENETIC ALGORITHM      :::::::
       ::::::::        ============================      :::::::
       ::::::::   Kalyanmoy Deb and his students at KanGAL :::::::
       ::::::::             All rights reserved.         :::::::
       :::::::::::::::::::::::::::::::::::::::::::::::::::::::::

How many generations ? ------------- : 100

Population Size ? ------------------ : 10

Actual value of the input  ---- : 0.7

Number of binary-coded variables (Maximum 30) ---- : 0

Number of real-coded variables (Maximum 30) ---- : 2

Lower and Upper bounds of xreal[1] ----- : 0.00002
0.0175

Lower and Upper bounds of xreal[2] ----- : 225
450

 Are the real-parameter bounds rigid ? (y/n) y

parameter-space niching to be done ? (y/n) --------- : y

Niching parameter value  ? --------------- : 0.1

 Reports to be printed ? (y/n) y

 How many runs ? 10

Cross Over Probability? (0 to 1): 0.45

Mutation Probability for real variables? (0 to 1) : 0.15

 Give distr. index for SBX and mutation? 0.1
0.1

 Give random seed (0 to 1.0) 0.0001_
```

Figure 48: Screenshot of the user interface for the GA [67]

4 different strategies have been used for experimentation. This discussion would include only strategy 1 as the performance using this set of parameters was found to be better. The parameters in this strategy are illustrated in table 9.

Table 9 : Strategy 1

Number of generations	Population size	Range for $\dot{\varepsilon}$	Range for T	Niching Parameter	Number of Trials	Crossover Probability (Cp)	Mutation Probability (Mp)	Seeding Factor
100	10	(0.0175, 0.000015)	(225, 450)	0.1	10	0.45	0.15	0.0001

ANN is trained to predict the strain rate sensitivity corresponding to the strain rates and the temperatures. The weights and the biases used during learning are retrieved and a mathematical expression is formulated which represents m as a function of strain rate and the temperature. The error which is the relative difference between the actual *m* and the GA predicted *m* is used as an objective function. The results which are the temperatures and the strain rates corresponding to a specific value of *m* are presented in the following plots and in table 10.

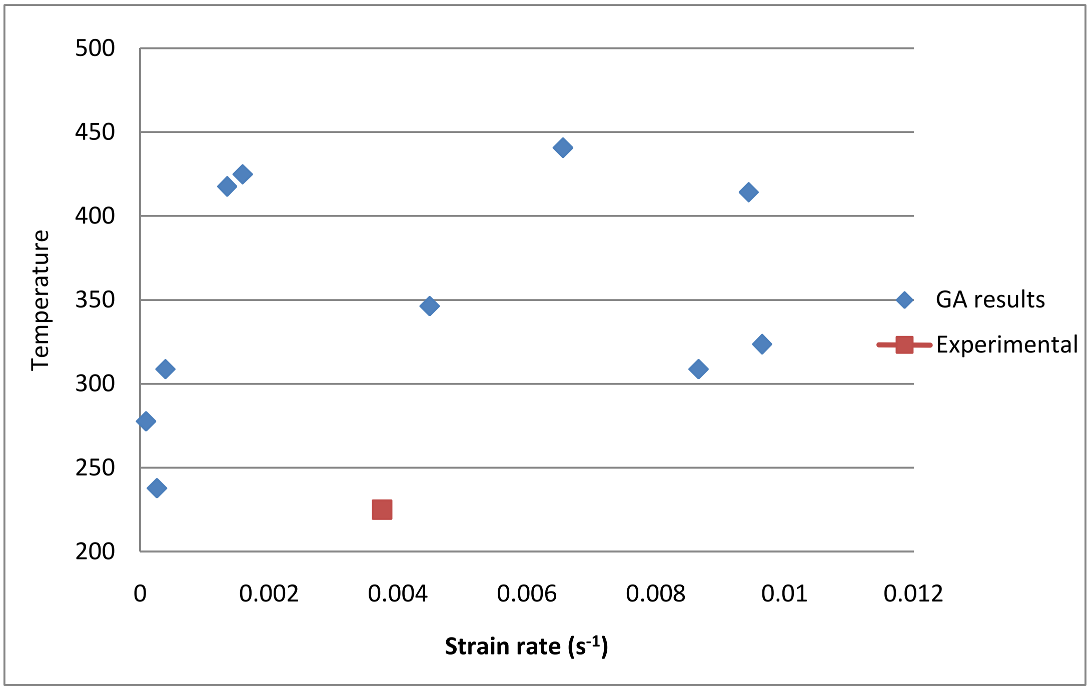


Figure 49 : Comparison of GA predicted values for *m* = 0.14 and the experimental values

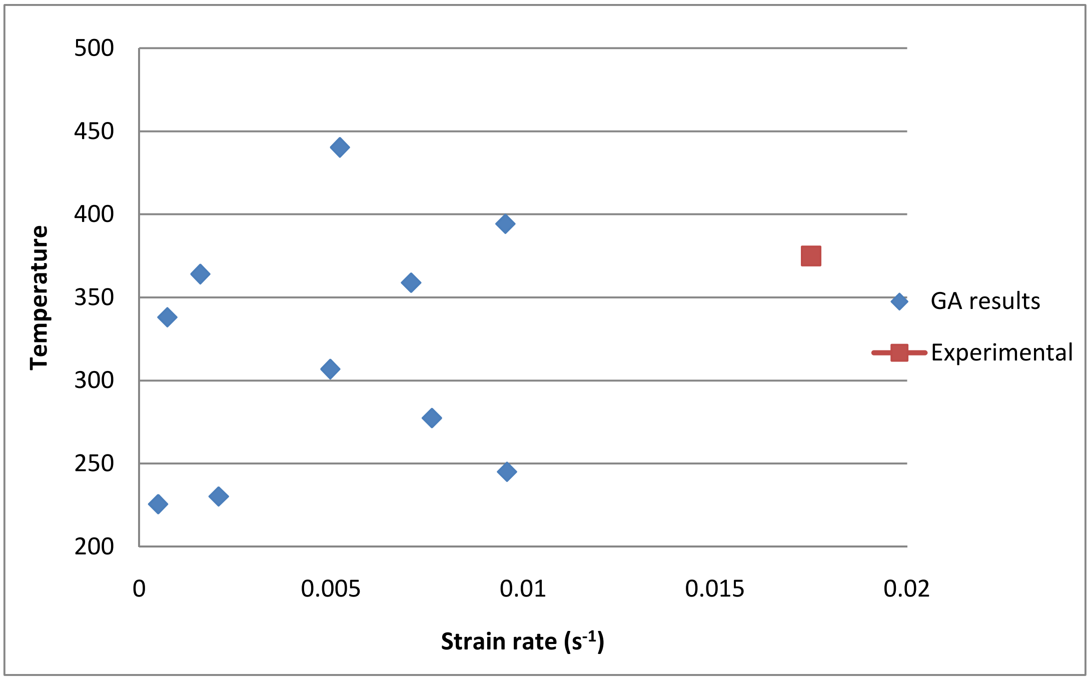


Figure 50 : Comparison of GA predicted values for *m* = 0.195 and the experimental values

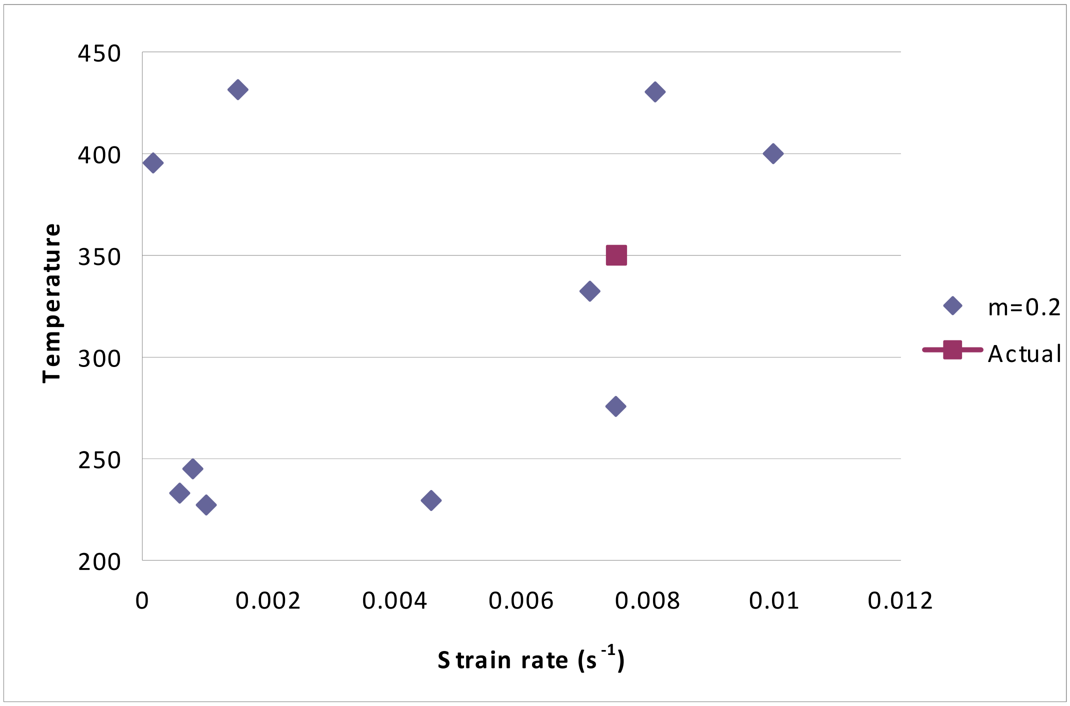


Figure 51 : Comparison of GA predicted values for *m* = 0.2 and the experimental value

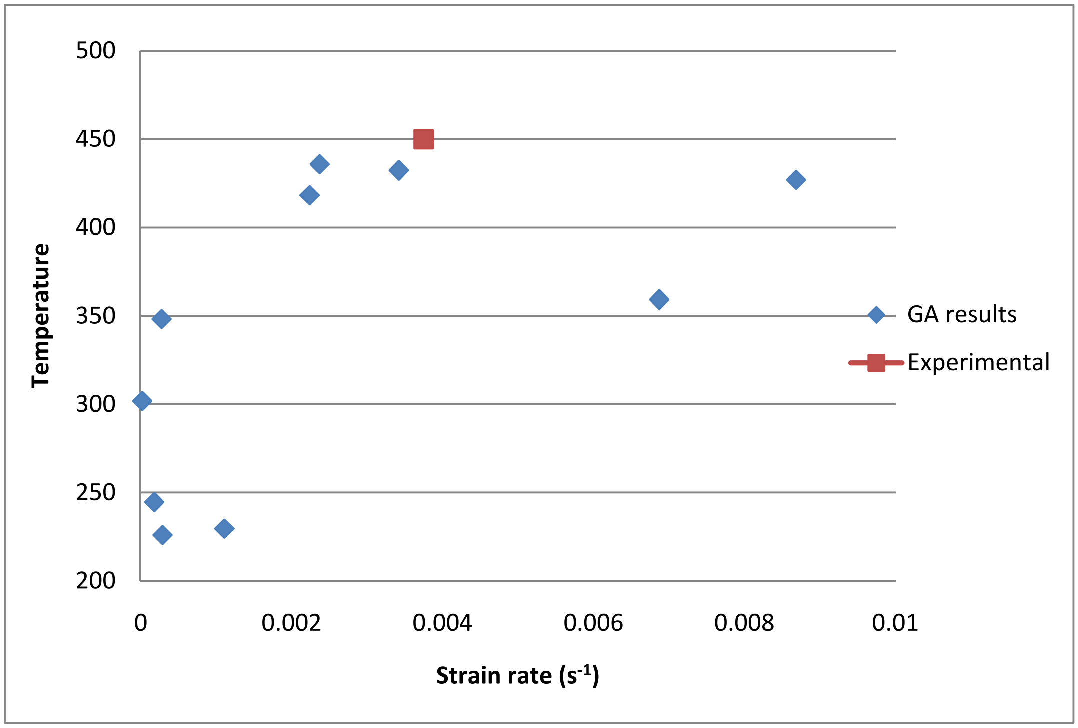


Figure 52 : Comparison of GA predicted values for *m* = 0.304 and the experimental value

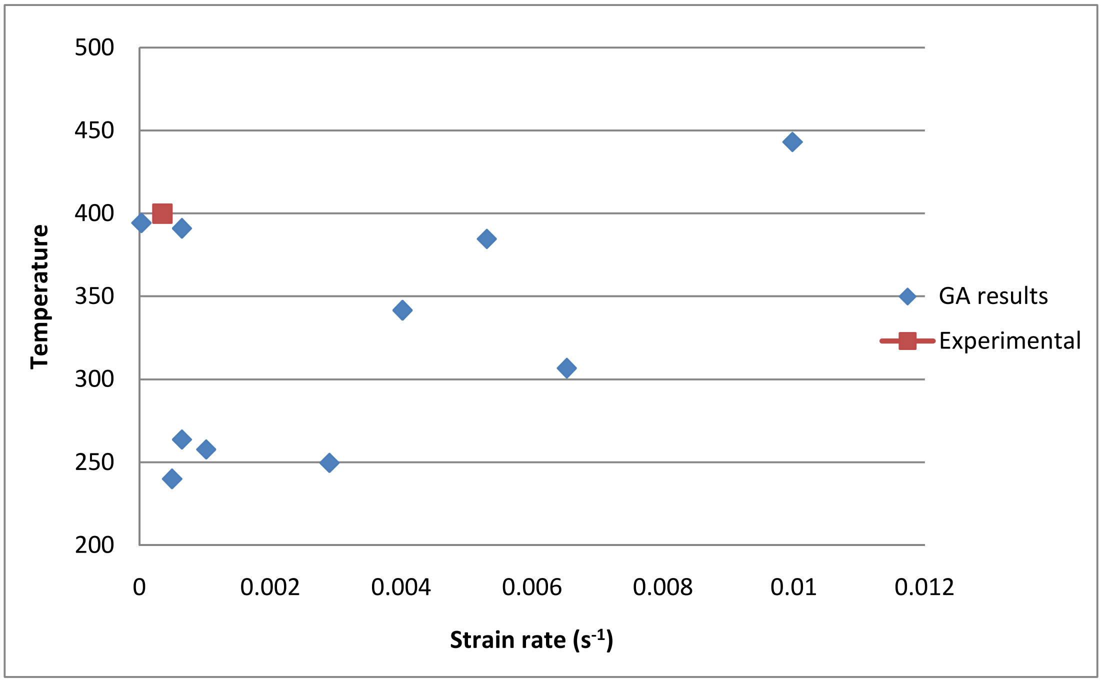


Figure 53 : Comparison of GA predicted values for *m* = 0.54 and the experimental value

