DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

NEURAL NETWORKS APPLICATIONS IN PARAMETER SETTING OF TUBE HYDROFORMING AND METAL CUTTING PROCESSES

by

Sezgi ÖZEN

August, 2011

İZMİR

NEURAL NETWORKS APPLICATIONS IN PARAMETER SETTING OF TUBE HYDROFORMING AND METAL CUTTING PROCESSES

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of Requirements for the Degree of Doctor of Philosophy
in Industrial Engineering, Industrial Engineering Program

by
Sezgi ÖZEN

August, 2011
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"NEURAL NETWORKS APPLICATIONS IN PARAMETER SETTING OF TUBE HYDROFORMING AND METAL CUTTING PROCESSES"** completed by **SEZGİ ÖZEN** under supervision of **Prof. Dr. G. MİRAÇ BAYHAN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. G. Miraç BAYHAN

Supervisor

Prof. Dr. Nihat BADEM

Thesis Committee Member

Asst. Prof. Güleser KALAYCI DEMİR

Thesis Committee Member

Doç. Dr. Aştımır GÜNGÖR

Examining Committee Member

Yrd.Doç.Dr. Gonca TUNÇEL

Examining Committee Member

Prof.Dr. Mustafa SABUNCU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost I would like to express my deepest gratitude and thanks to my supervisor Prof. Dr. G. Miraç Bayhan for her support, patience, friendly encouragement and valuable advice throughout the progress of this dissertation. Her guidance helped me in all the time of research and writing of this thesis.

I would also like to thank to the members of my Thesis Progress Committee Prof. Dr. Nihat Badem and Asst. Prof. Dr. Güleser Kalaycı Demir for their helpful comments and suggestions. Their insights improved this work.

I would like to express my thanks to all the professors and colleagues in the Department of Industrial Engineering for their support.

I would like to express my special thanks to my friend Seval Askar for helping me get through the difficult times, and for all the emotional support, entertainment, and caring she provided. Last and most importantly; I am extremely grateful to my parents, Aydın Özen and Nevcihan Özen for their love, confidence, encouragement and endless support in my whole life. To them I dedicate this thesis.

Sezgi ÖZEN

NEURAL NETWORKS APPLICATIONS IN PARAMETER SETTING OF TUBE HYDROFORMING AND METAL CUTTING PROCESSES

ABSTRACT

The objective of this research is to test and conclude about the efficiency of ANNs for optimization of manufacturing processes. For this purpose, ANN based methods are proposed to deal with two different manufacturing processes. The first problem is the tube hydroforming process with two conflicting objectives. A two-stage neural network approach and a hybridization of ANNs with genetic algorithm are proposed for the solution of the problem. Simulation outcomes of the proposed approaches are compared with Taguchi approach. The results show that ANNs need to be integrated by other solution techniques since combining neural networks with genetic algorithm provide the best process performance for tube hydroforming process under consideration. The second problem is the process parameters optimization of a metal cutting process with unit cost minimization. The original integer programming model of the problem given in the literature is used to construct the energy function by using penalty approach. For this problem, a maximum and continuous neural network interacting with each other are proposed. The results are compared with optimum results of dynamic programming, integer programming and non-linear programming. The results show that neural networks are an effective alternative to operation research techniques and combining Hopfield-type networks with penalty approach gives the advantage of obtaining optimal solution in an extremely large solution space within a reasonable computation time.

The contribution of this thesis is two fold. One is to the manufacturing process literature as this study is the first attempt to solve the parameter optimization problems of the manufacturing processes under consideration. This thesis also makes contribution to ANN literature as combining ANNS with different techniques for optimization of manufacturing processes.

Keywords: Neural networks, Manufacturing process, Parameter optimization

TÜP ŞEKİLLENDİRME VE METAL İŞLEME PROSESLERİNDE PARAMETRE BELİRLEME İÇİN SİNİR AĞI UYGULAMALARI

ÖZ

Bu çalışmanın amacı yapay sinir ağlarının üretim proseslerinin optimizasyonu için kullanılabilirliğini test etmektir. Bu amaç için iki üretim prosesine sinir ağı tabanlı çözüm yöntemleri önerilmiştir. Dikkate alınan ilk problem çelişen iki amaca sahip tüp şekillendirme prosesidir. Problemin çözümü için iki aşamalı yapay sinir ağı ve melez bir yapay sinir ağı-genetik algoritma yaklaşımı önerilmiştir. Önerilen yaklaşım kullanılarak elde edilen sonuçlar, Taguchi yaklaşımı ile karşılaştırılmıştır. Elde edilen sonuçlara göre yapay sinir ağlarının diğer çözüm yöntemleri ile birlikte kullanılması tüp şekillendirme prosesinin performansında daha iyi gelişmeler sağlamaktadır. İkinci problem, amacı birim maliyeti minimize etmek olan bir metal işleme prosesinin optimizasyonudur. Enerji fonksiyonunu oluşturmak için penaltı yaklaşımı kullanılmış ve problemin çözümü için literatürde önerilmiş tamsayı formülasyonu kullanılmıştır. Bu problem için, birbirini etkileyen bir maksimum ve bir sürekli sinir ağı modeli önerilmiştir. Önerilen yaklaşım bir üretim prosesinin optimizasyonu probleminde test edilmiş ve sonuçlar dinamik programlama, tamsayı programlama ve doğrusal olmayan programlama ile kıyaslanmıştır. Elde edilen sonuçlara göre yapay sinir ağları metal işleme prosesleri için yöneylem tekniklerine karşı etkin bir alternatif oluşturmaktadır ve Hopfield türevi ağların penaltı yaklaşımı ile birlikte kullanılması sonucu çok geniş çözüm uzayı içerisinde son derece kısa sürelerde istenilen sonuçlara ulaşma avantajı sağlamaktadır.

Bu çalışmanın sağladığı katkılar iki yönlüdür. Bunlardan biri, dikkate alınan prosesler için ilk yapay sinir ağları uygulaması olması sebebi ile üretim prosesleri literatüre sağlanan katkıdır. Bu çalışma ile sağlanan bir diğer katkı ise, yapay sinir ağlarının üretim proseslerinde kullanımı için başka çözüm yöntem yöntemleri ile birlikte kullanılması ile yapay sinir ağları literatürüne sağlanmıştır.

Anahtar sözcükler: Sinir ağları, Üretim prosesi, Parametre optimizasyonu

CONTENTS

Page

CHAPTER ONE
INTRODUCTION

In this chapter, the background, motivation and objectives of this work are stated, and the organization of this dissertation is outlined.

1.1 Background and Motivation

Quality and productivity are essential factors for achieving success. A properly designed manufacturing process can significantly affect overall production costs and quality levels. Thus, process planning is very important to ensure the quality of machining products, and to reduce the process costs and increase the process effectiveness. Process planning involves determination of appropriate machines, tools and process parameters under certain conditions for each operation. The success of a manufacturing process will depend on the selection of process parameters. The effective optimization of these parameters dramatically minimize the cost of the manufacturing process as well as the increase the quality of the final product (Cus & Balic, 2003; Baskar, Asokan, Saravanan & Prabhaharan, 2006).

In industry, optimal parameter design problem frequently occurs in product development, process design and operational condition setting stages. The problem consists of finding the optimum process parameter settings which provide the best process performance. In other words, a parameter design is desired to obtain a set of operating conditions (process parameters) in such a manner that the process performance is kept in a desired range. Since a manufacturing process requires optimizing more than one objective, usually many conflicting responses must be optimized simultaneously with process parameters. In the lack of systematic approaches, the optimization of multiple responses was done by "trial-and-error" or by changing one control variable at a time while holding the rest constant. In the last decades, different solution methods such as operational research techniques, design of experiments, simulation and artificial intelligence have been proposed for modeling and solution of process optimization problems. Due to the enormous complexity of many processes and the high number of influencing parameters, conventional approaches to the optimization of manufacturing processes are no longer sufficient. Since such methods are not efficient in finding the true optimum, many researchers try to find more efficient methods for optimization of multiple responses. After the success of Hopfield & Tank (1985), despite a vast amount of work existing in the literature, to find an efficient method for obtaining optimal solutions in polynomial time motivated the researchers to apply neural networks to optimization problems and to compare their performance with other techniques'. The motivation behind the Hopfield & Tank neural network model was to take advantage of the great speed associated with the massively parallel computing capabilities of neural networks for fast solution of combinatorial optimization problems.

Neural-network models are powerful tools when modeling data sets that are non-linear and highly correlated. A neural network model is developed to predict the value of critical parameters in a complex manufacturing process, on the basis of process operating parameters or conditions. This gives manufacturer valuable information about the process parameter values that are required under various operating conditions and at various stages of the process in order to reach desired response values. Here, the motivation behind this research is to test the success of neural networks in solving optimization problems for tube hydroforming and metal cutting processes and to conclude about their performance.

1.2 Research Objective

In this thesis, we deal with optimization of two manufacturing processes. The first one is the tube hydroforming process with optimization of two conflicting objectives: minimization of thinning ratio and maximization of bulge ratio. For the solution of this problem, we proposed a two-stage artificial neural network (ANN) approach in which a back propagation network is employed in each stage. The network in the first stage is built for parameter searching while the network in the second stage is used for response estimating. To compare performance of the proposed network, a two stage genetic algorithm (GA) approach is also proposed for optimization of the tube hydroforming process. In the first stage a metamodel is built to model the relationship between forming parameters and process responses. ANNs and response surface analysis are employed to build the metamodel.

The second problem solved in this thesis is process parameters optimization of a metal cutting process with unit cost minimization. For the solution of this problem, we proposed a dynamical gradient network. The original integer programming model given by Gupta, Batra & Lal (1995) is used to construct the energy function. The appropriate energy function is constructed by using a penalty function approach. Due to the tradeoff problem among the penalty terms, it becomes very difficult to find the values of the penalty parameters that result in feasible and good solutions. Some of the penalty terms are tried to be eliminated by the proposed network. Therefore, log-sigmoid and maximum networks are used to drop some of the penalty terms from the energy function. By this way, it is aimed to reduce the network complexity and to obtain a simplified energy function. Some of the binary constraints are satisfied using hard limit transfer functions, some binary constraints are satisfied using maximum networks.

The objectives of this thesis are listed below:

- To present a detailed evolutionary path of ANNs in manufacturing process optimization, review the current research literature, classify the approaches according to their architectures and to discuss several future research directions.

- To present a literature review on manufacturing processes optimization

- To propose and evaluate artificial neural network models for solving two manufacturing process optimization problems; the tube hydroforming process with thinning ratio minimization and bulge ratio maximization, and metal cutting process with unit cost minimization.

- To illustrate the use of genetic algorithms and response surface analysis in conjunction with the proposed approaches.

- To compare the results of the proposed approaches with other solution methods commonly used in the literature

- To discuss the use of ANNs for solving problem of optimization of manufacturing processes.

1.3 Contribution of the Thesis

The contribution of this thesis to the literature is two fold. The first one is to introduce an artificial neural network approach to solve the problems described above. Although there are different techniques already used for solving these problems, to the best of our knowledge, this thesis will be the first attempt to use neural networks for solving optimization problem of selected manufacturing processes. Therefore, this thesis will make a contribution to the relevant literature in terms of solution approaches.

On the other hand, the proposed approaches are combined with other solution techniques such as genetic algorithm and penalty approach. In the literature, stand alone neural networks have been commonly used for solving parameter optimization problem of manufacturing processes. Therefore, the second contribution of this thesis is to the artificial neural network literature in terms of their applicability to manufacturing optimization problems when they combined with other techniques.

1.4 Organization of the Thesis

The organization of this dissertation is as follows.

Chapter 2 is an introduction to manufacturing processes. Types of manufacturing processes are provided along with an overview of solution approaches used for solving optimization of manufacturing processes.

Chapter 3 presents a comprehensive review on neural networks and its implementations in industry. The basic concepts of neural networks and their attractions for solving process optimization problems are given. Different types of ANNs, backpropagation networks and Hopfield networks, are described in detail. The advantages, disadvantages and suitability of approaches in each category for solving process optimization problems are discussed and possible future research directions are given.

In Chapter 4, it is aimed to describe how GAs work. Basic concepts of GAs are given and a simple GA algorithm is described. Advantages of GA over other methods are presented and applications of GA in manufacturing processes are illustrated.

In Chapter 5, the problem of parameters optimization in tube hydroforming process is introduced and the relevant literature review is given. The proposed artificial intelligence approaches are explained and the performances of proposed approaches are compared with other existing solution methods for the optimization of tube hydroforming process.

In Chapter 6, the problem of process parameters optimization in metal cutting is studied and the relevant studies in the literature are reviewed. The objective in this problem is minimization of unit production cost subject to several constraints. The proposed interconnected network is presented and the convergence of the network is presented. The proposed approach is illustrated by an example and simulation results are compared with existing solution methods used to solve optimization of metal cutting processes.

Chapter 7 contains the concluding remarks of this research and identifies future research directions.

CHAPTER TWO

OPTIMIZATION OF MANUFACTURING PROCESSES

In this chapter, we present an overview of optimization techniques used in solving optimization of parameters for tube hydroforming and metal cutting processes.

2.1 Manufacturing Processes

A manufacturing process or system is defined as the use of one or more physical mechanisms to transform the shape of a material's shape and/or form and/or properties. Designing manufacturing process is a difficult task because:

1. Manufacturing processes are large and have many interacting components.

2. Manufacturing processes are dynamic.

3. The relationships between performance measures and decision variables cannot usually be expressed analytically.

4. Data may be difficult to measure in a harsh processing environment.

5. There are usually multiple performance requirements for a manufacturing process and these may conflict.

In general, a manufacturing system design can be conceptualized as the mapping from the performance requirements of a manufacturing system, as expressed by values of certain performance measures, onto suitable values of process parameters, which describe the physical design or the manner of operation of the manufacturing system A performance measure is a variable whose value quantifies an aspect of the performance of a manufacturing system. Performance measures are either benefit measures or cost measures. They can be divided into four categories: time, quality, cost, and flexibility. In general, a number of performance measures will be relevant for a given manufacturing system. However, they will differ from one manufacturing system to another (Chryssolouris, 2006).

A significant improvement in process performance may be obtained by optimization in process planning. Tool selection, machine selection, process selection and tool path selection, process parameter selection are the most important areas for optimization in manufacturing process planning. The fundamental activity in the planning of a manufacturing process is deciding the values of the process parameters that identify and determine the regions of critical decision variables leading to desired responses with acceptable variations.

The optimization of manufacturing process parameters takes into account a number of factors such as the shape and size of the workpiece, the required tolerances, surface quality, the material the workpiece is made of, and the quantity to be made. Then, the factors affecting the performance of a manufacturing process can be categorized into three groups:

- *Operating constraints* such as manufacturing practice, the manufacturing process, machine tool characteristics and capability and available processing time as specified by production planning.

- *Operating requirements* such as the workpiece material and geometry, the operation being performed and the tooling data.

- *Tool performance factors* such as the tool material and geometry and the use of cutting fluids (Scallan, 2003).

2.2 Classification of Manufacturing Processes

Manufacturing processes can be classified into four main categories:

2.2.1 Forming

Forming processes cause a material to take the shape of a die using an external force. Forming processes change the size and shape but not the volume of the material by forcing the material over, between, into a forming device. Forming processes use a forming force and a forming device. The force may be generated by a hammer, press, or rolling machine. The forming device may be a die with a shaved cavity, a mold with an external shape, or a set of smooth or shaped rolls. Forming may be done either hot or cold. Types of forming processes are:

- Compressive forming
- Tensile forming
- Bending
- Forming by shearing

Forming processes generally have high tooling costs due to the complicated die geometries required. Because the cost of tooling is high, forming processes are usually applied to lot sizes large enough to economically justify the high cost of tools and machinery required. Finally, highly-skilled workers are usually not needed to operate deforming processes, so labor costs are also relatively low, compared with other manufacturing processes. In terms of part quality, the deformation process produces work hardening, which increases the mechanical strength of the part. However, excessive material deformation may lead to crack and overlap formation in the workpiece. Forming processes have a relatively low degree of flexibility compared with other manufacturing processes, since the kinematics of forming machines are constrained by motion, force or energy. The geometry of the part is governed solely by the tool geometry. However, since forming dies must move relatively to the workpiece, the geometric features that are producible, are limited (Patton, 1970; Ulrich and Eppinger, 2003).