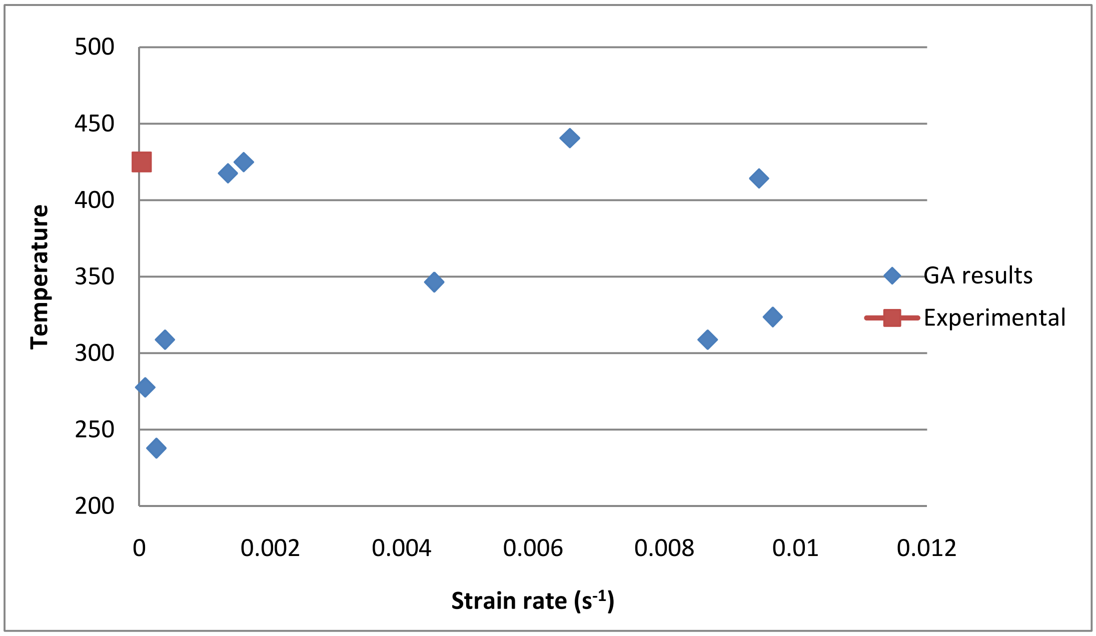


Figure 54 : Comparison of GA predicted values for *m* = 0.7 and the experimental value

Table 10: GA predicted strain rates and temperatures

Trial	m=0.14		m=0.2		m=0.195	
	SR	Temp	SR	Temp	SR	Temp
1	9E-05	389.9614	0.000123	392.7945	0.000136	418.524
2	0.0003	415.6743	0.000202	441.5649	0.000177	387.1938
3	0.0005	353.9275	0.00025	343.0425	0.000181	318.3776
4	0.0006	356.3202	0.000319	325.5796	0.000334	313.3751
5	0.0011	365.9735	0.000378	293.2905	0.000595	330.6296
6	0.0013	376.6664	0.000434	357.1834	0.000786	428.2114
7	0.0014	413.3571	0.000643	448.2808	0.000843	371.0082
8	0.0014	414.1958	0.000764	395.2515	0.001021	408.3773
9	0.0016	421.5216	0.000942	366.0753	0.001115	434.7239
10	0.0021	446.3404	0.001478	431.8206	0.001909	429.3818
Average	0.001	395.3938	0.000553	379.4884	0.00071	383.9803
Actual	0.0038	225	0.0075	350	0.0175	375

Trial	m=0.54		m=0.7		m=0.304	
	SR	Temp	SR	Temp	SR	Temp
1	0.000067	366.1785	0.000307	335.1543	0.000089	395.0438
2	0.000069	277.2722	0.000503	349.2239	0.000181	319.7039
3	0.000254	410.4168	0.000556	321.1812	0.000187	307.7927
4	0.00031	360.9239	0.000676	387.3453	0.000307	335.1543
5	0.000358	359.2024	0.000762	387.7812	0.001018	428.5065
6	0.000771	360.5737	0.001143	404.2414	0.001145	401.3186
7	0.000848	363.7031	0.001177	362.9292	0.001509	401.07
8	0.001093	365.9514	0.001222	403.2475	0.001566	397.7471
9	0.00151	414.0023	0.001254	393.1675	0.001706	438.7524
10	0.010493	409.6965	0.002052	446.2812	0.003065	267.4946
Average	0.001577	368.7921	0.000965	379.0553	0.001077	369.2584
Actual	0.00035	400	0.000035	425	0.00375	450

By comparing the experimental strain rate and temperature values corresponding to each of these six strain rate sensitivity index (*m*) values with the GA generated strain rate and temperature values, it can be observed that in most of the cases the GA predictions were close to the actual experimental values though they could not give the best fit. And also a unique solution to these problems cannot be anticipated using a GA. This is one drawback of the approach.

The experimentation using GA is carried out for four different strategies as mentioned previously. These experiments were done for 10, 20 and 50 trials. The purpose of which was to check for the repeatability of the GA and thus establish a guideline on how predictable the GA is. The results presented in the plots are for 10 trials. The results for 20 and 50 trials are included in the Appendix.

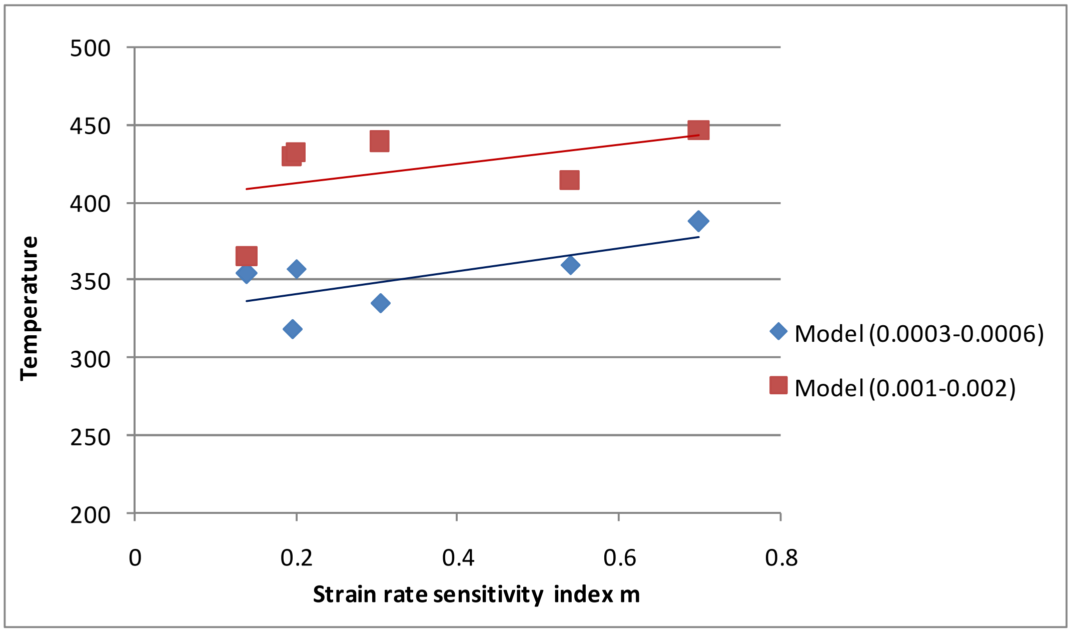


Figure 55 : Results from the ANN-GA model corresponding to strain rates (0.0003-0.0006) and (0.001-0.002)

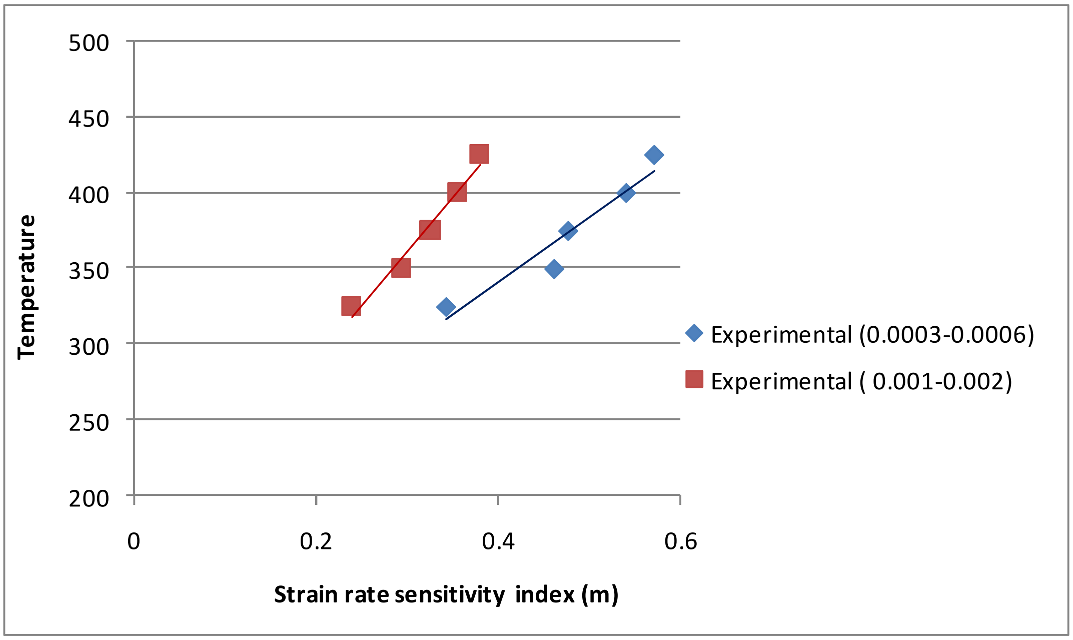


Figure 56 : Experimental results for strain rates (0.0003-0.0006) and (0.001-0.002)

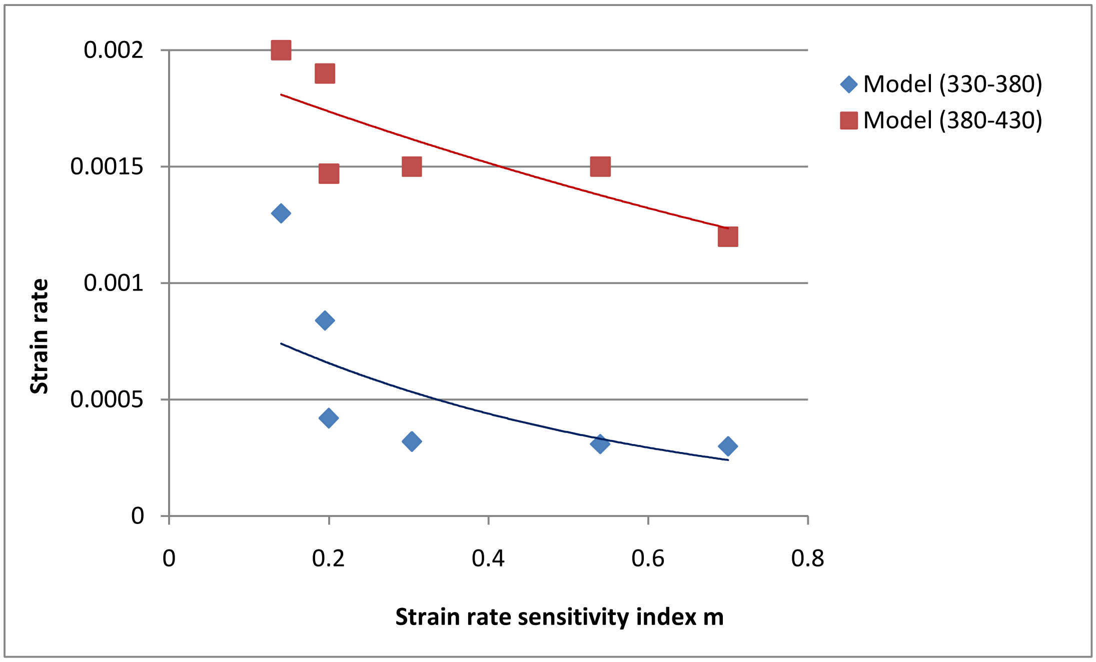