2.2.2 Casting & Molding

Casting and molding cause molten or liquid materials to enter a mold where it solidifies before being extracted. Casting and molding produce parts that have a desired size and shape by introducing a material into an existing mold cavity. The material may be a liquid or may be made molten by heating it. The material is then introduced into the mold by gravity (pouring) or with force (injecting). Once in the mold, the material is solidified by cooling, drying, or chemical action. The casting is then extracted by opening or destroying the mold. The selection of an appropriate casting process will depend on a number of factors which include the material, size, weight and complexity of the geometry, labor, equipment and tooling costs, tolerances and surface finish required, strength and quantity and production rate required and the overall quality requirements. Type of casting and molding processes are:

- Sand casting
- Die casting
- Investment casting
- Injection molding

Casting processes are limited in terms of surface quality, porosity, and, consequently, the strength of the parts produced is also limited. Generally, casting processes are used in production with relatively large lot sizes so that the high capital cost can be justified. The part quality can be influenced by process parameters such as die temperature, cooling time and cooling rate, as well as the design of die or mold features. The flexibility of casting and molding processes is limited. Only one part geometry can be produced for a die geometry and the part geometry cannot be changed through workpiece-tool motions. However, these processes have the potential to produce parts with very intricate geometric features, especially internal features, and workpiece thicknesses (Patton, 1970).

2.2.3 Machining

Machining gives a material size and shape by removing excess material. This process uses a cutting element (tool, burning gases, electric spark, etc.). They include mechanisms to develop cutting motion (causing a cut to form) and feed motion (bringing new material into the cut). Machining processes involve the removal of material from the workpiece and there is a variety of processes that fall into this category. Main machining processes are:

- Milling
- Turning
- Drilling
- Grinding

Machining processes are, by far, the most commonly used of manufacturing processes. This is due to the diversity of shapes and degree of accuracy that can be obtained with many machining processes compared to other manufacturing processes. Specific reasons for the use of machining processes are:

- The need for closer dimensional accuracy than is achievable from casting or forming processes alone;

- The need for improved surface finish than is achievable from casting or forming processes alone;

- The need for part finishing due to heat treatment;

- In the manufacture of small lots, machining may be the most economical method of production.

There are, however, a number of distinct disadvantages of using machining processes:

- By their very nature there is waste material;

- They require more capital, energy and labor than casting and forming processes per volume of production;

- Removing material generally takes longer than casting and forming processes per volume of production.

Regardless of these disadvantages, machining processes are widely used and play an indispensable role in manufacturing.

The machine and tooling costs, associated with mechanical material removal processes, are low compared with other manufacturing processes. However, the skill level involved for programming or manually setting the tool and the workpiece kinematics is relatively high, thus, labor costs for operating material removal processes are also correspondingly high. Machining processes are therefore, better suited for low to medium volume production. The production rates for machined parts are much lower than those for casting or forming processes, since it is the tool that is required to make multiple passes over the workpiece surface in order to produce the final shape. The material removal rate is dependent on the surface quality desired, the workpiece material, the cutting tool material and the cutting fluid used. Surface quality and surface technology are clearly very important aspects of material removal processes. Surface effects are caused both by the process itself and the workpiece material properties. These effects have a direct influence on the mechanical characteristics of the workpiece and eventually on the reliability of the component. Material removal processes are among the most flexible of the manufacturing processes. Since the geometry of the finished part is defined by the geometry and the kinematics of the tool and workpiece, material removal processes can produce parts with a wide range of sizes, shapes and surface quality (Patton, 1970; Ulrich and Eppinger, 2003).

2.2.4 *Joining*

Joining or assembly process is used to temporarily or permanently fasten pieces together. It is focused on the formation of specific geometries. Joining processes enable the manufacturing of a product in individual components and then combine them into a single product, which may be easier and less expensive to manufacture than the whole product at once. Joining processes also allow the inclusion of features and properties in a particular product, which may differ from the majority of the components used in the product. Type of casting and molding processes are:

- Welding
- Brazing
- Soldering
- Adhesive Bonding
- Mechanical joining

The capital and tooling costs, associated with most joining operations are relatively low compared with those of other processes, since most joining equipment is inexpensive. Joining processes are very labor-intensive, especially in case of adhesive bonding or joining parts with complex geometries. Most joining processes require pre-processing of the joining surfaces in order to minimize surface roughness and a period of time after joining, for curing the bond or cooling the weld. These factors result in low production rates for joining processes compared with those for forming, casting, molding or removing processes. Some types of defects such as porosity, entrapment of contaminants in the joint, incomplete fusion or penetration, crack formation, surface damage and residual stresses may occur during joining processes. However, effective use of joining techniques can produce joints with mechanical strength exceeding that of its joining members. Joining processes also have a high degree of flexibility in part geometry and lot size (Patton, 1970; Ulrich and Eppinger, 2003).

The main characteristics of manufacturing processes are summarized in Table 2.1.

Table 2.1 Main characteristics of manufacturing processes

Process	Cost	Production Rate	Quality	Flexibility
Forming	High	High	Low-Medium	Low
Casting/Molding	High	High	Low-Medium	Low
Machining	High	Medium-High	High	High
Joining	Low	Low-Medium	Low-Medium	High

2.3 Optimization and Manufacturing Processes

Optimization of manufacturing processes can increase the quality and quantity of products and decrease production cost simultaneously. Optimization methods can find the compromised solutions for the conflicting objectives of different design features or aspects to reach the maximum capability of a manufacturing system.

Optimization problems in manufacturing process planning are determining the optimal configuration of process factors to increase the process performance in terms of performance measures. The ranges of process factors restrict the possible alternatives that are considered to be feasible. In most of the manufacturing processes, more than one response has to be considered for optimization of process parameters making it necessary to tackle these problems in such a way that several approaches can be simultaneously optimized. Thus, problem of parameter optimization can be concluded as a multiple response optimization problem.

Multi-objective optimization is the process of maximizing or minimizing more than one objective function while satisfying the prevailing constraints or bounds.

$$\min F(X) = \left[f_1(X),\ldots,f_i(X),\ldots f_m(X)\right]$$

$$\textit{subject to}:$$

$$\begin{aligned} &h_i(X) = 0 && i = 1,\ldots,q \\ &g_j(X) \leq 0 && j = 1,\ldots,p \\ &x_k^l \leq x_k \leq x_u^k && k = 1,\ldots,n \end{aligned}$$

where the components of the objective function vector, *F(X),* are in conflict with one another. Since the components of objective function vector are competing in general, there is no unique solution to this problem. The purpose of this problem is to search for a best compromise solution to ensure objectives are close to their corresponding preference points as much as possible.

2.4 Solution Approaches

The methods and tools for the optimization of manufacturing processes fall into four broad categories: *operations research, design of experiment, artificial intelligence,* and *simulation.* The divisions among these categories are fuzzy.

2.4.1 Operations Research

Operations research methods use an appropriate mathematical description of the problem. They do not try to investigate all of the possible feasible solutions, which would be practically impossible, and they reduce the search space and CPU time required to obtain a solution while satisfying the constraints.

2.4.1.1 Mathematical Programming

In mathematical programming parameter decisions are modeled using integer or continuous variables and the process planning problem is represented as an optimization problem in which a mathematical function has to be minimized or maximized subject to some linear and non-linear algebraic constraints. If the objective function is linear and the constraints are a combination of linear equalities or inequalities, the problem is called a linear programming problem. In a linear programming problem, the decision variables involved in the problem are also nonnegative. The most widely applied method for the solution of linear programming problems is the simplex method developed by George Dantzig in 1947. It is an iterative procedure for generating and examining different extreme points of a linear program, each one improving the current value of the objective function until an optimum is found. If some of the variables in a linear programming model are required to have integer values, this model is referred to as mixed integer programming (MIP) and if all the variables are integers, it is called a pure integer programming problem (Akyol, 2006).

Mathematical programming is commonly applied for machining operations. Kusiak (1985), Tan & Creese (1995) and Gupta, Batra & Lal (1995) used linear programming approach for multi-pass turning operations. Some studies combine several mathematical programming approaches such as linear programming, geometric programming and dynamic programming (Prasad, Rao & Rao, 1997; Chen, Lee & Fang, 1998; Liang, Mgwatu & Zuo, 2001). Many other iterative mathematical search algorithms with their applications are reported in the literature, such as geometric programming approach (GopalaKrisnan & Al-Khayyal, 1991; Sönmez, Baykaşoğlu, Dereli & Filiz, 1999) and Nelson-Mead simplex search approach (Agapious, 1992 a,b & c). Sönmez, Baykaşoğlu, Dereli & Filiz (1999) and Mukherjee & Ray (2006) provide a good survey on applications of mathematical programming models in machining operations.

For other processes such as casting and forming, applications of mathematical programming can be found in Miettinen, Makela & Mannikkö (1998) and Naceur, Guo, Batoz & Lenoir (2001). The detailed survey of use of quadratic programming for metal forming processes has been presented by Zhang, Xu, Di & Thomson (2002).

2.4.1.2 Dynamic Programming

Dynamic programming is a method based on Bellman's principle of optimality for solving problems that can be viewed as multistage decision processes. A multistage decision problem is a problem that can be separated into a number of subproblems referred as sequential steps, or stages, which may be completed in one or more ways. It is an enumeration method that uses a "*divide and conquer*" approach, and finds optimal solutions to subproblems. Then, according to the principle of optimality, it solves the problem recursively. Since it performs an intelligent enumeration of all feasible points, it resembles the branch-and bound method (Akyol, 2006).

As the first attempt, Iwata, Muratsu, Iwatsubo & Fujii (1972) presented a dynamic programming approach for machining operations. The further applications of dynamic programming to solve optimization problem of manufacturing processes are Hayers & Davis (1979), Sekhon (1982), Yehuda, Feldman, Pinter & Wimer (1989), Daskin, Jones & Lowe (1990), Shin & Joo (1992) and Agapious (1992 a,b &c) (Mukherjee & Ray, 2006).

2.4.2 Simulation

2.4.2.1 Finite element method (FEM)

Finite element method (FEM) is a numerical method for solving a differential or integral equation. The method essentially consists of assuming the piecewise continuous function for the solution and obtaining the parameters of the functions in a manner that reduces the error in the solution. In finite element analysis, the domain of a problem is broken into many smaller zones called elements. At this point, finite element analysis can be used to calculate an approximate solution—element by element—to this problem. Visualization software can then be used to put this collection of information into an intuitive and coherent picture.

There are generally two types of FEM that are used in industry: 2-D modeling, and 3-D modeling. While 2-D modeling conserves simplicity and allows the analysis to be run on a relatively normal computer, it tends to yield less accurate results. 3-D modeling, however, produces more accurate results while sacrificing the ability to run on all but the fastest computers effectively. Within each of these modeling schemes, the programmer can insert numerous algorithms (functions) which may make the system behave linearly or non-linearly (Tekkaya, 2000).

The key idea is to simulate the performed experiment, trying to adapt material parameters in order to compute with FEM the same results as the experimental results. This problem is mathematically called an inverse problem and can be seen as an optimization problem where the objective function is to minimize the gap between experimental and FEM results. The optimization variables are the material parameters that appear in the proposed model.

Finite element analysis relies on breaking a complicated problem into a large number of less complex problems. When the solution to a problem exhibits very complicated behavior, it is sometimes acceptable to apply simplifications. Often times, though, a broad simplification introduces too much error to be useful. This is when breaking up the problem into many separate problems can help. Simplified solutions to each element of a problem can be integrated together to give a highly accurate general solution.

In the literature, FEM is commonly used for simulation of metal forming. Pioneering studies of FEM to process optimization are made to sheet metal processes by Wifi (1976), Gotoh & Ishise (1978) and Wang & Budiansky (1978). The first 3D applications are known by Tang, Chu & Samanta (1982) and Toh & Kobayashi (1983).

Further studies on metal forming include Tekkaya (2000), Huh & Kim (2001), Ghouati & Gelin (1999), Santos, Duarte, Reis, Rocha, Neto & Paiva (2001) and Fourment & Chenot (1996). One can refer to Ponthot & Kleinermann (2006) for a detailed review of FEM applications to forming processes.

2.4.3 Design of Experiments (DoE)

These strategies were originally developed for the model fitting of physical experiments, but can also be applied to numerical experiments. The objective of DoE is the selection of the points where the response should be evaluated.

2.4.3.1 Taguchi method

Taguchi method, employing design of experiments, is one of the most important statistical tools of total quality management for designing high quality systems at reduced cost. Taguchi method is an efficient problem solving tool, which achieves continuous quality improvement of the performance of the product, process, design and system by minimizing the variation in product and process performance. The objective is to determine the optimal combination of process parameters so that the product or process is most robust with respect to noise factors.

G. Taguchi is the developer of the Taguchi method and he proposed that the engineering optimization of a process or product should be carried out in a three-step approach (Tarng & Yang, 1998):

1. ***System design:*** System design involves the development of a system to function under an initial set of nominal conditions. System design requires technical knowledge from science and engineering. Since the system design is an initial functional design, it may be less than optimum in terms of quality and cost.

2. ***Parameter design:*** After the system architecture has been chosen, the next step is parameter design. The objective of the parameter design is to optimize the settings of the process parameter values for improving quality characteristics and to identify the product parameter values under the optimal process parameter values.

3. ***Tolerance design:*** When parameter design is not sufficient for reducing the output variation, the last phase is tolerance design. It involves tightening tolerances on the product parameters or process parameters whose variations result in a large negative influence on the required product performance.

The Taguchi method is based on statistical design of experiments and the number of experiments increases with the increase of process parameters. To solve this complexity, the Taguchi method uses a special design of orthogonal array to study the entire process parameter space with a small number of experiments only. The experimental results are then transformed into a signal-to-noise (S/N) ratio. The signal-to-noise ratio can be used to measure the quality characteristics deviating from the desired values. Depending on the particular type of characteristics involved, different S/N ratios may be applicable, including “lower is better” (LB), “nominal is best” (NB), and “higher is better” (HB).

To summarize, the parameter design of the Taguchi method includes the following steps (Tarng & Yang, 1998):

1. Identify the quality characteristics and process parameters to be evaluated.

2. Determine the number of levels for the process parameters and possible interactions between the process parameters.

3. Select the appropriate orthogonal array and assign the process parameters to the orthogonal array.

4. Conduct the experiments based on the arrangement of the orthogonal array.

5. Analyze the experimental results using the signal-to-noise ratio and statistical analysis of variance.

6. Select the optimal levels of process parameters.

7. Verify the optimal process parameters through a confirmation experiment.

Taguchi’s technique of parameter design has been successfully applied in a number of machining problems by researchers (Youssef, Beauchamp & Thomas, 1994; Lin, 2002; Singh, Shan & Pradeep, 2002). Research works applying Taguchi on joining processes can be found in Tarng & Yang (1998), Tarng, Yang & Juang (2000) and Lakshminarayanan & Balasubramanian (2008). Other typical applications of Taguchi method include the optimization of molding processes (Fox & Lee, 1990; Chen, Lee & Yu, 1998; Reddy, Nishina & Babu, 1998; Rowlands, Antony & Knowles, 2000), casting processes (Syrcos, 2003; Wu & Chang, 2004) and forming processes (Tsui, 1999; Li, Nye & Metzger, 2006).

More detailed review of Taguchi method for optimization of manufacturing processes has been presented in Sukthomya & Tannock (2004).

2.4.3.2 Response Surface Methodology (RSM)

As an important subject in the statistical design of experiments, the Response Surface Methodology is a collection of mathematical and statistical techniques useful for the empirical modeling and analysis of problems in which the objective is to optimize a response (output variable) which is influenced by several independent variables (input variables).

The method was introduced by Box and Wilson in 1951 to model experimental responses and then it is migrated into the modeling of numerical experiments. The main idea of RSM is to use a sequence of designed experiments to obtain an optimal response.

RSM explores the relationship between variables and responses. Generally, the structure of the relationship between the response and the independent variables is unknown. The response surface designs are types of designs for fitting response surface and the first step in RSM is to find a suitable approximation to the true relationship. The most common forms are low-order polynomials (first or second-order).

The construction of response surface models is an iterative process. Once an approximate model is obtained, the goodness-of-fit determines if the solution is satisfactory. If this is not the case, the approximation process is restarted and further experiments are made. R^2 is a statistic that will give some information about the goodness of fit of a response surface model. The R^2 coefficient of determination is a statistical measure of how well the model approximates the real data points. An R^2 of 1.0 indicates that the model perfectly fits the data. Adjusted R^2 is a modification of R^2 that adjusts for the number of explanatory terms in a model. Unlike R^2, the adjusted R^2 increases only if the new term improves the model more than would be expected by chance. The adjusted R^2 can be negative, and will always be less than or equal to R^2.

Many researchers and practitioners use RSM in metal cutting process parameter optimization problems. The first attempt of optimization of cutting parameters by RSM has been presented by Taramen (1974).

The further studies on determining optimal parameters of metal cutting processes by RSM can be found in El Baradie (1993), Lee, Shin & Yang (1996), Fuh & Chang (1997) and El-Axir (2002). An application of RSM to wire electrical discharge machining has been shown in Hewidy, El-Taweel & El-Safty (2005). Jeang, Li & Wang (2010) used RSM combining with a mathematical programming model to optimize process parameters in cutting operations. An example of application of RSM on forming processes has been presented by Jansson, Andersson & Nilsson (2005).

In this thesis, response surface analysis is used to map the relationship between process parameters and responses for solving the parameter optimization problem of manufacturing process under consideration.

2.4.4 Artificial Intelligence (AI)

The field of artificial intelligence may be defined as the study of ideas that enable computers to be intelligent. Its main goals are to make computers more useful, and to understand the principles that make intelligence possible. AI can be seen as an attempt to model aspects of human thought on computers. It is also sometimes defined as trying to solve by computer any problem that a human can solve faster.

2.4.4.1 Artificial Neural Network (ANN)

ANNs were originally designed for simulating the brain behavior. They have emerged as efficient approaches in a variety of engineering applications where problems are difficult to formulate or hardly defined. They are computational structures that implement simplified models of biological processes, and are preferred for their robustness, massive parallelism and ability to learn. In metaheuristics literature, neural networks are put into local-search based metaheuristics category. The reason is their iterative master process characteristic, that is, they guide and modify the operations of subordinate heuristics to efficiently produce high quality solutions, and provide decision makers with fast and robust tools for obtaining high quality solutions in reasonable computation times to many real life problems.

From a modeling viewpoint, they are mathematical representations of biological nervous systems that can carry out complex cognitive and computational tasks. They are composed of many nonlinear interconnected processing elements that are analogous to neurons, and connected via weights that are analogous to synapses. The modern age of neurocomputing started with the work of McCulloch & Pitts (1943) in which the first mathematical model of a single biological neuron was presented. Although McCulloch and Pitts' study showed that simple type of neural Networks were able to learn arithmetic or logical functions, ANN algorithms have been successful enough for many applications in the mid 1980s (Potvin & Smith, 2003).

ANNs has received considerable attention in the last years and has been applied to optimization of manufacturing (Hsieh, 2006; Su & Hsieh, 1998; Tong & Hsieh, 2001; Cook, Ragsdale & Major, 2000; Zuperl & Cus, 2003; Ko, Kim, Kim & Choi, 1998).

Recent comprehensive review of ANN applications in manufacturing, Zhang & Huang (1995) and Sukthomya & Tannock (2005) cited such diverse venues as machining, cutting, molding, welding etc.

2.4.4.2 Genetic Algorithm (GA)

Genetic algorithm, being one of the most popular combinatorial algorithms and AI techniques, is a search technique for solving optimization problems based on the mechanics of the survival of the fittest.

GA starts with the creation of an initial population of possible solutions to the problem called individuals or chromosomes, and the genes within the chromosomes determine the individual features of the child. Each chromosome is associated with a fitness value, which represents the probability of a chromosome being selected to be a parent. From the individual population, a new population is generated using one of the specific operators such as reproduction, crossover or mutation. By the reproduction operator, the solutions in the old population are copied to the next population with a probability depending on the fitness of the solution which corresponds to the value of the objective function for that solution. Using the crossover operator, new solutions are generated from pairs of individuals, and by mutation one or more of the genes in a chromosome are altered in a random way which helps the GA to explore a new, perhaps a better feasible region than the previously considered. The process is repeated until some stopping rule is satisfied and the individual with the most favorable fitness is the solution to the problem.

Several applications of GA in machining processes have been reported in the literature as Suresh, Rao & Deshmukh (2002), Amiolemhen & Ibhadode (2004), Liu

& Wang (1999), Onwubolu & Kumalo (2001), Chen & Tsai (1996), Cus & Balic (2003), Solimanpur & Ranjdoostfard (2008), Sardinas, Santana & Brindis (2006), and Sreeram, Kumar, Rahman & Zaman (2006). Previous works dealt with the optimization of casting processes are Vijian & Arunachalam (2007) and Filipic & Laitinen (2005).

To enhance the performance of GA, there has been an explosive growth in the successful use of hybrid GAs in process optimization (Su & Chiang, 2003; Shen, Wang & Li, 2007; Ozcelik & Erzurumlu, 2006; Li, Su & Chiang, 2003; Yang, Lin & Chen, 2006)

In the literature, although a large number of approaches such as mathematical programming, design of experiments and FEM to solve the manufacturing process optimization problems, recently there has been an explosion of interest in using artificial intelligence. In this thesis, GA and ANN are used to solve the problems under consideration. Details and a comprehensive review of these solution methods will be given in the following chapters.

CHAPTER THREE
GENETIC ALGORITHMS

The earliest instances of what might today be called as genetic algorithms (GAs) appeared in the late 1950s and early 1960s. As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments. This foundational work established more widespread interest in evolutionary computation. By the early to mid-1980s, genetic algorithms were being applied to a broad range of subjects, from abstract mathematical problems like bin-packing and graph coloring to tangible engineering issues such as pipeline flow control, pattern recognition and classification, and structural optimization (Akyol, 2006).

The purpose in this chapter is to give a survey of recent research where GAs were used for optimization of tube hydroforming and metal cutting processes.

3.1 Genetic Algorithms

The GA originally developed by Holland in the 1970s is a stochastic search method based on evolution and genetics and exploits the concept of survival of the fittest. They represent an intelligent exploitation of a random search used to solve optimization problems. Although randomised, GAs are by no means random, instead they exploit historical information to direct the search into the region of better performance within the search space. The basic techniques of the GAs are designed to simulate processes in natural systems necessary for evolution, based on the Darwinian principle of “survival of fittest”. As in nature, competition among individuals for scanty resources results in the fittest individuals dominating over the weaker ones. GAs simulate the survival of the fittest among individuals after a series of iterative computations for solving a problem (Akyol, 2006).

GAs differ from conventional search techniques that conduct a point-to-point search in the solution space. Each generation consists of a population of character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

GAs are based on an analogy with the genetic structure and behavior of chromosomes within a population of individuals using the following foundations (Goldberg, 1989):

- Individuals in a population compete for resources and mates.

- Those individuals most successful in each 'competition' will produce more offspring than those individuals that perform poorly.

- Genes from `good' individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.

- Thus each successive generation will become more suited to their environment.

The GA approach represents a powerful, general-purpose optimization paradigm in which the computational process mimics the theory of biological evolution. The power of these algorithms is derived from a very simple heuristic assumption that the best solution will be found in the regions of solution space containing high proposition of good solution, and that these regions can be identified by judicious and robust sampling of the solution space. As a local search technique, GA can find solutions for a wide range of application. It has been successfully used in job-shop scheduling, production planning, line balancing, lumber cutting optimization, and process optimization (Cook, 2000). The basic concepts of GAs and a simple GA algorithm are described in the next section.

3.1.1 Basic Concepts

The solution of an optimization problem with GA begins with a set of candidate solutions called population. To achieve the desired response, GAs generate a successive population of alternate solutions in which a candidate solution is represented by a sequence of numbers known as chromosome or string. A chromosome's potential as a solution is determined by its fitness function, which evaluates a chromosome with respect to the objective function of the optimization problem under consideration. The GA then iteratively creates new populations from the old by ranking the strings and uses the fittest to create new strings which are closer to the optimum solution to the problem at hand. GAs consist of thee main operations that randomly impact the fitness value: reproduction (selection), crossover and mutation. The reproduction-evaluation cycle used by GA is referred as a generation.

There are basically six steps to be taken in a genetic algorithm optimization (Correia, Gonçalvez, Cunha & Ferraresi, 2005):

(1) *GA Parameters:* The main parameters of GA are population size, crossover probability, mutation probability and generation number.

- *Population size:* Indicates how many chromosomes exist in the population. If there are too few chromosomes, GA have a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to increase population size, because it does not make solving the problem faster.

- *Crossover probability:* Indicates how often the crossover will be performed. If there is no crossover, offspring is exact copy of parents. If there is a crossover, offspring is made from parts of parents' chromosome.

- *Mutation probability:* Indicates how often the parts of the chromosome will be mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed.

- *Maximum number of generations:* Indicated the termination criteria of the algorithm.

(2) *Creation of initial population:* An initial set of individuals is created by a random generator. Each individual in the population needs to be described in a chromosome representation that plays a vital role in the development of a GA. A problem can be solved once it can be represented in the form of a solution string (chromosome). The genes in the chromosome can be binary or real integer number. The chromosome length is the vector length of the solution to the problem. In real coded GAs, each gene represents a variable of the problem. Once, the initial population is created, the next step is to select the strings to generate new population.

(3) *Fitness evaluation:* A fitness function that describes the relationship between inputs and outputs is a particular type of objective function that prescribes the optimality of a solution in a genetic algorithm so that the particular chromosome may be ranked against all the other chromosomes.

(4) *Selection:* After formation of chromosomes, the individuals should be selected for creation of the new generation. The selection operator allows individual strings to be copied for possible inclusion in the next generation. Selection is based on the fitness value of each member of a generation. According to Darwin's evolution theory the best ones should survive and create new offspring. There are several methods of selection such as roulette wheel selection, rank based selection, tournament selection etc.

- *Roulette Wheel Selection:* This is a way of choosing members from the population of chromosomes in a way that is proportional to their fitness. It does not guarantee that the fittest member goes through to the next generation; however it has a very good chance of doing so. This could be imagined similar to a Roulette wheel in a casino. Usually a proportion of the wheel is assigned to each of the possible selection based on their fitness value. This could be achieved by dividing the fitness of a selection by the total fitness of all the selections, thereby normalizing them to 1. Then a random selection is made similar to how the roulette wheel is rotated as in Figure 3.1.

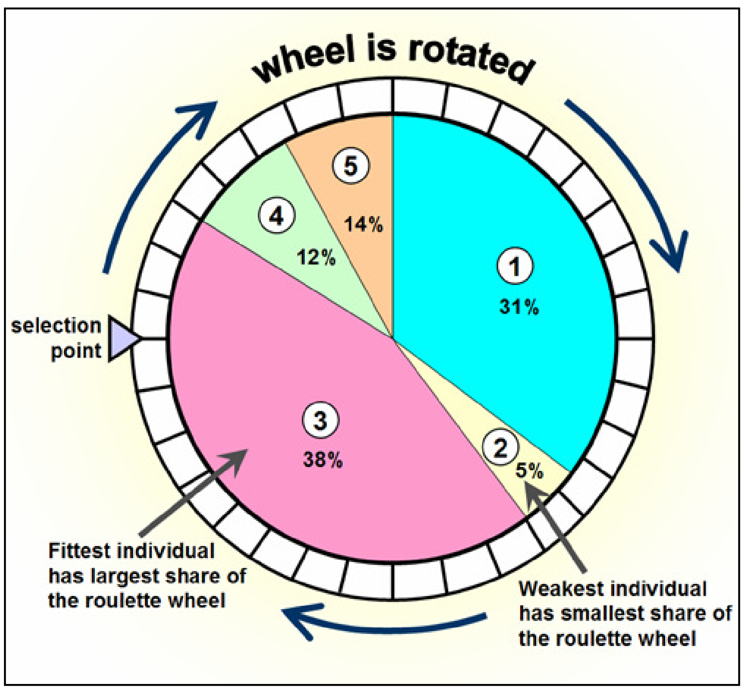


Figure 3.1 Roulette wheel selection: based on fitness
(from Engineering Design Centre)

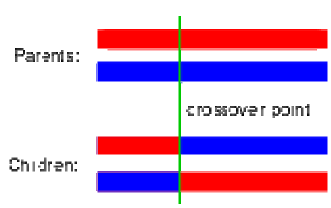
- *Tournament Selection:* Tournament selection involves running several "tournaments" among a few individuals chosen at random from the population. The winner of each tournament (the one with the best fitness) is selected for crossover. Selection pressure is easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

- *Rank Selection:* Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N where N is the number of chromosomes in population.

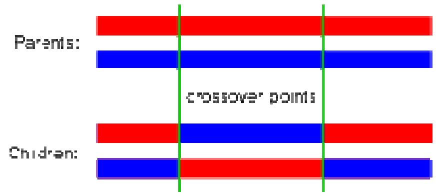
- *Truncation Selection:* With truncation selection that has a threshold of T between 0 and 1, only the fraction T best strings can be selected. They all have the same selection probability.

(5) <u>*Crossover:*</u> After creating the mating pool, the population is enriched with good strings from the previous generation but does not have any new string. A crossover operator is applied to the population to create better strings. All individuals in the mating pool are randomly selected for crossover to generate the offspring. The total number of participative strings in crossover and whether crossover should take place are controlled by crossover probability. If GA decides not to perform crossover, the selected strings are simply copied to the new population. If crossover does take place, then a random splicing point is chosen in a string, the two strings are spliced and the spliced regions are mixed to create two new strings. These child strings are then placed in the new population. The main types of crossover operation are (Knight, D. from www.ivoryresearch.com).

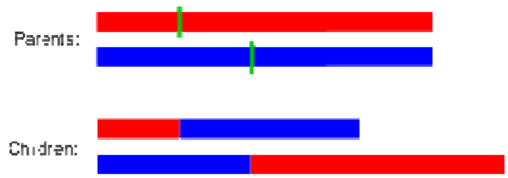
- *Single point crossover:* One crossover point is selected.



- *Two point crossover:* Two points are selected.



- *Cut and splice:* Results in a change in length of the children strings.



(6) *Mutation:* Selection and crossover alone can obviously generate a staggering amount of differing strings. However, depending on the initial population chosen, there may not be enough variety of strings to ensure the GA sees the entire problem space. Or the GA may find itself converging on strings that are not quite close to the optimum it seeks, due to a bad initial population. The need for mutation is to improve the local research ability and keep diversity in the population. Mutation operator creates an offspring by applying a random change to a single individual in the current generation. The GA has a mutation probability which dictates the frequency at which mutation occurs. Mutation can be performed either during selection or crossover. The mutation operator includes, uniform mutation, non-uniform mutation.

After applying the GA operators, a new set of population is created. Then, if necessary, they are decoded and fitness values are calculated. This completes one generation of GA. Such iterations are continued till the termination criterion is achieved.

Then the basic genetic algorithm can be outlined as follows:

1. Set GA parameters (population size, maximum number of generation, parameter number, crossover rate, mutation rate etc.)
2. Create initial population
   a. Chromosome representation
3. Evaluate fitness of each chromosome
4. Create a new population:
   a. Selection
   b. Crossover
   c. Mutation
5. Use new population for further run
6. Return to Step 3 until termination criteria is met.

3.2 Genetic Algorithms in Manufacturing Process Optimization

Genetic algorithms are an important problem solving technique. The algorithm uses a strategy of a directed search through a problem state space from a variety of points in that space. For this reason, three main advantages of the genetic algorithm in optimization are identified as (Akyol, 2006):

- They generally find nearly global optima in complex spaces. This is important because the search spaces for our problems are highly multimodal and GA has the ability to solve convex, and multi-modal function, multiple objectives and non-linear response function problems, and it may be applied to both discrete and continuous objective functions.