Figure 57 : Results from the ANN-GA model corresponding to temperatures (330-380) and (380-430)

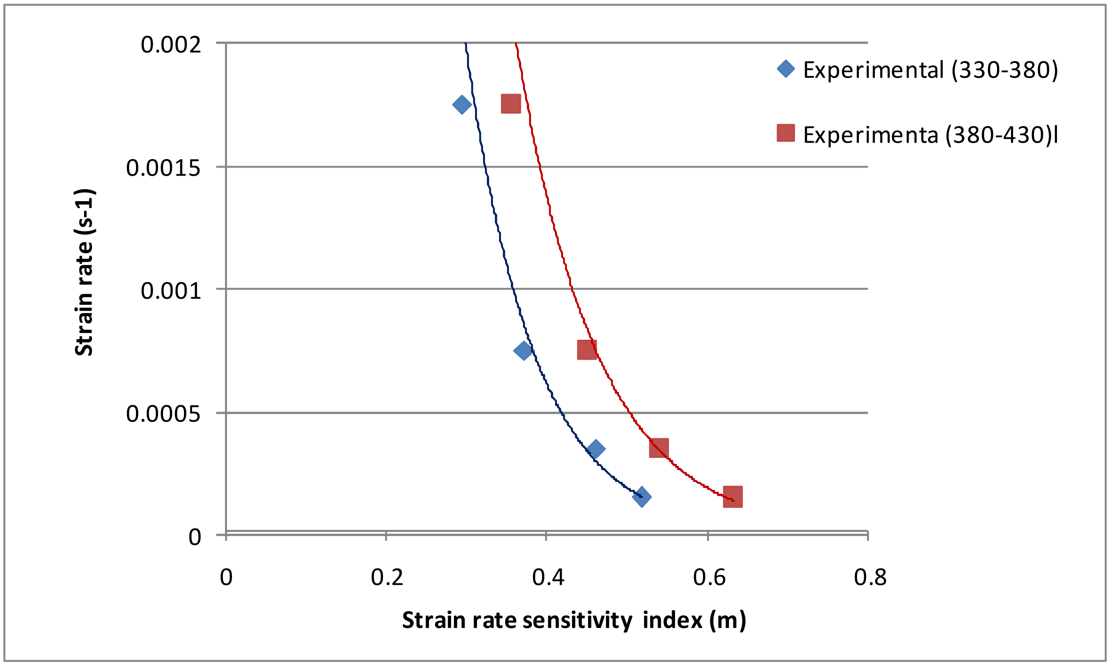


Figure 58 : Experimental results for temperatures (330-380) and (380-430)

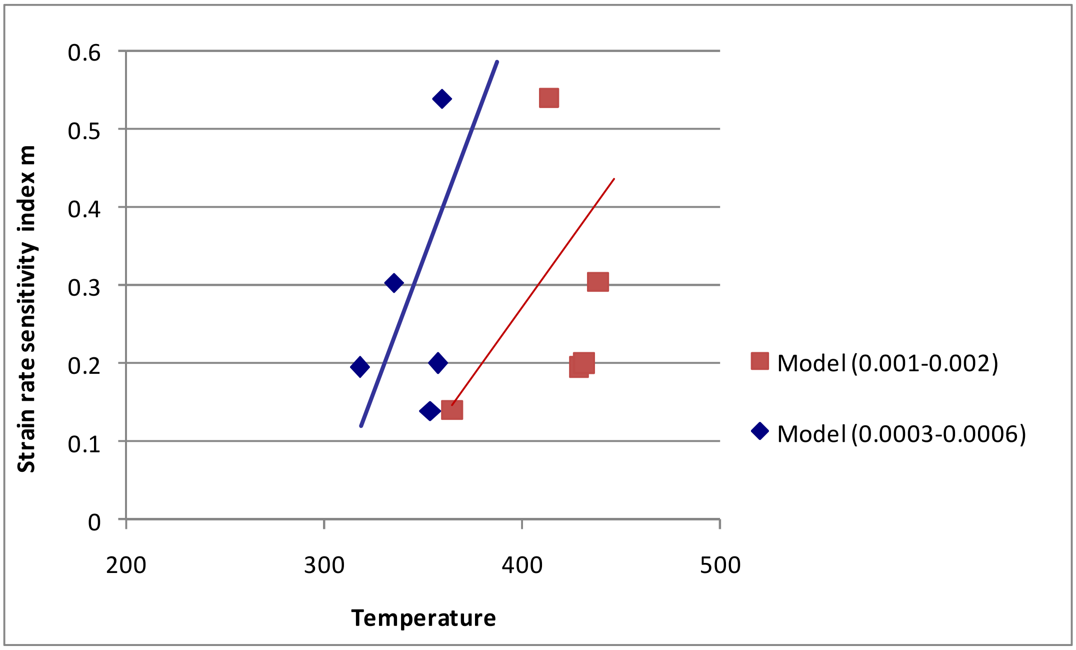


Figure 59: Results from the ANN-GA model for strain rates (0.001-0.002) and (0.0003-0.0006)

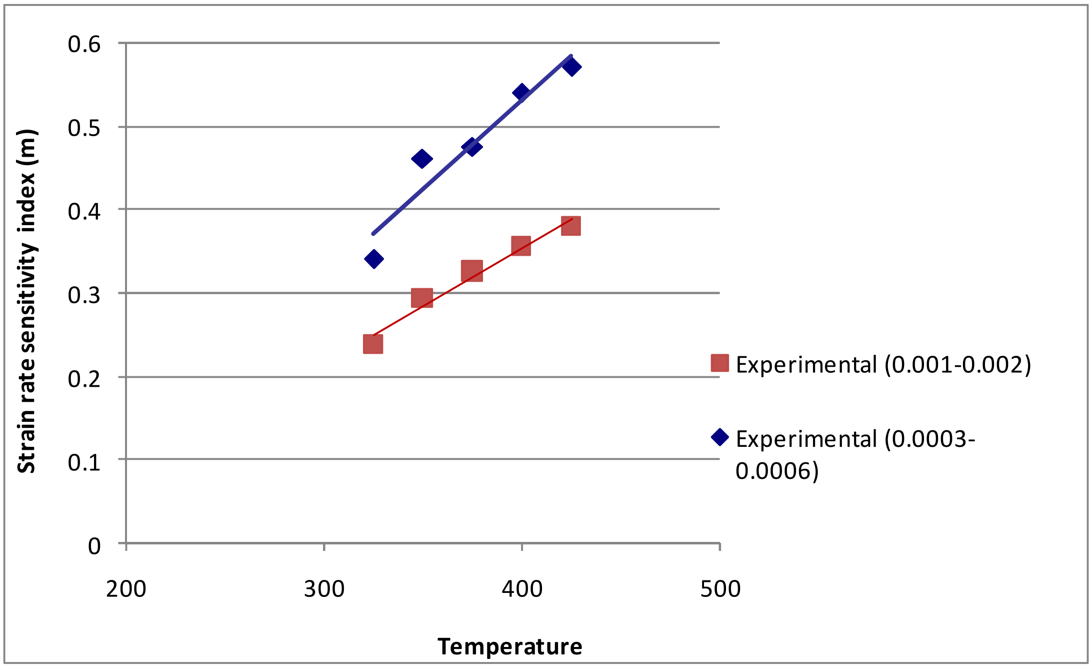


Figure 60 : Experimental results for strain rates (0.001-0.002) and (0.0003-0.0006)

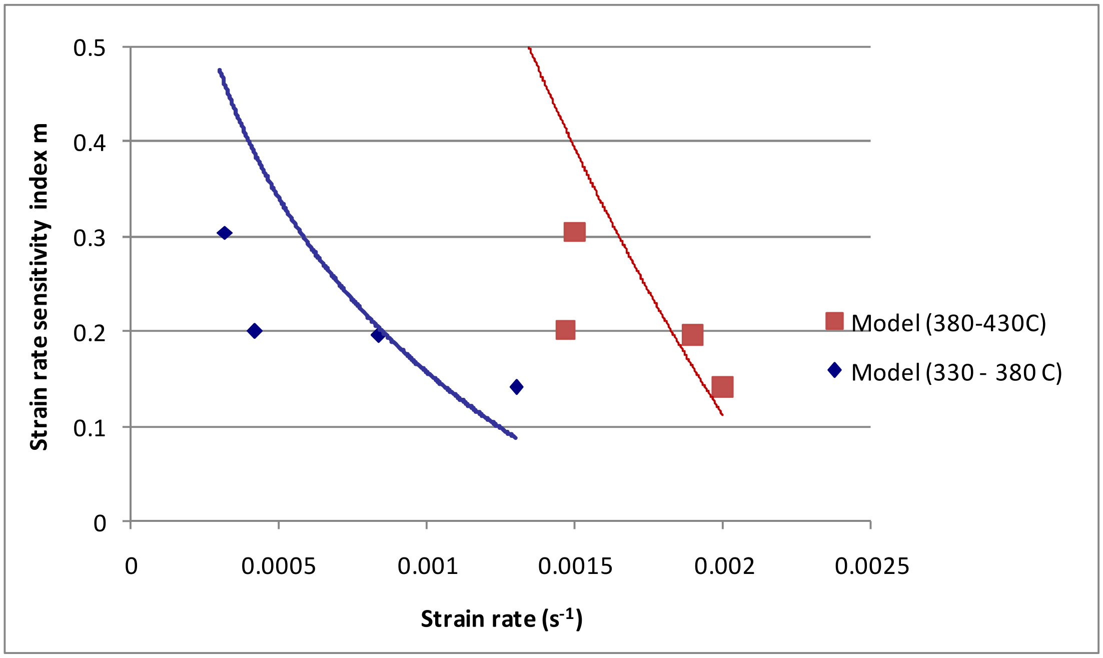


Figure 61 : Results from the ANN-GA model for temperatures (330-380C) and (380-430C)

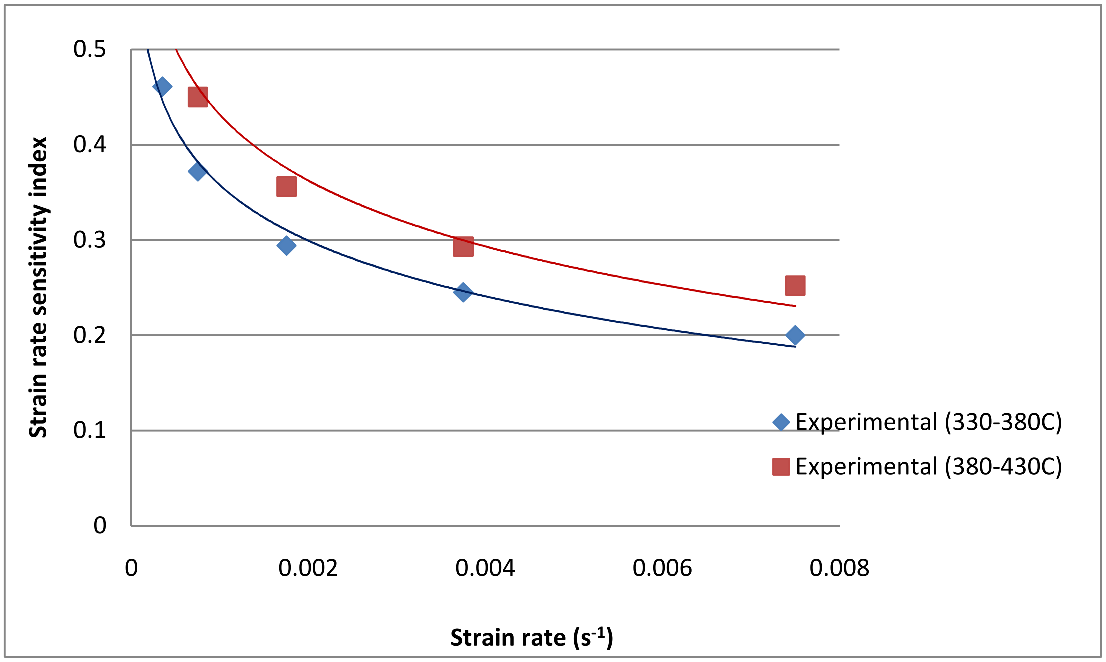


Figure 62 : Experimental results for temperatures (330-380C) and (380-430C)

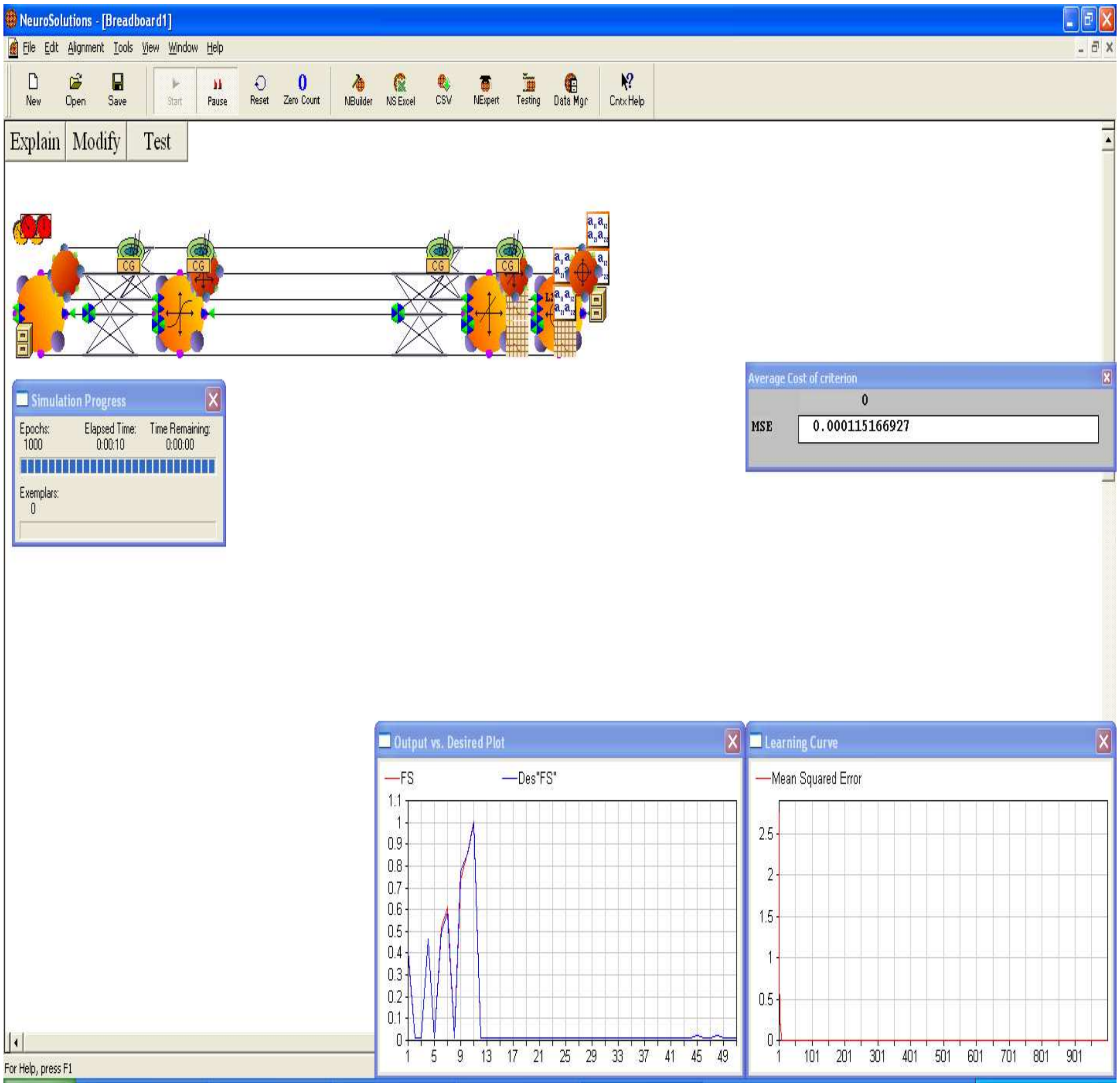
The plots 55, 56 represent the results generated by the GA for 2 different strain rate ranges. These results are plotted in order to address the concern that the GA results did not give a best fit when compared to the experimental results. Observing the results in these plots indicated that the trend followed by the results from the ANN-GA model is similar to the experimental results.

Similarly plots 57 and 58 give a comparison of the behavior of the ANN-GA results to the experimental results for 2 different range of temperatures. It is again observed that both these sets follow the same trend.

A similar comparison of results at a specific range of strain rates and temperatures is presented in plots 59 through 62 too.

5.2. Problem 2

Among the six gradient search methods used for training the model for this problem, it was found that conjugate gradient search method outperformed all the other GS methods. The screenshot of the ANN model using Conjugate GS method is presented in figure 55.



Epochs: 1000
Elapsed Time: 0:00:10
Time Remaining: 0:00:00
Exemplars: 0

0
MSE 0.000115166927

Figure 63: Screenshot of the ANN model used in predicting the flow stress

Table 11 : Experimental parameters for ANN training

- GS Method: Conjugate
- No. of input parameters : 2
- No. of Output parameters: 1
- No. of hidden layers : 1
- No. of PEs in hidden layer : 3
- Termination Criteria : Epochs
- No. of Epochs: 1000

The best performing GS method was chosen on the basis of the mean square error at the end of the training and the proximity of the predicted values to the actual experimental values. The same has been presented in the following plot and the table.

Table 12 : The results obtained using different GS methods

Actual σ	Conjugate	Levenberg-Marquardt	Momentum	Quick Propagation	Step	Delta-Bar-Delta
81	87.29923	66.25482	99.65938	95.56296	95.858733	95.07692
13	11.69232	16.67334	12.98519	18.46343	18.280565	11.06591
4.75	4.446782	5.222236	6.415669	10.86201	10.695521	4.690423
1.6	1.17853	1.181354	2.751108	5.37917	5.2817556	1.697771