- Considering their ability to find global optima, genetic algorithms are fast, especially when tuned to the domain on which they are operating. It can explore large search space and its search direction or transition rule is probabilistic, not deterministic, in nature, and hence, the chance of avoiding local optimality is more,

- It works with a population of solution points rather than a single solution point as in conventional techniques, and provides multiple near-optimal solutions. This contributes much to the robustness of genetic algorithms. It improves the chance of reaching the global optimum and, vice versa, reduces the risk of becoming trapped in a local stationary point.

- GAs do not require any form of smoothness. As it is not based on gradient-based information, it does not require the continuity or convexity of the design space.

- According to Goldberg, the simulated evolution of a solution through genetic algorithms is more efficient and robust than the random search, enumerative or calculus based techniques. The main reasons given by Goldberg are the probability of a multi-modal problem state space in non-linear problems, and that random or enumerative searches are exhaustive if the dimensions of the state space are too great.

- The problem solving strategy involves using "the strings' fitness to direct the search; therefore they do not require any problem-specific knowledge of the search space, and they can operate well on search spaces that have gaps, jumps, or noise.

- Another advantage of genetic algorithms is their inherently parallel nature, i.e., the evaluation of individuals within a population can be conducted simultaneously, as in nature.

As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments; most notably, Holland was also the first to explicitly propose crossover and other recombination operators.

The foundational works established more widespread interest in evolutionary computation. By the early to mid-1980s, genetic algorithms were being applied to a broad range of subjects, from abstract mathematical problems like bin-packing and graph coloring to tangible engineering issues such as pipeline flow control, pattern recognition and classification, and structural optimization. Today, evolutionary computation is a thriving field, and genetic algorithms are solving problems of everyday interest in areas of study as diverse as stock market prediction and portfolio planning, aerospace engineering, microchip design, biochemistry and molecular biology, and scheduling at airports and assembly lines.

Several applications of GA-based technique in process parameter optimization problems have been reported in the literature. Liu & Wang (1999) claim that by reducing the operating domain of GA, by changing the operating range of decision variables, convergence speed of GA increases along with significant increase in milling process efficiency. Shunmugam, Bhaskara & Narendran (2000) optimized the machining parameters such as number of passes, depth of cut in each pass, and speed and feed obtained using a GA, to yield minimum total production cost while considering technological constraints such as allowable speed and feed, dimensional accuracy, surface finish, tool wear and machine tool capabilities in face-milling operations. Dereli, Filiz & Baykasoglu (2001) optimized cutting parameters for milling operations taking unit cost as an objective function by using genetic algorithm.

Onwubolu & Kumalo (2002) propose a local search GA-based technique in multi-pass turning operation with mathematical formulation in line with work by Chen & Tsai (1996) with simulated annealing-based technique. Krimpenis & Vosniakos (2002) use a GA-based optimization tool for sculptured surface CNC milling operation to achieve optimal machining time and maximum material removal. Chowdhury, Pratihar & Pal (2002) apply a GA-based optimization technique for near optimal cutting conditions selection in a single-pass turning operation, and claim that GA outperform goal programming technique in terms of unit production time at all the solution points. Wang, Da, Balaji & Jawahir (2002) apply GA-based technique for near-optimal cutting conditions for a two-and three-pass turning operation having multiple objectives. Schrader (2003) illustrates the usability of GA based technique for simultaneous process parameter optimization in multi-pass turning operations.

Cus & Balic (2003) use GA-based technique to determine the optimal cutting conditions in NC-lathe turning operation on steel blanks that minimize the unit production cost without violating any imposed cutting constraints. The results obtained are compared with those taken from recent literature and the comparisons proved the effectiveness of the proposed approach. Amiolemhen & Ibhadode (2004) proposed a GA approach similar to Onwubolu & Kumalo (2002). They developed a user friendly computer package based on GA approach for determining the optimal machining parameters for multi pass machining operation.

To illustrate the efficiency of non-conventional manufacturing processes, Baskar, Asokan, Saravanan & Prabhaharan (2005) proposed optimization procedures based on genetic algorithm, ant colony algorithm, tabu search and particle swarm optimization for the optimization of machining parameters for milling operations. The objective of the problem was to maximize profit for the inputs as cutter diameter, cutting length of workpiece, number of machining operation and number of cutting teeth of the tool. They concluded that particle swarm optimization always yielded better results. In their latter research, Baskar, Asokan, Saravanan & Prabhaharan (2006) developed a strategy based on the same objective constraints to determine the optimum cutting parameters of milling operations. They proposed three optimization procedures based on genetic algorithm, hill climbing algorithm and memetic algorithm. To compare these optimization procedures, they presented an example problem. They concluded that all procedures provided significant improvement but they suggested using memetic algorithm for solving the optimization problem of the milling operations.

Al-Aomar & Al-Okaily (2006) utilized a simple genetic algorithm based search as an alternative to Taguchi's experimental design for the solution of parameter design problem of turning process. They applied the proposed approach to a CNC lathe machine and the results indicated that simple GA provided lower cost than design of experiment approach. However, overhead costs did not taken into consideration by the researchers. Sardinas, Santana, & Brindis (2006) presented a multi-objective optimization technique based on GAs to optimize cutting parameters, cutting speed, feed and speed, in turning processes. They tried to optimize two conflicting objectives, tool life and operation time, simultaneously. The researchers concluded that the proposed approach offered greatest amount of information in order to make a decision on selecting cutting parameters of turning.

Palanisamy, Rajendran & Shanmugasundaram (2007) used GA to minimize machining time where constraints of the process are assumed as feed rate, depth of cut, cutting speed, surface roughness and cutting force with a constant material removal rate. They concluded that GAs are efficient solution methods for complex optimization problems since they converge very quickly. Solimanpur & Ranjdoostfard (2008) presented a new optimization technique based on genetic algorithms for determination of cutting parameters in machining operations. The proposed approach had multiple fitness functions and the algorithm found multiple solutions. The results are compared with another GA approach and an ANN approach. These comparisons indicated that the proposed approach is both effective and efficient.

In the literature reviewed, it is observed that optimization ability of GAs is strengthened when combined with other techniques. Most commonly, hybrid approaches combining GAs with ANN, Taguchi and RSM are presented for solving manufacturing process optimization problems.

Ozcelik & Erzurumlu (2005) achieved the minimization of the objective function for an injection molding process by employing a hybrid approach involving finite element analysis, statistical design of experiment, response surface methodology and genetic algorithms. Finite element analyses are conducted for combination of process parameters organized using statistical three-level full factorial experimental design. By using the results of finite element analysis, they created a predictive model by response surface methodology. By interfacing this response surface model with genetic algorithm, they obtained the optimum process parameter values.

The parameter design of the Taguchi method, RSM and GA are integrated by Hou, Su & Liu (2007) to set the optimal parameters of a milling process. The orthogonal array experiment is conducted to obtain the response measurements and ANOVA is used to determine the significant parameters. The RSM is then used to build the relationship between the input parameters and output responses and used as the fitness function of the GA approach. The process has two response and these responses are converted into a single fitness function. Finally, GA is applied to find the optimal parameters for the milling process. The results of the experiments indicated that there was a conflict between these responses the proposed method was insufficient to deal with this conflict.

The combination of GA with ANN is commonly used for optimization problems and the use of this hybrid approach in manufacturing process optimization is reviewed in the next section. The more detailed review of GA applications in tube hydroforming process and metal cutting process is given in chapter five and chapter six, respectively.

CHAPTER FOUR
ARTIFICIAL NEURAL NETWORKS

Since the introduction of the first formalized model of a neuron in 1943 by McCulloch and Pitts, there has been a great progress of neurobiology (McCulloch & Pitts, 1943). This progress allowed researchers to build mathematical models of neurons to simulate neural behavior. ANNs can be defined as networks of elementary nodes called artificial neurons or processing elements that are interconnected by direct links called connections and the neurons cooperate to perform parallel distributed processing to solve a desired computational task.

The purpose of this chapter is to give details of ANNs and a survey of recent research on ANN applications in tube hydroforming and metal cutting processes.

4.1 Artificial Neural Networks (ANNs)

Artificial Neural Networks can be put into local search based metaheuristics category which includes simulated annealing, noisy methods, guided local search methods, iterated local search, tabu search, threshold accepting, and variable neighborhood search (Osman, 2002). From a modeling viewpoint, they are mathematical representations of biological nervous systems that can carry out complex cognitive and computational tasks. They are composed of many nonlinear interconnected simple processing elements that are analogous to neurons, and connected via weights that are analogous to synapses. The concepts of distributed, adaptive and nonlinear computing are the core of neural computation. Distributed computation strengths the reliability of the neural network and it enables fault tolerance and high throughput by taking the advantage of co-operative computing. Adaptive computing is the ability to change a system's parameters according to some rule. Since it is an efficient way to search for optimal performance, the network can respond in a repetitive manner to absolute quantities.

The modern age of neurocomputing started with the work of McCulloch & Pitts (1943) in which the first mathematical model of a single biological neuron was presented. Although McCulloch and Pitts' study showed that simple type of neural networks were able to learn arithmetic or logical functions, ANN algorithms have been successful enough for many applications in the mid 1980s (Potvin & Smith, 2003). The field attracted the attention of many researchers from different disciplines such as engineering, physics, mathematics, computer science and medicine. In recent years, ANNs have become popular in various real world applications including prediction and forecasting, function approximation, clustering, speech recognition and synthesis, pattern recognition and classification, and many others.

4.1.1 Basic Concepts

A neural net consists of a large number of simple processing elements called neurons. Each neuron is connected to other neurons by means of directed communication links, each with an associated weight. The weight represents information being used by the net to solve a problem.

Each neuron has an internal state, called its activation or activity level, which is a function of the inputs it has received. Typically, a neuron sends its activation as a signal to several other neurons.

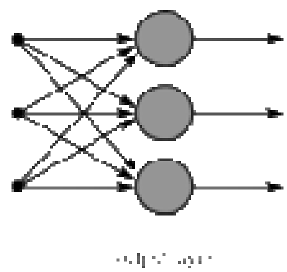
Often, it is convenient to visualize neurons as arranged in layers. Typically, neurons in the same layer behave in the same manner. Key factors in determining the behavior of a neuron are its activation function and the pattern of weighted connections over which it sends and receives signals. The arrangement of neurons into layers and the connection patterns within and between layers is called the network architecture. Many neural nets have an input layer in which the activation of each unit is equal to an external input signal (Akyol,2006).

Neural nets are often classified as single layer, multi layer or competitive networks. Figure 4.1 are examples of these networks.

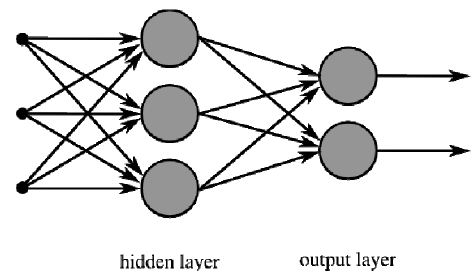
- Single-layer networks: A single-layer network has one layer of connection weights. Often, the units can be distinguished as input units, which receive signals from the outside, and output units, from which the response of the network can be read.

- Multi-layer networks: A multi-layer network is a network with one or more layers between the input layer and the output layer. Multi-layer networks can solve more complicated problems than can single-layer networks, but training may be more difficult.

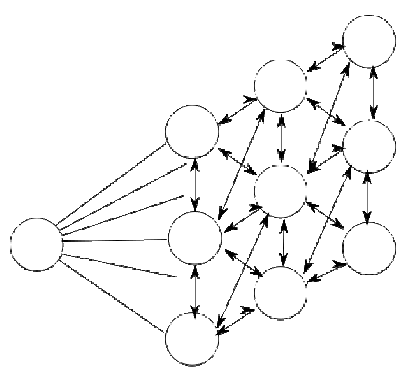
- Competitive layer networks: A competitive layer forms a part of a large number of neural networks. Typically, the interconnections between neurons in the competitive layer are not shown in the architecture diagrams.



(a) Single layer



(b) Multi layer



(c) Competitive

Figure 4.1 Examples of ANNs (from Fausett, 1994)

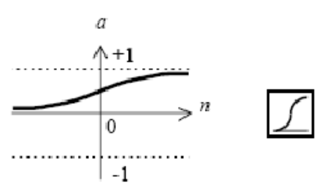
In addition to the architecture, the method of setting the values of the weights (training) is an important characteristic of neural networks. There are two types of training:

- Supervised training: Training is accomplished by presenting a sequence of training data set. The weights are then adjusted according to a learning algorithm.

- Unsupervised training: Self-organizing neural networks group similar inputs together without the use of training data. A sequence of inputs is provided but no targets are specified. The network modifies the weights so that the most similar inputs are assigned to the same output.

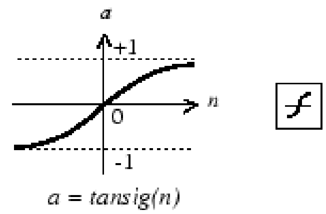
The basic operation of a neuron involves summing its weighted input signal and applying an activation/transfer function. Typically, the same activation function is used for all neurons in any particular layer of a neural network, although this is not required. Activation functions commonly used are:

- *Logarithmic-sigmoid transfer function:* This function generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity.



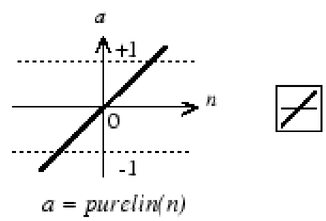
$a = logsig(n)$
Log-Sigmoid Transfer Function

- *Tangent-sigmoid transfer function:* This function generates outputs between -1 and 1 as the neuron's input goes from negative to positive infinity.



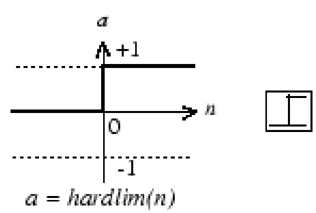
Tan-Sigmoid Transfer Function

- *Linear transfer function:* The outputs of this function ranges between positive and negative infinity.



Linear Transfer Function

- Hard-limit transfer function: this function limits the output of the neuron to either 0, if the net input argument *n* is less than 0; or 1, if *n* is greater than or equal to 0.



Hard-Limit Transfer Function

In the literature of optimization of manufacturing process parameters, ANNs have attracted much attention because of their characteristics listed below.

- ANNs capture the complex relationship between the input and output variables that are difficult or impossible to analytically relate after they are exposed to examples of the relationship, that is, after they learned. After they learned the unknown correlation between the input and output data, they can generalize to predict or classify for cases they were not exposed to.

- In some cases of designing manufacturing systems, ANNs are preferred to time consuming simulation approaches.

- Backpropagation Networks (BPNs) are used to select a manufacturing strategy to achieve accurate estimations of parameters such as values of machining parameters. They are used to estimate the system performance measures such as production cost, production rate etc.

- In static environments, it is possible to obtain the optimal or near optimal parameter settings by mathematical modeling, dynamic programming or other advanced methods. But, since real manufacturing environments are dynamic, flexible methods are needed to react any change in the system that varies with time. So, in dynamic manufacturing environments, ANNs are employed to reduce the need for re-optimizing parameters.

- While optimizing networks such as Hopfield network and its extensions are involved directly in the optimization by mapping the objective functions of manufacturing processes to be optimized and constraints of the problems on to these networks, competitive networks can detect regularities and correlations in input vectors and adapt future responses accordingly (Min, Yih, & Kim, 1998).

Problem of manufacturing process parameters is a non-linear optimization problem with constraints, so it is difficult for the conventional optimization algorithms to solve such problems because of problems of convergence speed or accuracy. Several applications of ANN-based solution approaches for solving manufacturing process optimization problems are reported in the literature.

4.2 Types of Artificial Neural Networks

4.2.1 Backpropagation Neural Networks

One of the important types of networks used in manufacturing applications is backpropagation neural network which is a multi-layer, feed forward network trained by backpropagation. It consists of an input layer, one or more hidden layers and an output layer. Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs. This is why they are called feedforward neural networks.