49	52.43645	56.86877	51.89711	50.78108	50.864955	52.486

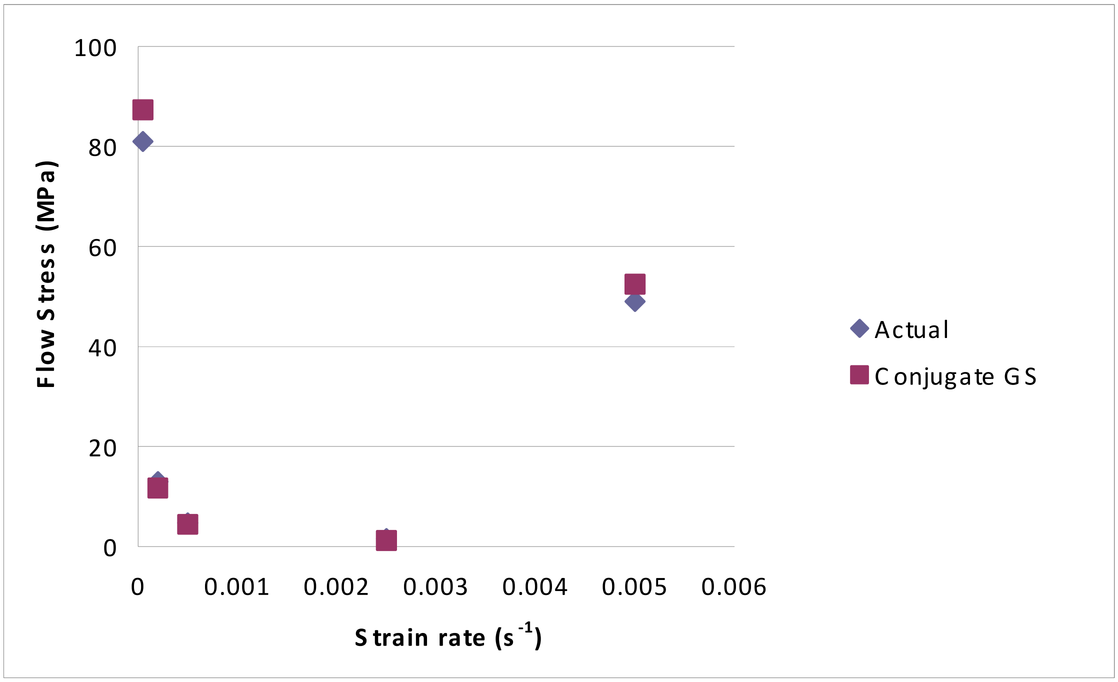


Figure 64 : Performance of Conjugate gradient search method

The weights and biases saved at the end of the training using Conjugate gradient search are used to formulate a mathematical relationship. This relationship is used as an objective function for the GA.

Similar to problem 1, four different strategies were used for the GA experimentation. Results obtained using strategy 1 is discussed here. The following table lists the parameters included for strategy 1.

Table 13: Parameters used in strategy 1

Number of generations	Population size	Range for $\dot{\varepsilon}$	Range for T	Niching Parameter	Number of Trials	Crossover Probability (Cp)	Mutation Probability (Mp)	Seeding Factor
100	10	(0.00002, 0.000015)	(225, 450)	0.1	10	0.45	0.15	0.0001

The results from the GA simulations using the above mentioned strategy are represented in table 12 and the plots following it.

Table 14: GA predicted strain rates and temperatures

	$\sigma = 1.6$ Mpa		$\sigma = 4.75$ Mpa		$\sigma = 13$ Mpa	
Trial	$\dot{\varepsilon}$	T	$\dot{\varepsilon}$	T	$\dot{\varepsilon}$	T
1	0.00002	447.4554	0.00006	430.31766	0.00092	354.28539
2	0.0004	234.1116	0.00012	372.85297	0.00198	290.84511
3	0.00207	425.4281	0.00175	228.7983	0.00349	311.0736
4	0.00226	429.8928	0.00233	436.73114	0.00458	426.18579
5	0.00229	394.3834	0.00302	226.51111	0.0049	387.64135
6	0.00497	263.3338	0.00432	404.3306	0.00648	361.83757
7	0.00813	390.2924	0.00483	344.57966	0.00665	278.71366
8	0.00849	441.7477	0.00675	245.57541	0.00895	392.32321
9	0.00927	299.3539	0.00959	390.71399	0.00959	287.06816
10	0.00959	387.9649	0.0098	365.44214	0.00964	431.78109
Average	0.004749	371.3964	0.004257	344.5853	0.005718	352.175493
Actual	0.00005	425	0.0002	400	0.0005	375

	σ = 48 Mpa		σ = 81Mpa	
Trial	$\dot{\varepsilon}$	T	$\dot{\varepsilon}$	T
1	0.00029	327.3049	0.00371	342.5149
2	0.0003	241.2822	0.00386	328.0034
3	0.0003	244.3431	0.00504	337.2322
4	0.00038	342.3936	0.00505	318.6432
5	0.00182	372.0016	0.00667	449.5732
6	0.00222	274.1863	0.00755	373.597
7	0.00391	319.1848	0.00867	348.6764
8	0.00545	418.5309	0.0093	429.698
9	0.00657	329.0017	0.00961	296.9257
10	0.00771	343.8835	0.00966	405.0093
Average	0.002895	321.2112	0.006912	362.9873
Actual	0.005	325	0.0025	225

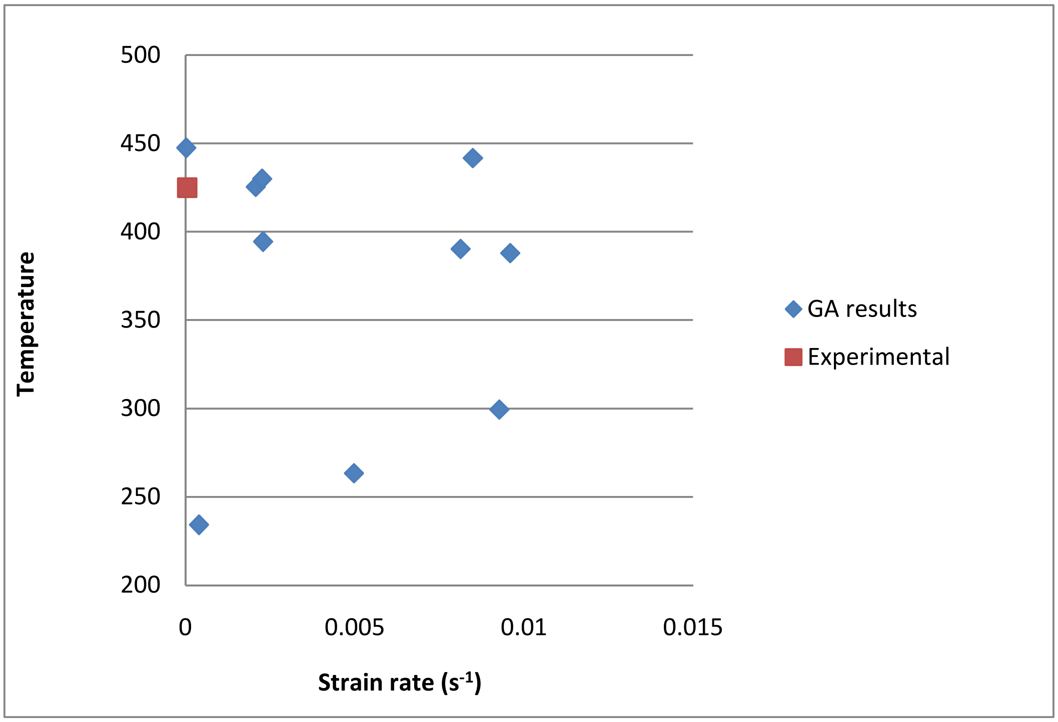


Figure 65: Comparison of GA predicted values for $\boldsymbol{\sigma}$ = 1.6 MPa and the experimental values

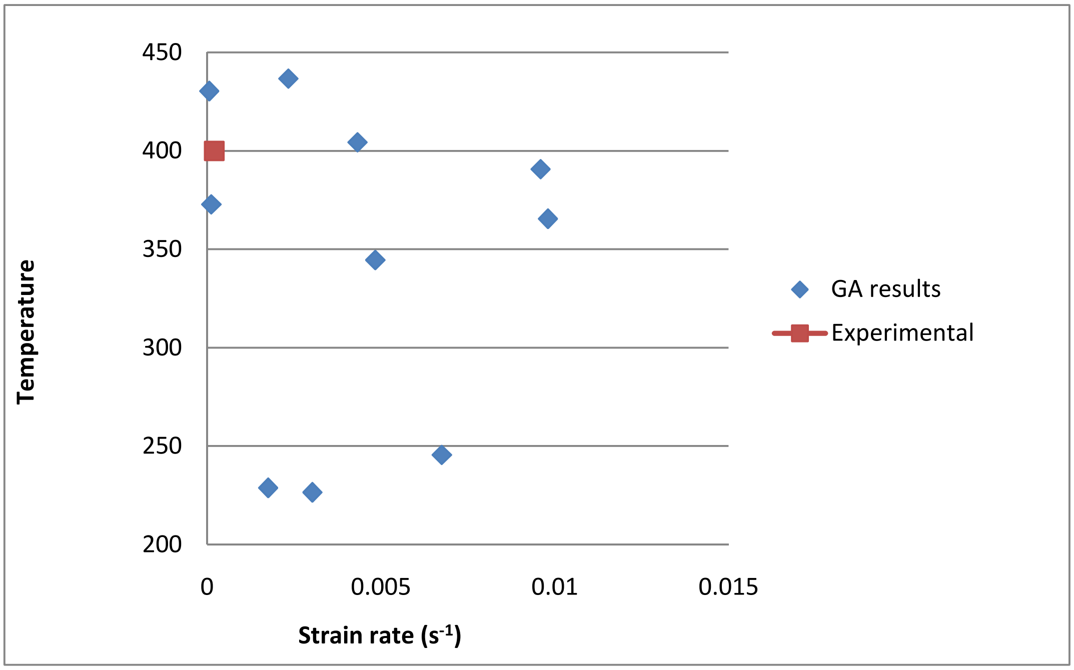


Figure 66 : Comparison of GA predicted values for $\boldsymbol{\sigma}$ = 4.75 MPa and the experimental values

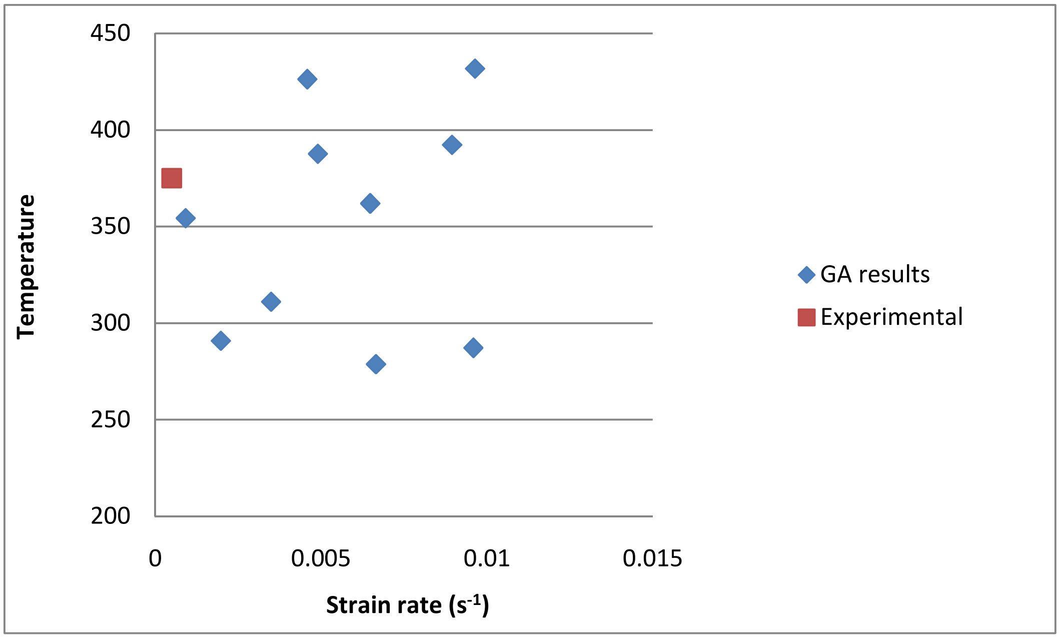


Figure 67 : Comparison of GA predicted values for $\boldsymbol{\sigma}$ = 13 MPa and the experimental values

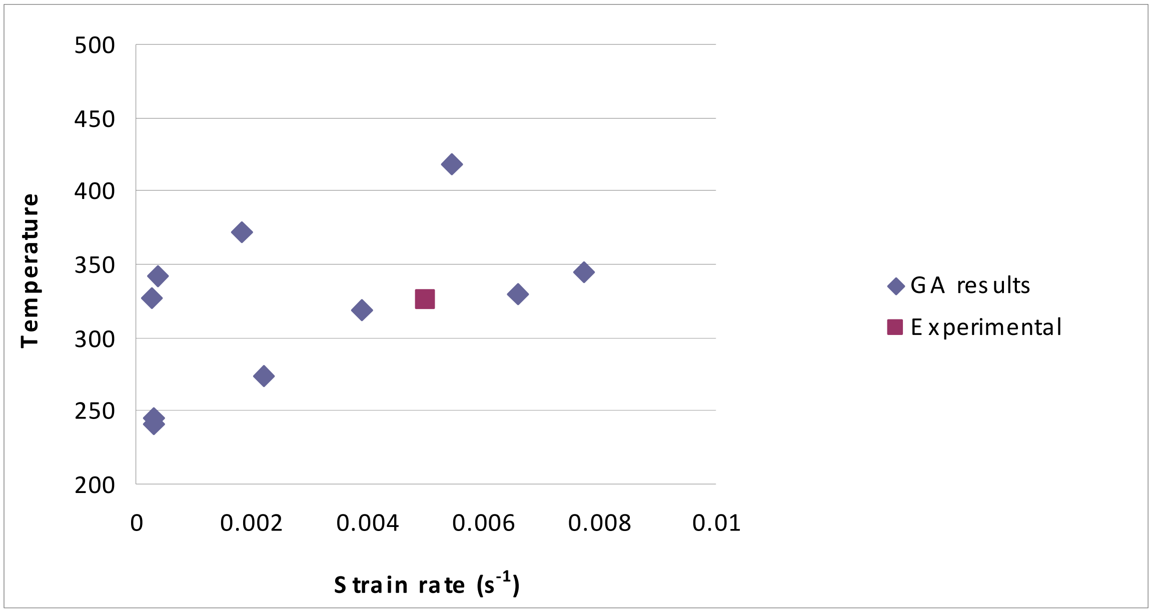


Figure 68 : Comparison of GA predicted values for σ = 48 MPa and the experimental values

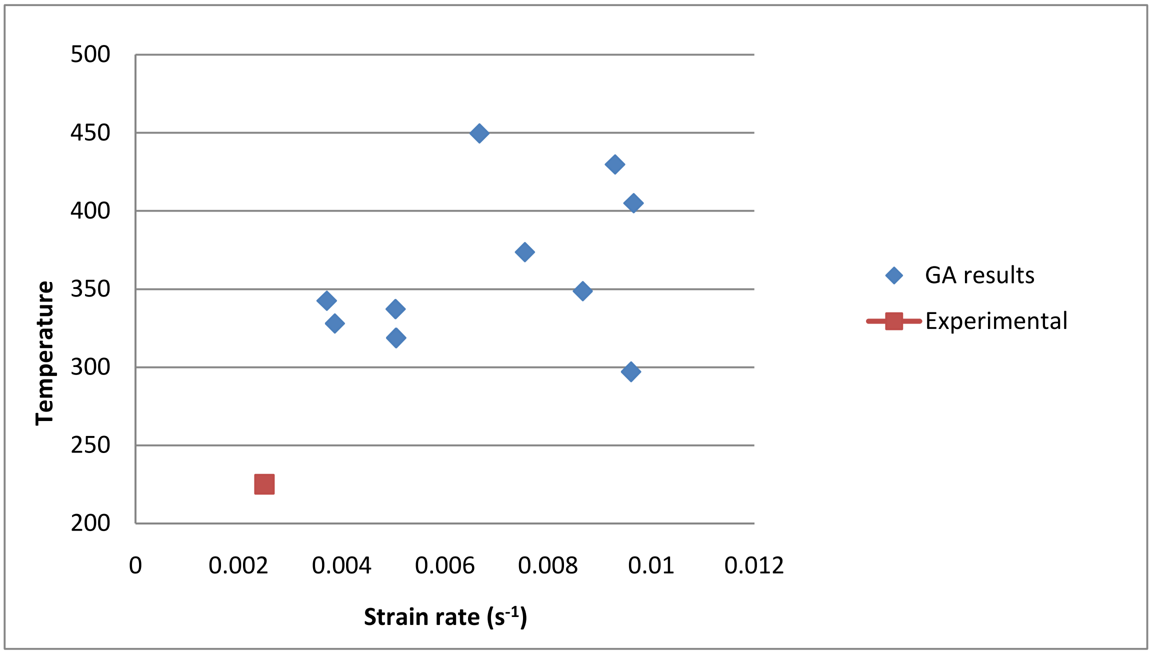


Figure 69 : Comparison of GA predicted values for σ = 81 MPa and the experimental values

The GA simulations were done for 5 different flow stress values. It is evident from the plots that the strain rate and the temperature values predicted using the GA was close to the actual experimental values. But similar to the problem 1 a unique pair of strain rate and temperature corresponding to specific value of flow stress could not be obtained.

The results presented in the plots 57 through 61 are simulations carried out for 10 trials. The GA has been checked for repeatability by running the simulations for 20 and 50 trials. The results from these trials are included in the Appendix.

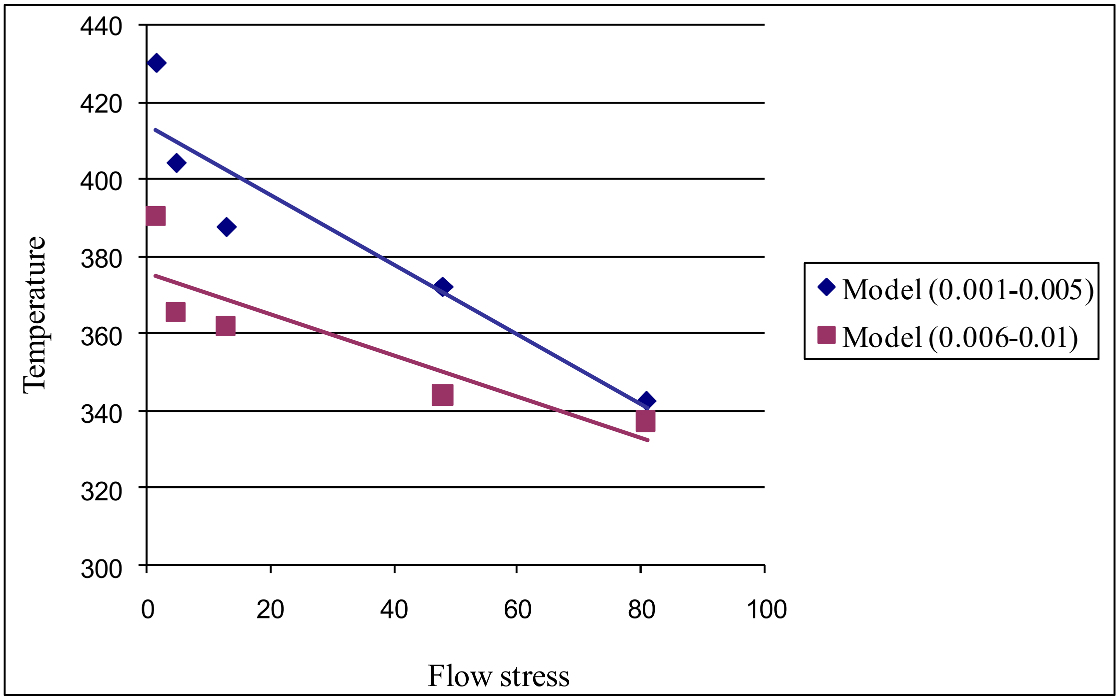


Figure 70 : Results from the ANN-GA model for strain rates (0.001-0.005) and (0.005-0.01)

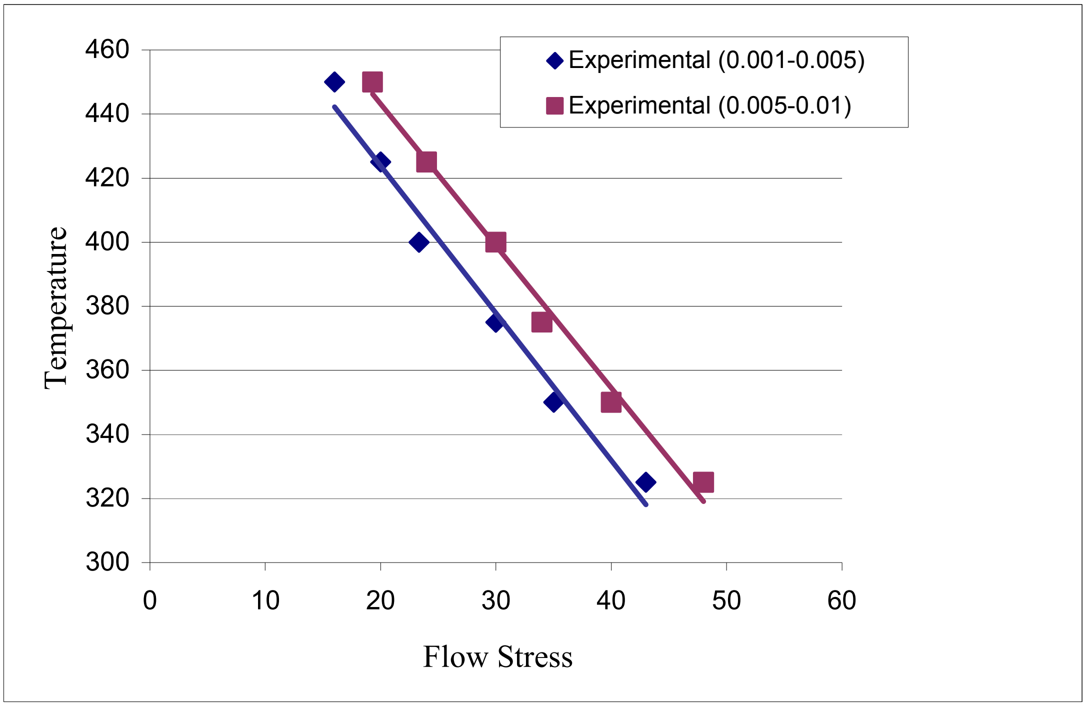


Figure 71 : Experimental Results for strain rates (0.001-0.005) and (0.005-0.01)

The results obtained from the ANN-GA model for strain rate ranges of (0.001-0.005) and (0.005-0.01) are plotted and these when compared to the experimental results from the same strain rate ranges indicate that they have a similar trend. This is one approach to validate the ANN-GA model developed in this research.

5.3. Summary

The results obtained by applying the proposed approach to the two problems mentioned in chapter 3 and 4 have been illustrated in this chapter. A discussion on the results is also presented.

Chapter 6: Conclusions and Future Recommendations

6.1. Conclusions

A hybrid model integrating predictive capabilities of ANN and optimization feature of GA is developed for the purpose of inverse modeling. The proposed approach is applied to Superplastic forming of materials to predict the material properties which characterize the performance of a material. The study is conducted using two problems. For the first problem, ANN is trained to predict the strain rate sensitivity index *m* given the temperature and the strain rate. The performance of different gradient search methods used in training the ANN model is demonstrated. The weights and the biases from the best Gradient Search method which is the Levenberg-Marquardt in this case are used to formulate the objective function of the GA. GA simulations are performed using 4 different strategies.

Similar approach is used for the second problem. The objective of this problem is to predict the input parameters, i.e. strain rate and temperature corresponding to a given flow stress value. The best performing GS in this case is the Conjugate GS method. Similar to the first problem the experiments using the GA are conducted using 4 different strategies. These strategies mainly differed in the number of trials, crossover and the mutation probabilities.