A backpropagation feedforward neural network, as given in Figure 4.2, has the following characteristics:

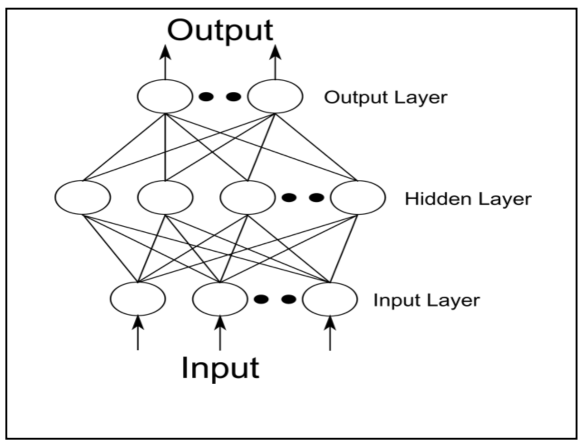


Figure 4.2 A feedforward backpropagation network
(from Su & Wu, 2001)

- Neurons are arranged in layers, with the first layer taking in inputs and the last layer producing outputs. The middle (hidden) layers have no connection with the external world, and hence are called hidden layers. While the number of neurons in input and output layers can be determined exactly according to the dimensions of input and output data set, the number of neurons in hidden layer is selected by trial-and-error.

- Each neuron in one layer is connected to every neuron on the next layer. These connections are not all equal; each connection may have a different strength or weight. Hence information is constantly "fed forward" from one layer to the next and this explains why these networks are called feed-forward networks.

- There is no connection among neurons in the same layer.

Training in feedforward networks is supervised learning, in which pairs of input and output values are fed into the network for many cycles, so that the network 'learns' the relationship between the input and output. In supervised learning the most popular learning law is backpropagation. Backpropagation, which was first introduced by Werbos (1974) and was later rediscovered independently by Parker (1985) and Rumelhart, Hinton, & Williams (1986), and then modified in various manners by numerous researchers in order to overcome its deficiencies, is one of the most popular algorithms for training multilayer perceptrons. This learning rule is a kind of gradient descent technique with backward error propagation, used to adjust the neural weights of a multilayer perceptron. Multilayered perceptrons trained with backpropagation learning algorithm are generally referred to as backpropagation networks. In backpropagation training, the weights of the network are randomly initialized before training starts. Every time an input of a training sample is presented, by propagating through the network layer by layer, a set of outputs is produced as the actual outputs of the network. At the output layer, the actual outputs are compared to the desired outputs, and an error signal is computed by getting the difference between the actual value and the desired value. This error signal is propagated backward through the network and the weight values are then adjusted by a magnitude proportional to the negative gradient of the error function, which is generally equal to the sum of squared errors. By this way, the difference (mean square error) between the actual and the desired outputs is minimized.

A successful simulation of the backpropagation networks requires the determination of some parameters:

- *Size of data Set:* The available data set is generally separated into three subsets; called training set, testing set and cross validation set. The first subset is the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The validation error will normally decrease during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise and training terminates. The last subset, testing set is used to test the network's performance.

- *Layer Configuration:* It is necessary to define layers and neurons in each layer. The number of neurons in the input (output) layer will be equal to the number of inputs (outputs). Since, the number of hidden layers will be determined by trial-and-error, it is highly recommended to start with single hidden layer.

- *Transfer Function:* Transfer function describes the way in which information, or data, flows through the network. Each neuron receives weighted input from every other neuron in the network, applies a non-linear threshold and presents its output for the others to input.

- *Learning rule:* Learning rule means the correction term by which the weights are changed based on their previous value.

    - *Step size:* Most gradient search procedures require the selection of a step size. The idea is that the larger the step size the faster the minimum will be reached. However, if the step size is too large, then the algorithm will diverge and the error will increase instead of decrease. If the step size is too small then it will take too long to reach the minimum, which also increases the probability of getting caught in local minima.

- *Learning rate:* It is a changeable value used by several learning algorithms, which effects the changing of weight values. The greater the learning rate, the more the weight values are changed. Is usually decreased during the learning process.

- *Stopping Criteria:*

  - *Epoch:* Epoch is a term that is often used in the context of machine learning. An epoch is one complete presentation of the *data set to be learned* to a network. Learning machines like feedforward neural nets that use iterative algorithms often need many epochs during their learning phase.

  - *Termination:* The stop criterion for supervised training is usually based on the mean squared error (MSE). Most often the training is set to terminate when the MSE drops to some threshold. Even though the MSE of the training set will keep decreasing throughout the simulation, at some point the MSE of the test set will begin to rise. This is an indication that the network has begun to overtrain or "memorize" the training patterns. Thus, cross validation is a highly recommended criterion for stopping the training of a network.

- *Performance Functions:* This parameter is used to evaluate the performance of the network. Most commonly used and default performance function is MSE that is the average squared error between the network outputs and the target outputs.

  - *Mean Square Error:* The mean squared error is simply two times the average cost. The formula for the mean squared error is:

$$MSE = \frac{\sum_{j=0}^{P}\sum_{i=0}^{N}(d_{ij} - y_{ij})^2}{N\ P} \tag{4.1}$$

where P is the number of output processing elements, N is the number of exemplars in the data set, y_{ij} is the network output for exemplar i at neuron j and d_{ij} is the desired output for exemplar i at neuron j.

- *Normalized Mean Square Error:* The normalized mean squared error is defined by the following formula:

$$NMSE = \frac{P\ N\ MSE}{\sum_{j=0}^{P}\frac{N\sum_{i=0}^{N}d_{ij}^2 - \left(\sum_{i=0}^{N}d_{ij}\right)^2}{N}} \tag{4.2}$$

where P is the number of output processing elements, N is the number of exemplars in the data set, MSE is the mean squared error and d_{ij} is the desired output for exemplar i at neuron j.

- *Correlation Coefficient:* The size of the mean square error (MSE) can be used to determine how well the network output fits the desired output, but it doesn't necessarily reflect whether the two sets of data move in the same direction. The correlation coefficient (r) solves this problem. By definition, the correlation coefficient between a network output x and a desired output d is:

$$r = \frac{\frac{\sum_i (x_i - \bar{x})(d_i - \bar{d})}{N}}{\sqrt{\frac{\sum_i (d_i - \bar{d})^2}{N}}\sqrt{\frac{\sum_i (x_i - \bar{x})^2}{N}}} \tag{4.3}$$

The basic algorithm of backpropagation training is as follows (Fausett, 1994):

Step 0. Initialize weights (Set to small random numbers)

Step 1. While stopping condition is false, do Steps 2-9.

Step 2. For each training data, do Steps 3-8.

Feedforward:

Step 3. Each input unit (X_i, $I=1,2,\ldots,n$) receives input signal x_i and broadcast this signal to all units in the layer above (the hidden neurons).

Step 4. Each hidden unit (Z_j, $j=1,2,\ldots,p$) sums its weighted input signals,

$$z\_in_j = v_{0j} + \sum_{i=1}^{n} x_i v_{ij} \tag{4.4}$$

applies its activation function to compute its output signal.

$$z_j = f(z\_in_j) \tag{4.5}$$

and sends this signal to all units in the layer above (the output neurons)

Step 5. Each output unit (Y_k, $k=1,2,\ldots,m$) sums its weighted input signals,

$$y\_in_k = w_{0k} + \sum_{j=1}^{p} z_j w_{ik} \tag{4.6}$$

and applies its activation function to compute its output signal,

$$y_k = f(y\_in_k) \tag{4.7}$$

Backpropagation of error:

Step 6. Each output unit (Y_k, $k=1,2,\ldots,m$) receives a target (desired) pattern corresponding to the input training pattern, computes its error information term (error signal),

$$\delta_k = (t_k - y_k) f'(y\_in_k) \tag{4.8}$$

calculates its weight correction term (used to update w_{jk} later),

$$\Delta w_{jk} = \alpha \delta_k z_j \tag{4.9}$$

calculates its bias correction term (used to update w_{0k} later),

$$\Delta w_{0k} = \alpha \delta_k \tag{4.10}$$

and sends δ_k to units in the layer below.

Step 7. Each hidden unit (Z_j, $j=1,2,\ldots,p$) sums its delta inputs (from neurons in the layer above)

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \tag{4.11}$$

multiplies by the derivative of its activation function to calculate its error information term,

$$\delta_j = \delta\_in_j f'(z\_in_j) \tag{4.12}$$

calculates its weight correction term (used to update v_{ij} later),

$$\Delta v_{ij} = \alpha \delta_j x_i \tag{4.13}$$

and calculates its bias correction term (used to update v_{0j} later),

$$\Delta v_{0j} = \alpha \delta_j \tag{4.14}$$

Update weights and biases:

Step 8. Each output unit (Y_k, $k=1,2,\ldots,m$) updates its bias and weights ($j=0,1,\ldots,p$):

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{kj} \tag{4.15}$$

Each hidden unit (Z_j, $j=1,2,\ldots,p$) updates its bias and weights ($i=0,1,\ldots,n$):

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \tag{4.16}$$

Step 9. Test stopping condition (Fausett, 1994).

Backpropagation networks have been successfully used in modeling, classification, forecasting, design, control, and pattern recognition. Their improved generalization capabilities over competing machine learning tools and their easy mechanism made them attractive to be utilized in optimization of manufacturing processes.

In 1986, the modern era of neural networks was ushered in by the derivation of back propagation by Rumelhart, Hilton & Williams (1986). Then, Rangwala & Dornfeld (1989) applied backpropagation network to predict optimal conditions (cutting parameters such as cutting speed, feed rate and depth of cut) in turning operations by minimizing a performance index.

The first application of neural networks in monitoring reported in the literature could be that of Gövekar, Grabec & Peklenik (1989). The authors applied a back-propagation network for the monitoring of a drilling process. Their results show that the recognition ability was influenced not only by the neural network but also by the properties of the detected system. Later applications (Monostori & Nacsa, 1990; Nacsa & Monostori, 1990) show that neural networks can advantageously be used in real-time monitoring of manufacturing processes and other technical processes.

Anderson, Cook & et al. (1990, 1991) applied the neural network approach in arc welding that can generally be viewed as a multiple-input/multiple-output system. The back-propagation networks were used for the modeling and control of the process. Various configurations, in terms of the number of layers and the number of network nodes, were tested. For the application presented, two-layer (not counting the input layer) networks consisting of a single hidden layer and an output layer have been proved to be adequate. Smartt, Johnson, Einerson & Cordes (1991) also applied the neural network approach in arc welding. Instead of using neural network to model the arc welding processes, they developed a new approach to quantify conditional logic rules and represent them in a neural network.

Smith (1991) reported the use of back-propagation neural networks in quality control in an injection molding corporation. They proposed that since neural networks are especially applicable when the data considered do not follow a known distribution or pattern, they are well suited for the quality control of injection molding. The results show that the neural network approach is comparable to other quality control methods, including control charts and statistical techniques, in goodness of output for quality control. They concluded that an advantage of the proposed neural network approach is the convenience of learning to establish the relationships directly, rather than through analysis and assumptions. Using a single network to monitor multiple products and/or quality parameters is an additional advantage.

As further study, Guillot & El Ouafi (1991) applied a three-layer feedforward neural network in the identification of tool breakage in metal cutting processes. Wu, Liou & Pi (1991) presented a neural network approach to diagnose processing damages in injection molding. Sathyanarayanan, Lin & Chen (1992) employed artificial neural networks with backpropagation for modeling a typical creep feed super alloy-grinding.

Cook & Shannon (1992) presented a methodology to predict the occurrence of out-of-control process conditions in a composite board manufacturing facility. This method was developed using neural network theory and the neural network, using back-propagation method, was successfully trained to represent the process parameters. The trained neural network was able to successfully predict the state of control of the specific manufacturing process parameters with 70% accuracy. The learning rule used in this research was the generalized delta rule which is an error-correcting rule that has been used in various applications including converting printed text to speech, controlling robot arms, and selecting good loan applications. Hou & Lin (1993) designed a monitoring system for identifying process signals using neural networks. Two examples were presented to demonstrate the feasibility of the monitoring system and its recognition ability. The results are quite promising and show that the neural network based system seems to have a good potential in monitoring automatic manufacturing processes. Hyun & Cho (1994) predicted the forming pressure for hydroforming process using artificial neural network. In the article, a back-propagation network was implemented to learn the mapping characteristics and then estimated the forming pressure in the chamber from the geometric variables of the punch. Anjum, Tasadduq & Khaled (1997) proposed two-stage procedure for obtaining the best parameter design based on implementing response surface methodology via neural network. Applying the method, the neural network was trained by the results of a fractional factorial design, and was then used to estimate the response values for the full factorial design.

Coit, Jackson & Smith (1998) aimed at the technology transfer aspects of neural networks to manufacturing process modeling and optimization by focusing on two highly non-linear processes where there are many variables which affect the ultimate outcome. These are wave soldering of printed circuit boards and slip casting of large ceramic products. After careful validation of the prediction accuracy over the entire range of anticipated operating conditions, the final neural network models have been implemented at the manufacturing plants. To map the relationship between die casting process parameters and the injection time, Yarlagadda & Chiang (1999) have developed a multi-layer feed-forward network using three different algorithms, namely the error back-propagation algorithm, the momentum and adaptive learning algorithm, and Levenberg–Marquardt approximation algorithm. The characteristics of the three algorithms were analyzed.

Smith (2000) offered neural networks for the solution powder metallurgy problems that are too complex for standard statistical methods because of the numerous variables involved and the non-linearity of the relationships. In a further study, Yarlagadda (2001) analyzed the effect of process parameters on injection molding process based on governing equations of the mold filling process by using neural networks. Ohdar & Pasha (2003) employed a three layer neural network with backpropagation algorithm for controlling the properties, particularly the density, of metal powder perform. Once they predicted the density for various combinations of input variables such as compacting pressure, sintering temperature and percent reduction, they selected the appropriate combination of input variables corresponding to the desired density.

Risbood, Dixit & Sahasrabudhe (2003) and Grzesik & Brol (2003) examined the use of artificial neural networks to estimate and control the surface roughness during machining processes where Bisht, Gupta, Pal & Chakraborty (2005) proposed a multi-layer feedforward neural network model employing backpropagation for the prediction of flank wear in turning operations by inputting cutting force ratio, feed rate and cutting speed.

Cus, Zuperl & Milfelner (2006) used supervised networks to successfully estimate the cutting forces developed during the milling process. The predictive capability of using analytical and neural network approaches is compared. They built a multi-layer feedforward network with backpropagation training method. They made an extensive number of tests on the milling machine to confirm the neural model with different cutting parameters and concluded that the proposed model can be used for simulation purposes and for monitoring and optimization of the cutting parameters in machining process.

Karunakar & Datta (2007) developed a neural network to simulate the relationship between sand casting parameters like green compression strength, permeability, moisture percent, composition of the charge, and melting conditions as inputs and the presence/absence of defects as outputs. The set of inputs of the casting that is going to be made was fed to the network, and the network could predict whether the casting would be sound or defective. The causes for the defects were investigated, and the defects were prevented by altering the molding parameters. Belhadj, Abbassi, Mistou & Zghal (2010) proposed a multi-layered feed forward employing backpropagation to predict the thickness and the forming pressure tube hydroforming process. They concluded that the variation thickness is very sensitive to the variation of the input parameters and they proposed to improve training algorithms or to make training especially in the desired zone based on a numerical experimental design.

When the articles reviewed above are considered, the success of most of the studies are the result of the good generalization capabilities of backpropagation networks which are used to capture the complex relationship between the input and output variables of the considered manufacturing process. Thus, it can be said that backpropagation networks are not directly involved in the optimization problem. That is, actual optimization of manufacturing processes is not performed. However, there are few studies aimed to optimize manufacturing processes by using artificial neural networks.