Results from strategy 1 (10 trials) are discussed in detail. The GA generated strain rate and the temperatures corresponding to a given *m* in the first case and the flow stress in the second case were compared to the experimental results. Results obtained in both the problems indicate proximity to the experimental results.

One of the drawbacks of this approach is that a unique solution cannot be obtained. The results generated using this approach is near optimal. Therefore this cannot be applied to applications where a unique solution to the problem is desired.

6.2. Unique Features

- Integrating the ANN and the GA for inverse design

ANN model is trained to learn the relationship between the process parameters. The relationship developed is used in the objective function of the GA to generate near optimal solutions.

- Application of ANN-GA model for Material Informatics

Materials informatics studies the collection, classification, storage, retrieval, transfer, and dissemination of material science information to support the advancement of knowledge in this domain. This in the present approach is obtained through the application of the ANN-GA model.

- Model validated to be used for design and optimizing processes

The working of the ANN-GA model is demonstrated using an example from Superplastic forming.

6.3. Future Work

The proposed approach can be further validated by applying it on better data sets. Better in terms of the number of exemplars and also the number of process parameters.

The performance of the ANN-GA model can be tested by establishing physical boundaries to the process parameters.

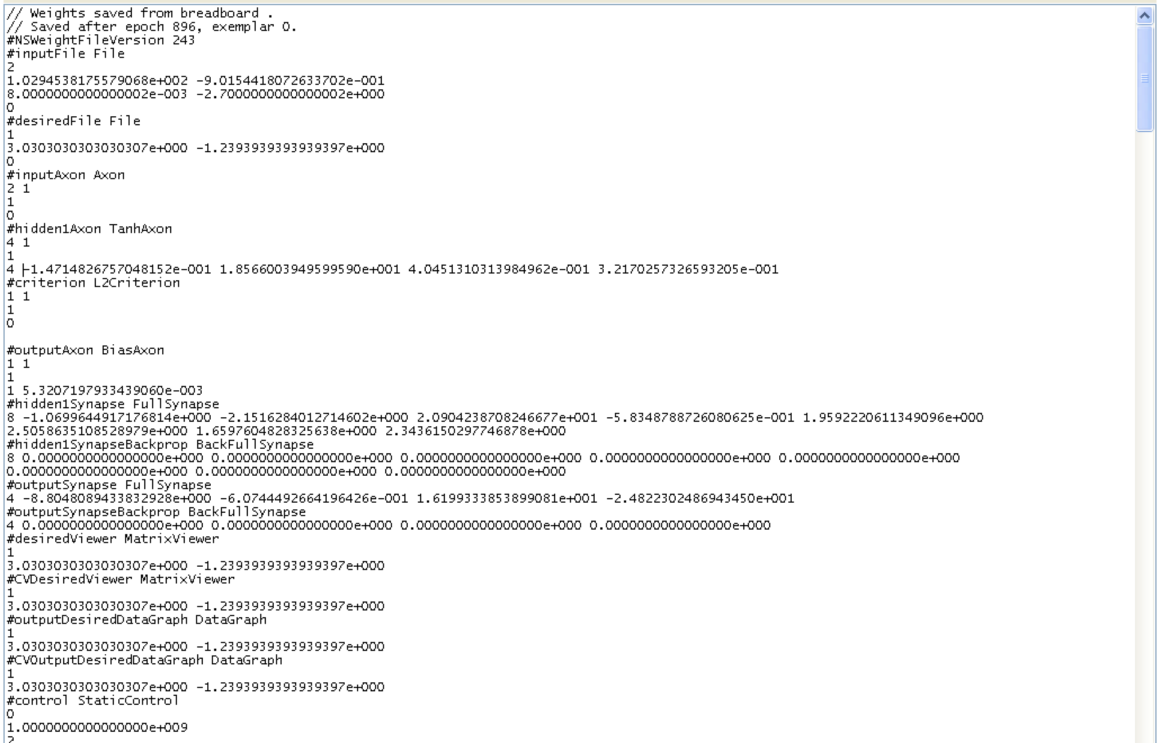
One of the drawbacks of this approach is that it cannot give a unique solution. The solutions thus obtained are near optimal. Therefore a method to filter the outcome of this model can be one scope for further research.

GA simulations can be performed with more strategies by varying the different parameters such as the crossover probability, mutation probability, Niching, Seeding parameters.

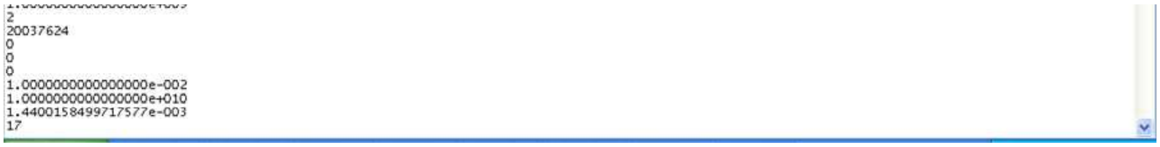
APPENDIX I

Weights File from NeuroSolutions Breadboard: Problem 1

(Only data used in the computations are shown)



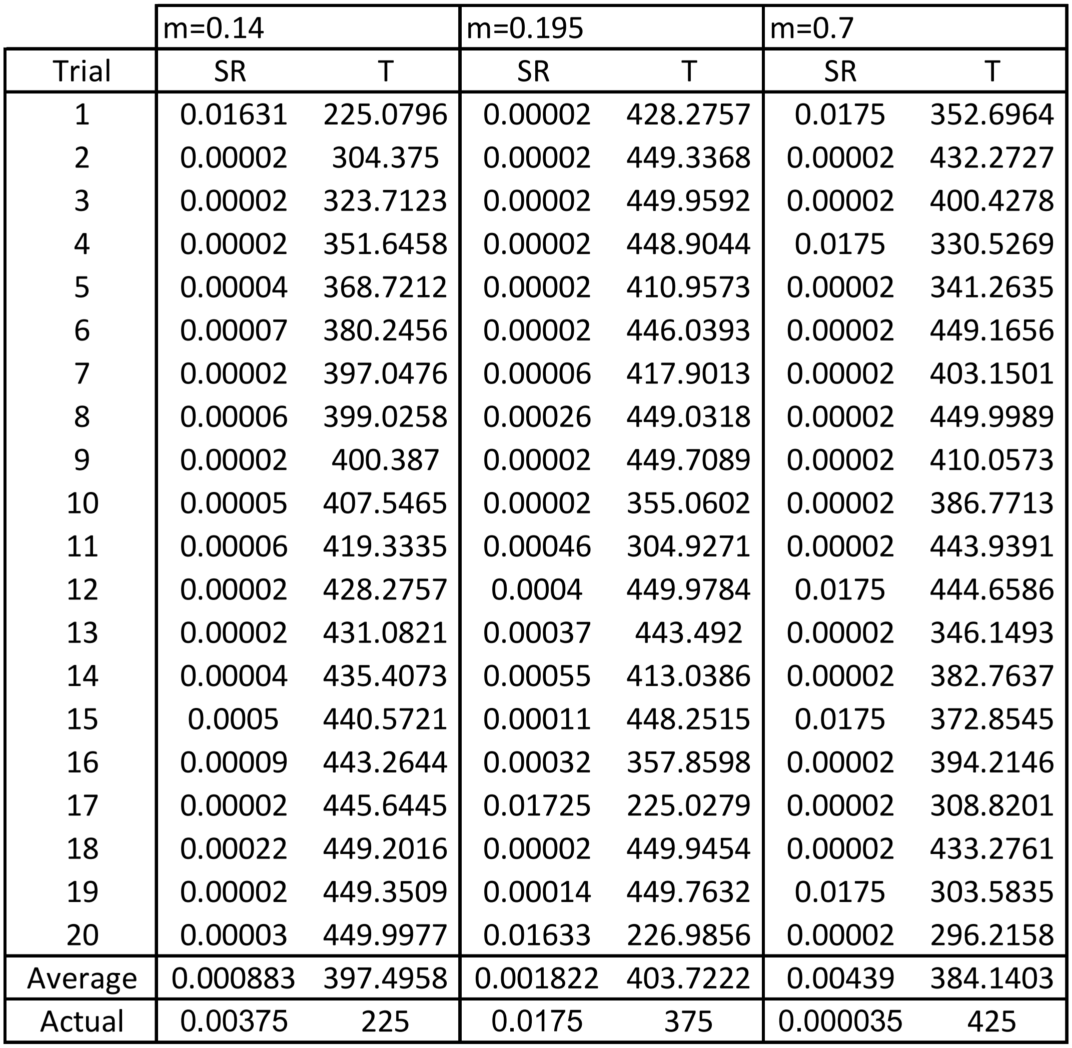
```
// Weights saved from breadboard .
// Saved after epoch 896, exemplar 0.
#NSWeightFileVersion 243
#inputFile File
2
1.0294538175579068e+002 -9.0154418072633702e-001
8.0000000000000002e-003 -2.7000000000000002e+000
0
#desiredFile File
1
3.0303030303030307e+000 -1.2393939393939397e+000
0
#inputAxon Axon
2 1
1
0
#hidden1Axon TanhAxon
4 1
1
4 |-1.4714826757048152e-001 1.8566003949599590e+001 4.0451310313984962e-001 3.2170257326593205e-001
#criterion L2Criterion
1 1
1
0