Su & Hsieh (1998) proposed an approach based on neural networks to solve the quality optimization problem in Taguchi's dynamic experiment. However, only a single response is addressed and the effects of the control factors on response are not presented. As a further study, Tong & Hsieh (2001) proposed a novel means of applying artificial neural networks to solve the multi-response optimization problem by combining the quantitative and qualitative response. Although parameter optimization can be obtained, they were not able to achieve to analyze the effect of control factors on multiple responses. In lieu of the above methods, Hsieh (2006) proposed a complete procedure based on an artificial neural network to perform optimization of the multi-response problem in arbitrary experimental design. They studies two cases: one for level combination of control factors, the other for both mixed and level settings of control factors. They concluded that no matter whether the control factors are due to the level form or the real value, the proposed procedure can be utilized. At the same time, the effect of the control factors on responses is also obtained by the proposed approach. However, their study is focused on only the quantitative responses and is not applicable for qualitative responses.

Zuperl & Cus (2003) and Zuperl & Cus (2006) developed a neural optimization algorithm to reach higher precision of predicted results and efficient optimization of turning parameters. For assessing the multi-attribute value function, they built a feedforward neural network and compared the performance of this network with radial basis networks. They employed adaptive learning algorithm and for the optimization phase, they defined the area, in which extreme of the function is reached, by using limitation equations. According to the experiments done, they concluded that feedforward neural networks provide more accurate results but they require more time for training and testing.

Although the popularity of backpropagation networks has grown significantly in the past few years, some problems still exist with the application of the backpropagation networks. That is, these networks are trained by a gradient based search technique which has the risk of getting stuck in local optimum and the starting point of the connection weights becomes an important issue to reduce the possibility of being trapped in local optimum. Another difficulty with the construction of these types of networks is the necessity of generating a training set which is time consuming. Therefore, in recent years, the performance of these networks is tried to be enhanced by combining them with different heuristics or meta-heuristics as described in Section 4.3.

4.2.2 Hopfield Networks

Hopfield networks are one of the well-known dynamic systems used for optimization problems. The original Hopfield NNs, which consist of a fully connected network of neurons capable of performing computational tasks, were introduced by Hopfield (1982). Using binary state neurons and a stochastic algorithm to update the neurons, this network served as a content addressable memory that allows for the recall of data based on the degree of similarity between the input pattern and the patterns stored in the memory. This model is known as the discrete and stochastic Hopfield model. An example of Hopfield network is given in Figure 4.3.

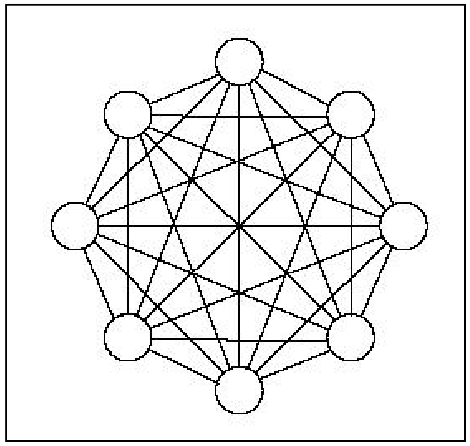


Figure 4.3 An example of Hopfield network
(from www.learnartificialneuralnetworks.com)

In this non-layered recurrent network, the connection weights are assumed to be symmetric ($w_{ij}=w_{ji}$) and they store information about the states of the network. The Hopfield network employs Hebbian learning where weight between two neurons is strengthened/weakened in case an excitatory/inhibitory connection exists. Thus, the weight will be zero in case of no interaction. . In the case of an excitatory connection, the weights will take positive values; they will be negative in the case of an inhibitory connection or they will be zero in the case of no interaction (Akyol, 2006).

Each neuron (i) is described by an internal and an external state. The internal state (net input value) of each neuron is represented by u_i, while the external state (output value) by v_i. In this model, the internal states are continuous and the external states are binary. The input of each neuron comes from two sources, external inputs I_i and inputs from other connected neurons. The relationship between the internal and external states of the neurons is represented by the following McCulloch and Pitts dynamics rule (Akyol, 2006).

$$u_i(t+1) = \sum_{j=1}^{n} w_{ij} v_j(t) + I_i \tag{4.16}$$

$$v_i(t+1) = f(u_i) = \begin{cases} 1 & \text{if } u_i > 0 \\ 0 & \text{if } u_i \le 0 \end{cases} \tag{4.17}$$

The internal state of a neuron is found by taking the weighted sum of the external states of all connecting neurons with a constant external input to that neuron. In Eq. (4.16), t is a discrete time; w_{ij} is the synaptic interconnection strength from neuron j to i, f is the activation function between u_i and v_i and can take several forms. It can be the unit step function as defined by the Eq. (4.17).

In a Hopfield model, the states of the neurons are updated in a random manner. The objective function and the problem constraints are mapped onto a quadratic function that represents the energy of system of neurons.

$$E = \frac{-1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} v_i v_j - \sum_{i=1}^{N} I_i v_i \tag{4.18}$$

The aim is to obtain a configuration minimizing the energy function. Hopfield has proved that with symmetrical weight matrix and non-negative elements on the diagonal of the weight matrix, the energy function, by performing gradient descent, minimizes until convergence to stable states which represent the local minimum values of the energy function.

After the original discrete stochastic model based on McCulloch-Pitts neurons was introduced, in a later work, Hopfield (1984) proposed a deterministic model based on continuous neurons. The idea was inspired by the fact that the neurons of the original model were different than the real biological neurons and from the realistic functioning of electronic circuits. So by maintaining the important properties such as content- addressable memory of the original model, a new model is constructed. The continuous Hopfield model given in Hopfield (1984) is represented by the following resistance-capacitance differential equation to model the capacitance and resistance of a real neuron's cell membrane. In this model, the dynamics of each neuron i may be defined as below.

$$\frac{du_i}{dt} = \sum_{j=1}^{N} w_{ij} v_j - \frac{u_i}{\tau} + I_i \tag{4.19}$$

$$v_i = f_i(u_i) = \frac{1}{2}\left(1 + \tanh\left(\frac{u_i}{T}\right)\right) \tag{4.20}$$

$$\tau = RC \tag{4.21}$$

where t is a continuous time, f is a continuous sigmoidal transfer function that determines the relationship between the internal state of a neuron and its output level, R is the trans-membrane resistance, C is the input capacitance, T is a parameter to control the slope of the transfer function and τ is the value of time constant of the amplifiers. In this model, the external states are ranged between 0 and 1, and are continuous.

The idea of using ANNs to provide solutions to NP-hard optimization problems was pioneered by Hopfield & Tank (1985) with the use of their network for solving the Traveling Salesman Problem (TSP). In their paper, Hopfield & Tank showed that if an optimization problem can be represented by an energy function, then a Hopfield network that corresponds to this energy function can be used to minimize this function and thus provides an optimal or near-optimal solution. Since then, because of the advantages of using Hopfield networks, extensive research has been carried out on the application of the Hopfield networks for solving different optimization problems. Massive parallelism and convenient hardware implementation of the network architecture are among the most important advantages of Hopfield networks.

In this network, objective function and the problem constraints are encoded in terms of an appropriate energy function. The aim is to obtain a configuration minimizing the energy function. Translation of the optimization problem into an appropriate energy function is in general, a difficult task. It must be in a quadratic form to meet the form of the energy function of the Hopfield network. Applying the most common method, penalty function approach, the energy function of the network is set equivalent to the objective function of the problem, and the problem is reduced to an unconstrained form by including the constraints of the problem in the energy function as penalty terms. By this way, the constraint violations are penalized.

Since there is a risk that backpropagation networks get stuck in local optimum and the success of Hopfield &Tank in solving a Traveling Salesman Problem encouraged many optimization researchers to employ Hopfield networks in optimization. Any optimization problem that can be defined by a quadratic form can be tackled with Hopfield networks. However, Hopfield NNs have some shortcomings:

- They do not guarantee the feasibility. A Hopfield network whose energy function reaches its minima at the same points with the cost function that describes the optimization problem must be designed. By performing gradient descent on the energy function, the Hopfield model gets easily trapped in local minimum states, and this causes decreasing efficiency especially in large sized problems.

- The performance of Hopfield network is very sensitive to the initial configuration of the network. Determining the penalty coefficients requires a tedious trial and error process. It requires a large number of iterations to converge to a solution.

As the earliest study, Hopfield (1982) used binary state neurons and a stochastic algorithm to update the neurons and the proposed network served as a content addressable memory that allows for the recall of data based on the degree of similarity between the input pattern and the patterns stored in the memory. This model is known as the discrete and stochastic Hopfield model. In a later work, Hopfield (1984) proposed a deterministic model based on continuous neurons. The idea was inspired by the fact that the neurons of the original model were different than the real biological neurons and from the realistic functioning of electronic circuits. So by maintaining the important properties such as content-addressable memory of the original model, a new model is constructed. In order to get better results and to reduce the number of neurons required to solve the same problem, Foo & Takefuji (1988c) introduced integer linear programming networks as extensions of the original Hopfield network, and achieved better solutions.

In the relevant literature, it is seen that Hopfield networks are commonly employed for scheduling problems. To the best of our knowledge, there is no established research based on Hopfield networks for the optimization of manufacturing process parameters.

4.2.3 Competitive Networks

The works by Grossberg (1972), von der Malsburg (1973), Fukushima (1975), Willshaw & von der Malsburg (1976), and Grossberg (1976 a, b) are the first in the area of competitive learning. Unlike Hopfield networks, the winner take all strategy forms the base of the competitive networks. Winner-take-all is a computational principle applied in computational models of neural networks by which neurons in the output layer compete with each others to be activated with the result that only one output neuron, is on at any time.

In this unsupervised network, as in Figure 4.4, there is a single layer of output neurons fully connected to the input neurons of the network. In this output layer known as the competitive layer, lateral inhibition occurs among the neurons and output nodes in the network mutually inhibit each other, while simultaneously activating themselves through reflexive connections. After some time, only one node in the output layer will be active, namely the one corresponding to the strongest input. For an input pattern presented to the network, the neuron with the weight vector at the least distance from the input vector is called the strongest or winner and its output is set to one.

In the literature reviewed, it is seen that most applications of competitive networks in manufacturing focus on grouping technology for manufacturing cell problems. The application of maximum network is studied by Knapp & Wang (1992)

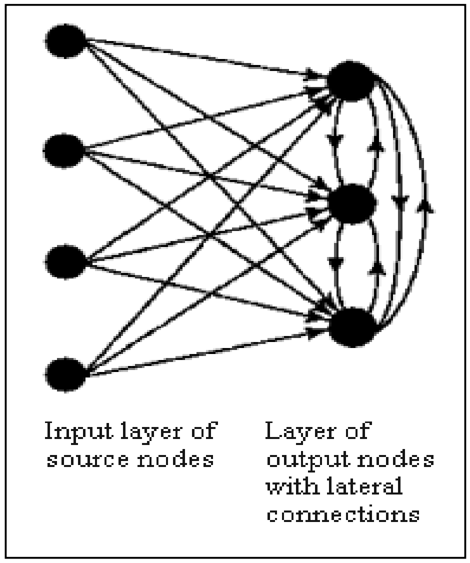


Figure 4.4 an example of competitive network

(from rslab.movsom.com)

to automate the process selection and task sequencing in machining processes. In their approach, two cooperating neural networks are utilized. The primary network is a three-layer backpropagation network. The second fixed-weight network utilizes the MAXNET architecture. The primary network is a three-layer backpropagation network built for generation of the operation alternatives. The second fixed-weight network utilizes the MAXNET architecture that was used to make a decision among competing operation alternatives. In the last stage, the output of the MAXNET was fed back to the input layer of the primary network to provide a basis for deciding the next operation in the machining sequence.

4.3 Hybrid Approaches

In recent years, besides their advantages of parallelism, learning, generalization capability, nonlinearity, and robustness, several limitations of ANNs such as settlement into local minima, trial and error parameter determination process, long learning time are perceived. To compensate their disadvantages, hybrid systems in which ANNs are combined with different computing techniques have been proposed in the literature. As a result, to enhance the performance of the neural networks, there has been an explosive growth in the successful use of hybrid neural networks in optimization. In this section, we review the hybrid studies in manufacturing exploiting the combinations of neural networks with different approaches.

One of the most common hybrid approaches involves artificial neural networks and Taguchi method. Taguchi approaches are used to provide input data to the neural network models. Rowlands, Packianather & Oztemel (1996) presented an integrated method to illustrate how optimum parameter design can be achieved by using design of experiments in conjunction with neural networks. They used NN to provide data for a full-factorial experiment, using the results of a fractional factorial designed experiment for training. As a similar study, Chiu, Su, Yang, Huang, Chen & Cheng (1997) used the neural network model and the Taguchi method to determine the optimum parameter setting in an injection molding process. The results showed that the integrated method is capable of treating continuous parameter values. The ANN and the Taguchi method have also been implemented for minimizing an objective function relevant to the forming process by Ko, Kim, Kim & Choi (1998). The combinations of process parameters are selected by orthogonal array. The results of simulation corresponding to orthogonal array are used as training data for the ANN to obtain optimal conditions. Their study concluded that the proposed method gave more systematical and economically feasible strategies to design metal forming processes.

Sukthomya & Tannock (2005) utilized historical process data, to train a NN to model the actual production process. The NN is trained to learn the relationship between process parameters and process response, using process parameters as training input, and process response as training output. After suitable training, the NN is used to provide a reasonable approximation of output for previously unseen process input. They concluded that the proposed NN-Taguchi approach could be used to estimate the results of Taguchi experiment settings, without conducting any actual experimentation and could be used in integration with optimization techniques to find the optimum process parameter settings. To optimize multiple responses simultaneously, an N-D method by using artificial neural network and data envelopment analysis is presented by Liao (2005). In the first phase, ANNs are primarily used to predict the multi-response values of all factors/levels combinations and in the second phase, data envelopment analysis is primarily used to multi-choose the optimal factors/levels combination. They resolved two case studies available in the literature, and results obtained indicated that the proposed approach provided an efficient and feasible solution to the multiple-response problem considered. For solving multiple response problems in parameter design problems, Antony, Anand, Kumar & Tiwari (2006) proposed a four-step procedure. They optimized three responses for an electronic assembly operation. Multiple S/N ratios are mapped into a neuro-fuzzy model to identify the optimal level settings for each parameter. Finally, they employed ANOVA analysis to identify parameters significant to the process.

In order to handle one of the most important shortcomings of ANN, trapping in local minima, ANNs are commonly combined with GAs. As one of the major studies, Cook, Ragsdale & Major (2000) showed the use of an integrated NN-GA approach to determine the optimal process parameter values needed under different conditions and at various stages of the process. A neural-network model was developed to predict the value of critical strength parameters in a particle- board manufacturing process, on the basis of process operating parameters and conditions. A genetic algorithm then used the trained neural-network prediction model to determine the process parameter values that would result in the nearest to optimal value of the strength parameters that could be obtained under various operating conditions.

Another illustration of ANN and GA hybrid methods is presented by Su & Chiang (2003) on wire bonding process. The backpropagation neural network is used to provide the non-linear relationship between factors and the response based on the experimental data. Then, they applied GA to obtain the optimal factor settings of the process. The comparison between the proposed approach and the Taguchi method demonstrated the superiority of the proposed approach in terms of process capability. Following this work, Li, Su & Chiang (2003) integrated desirability functions into the proposed network since most of manufactured products have more than one quality characteristics and the quality characteristics are generally correlated with each other. By an illustrative example on silicon manufacturing process, it is concluded that the proposed approaches provided a compromise and composite solution.

To decrease the number of experiments Huang & Tang (2006) developed a decision making system combining ANN, GA and Taguchi method. They proposed a systematic approach for parameter values in melt spinning to arrive at the optimal product quality. The experimental layout in the Taguchi method provided training samples of neural network and optimal parameter conditions are decided by genetic algorithm. Yang, Lin & Chen (2006) presented a new method to search the optimal loading paths of forming process using ANN and GA. In the study, ANN was used to map the relationship between the design variables and thickness variation. Then GA was applied to search the optimal internal pressure and axial displacement by using ANN as a solver of the proposed objective function. However, the final part should be formed completely and without any failures at the final step, thus the constraint functions in this case needed to be applied to control the final part dimensional accuracy and failures. To deal with this problem, Yong, Chan, Chunguang & Pei (2009) proposed a hybrid method to optimize loading path of forming process. They proposed two constraint functions: One is for checking wrinkling and the final part dimensional accuracy, and another is for controlling the fracture and excessive thinning. On the basis of the given parameter spaces, they used GA to search the global optimum of loading paths in combination with the trained ANN.

In their study, Sen & Shan (2006) proposed a hybrid approach for optimal selection of machining conditions in drilling process. The proposed approach first used a backpropagation neural network to formulate a fitness function for predicting the response parameter of the process. From the network output, the desirability method obtained a composite fitness function for further use in the genetic algorithm. Then, the genetic algorithm predicted the optimal input combinations and simultaneously optimizes the multi-response characteristics of the process. However, the simulation results indicated that the proposed approach is largely sensitive to the relative preference of weighting factor used in forming the composite function and the optimal solution obtained by this approach is quite subjective.

As an example of combination of ANNs with desirability functions, Hsu, Su & Liao (2004) presented an integrated procedure using neural networks and exponential desirability functions to solve a multi-response parameter design problem. They used a backpropagation neural network to build response model of the dynamic multi response experimental data. As an extension of their study, Hsu (2004) presented a hybrid approach involving Tabu search to optimize the manufacturing process. The confirmation results demonstrated the practicability and effectiveness of the proposed approach. The approach proposed by Chang (2006) employed a backpropagation network to construct the response model of the dynamic multi-response system by training the experimental data. The response model is then used to predict all possible responses of the system by presenting full parameter combinations. Finally, the best parameter setting is obtained by maximizing the objective value and by an illustrative example they demonstrated the effectiveness of the proposed approach.

For metal forming processes, ANNs are combined with a most commonly used method, Finite element method to analyze the forming process. Raj, Sharma, Strivastava & Patvardhan (2000) investigated the applicability and relative effectiveness of ANN based models for rapid estimation. The results obtained are found to correlate well with the finite element simulation data in cases of metal forming, and experimental data in cases of metal cutting. They concluded that their study had considerable implications in selection of the tools and on-line monitoring of tool wear and thus prevent damage to the tool and work piece during the course of manufacturing.

Besides backpropagation neural networks, Hopfield networks also combined with different approaches to overcome its shortcomings. From optimization viewpoint, the Hopfield neural network and its extensions belong to the penalty method for solving the constrained real optimization into which a combinatorial optimization is converted. The penalty function requires the weighting factors for the penalty terms to be sufficiently large in order to converge to a feasible solution. But as the penalty terms become stronger, the original objective function becomes weaker, and as they become larger and larger, the problem becomes ill conditioned. To deal with this problem, Li (1996) combined the augmented Lagrange multiplier method and the penalty methods of the Hopfield networks to obtain the augmented Lagrange Hopfield network. By this way, both the solution quality and the convergence properties of the Hopfield network are improved. Thus, the proposed approach helps to overcome the problems associated with the penalty method or the Lagrange multiplier method when used alone (Li, 1996). Following this work, Luh, Zhao, & Wang (2000) proved the convergence of Lagrangian Relaxation Neural Networks (LRNN) for separable convex problems, and constructed LRNN for separable integer programming problems.

4.4 Summary and Future Research

Over the last decade, ANNs have been applied to an increasing number of real-world problems of considerable complexity and to the theoretical test problems. In this chapter, we tried to provide an extensive literature review on the applications of ANNs in manufacturing process optimization. In order to see the gradual development in these works, the recent research studies are summarized in a chronological order. Our survey is limited with the publications appearing in refereed journals and conference proceedings till 2010. Table 4.1 summarizes the manufacturing process applications considered in this research. The conclusions drawn from this detailed review are summarized below:

- Most of the approaches proposed in the reviewed articles are based on artificial neural networks and genetic algorithms and a great emphasis has been given on the optimization or planning of manufacturing processes. The literature presents many variants of traditional ANN approaches and their combinations with different techniques to improve their performance by trying to escape from the local minima, by reducing the computational effort required, by speeding convergence and by decreasing the number of neurons and interconnections.

- Although widely preferred in the literature because of their highly parallel computational capabilities, one of the major problems in the application of artificial neural networks is determining the parameters of network architecture which is commonly achieved by trial and error. Thus, we believe that an important direction of future research is to search for the methods to overcome this trial and error experiments.

- In the last years, ANNs have either been combined with artificial intelligence techniques such as expert systems, with metaheuristics such as GAs, tabu search, simulated annealing or with some heuristic procedures to form hybrid approaches providing superior solutions. As a global search technique, the combination of GAs with ANNs is widely used in obtaining optimal solutions, and considerable success is achieved by overcoming the slow convergence property of GAs and the local minima problem of ANNs. In this thesis, we also used a hybrid optimization approach involving ANN and GA.

- In the neural network design, setting of the parameters, initialization of the weights, configuration of the network are often problem specific and the correct value of these parameters however is not known a priori. Therefore, for any given problem, a wide variety of parameters must be tried to generate confidence that a best solution has been found. Sensitivity of the ANNs to their initial configuration and inability of the gradient based search techniques to find global solutions motivated the researchers to employ EAs together with ANNs for the automatic adjustment of the parameters and the topology of the ANNs.

We believe that in the near future the researchers will benefit from the use of the recent advances in ANNs, metaheuristics, and their combinations. It can be concluded that, the future of ANNs not only lies in their explicit use but also lies in its conjunction with other advanced technologies.

Table 4.1 Evolution of ANNs for optimization of manufacturing processes

Year	Author(s)	Approach	Application area
1986	Rumelhart, Hilton & Williams	Backpropagation network	General application
1989	Rangwala & Dornfeld	Backpropagation network	Turning operation
1989	Govekar et al.	Backpropagation network	Drilling operation
1990	Monostori & Nacsa	Artificial neural networks	Drilling process
1990	Nacsa & Monostori	Artificial neural networks	Drilling process
1990	Anderson et al.	Backpropagation network	Arc welding
1991	Anderson et al.	Backpropagation network	Arc welding
1991	Smartt et al.	Artificial neural networks	Arc welding
1991	Smith	Backpropagation network	Injection molding
1991	Guillot & El Quafi	Multi-layer feedforward network	Metal cutting process
1991	Wu et al.	Artificial neural networks	Injection molding
1992 (a,b)	Knapp & Wang	Multi-layer backpropagation network & Maximum network (MAXNET)	Machining process
1992	Sathyanarayanan, Lin & Chen	Backpropagation network	Alloy-grinding operation
1992	Cook & Shannon	Backpropagation network	Composite board manufacturing

1993	Wang et al.	Artificial neural networks	Wire bonding process
1993	Hou & Lin	Artificial neural networks	Automatic manufacturing processes
1994	Hyun et al.	Backpropagation network	Tube hydroforming process
1996	Rowlands et al.	(A Hybrid approach) Artificial neural network combined with Taguchi approach	Manufacturing processes with multiple responses
1997	Anjum et al.	Two stage artificial neural network	Manufacturing processes with multiple responses
1997	Chiu et al.	(A Hybrid approach) Artificial neural network model and Taguchi method	Injection molding
1998	Coit, Jackson & Smith	Artificial neural networks	Non-linear processes (Wave soldering & Casting)
1998	Su & Hsieh	Two stage feedforward network	Semiconductor manufacturing with one response
1998	Ko, Kim, Kim & Choi	(A Hybrid approach) Artificial neural network combined with Taguchi approach	Forming process
1999	Ko, Kim & Kim	(A Hybrid approach) Artificial neural network combined with Taguchi approach	Forming process

1999	Yarlagadda & Chiang	Multi-layer feedforward using three different algorithms (Backpropagation, Momentum & Adaptive learning, Levenberg-Marquardt approximation)	Die casting
1999	Jain, Jain & Kalra	Artificial neural networks	Machining process
2000	Smith	Artificial neural networks	Powder metallurgy
2000	Cook, Ragsdale & Major	(a Hybrid approach) Combination of neural networks with GAs	Particle board manufacturing
2000	Raj et al.	(a Hybrid approach) Multi-layer network with Levenberg-Marquardt algorithm and Finite element simulation	Metal forming & Machining
2000	Sadeghi	Backpropagation network	Injection molding
2001	Yarlagadda	Artificial neural networks	Injection molding
2001	Hsieh & Tang	Two stage feedforward network with backpropagation	IC manufacturing
2002	Heider, Piovoso & Gillespie	Multi-layer feedforward network with backpropagation	Thermoplastic tow-placement process
2002	Benardos & Vosniakos	(A hybrid approach) Artificial neural networks combined with Taguchi method	Milling process
2002	Feng, Wang & Yu	Artificial neural networks	Honing process
2003	Ohdar & Pasha	Multi-layer feedforward network with backpropagation	forging process
2003	Risbood et al.	Artificial neural networks	Machining process
2003	Grzesik & Brol	Artificial neural networks	Machining process
2003	Zuperl & Cus	Feedforward network & Radial basis network	Turning operation

2003	Su & Chiang	(a Hybrid approach) Backpropagation network combined with GA	Wire bonding process
2003	Li, Su & Chiang	(a Hybrid approach) Hybrid ANN-GA approach integrated with desirability functions	Silicon manufacturing process
2004	Hsu et al.	(a Hybrid approach) Combination of backpropagation network with exponential desirability functions	Manufacturing processes with multiple responses
2004	Hsu	(a Hybrid approach) Combination of backpropagation network with exponential desirability functions and Tabu search	Optical coupler manufacturing process
2005	Bisht et al.	Multi-layer feedforward network	Turning operation
2005	Sukthomya & Tannock	(A Hybrid approach) Artificial neural network combined with Taguchi approach	Super-plastic forming process
2005	Liao	(A Hybrid approach) Combination of backpropagation networks with data envelopment analysis	Manufacturing processes with multiple responses
2005	Kurtaran, Ozcelik & Erzurumlu	(A hybrid approach) Integrated FEM, Design of experiment, ANN and GA	Injection molding
2006	Cus, Zuperl & Milfelner	Multi-layer feedforward network with backpropagation	Milling process
2006	Hsieh	Two stage feedforward network	Lead frame manufacturing

2006	Cus& Zuperl	Feedforward network & Radial basis network	Turning operation
2006	Antony et al.	(A hybrid approach) Neuro-fuzzy model based on S/N ratios	electronic assembly operation with three responses
2006	Huang & Tang	(a Hybrid approach) Decision making system combining ANN, GA and Taguchi method	Melt spinning
2006	Yang, Lin & Chen	(a Hybrid approach) Artificial neural networks combined with Genetic algorithms	Forming process
2006	Sen & Shan	(a Hybrid approach) Backpropagation network combined with GA and desirability functions	Drilling process
2006	Chang	(a Hybrid approach) Artificial neural networks combined with the structure of the optimization problem	Manufacturing processes with multiple responses
2006	Karkoub	Random neural network	Tube Hydroforming
2007	Changyu, Lixia & Qian	(A Hybrid approach) Combination of artificial neural networks with genetic algorithms	Injection molding
2009	Yong et al.	(a Hybrid approach) Artificial neural networks combined with Genetic algorithms	Forming process with two constraints
2010	Belhadj et al.	Multi-layer feedforward network with backpropagation	Tube hydroforming process

CHAPTER FIVE
OPTIMIZATION OF FORMING PARAMETERS FOR TUBE HYDROFORMING PROCESS (THP) USING ARTIFICIAL NEURAL NETWORKS

5.1 Introduction

Hydroforming basically is a technique that uses a fluid either to form or aid in forming a part from ductile metal. The most common type of hydroforming used, tube hydroforming, changes the cross-sectional shape of a tube from the normal round to other shapes that change along the part's length.

Tube hydroforming offers several advantages as compared to conventional manufacturing via stamping and welding. These advantages include:

(a) Part consolidation (stamped and resistance welded two or more pieces of a box section can be manufactured in one operation from a hollow component),

(b) Weight reduction through more efficient section design and tailoring of the wall thickness,

(c) Improved structural strength and stiffness,

(d) Lower tooling cost due to fewer parts,

(e) Fewer secondary operations (no welding of sections required and holes may be punched during hydroforming),

(f) Reduced dimensional variations, and

(g) Reduced scrap.

In recent years, because of these enormous advantages over conventional processes, tube hydroforming has become a popular method in making tubular parts of different configurations used in automotive industry (Aue-U-Lan, Vree, Brekelmans, Geers, Sillekens & Werkhoven, 2005).

A classical example of tube hydroforming process is shown in Figure 5.1. The process is controlled by two types of loads, internal pressure (P) and axial force (F). After hydroforming process the initial length of tube (L_i) turns to L_f.

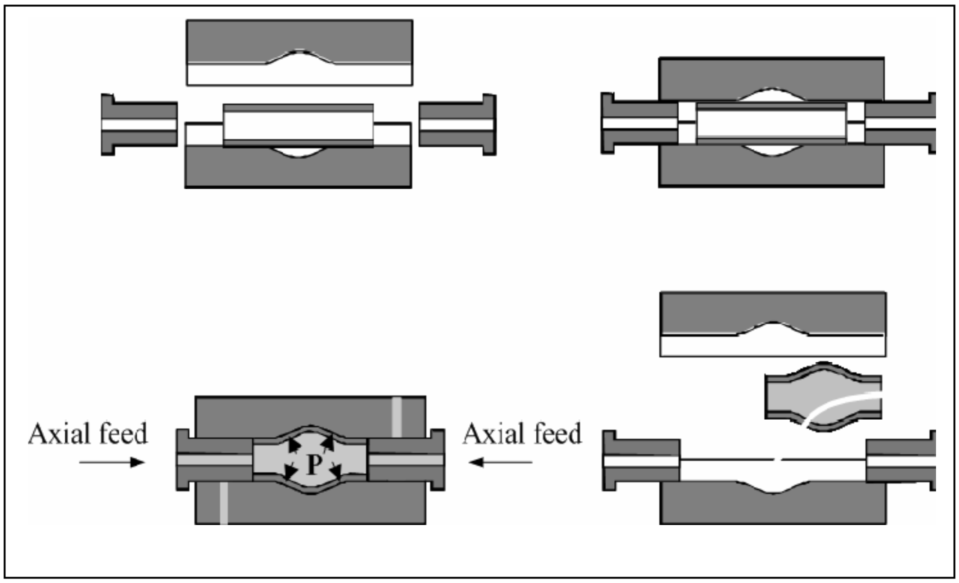


Figure 5.1 Tube hydroforming process (from Shengdun, 2007)

The aim of the process is to obtain high forming performance without any failure happening such as bursting, buckling and wrinkling. For example, bursting takes place when pressure is applied too rapidly without enough material feeding, while too much feeding of material tends to cause buckling. Eventually, these failures cause fracture and there is no clearly preferred approach to predict fracture in THP. Then, the quality of the hydroformed material can be measured by some characteristics such as thinning ratio and bulge ratio as commonly used in the literature. The performance of the process can be maximized by minimizing thinning ratio and maximizing bulge ratio simultaneously. These characteristics are highly dependent on by a number of forming parameters such as geometry dimensions, mechanical properties of the material and process parameters (Yong, Chan, Chunguang & Pei, 2009). Therefore, it is important to set up the forming parameters in such a way that the quality variations of the products and process are minimized without any failure happening.

The experimental optimization of any THP is often a very costly and time consuming task due to many kinds of non-linear events involved. In the relevant literature, optimization of forming parameters has often been considered as a multi response optimization problem, since there exist more than one responses should be optimized simultaneously. Various optimization approaches are employed for multi-response optimization of THP. Li, Nye & Metzger (2006) studied the effects of forming parameters on THP by using Taguchi and finite element method (FEM) in order to determine the optimal combination of forming parameters for the process. Yang, Zhang & Li (2006) and Fann & Hsiao (2003) used a gradient-based optimization method with FEM to check the hydroformed tube quality about its thickness uniformity and to minimize the thickness variation in tube hydroforming. Aue-U-Lan, Ngaile & Altan (2004) evaluated different optimization approaches and conducted FE simulations and experiments of a closed-die tube hydroforming. Imaninejad, Subhash & Loukus (2005) employed finite element simulation and response surface method to determine the loading paths. They employed the optimization software LS-OPT to optimize the internal pressure and axial feed in which minimum thickness variation was chosen as a design objective and maintaining the effective stress below the ultimate stress. Manabe & Amino (2002) investigated the parameters influencing THP by means of FEM simulations and experimental work. Koç & Altan (2002) investigated the effects of the geometry parameters in THP by a series of 2D FEM simulations. Abedrabbo, Worswick, Mayer & Riemsdijk (2008) proposed an optimization method linked with the finite element analysis which employed forming limit diagrams as a failure prediction tool. They employed the optimization software HEEDS which uses genetic algorithm search methods. Johnson, Nguyen, Davies, Grant & Khaleel (2004) proposed a numerical control based on incremental method for process parameters in order to obtain stable deformation and maximum formability during tube hydroforming. Jannson, Nilsson & Simonsson (2007) presented suggestions on how to perform parameter optimization in tube hydroforming process, and proposed an adaptive optimization method based on response surface methodology.