#outputAxon BiasAxon
1 1
1
1 5.3207197933439060e-003
#hidden1Synapse FullSynapse
8 -1.0699644917176814e+000 -2.1516284012714602e+000 2.0904238708246677e+001 -5.8348788726080625e-001 1.9592220611349096e+000
2.5058635108528979e+000 1.6597604828325638e+000 2.3436150297746878e+000
#hidden1SynapseBackprop BackFullSynapse
8 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
#outputSynapse FullSynapse
4 -8.8048089433832928e+000 -6.0744492664196426e-001 1.6199333853899081e+001 -2.4822302486943450e+001
#outputSynapseBackprop BackFullSynapse
4 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000 0.0000000000000000e+000
#desiredViewer MatrixViewer
1
3.0303030303030307e+000 -1.2393939393939397e+000
#CVDesiredViewer MatrixViewer
1
3.0303030303030307e+000 -1.2393939393939397e+000
#outputDesiredDataGraph DataGraph
1
3.0303030303030307e+000 -1.2393939393939397e+000
#CVOutputDesiredDataGraph DataGraph
1
3.0303030303030307e+000 -1.2393939393939397e+000
#control StaticControl
0
1.0000000000000000e+009
```



```
2
20037624
0
0
0
1.0000000000000000e-002
1.0000000000000000e+010
1.4400158499717577e-003
17
```

APPENDIX II

GA Results from Problem 1 using Strategy 2:



	m=0.14		m=0.195		m=0.7	
Trial	SR	T	SR	T	SR	T
1	0.01631	225.0796	0.00002	428.2757	0.0175	352.6964
2	0.00002	304.375	0.00002	449.3368	0.00002	432.2727
3	0.00002	323.7123	0.00002	449.9592	0.00002	400.4278
4	0.00002	351.6458	0.00002	448.9044	0.0175	330.5269
5	0.00004	368.7212	0.00002	410.9573	0.00002	341.2635
6	0.00007	380.2456	0.00002	446.0393	0.00002	449.1656
7	0.00002	397.0476	0.00006	417.9013	0.00002	403.1501
8	0.00006	399.0258	0.00026	449.0318	0.00002	449.9989
9	0.00002	400.387	0.00002	449.7089	0.00002	410.0573
10	0.00005	407.5465	0.00002	355.0602	0.00002	386.7713
11	0.00006	419.3335	0.00046	304.9271	0.00002	443.9391
12	0.00002	428.2757	0.0004	449.9784	0.0175	444.6586
13	0.00002	431.0821	0.00037	443.492	0.00002	346.1493
14	0.00004	435.4073	0.00055	413.0386	0.00002	382.7637
15	0.0005	440.5721	0.00011	448.2515	0.0175	372.8545
16	0.00009	443.2644	0.00032	357.8598	0.00002	394.2146
17	0.00002	445.6445	0.01725	225.0279	0.00002	308.8201
18	0.00022	449.2016	0.00002	449.9454	0.00002	433.2761
19	0.00002	449.3509	0.00014	449.7632	0.0175	303.5835
20	0.00003	449.9977	0.01633	226.9856	0.00002	296.2158
Average	0.000883	397.4958	0.001822	403.7222	0.00439	384.1403
Actual	0.00375	225	0.0175	375	0.000035	425

	m=0.304		m=0.2		m=0.54	
Trial	SR	T	SR	T	SR	T
1	0.00015	225.021	0.00002	362.2633	0.00002	443.0425
2	0.00039	438.0816	0.00002	445.2871	0.00075	449.5241
3	0.00007	449.1579	0.00001	447.909	0.00002	298.7531
4	0.00032	388.3095	0.00005	419.0644	0.00004	449.9983
5	0.00002	369.3845	0.00126	391.3193	0.00031	448.634
6	0.00002	448.7909	0.00002	443.6847	0.00005	448.2657
7	0.00007	446.4957	0.00002	435.5542	0.00004	437.9275
8	0.0001	444.2508	0.00002	404.7272	0.00001	438.5166
9	0.00002	444.9552	0.00004	359.466	0.00002	435.3308
10	0.00002	438.7029	0.00002	446.3157	0.00002	293.0021
11	0.00004	449.2372	0.00002	449.8764	0.00041	448.7764
12	0.0002	445.0237	0.00007	448.9482	0.00019	449.9975
13	0.01748	260.0993	0.00008	225.0743	0.00002	254.1494
14	0.00002	395.4827	0.00134	448.9146	0.00022	402.2176
15	0.00003	438.5407	0.0002	341.5551	0.00003	444.4388
16	0.00002	306.8473	0.00004	449.6852	0.00002	445.3001
17	0.00021	449.3605	0.00002	437.84	0.00001	440.7581
18	0.00003	449.3216	0.00002	429.3754	0.00051	393.6114
19	0.00067	446.9618	0.00013	447.5772	0.00056	448.3694
20	0.00002	432.6523	0.00002	435.3358	0.00002	445.6911
Average	0.000995	408.3339	0.000171	413.4887	0.000164	415.8152
Actual	0.00375	450	0.0075	350	0.00035	400

GA Results from Problem 1 using Strategy 3:

	m=0.2		m=0.7		m=0.195	
	SR	T	SR	T	SR	T
Trial 1	0.00004	430.6504	0.00002	345.5316	0.00002	437.0077
Trial 2	0.00003	426.4398	0.00005	441.2532	0.00005	444.455
Trial 3	0.00005	427.1583	0.00002	437.1593	0.00002	437.9822
Trial 4	0.00002	442.8729	0.01734	230.3529	0.00002	426.3098
Trial 5	0.00004	443.7792	0.00004	430.944	0.00004	422.7049
Trial 6	0.00165	448.5861	0.00002	449.7272	0.00002	445.7304
Trial 7	0.00002	436.2172	0.00002	303.3709	0.01748	226.2558
Trial 8	0.00002	440.7572	0.00005	441.0649	0.00002	449.9991
Trial 9	0.00002	447.8623	0.00008	439.884	0.00004	342.2148
Trial 10	0.00005	413.652	0.00003	449.8721	0.00006	449.9663
Trial 11	0.00001	449.6337	0.00002	449.4864	0.00011	419.2168
Trial 12	0.00003	225.115	0.00021	404.0005	0.00002	446.8349
Trial 13	0.01584	411.7664	0.00002	340.9164	0.00002	439.4591
Trial 14	0.00015	444.9511	0.01571	225.9177	0.00008	446.1457
Trial 15	0.00002	449.3738	0.00002	380.8889	0.00257	442.7468
Trial 16	0.0002	448.3926	0.01741	449.693	0.00002	446.0248
Trial 17	0.00002	449.7865	0.00003	378.4982	0.00005	449.2546
Trial 18	0.00029	393.5369	0.00023	443.4382	0.01712	262.0372
Trial 19	0.00002	449.984	0.00032	449.319	0.00149	449.9567
Trial 20	0.00094	438.6393	0.01625	409.9016	0.00004	416.2045
Trial 21	0.00002	442.8783	0.00002	436.2718	0.00002	442.0203
Trial 22	0.00002	312.5404	0.00001	433.5319	0.00002	449.8914
Trial 23	0.00002	433.6628	0.00004	449.3329	0.00004	430.8943
Trial 24	0.00002	449.5669	0.00005	449.9147	0.00003	447.151
Trial 25	0.0174	250.2052	0.00002	449.7502	0.00003	447.5728
Trial 26	0.00037	443.2658	0.00002	404.9225	0.01745	444.6479
Trial 27	0.00003	448.8032	0.00002	386.3574	0.00002	423.0024
Trial 28	0.00003	449.5013	0.00003	445.582	0.00005	445.8708
Trial 29	0.00002	444.0454	0.00008	442.9106	0.00002	403.1001
Trial 30	0.00008	449.7949	0.00201	231.8712	0.00002	443.0923
Trial 31	0.00055	443.4954	0.00002	353.2835	0.00002	435.2026
Trial 32	0.00002	391.3625	0.00002	374.777	0.00002	375.0012
Trial 33	0.01735	449.9443	0.00004	449.5552	0.00207	226.0529
Trial 34	0.00048	442.247	0.00048	449.4306	0.00042	435.3977
Trial 35	0.00005	428.2871	0.00005	435.9785	0.00019	424.0173
Trial 36	0.00002	449.9466	0.00005	393.0448	0.00003	449.8872
Trial 37	0.00002	424.4175	0.00002	449.342	0.00003	360.1835
Trial 38	0.0001	448.5243	0.00002	449.9966	0.00007	449.4759
Trial 39	0.00005	449.5967	0.00003	414.0121	0.00002	269.2925
Trial 40	0.00002	449.757	0.01726	229.2119	0.0002	443.6475
Trial 41	0.00002	449.8778	0.00002	416.7897	0.00002	449.9937
Trial 42	0.00005	382.6349	0.00003	449.8141	0.00002	446.1036
Trial 43	0.00011	449.9026	0.00002	447.1135	0.00013	445.3974
Trial 44	0.00002	423.7458	0.00002	443.6371	0.00002	449.2558
Trial 45	0.01744	225.2734	0.01735	225.5271	0.00024	449.1785
Trial 46	0.00002	421.7117	0.00007	365.7406	0.00002	410.1862
Trial 47	0.00002	415.8436	0.00013	449.5003	0.00002	447.6991
Trial 48	0.00002	449.325	0.01747	225.2336	0.01736	229.5344
Trial 49	0.00005	448.6066	0.00002	361.4775	0.00007	429.2197
Trial 50	0.00003	358.7793	0.00042	419.0126	0.00005	449.9512
Average	0.001478	420.9339	0.002475	396.6829	0.00156	415.0486
Actual	0.0075	350	0.000035	425	0.0175	375

	m=0.304		m=0.14		m=0.54	
	SR	T	SR	T	SR	T
Trial 1	0.00021	436.2064	0.00029	434.4437	0.00002	443.0818
Trial 2	0.00004	429.3591	0.00006	441.3638	0.00002	443.2977
Trial 3	0.00267	448.4805	0.00013	449.8012	0.00012	378.7243
Trial 4	0.00005	430.7244	0.00008	449.0495	0.00004	421.5831
Trial 5	0.00002	444.0592	0.00008	385.3709	0.00009	431.8546
Trial 6	0.00005	401.5406	0.00004	441.9221	0.00002	439.8508
Trial 7	0.00002	449.8243	0.00002	433.0627	0.00004	384.3332
Trial 8	0.00013	421.0245	0.00002	445.7575	0.00002	227.8881
Trial 9	0.00029	448.9975	0.00005	446.1227	0.00003	414.1627
Trial 10	0.00003	446.8052	0.00002	449.969	0.00003	449.9342
Trial 11	0.00002	336.3405	0.00121	449.819	0.01742	225.2764
Trial 12	0.00006	403.4891	0.01721	286.1481	0.00028	447.5662
Trial 13	0.00002	410.905	0.00004	448.4657	0.00002	447.1054
Trial 14	0.00002	448.3122	0.0003	449.994	0.00001	449.8122
Trial 15	0.00002	425.4038	0.00002	368.9324	0.00007	449.7799
Trial 16	0.00003	449.9503	0.00002	449.8023	0.00034	415.8289
Trial 17	0.00002	435.6053	0.01632	254.4224	0.00002	361.3482
Trial 18	0.00082	226.0666	0.00002	426.9629	0.00002	449.0369
Trial 19	0.00003	374.2643	0.00003	442.9463	0.00002	449.9734
Trial 20	0.00064	408.3079	0.00019	439.3564	0.00002	375.6875
Trial 21	0.00002	444.0064	0.00006	449.0873	0.00012	444.9424
Trial 22	0.0001	413.1855	0.00004	449.9993	0.00015	449.2363
Trial 23	0.00002	376.1961	0.00014	447.1258	0.00033	407.2452
Trial 24	0.00016	449.8672	0.00002	447.4184	0.00004	438.8346
Trial 25	0.00003	449.982	0.01716	225.1254	0.00002	344.3019
Trial 26	0.00002	432.7038	0.01053	353.0398	0.00003	445.6961
Trial 27	0.00002	448.3292	0.00002	419.1706	0.00014	449.7622
Trial 28	0.00008	412.1086	0.00005	443.98	0.00066	448.037
Trial 29	0.00011	377.9749	0.00007	449.0362	0.00006	413.2902
Trial 30	0.00003	437.7718	0.00002	449.0377	0.00002	410.4757
Trial 31	0.00003	449.8951	0.00005	445.816	0.00124	448.6352
Trial 32	0.00636	248.2501	0.00002	289.0735	0.00004	449.7572
Trial 33	0.00033	438.1701	0.00005	431.5263	0.00005	360.3479
Trial 34	0.01558	255.9067	0.00002	423.1748	0.00003	448.6897
Trial 35	0.00024	437.4805	0.00003	449.992	0.00004	449.9285
Trial 36	0.00004	448.1317	0.00002	426.9008	0.00002	414.5187
Trial 37	0.00002	449.9598	0.00012	448.4353	0.00002	432.9832
Trial 38	0.00002	434.7614	0.00017	427.3637	0.00002	449.4574
Trial 39	0.00004	311.5283	0.01462	228.3145	0.00002	447.9749
Trial 40	0.00002	441.4259	0.00003	447.9649	0.01743	235.517
Trial 41	0.00002	347.3537	0.00002	449.9695	0.00002	449.8977
Trial 42	0.00003	304.3679	0.00012	445.3544	0.01634	241.0427
Trial 43	0.00028	346.58	0.00007	447.0919	0.00004	449.8473
Trial 44	0.00003	444.7796	0.00002	440.8594	0.00004	449.9366
Trial 45	0.00002	442.2124	0.01717	225.4527	0.00011	386.7781
Trial 46	0.00002	447.7293	0.00002	438.8835	0.00437	226.8161
Trial 47	0.00002	449.9228	0.00017	440.1266	0.00013	436.1576
Trial 48	0.00002	404.971	0.00002	449.3426	0.00012	449.9958
Trial 49	0.01353	243.3575	0.00002	420.6423	0.00061	444.107
Trial 50	0.00002	434.426	0.00003	438.0274	0.00003	391.8147
Average	0.000849	405.98	0.001941	414.8209	0.001219	408.443
Actual	0.00375	450	0.00375	225	0.00035	400

GA Results for Problem 1 using strategy 4:

	m=0.14		m=0.2		m=0.195	
Trial	SR	Temp	SR	Temp	SR	Temp
1	0.00009	277.673	0.00017	395.4563	0.00049	225.6732
2	0.00026	237.7767	0.00059	233.0007	0.00073	338.0921
3	0.00039	308.7622	0.0008	245.04	0.00159	364.0583
4	0.00135	417.572	0.00101	227.2374	0.00207	230.2629
5	0.00159	424.8877	0.00151	431.4439	0.00497	306.8274
6	0.00449	346.3549	0.00457	229.4597	0.00522	440.3328
7	0.00656	440.6096	0.00708	332.312	0.00708	358.9652
8	0.00866	308.7936	0.00749	275.6735	0.00762	277.4538
9	0.00944	414.2815	0.00811	430.3569	0.00954	394.3941
10	0.00965	323.5482	0.00998	400.0518	0.00958	245.0889
Actual	0.00375	225	0.0075	350	0.0175	375

	m=0.54		m=0.7	
Trial	SR	Temp	SR	Temp
1	0.00003	394.3101	0.00009	277.673
2	0.0005	239.8838	0.00026	237.7767
3	0.00065	390.9645	0.00039	308.7622
4	0.00065	263.7095	0.00135	417.572
5	0.00102	257.7002	0.00159	424.8877
6	0.0029	249.6448	0.00449	346.3549
7	0.00402	341.5794	0.00656	440.6096
8	0.00531	384.732	0.00866	308.7936
9	0.00653	306.6556	0.00944	414.2815
10	0.00998	443.1387	0.00965	323.5482
Actual	0.00035	400	0.000035	425

References

1. Abu-Farha F. K., Deshmukh P. V., Khraisheh M. K., and Thuramalla N. V., (2003), "*Superplastic Forming: Stretching the Limits of Fabricating Medical Devices and Implants*", Proceedings of the ASM Materials & Processes for Medical Devices Conference, Anaheim, CA, ASM International Publications, pp 368-373.

2. Abu-Farha, F. K.,(2007), "Integrated Approach to the Superplastic Forming of Magnesium Alloys", PhD Dissertation.

3. Abu-Farha, F. K., Khraisheh, M.K., (2007), "Analysis of Superplastic Deformation of AZ31 Magnesium alloy" *Advanced Engineering Materials*, Vol.9, pp 777-783.

4. Aizawa, T., (1999), "Superplastic Forging Analysis by the Multi-level Modeling", *Materials Science Forum*, vol.304, pp 621-630.

5. Bariani, P. F., Berti, G., Bruschi, S., Dal, D. T., (2001), "Application of Neural Networks to Represent the Rheological Behavior of Nickel Based Superalloys Under Varying Hot Deformation Conditions" *Proceedings of the 4th International ESAFORM Conference on Material Forming*,pp 565-568

6. Barnes, A. J. , (2007), "Superplastic Forming 40 Years and Still Growing", *Journal of Materials Engineering and Performance*, vol. 16, no. 4, pp. 440-454.

7. Beliganur, N. K., (2007) "Application of Genetic Algorithms and CFD for Flow Control Optimization", Master's Thesis.

8. Bruschi, S., Dal Negro, T., (2005), "Hybrid Modeling to Represent the Material Flow Stress Under Hot Deformation Conditions" *ESAFORM 2005 Conference*, pp 37-40.

9. Casotto, S., Bruschi, S., Pascon, F., Habraken, A. M., (2005), "Development of a Neural Network to Predict the Final Geometry of Forged Rings After Cooling" *Proceedings of ESAFORM Conference*, pp 855-858.

10. Chen, D., Li, M., Wu, S., (2003), "Modeling of Microstructure and Constitutive Relation During Superplastic Deformation by Fuzzy - Neural Network." *Journal of Materials Processing Technology*, Vol.142, pp 197-202.

11. Chandra N., (1988), "Physical Modeling of Superplastic Forming", *Journal of Metals*, Vol.40, no.11.

12. Chandra, N., Rama, S.C. , Chen, Z., (1999), "Critical Issues in the Industrial Application of SPF-process Modeling and Design Practices", *Materials Transaction*, JIM, vol. 40, no. 8, pp 723-736.

13. Chandra, N., Rama, S. C., Chen, Z., (1999), "Critical Issues in the Industrial Application of SPF- Process Modeling and Design Practices", *Materials Transaction*, JIM, vol. 40, no. 8, pp. 723-736.

14. Cook D. F., Ragsdale C. T., Major, R. L., (2000), "Combining a Neural Network with Genetic Algorithm for Process Parameter Optimization", Engineering Applications of Artificial Intelligence, vol. 13, no. 4, pp 391-396.

15. Curtis R., "Face on" , *Materials World*, August 2005,pp 20-22.

16. Deb, K., Karthik, S., Okabe, T., (2007), "Self-Adaptive Simulated Binary Crossover for Real-Parameter Optimization" *GECCO'07*, pp 1187-1194.

17. Fausett L., (1994), “Fundamentals of Neural Networks” Prentice Hall, New Jersey.

18. Goldberg, D. E., "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addision-Wisley Company Inc: Reading, MA, 1989.

19. Grimes R., “Extended Brief”, *Materials World*, July 2005, pp 36-39.

20. Hajela, P., Lin, C. Y., (2000), "Real versus Binary Coding in Genetic Algorithms: A Comparative Study, in Computational Engineering Using Metaphors from Nature”, by Topping B. H. V, p. 77-83.

21. Hambli, R., Potiron, A., (2001), “Comparison of 2D and 3D Numerical Modeling of Superplastic Forming Processes” *Computer methods in applied mechanics and engineering*, Vol. 190, pp 4871-4888.

22. Hassan, O., Kurt, A., Arcaklioglu, E., (2007), “Artificial Neural network Applications to the Friction Stir Welding of Aluminum Plates” *Journal of Materials and Design*, Vol. 28, pp 78-84.

23. Holland, J. H., “*Adaptation in Natural and Artificial Systems*”, University of Michigan Press, Ann Arbor, 1975

24. Holter, T., Yao, X. Q., Rabelo, L. C., Jones, A., Yih, Y. W., (1995), “Integration of Neural Networks and Genetic Algorithms for an Intelligent Manufacturing Controller”, *Computers and Industrial Engineering*, vo.29, pp 211-215.

25. Hsiang, S. H., Kuo, J. L., (2005), “Applying ANN to Predict the Forming Load and Mechanical Property of Magnesium Alloy under hot extrusion”, *International Journal of Advanced Manufacturing Technology*, Vol.26, pp 970-977.

26. Hsiang, S. H., Kuo, J. L., Yang, F. Y., (2006), "Using Artificial Neural Networks to Investigate the Influence of Temperature on Hot Extrusion of AZ61 Magnesium Alloy" *Journal of Intelligent Manufacturing* , Vo. 17, pp 191-201.
27. Hsiang, S. H., Kuo, J. L., Yang, F. Y., (2006), "Using Artificial Neural Networks to Investigate the Influence of Temperature on Hot Extrusion of AZ61 Magnesium Alloy" *Journal of Intelligent Manufacturing* , Vo. 17, pp 191-201.
28. Jamal, S-A., Twomey, J., (2007), "ANN Constitutive Model for High Strain-rate Deformation of Al 7075-T6", *Journal of Materials Processing Technology*, vol.186, pp 339-345.
29. Javad, A. A., Tan, T. P., Zhang, M., (2003), "Neural Network for Constitutive Modeling in Finite Element Analysis" *Computer Assisted Mechanics and Engineering Sciences*, vo. 10, pp 523-529.
30. Jim K., (2007), "BMW's 3.0 Litre Engine Magnesium/Aluminium Composite Engine Block" *Canadian Driver*.
31. Jose. C. P., Neil R. E., Curt W. L., (1999), "Neural And Adaptive Systems: Fundamentals through Simulations", John Wilet & Sons, Inc. New York.
32. Kaifeng, Z., Qingyun, Z., Changwen, W., (1995), "Simulation of Superplastic Sheet Forming and Bulk Forming", *Journal of Materials Processing Technology*, vol. 55, no. 1, pp. 24-27.
33. Kanpur Genetic Algorithm Laboratory (KANgal), IIT Kanpur, India.
34. Klocke, F., Breuer, D., (2006), "Using the Finite Element Method and Artificial Neural Networks to Predict Ductile Fracture in Cold Forming Processes", *Proceedings AIP*.

35. Khraisheh, M. K., Zbib, H. M., (1999), “Optimum Forming Loading Paths for Pb-Sn Superplastic Sheet Materials”, *Journal of Engineering Materials and Technology*, Vol. 121.

36. Kohonen, T., (1988), “An Introduction to Neural Computing”, Neural Networks, vol. 1, pp. 3-16.

37. Kong, L. X., Hodgson, P. D., (1999), “Application of Constitutive and Artificial Neural Network Models to Predict the Hot Strength of Steels” *ISIJ International*, vol. 39, no. 10, pp 991-998.

38. Lefik, M., Schrefler, B. A., (2003), “Artificial Neural Network as an Incremental Non-linear Constitutive Model for a Finite Element Code” *Computer methods in applied mechanics and engineering*, vol. 192, pp 3265-3283.

39. Li, G.Y. , Tan, M. J. , Liew, K. M. , (2004),“Three-dimensional Modeling and Simulation of Superplastic Forming” *Journal of Materials Processing Technology*, vol. 150, pp 76-83.

40. Lin, W., Lee, T.C., (2006), “Prediction of Limiting Dome Height Using Neural Network and Finite Element Method”, *International Journal of Advanced Manufacturing Technology*, vo. 27, pp 1082-1088.

41. Lin, Q-Q., Peng, D-S., Zhu, Yuan, Z., (2005), “Establishment of Constitutive Relationship Model for 2519 Aluminum Alloy Based on BP Artificial Neural Network” *Journal of Central South University of Technology*, Vol. 12, no.4, pp 380-384.

42. Mok, S. L., Kwong, C. K., Lau, W.S., (2001), “A Hybrid Neural Network and Genetic Algorithm Approach to the Determination of Initial Process Parameters

for Injection Molding", *International Journal of Advanced Manufacturing Technology*, Vol. 18, pp 404-409.

43. Nazzal, M. A., Khraisheh, M. K., Darras, B. M. , (2004), "Finite Element Modeling and Optimization of Superplastic Forming Using Variable Strain Rate Approach" *Journal of Materials Engineering and Performance*, Vol 13, no.6, pp 691-699.
44. Nazzal, M. A., Khraisheh, M. K., Abu-Farha, F.K., (2007), "The Effect of Strain Rate Sensitivity Evolution on Deformation Stability During Superplasticity Forming." *Journal of Materials Processing Technology*, Vol. 191, pp 189-192.
45. NeuroSolutions 5 help
46. Onwubolu, G. C. (1999), "Manufacturing Features Recognition Using Back propagation Neural Networks", Journal of Intelligent Manufacturing, vol. 10, no. 3-4, pp. 289-299.
47. Osakada, K., Yang, G., (1991), "Application of Neural Networks to and Expert System for Cold Forging", *International Journal of Machine Tools and Manufacture*, vol. 31, no. 4, pp. 577-587.
48. Pathak, K.K., Panthi, S., Ramakrishnan, N., (2005), "Application of Neural Network in Sheet Metal Bending Process", *Defense Science Journal*, Vol. 55, no.2, pp 125-131.
49. Pavan Kumar, A., Karthik, S., Muttukaruppan, A., Battacharya, S.S., (2003), "Development of an ANN Controlled Genetic Algorithm for the Numerical Validation of a Model for Superplastic Forming." *Proceedings of the 21st IASTED Conference*, Applied Informatics, February.

50. Penumadu. D., Agrawal. G., Chameau. J. L., (1992), “Knowledge – Based Modeling of Material Behavior with Neural Networks”, *Journal of Engineering Mechanics*, vol.118, pp 1057-1059.

51. Pilling, J., Ridley, N., (1989), “Superplasticity in Crystalline Solids”, The Institute of metals, London, UK.

52. Porro R., Beatrice P., “The Importance of Weight Reduction for the Automobile Industry – Fiat Auto’s Experience in the use of Magnesium” *Proceedings of the Third International Magnesium Conference*, pp 167-176.

53. Sadeghi R., Pursell Z., (1997), “Finite Element Modeling of Superplastic Forming Using Analytical Contact Surfaces”, *Materials Science Forum*, vol. 243-245, pp 719-728.

54. Sathyanarayana, G., Joseph Lin, I., Chen, M-K., (1992), “Neural Network Modeling and Multi - objective Optimization of Creep Feed Grinding of Superalloys” *International Journal of Production Research*, vol.30, pp 2421-2438.

55. Singh, S. K., Kumar, D. R., (2004), “Application of a Neural Network to Predict Thickness Strains and Finite Element Simulation of Hydro-Mechanical Deep Drawing” *International Journal of Materials and Product Technology*, Vol. 21, pp 186-199.

56. Tikkaya, A. E., (2000), “State-of-the-art of Simulation of Sheet Metal Forming”, *Journal of Materials Processing Technology*, vol. 103, no. 1, pp. 14-22.

57. Vasin, R. A., Enikeev, F. U. , Safiullin, R. V., (1999), "Mathematical Modeling of Superplastic Forming of a Long Rectangular Box Section", *Materials Science Forum*, vol. 304, pp 765-770.

58. Xing, H. L., Wang, S., Makinouchi, A., (1993), "An Adaptive Mesh h-refinement Algorithm for the Finite Element Modeling of Sheet Forming" *Journal of Materials Processing Technology*, vol.91, pp 183-190.

59. Za'er Salem Abo-Hammour, "Advanced Continuous Genetic Algorithms and their Applications in the Motion Planning of Robot Manipulators and in the Numerical Solution of Boundary Value Problems", PhD Thesis, Pakistan Institute of Engineering and Applied Sciences, Quaid-i-Azam University, 2002.

60. Zeidenberg, M., (1990), "Neural Networks in Artificial Intelligence" Ellis Horwood, New York.

61. Zhang, H.C. , Huang, S. H., (1995), "Applications of Neural Networks in Manufacturing: a state-of-the-art survey" *International Journal of Production Research*, vol. 33, no. 3, pp 705-728.

62. "Applying the Perception to Three-Dimensional Feature Recognition" *Journal of design and manufacturing*, Vol.2, n 4, 187-198.

63. http://www.canadiandriver.com/articles/jk/051123 , Referred 10/2007

64. http://www.mse.mtu.edu/~drjohn/sp/intro/index.html, Referred 10/2007

65. http://cse.stanford.edu/class/sophomore-college/projects-00/neural-networks/, Referred 10/2007

66. http://www.superform-aluminium.com/, Referred 10/2007

VITA

Kirthi Bedida was born on January 14th, 1984 in Hyderabad, Andhra Pradesh, India. She received her bachelors' degree from Osmania University, Andhra Pradesh, India in June of 2005. She joined the University of Kentucky in the Fall of 2005 to pursue her MS in Mechanical Engineering and is expecting her degree in December of 2007. As a graduate student she served as a *Teaching Assistant* for MFS 512 (Manufacturing Systems) and ME101 (Introduction to Mechanical Engineering), and also as a *Research Assistant* at the UK AMFMP lab.

Scholastic Honors

Kentucky Graduate Scholarship, Fall 2005 - Spring 2007
University of Kentucky, Lexington, KY.

Papers and Conferences

Fazleena Badurdeen, Kirthi Bedida, Gursel Suer, Nishantha Dissanayake, "Capacitated Lot Sizing Problem Using Adaptive Multi-chromosome Crossover Strategy", *Proceedings of the Industrial Engineering research conference,* Nashville, Tennessee, May 2007.

Haritha Metta, Kirthi Bedida, Fazleena Badurdeen, Phil Marksberry, "From Maintenance to Sustainable Asset Management", Submitted to the *Journal of Quality in Maintenance Engineering.*

Haritha Metta, Kirthi Bedida, Fazleena Badurdeen, Phil Marksberry, "Total Productive Maintenance for a Sustainable Enterprise: Status and Future Directions", Selected for presentation, *IMECE 07*, Seattle, Washington, November 2007.

Kirthi Bedida, Haritha Metta, “Application of Meta-Heuristic methods to solve complex manufacturing problems,” Poster Presentation, *UK GS conference*, Lexington, KY, April 2007.

Kirthi Bedida, “Neural Network Modeling of the behavior of Mg AZ31”, Poster Presentation, *Annual TMS conference,* Orlando, Florida, Feb 2007.

Kirthi Bedida, “Artificial Intelligence in Manufacturing: A comparison of different neural network training algorithms to model superplastic forming,” *UK GS conference*, Lexington, KY, April 2007.

Kirthi Bedida, M.Lakshmi Sravanth, “*Jatropha carcus*, as a potential bio-diesel” *GMED’04*, Deccan College of Engineering, Hyderabad, India.