Instead of using mathematical models for optimization, some authors have proposed fuzzy logical approaches. Aydemir, Vree, Brekelmans, Geers, Sillekens & Werkhoven (2005) proposed an adaptive simulation approach to obtain an effective process control for tube hydroforming. They focused on optimization of process parameters to prevent wrinkles and bursting. The parameters are adjusted during simulation via a fuzzy knowledge based controller. Manabe, Suetake, Koyama & Yang (2006) proposed a database assisted fuzzy process control algorithm to determine optimal process parameters. For the virtual control system, an explicit dynamic finite element code was used in simulation.

New optimization techniques to reduce time consumptions and to determine the optimal forming parameters have been also used. Among these techniques artificial intelligence are highly demanded to model and optimize the THP with the purpose of manufacturing high quality parts. Artificial neural networks and genetic algorithm are two of the most promising artificial intelligence techniques for optimization and both of these two techniques are considered to be appropriate in the optimization of manufacturing processes. In recent years, ANN has become a very powerful and practical method to model very complex non-linear systems. It is used as a prediction model rather than an optimization tool for tube hydroforming problem. Hyun & Cho (2004) predicted the forming pressure for THP using ANN approach. Karkoub (2006) developed a model to predict the amount of deformation caused by hydroforming using random neural networks. Belhadj , Abbassi, Mistou & Zghal (2010) proposed an ANN approach to predict the thickness in the tube T-shape finished parts and to optimize the final part geometry.

The GA is a global optimization algorithm and the objective function does not need to be differentiable. This allows the algorithm to be used in solving difficult problems, such as multi-model, discontinuous or noisy systems. Mahanty, Agrawal, Shrin & Chakravarty (2007) presented two approaches by combining two-dimensional irregular-shaped polygonal elements with a real-encoded genetic algorithm and a hybrid algorithm using a real-coded genetic algorithm with a local optimization algorithm. Zafar (2002) used GA in combination with Finite Element packages to optimize the internal pressure and feed rate. Yong et al. (2009) proposed a hybrid method consisting of ANN, FEM and GA for optimizing the loading paths of THP. Shengdun, Yong, Zhiyuan & Chengwei (2007) applied genetic algorithm in combination with FE codes to search the optimal process parameters.

In this chapter, the objective is to apply ANNs to the optimization of forming parameters for THP and to compare its performance with GA. Since the relationship between process responses and process parameters is unknown, we used metamodeling approach to build the fitness function of proposed GA. Response surface analysis and artificial neural networks are used for metamodeling and their performances are compared. Although a comprehensive review for a better understanding of the ways of optimizing forming parameters of tube hydroforming process has been made, there appears to be no earlier study on the topic using ANNs as an optimization tool. This observation has been the motivation for the present work on parameter optimization problem of a THP.

A general outline of this chapter is as follows. A brief explanation of the process and optimization problem is given in subsection 5.2. Following the proposed method presented in subsection 5.3, a case study is performed in 5.4 Conclusions are pointed out in subsection 5.6.

5.2 Problem Statement

The purpose of the parameter selection problem in tube hydroforming is to find optimal parameter design in order to obtain high forming performance and to analyze the effects of parameters on the performance of the tube hydroforming process.

The principal factors, which influence the part quality, are the loading path of internal pressure and feeding during expansion and the size of start tube. A suitable combination of these factors is important to avoid part failure. Common failure modes that limit the tube hydroforming process are wrinkling, buckling and bursting as shown in Figure 5.2. If the axial force is very high while the internal pressure is too low, buckling and wrinkling may occur. If the axial force is too low and the internal pressure is very high, the tube may burst. Thus, successful tube hydroforming without instabilities highly depends on the combination of the internal pressure and axial feeding at the tube ends.

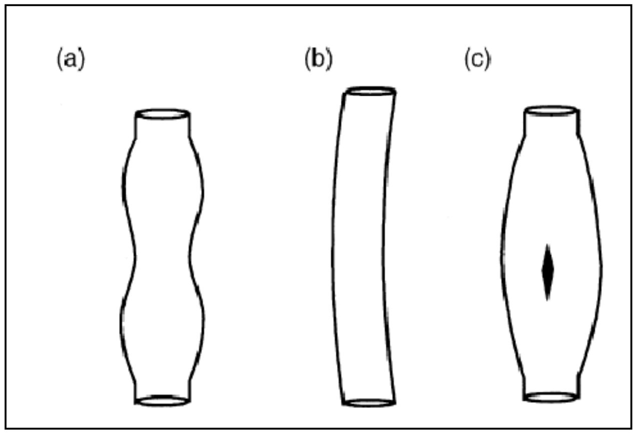


Figure 5.2 Failure modes a) Wrinkling; b) Buckling; c) Bursting (Koç and Altan, 2002)

5.2.1 Objectives

The tube hydroforming process possesses multiple responses and performance characteristics. However, the main objective of the process is to obtain high performance without any failure happening such as bursting, buckling and wrinkling. Among the three main failure modes involved in THP, bursting failure, as a consequence of necking, is irrevocable while other failure modes like buckling and wrinkling are recoverable. Eventually, these failures cause fracture and there is no clearly preferred approach to predict fracture in tube hydroforming process. Therefore, thinning ratio and bulge ratio are commonly used in the literature as a measure of forming quality. The objectives of the described process are: maximizing bulge ratio and minimizing thinning ratio (Li, Nye and Metzger, 2006).

$$Bulge\ ratio = \frac{r_1}{r_0} \tag{5.1}$$

where r_1 and r_0 represent maximum radius of the hydroformed tube and original radius of the tube before hydroforming, respectively.

$$Thinning\ ratio = \frac{t_0 - t_1}{t_0} \tag{5.2}$$

where t_0 and t_1 denote original thickness of the tube before hydroforming and critical thickness of the hydroformed tube, respectively.

5.2.2 Limitations

The conventional way of designing a THP starts with the definition of the basic parameters such as the die and tube geometries and the selection of the material. Then, a loading path is estimated to test the feasibility of the planned forming process.

The parameters influencing part quality in tube hydroforming process can be grouped into three categories: geometry parameters, material parameters and process parameters. There are several factors limiting the forming parameters. These factors originate usually from material properties and process specifications.

Table 5.1 illustrates the significant parameters of the process and each parameter has three levels. Thus, these parameters are limited with the bottom and top permissible limits.

The problem of optimizing forming parameters in THP requires developing a model capable of accurately describing the input-output behavior and capturing the range of these input-output parameters. Therefore, two artificial intelligence techniques are employed to optimize THP: ANN and GA.

5.3 Solution Methodology

In this section, we describe how artificial intelligence techniques can be used to solve the considered problem. Two artificial intelligence techniques are employed to optimize THP: ANN and GA. Firstly, a two-stage ANN is proposed to optimize the THP described in the previous section. Then, we employed a GA approach to optimize the same process. Finally, the comparison of these two artificial intelligence techniques is given by an illustrative example.

Table 5.1 Process parameters of tube hydroforming

Material Parameters	Density	ρ
	Young's modulus	E
	Hardening coefficient	K
	Hardening exponent	n
	Poisson's ratio	ν
	Yield strength	σ_y
	Ultimate tensile strength	σ_u
Geometry Parameters	Length of tube	L_0
	Outer radius of tube	r_0
	Thickness of tube	t_0
	Die entry radius	r_e
	Bulge width	W
Process Parameters	Internal pressure	P_f
	Nominal stress ratio	M
	Friction coefficient	μ

5.3.1 Design of the Proposed ANN

Proposed ANN approach involves two stages both employing a back propagation NN for parameter searching and response estimating. The reverse direction concept of Hsieh & Tong (2001) is explored to hold communication between these networks. The two–stage approach is displayed in Figure 5.2. As can be seen, the first back propagation network determines the optimal forming parameter combination. Hence, the first network can be interpreted as a procedure of parameter searching. Then, to obtain the estimates of the process responses, the second network is constructed by assigning the parameter combination and responses. Inputting the ideal parameter combination, the estimated results of responses can be obtained. Therefore, the second network can be interpreted as a procedure of response estimating.

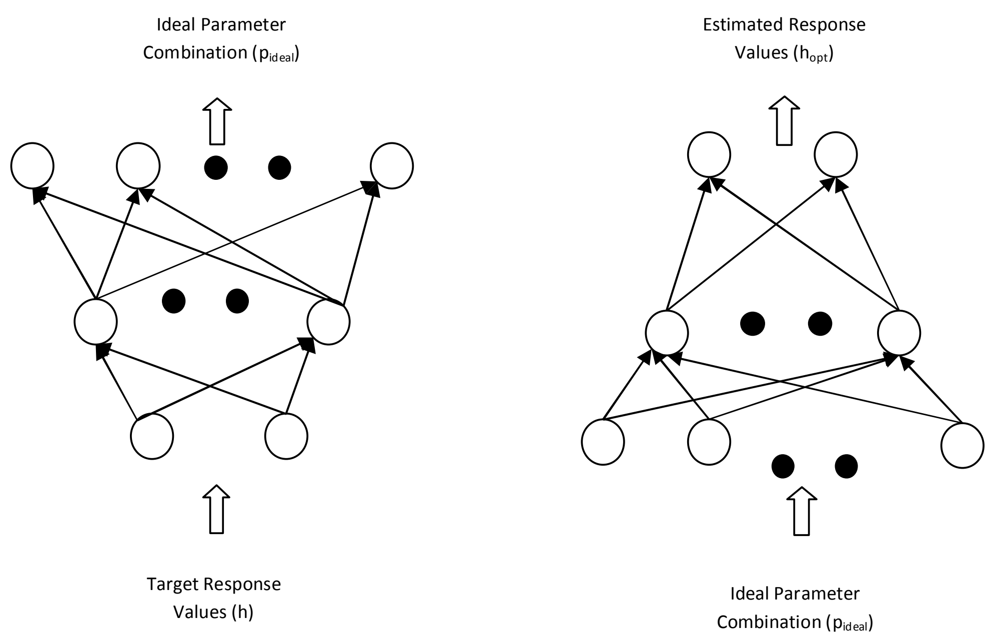


Figure 5.2 Topology of the proposed neural network approach

In the design of a neural network following steps are considered:

1. Data Acquisition
2. Determine Network architecture
3. Training and testing
4. Simulation

5.3.1.1 Data Acquisition

The following input data must be entered into the model:

- Forming parameters: Forming parameters may be fed into the model in two ways. The original values of the parameters or, if available, the level settings of the forming parameters may be used.

- Process responses: Since the objective of the process is getting a deformed tube with higher bulged height and less wall thickness reduction, the response of the process, y, may be expressed in different ways:

$$y = thinning\ ratio - bu\lg e\ ratio \tag{5.3}$$

$$y = \frac{thinning\ ratio}{bu\lg e\ ratio} \tag{5.4}$$

$$y = thinning\ ratio + \frac{1}{bu\lg e\ ratio} \tag{5.5}$$

For all the expressions given above, minimum process response may be obtained by minimizing thinning ratio and maximizing the bulge ratio. While thinning ratio takes values between 0 and 1, bulge ratio can take values in any range, commonly greater than 1. Since it is necessary to input target value of process response to the first network and the objective is to minimize the process response, the target of process response can be set as zero.

The data set used to train the proposed networks has to include forming parameter values and corresponding process response values. To improve generalization, the available data is divided into three subsets. The first subset is employed to train the network, that is, for computing the gradient and determining the optimum network weights and biases. The second subset is used for validation. The error on the validation set is monitored during the training process. When the validation error increases for a specified number of iterations, the training is stopped. The weights and biases corresponding to minimum validation and training errors are considered as the optimum values of weights and biases. The last subset is dedicated for testing, that is, for obtaining the overall accuracy of the network and for comparing the performance of different network structures. The percentage of cross validation data and testing data is set as 5% of the whole data, respectively.

5.3.1.2 Proposed Network Architecture

As mentioned previously, since the backpropagation network has the ability to map the complex relationship between input data and corresponding outputs, we adopted a feedforward BPN for optimization of tube hydroforming process.

Since the first network was built for parameter searching, process responses are treated as inputs of this network. Thus, the number of neurons in the input layer is determined by the number of process responses of the problem considered. If the values of these responses are not in the same scale, all data must be normalized. The outputs of this network are the forming parameters, thus the output layer has one neuron for each forming parameter under study. The second network is used for response estimating and the outputs of the first network are used as the inputs of this network. Then, this network has one neuron for each parameter under study in the input layer and the number of output neurons is determined by the number of process responses of the problem considered. A network with one hidden layer is selected as the starting network structure. In order to find the number of neurons in the hidden layer trial-and-error approach is used.

5.3.1.3 Network Training

In this phase, the network is trained with the training data set described above and the performance checked with the test set. First, network parameters affecting the training need to be determined. Parameters of the proposed networks to be determined are:

- *The number of hidden layers and number of neurons in the hidden layer(s):* As mentioned before, one hidden layer is selected as the starting network structure. In order to find the number of neurons in the hidden layer trial-and-error approach is used. Based on this approach, a random, small number of neurons are used in the hidden layer and if the error of the trained network does not meet the desired tolerance, the number of neurons in the hidden layer is increased and training cycle and performance evaluation is repeated. This procedure is continued until the trained network performs satisfactorily.

- *Transfer functions of neurons:* The transfer functions can be any differentiable transfer function such as tan-sigmoid transfer function (tansig), log-sigmoid transfer function (logsig) and linear transfer function (purelin). Since the outputs of both two networks to be created ranges between [-∞,+∞], "purelin" transfer function is used for output layer. However, there is not any restriction for the output of hidden layers. Therefore, both "tansig" and "logsig" functions are used for hidden layers for different trials.

- Learning algorithm, learning rate and step size: The most common algorithms used for backpropagation training are "Batch Gradient Descent with Momentum and Variable Learning Rate algorithms. These algorithms are very sensitive to the proper setting of step size and learning rate and the initial values of these parameters are set as the default values of the software used: 0.7 and 0.1, respectively. If the error of the trained network does not satisfy the given tolerance, different values are used to decrease the network error.

- *Stopping criteria:* The termination criteria to finish training may be one of the following:
    - As the network begin to overfit the data, the error on validation set also begins to rise. If the validation error is increases for a specified number of iterations, the training terminates.
    - If the performance function of the network training drops to some threshold defined, the training terminates. For the proposed networks, default performance function, mean square error, is used.
    - When maximum number of epochs is reached, the training terminates.

Once the parameters are set, the proposed networks are trained by the selected learning rate algorithm for the pre-determined number of epochs until the given termination criteria is met. Since there is no established method to determine the optimal parameters of network architecture, we performed trial-and-error experiments with different network parameters to select the optimal network configuration.

Selection of optimal network configuration takes place on the basis of trial-and-error because there are no general rules for the selection process. After the two networks in the proposed approach are trained, network performance indicators are calculated. Mean square error (MSE) and correlation coefficient (r) are used to evaluate the network configurations. These configurations differ from each other in terms of network parameters defined above. The network configuration with minimum MSE and appropriate R value is selected as the best network architecture.

5.3.1.4 Simulation

In this step, the selected network configuration is simulated. The target value of the process response is inputted to the first network to obtain the optimal parameter design. The optimal values of forming parameters obtained by the first network are now inputted to the second network and optimal response of THP is found.

5.3.2 Design of the Proposed GA

Since the relationship between process responses, thinning ratio and bulge ratio, and forming parameters cannot be represented analytically, the proposed GA approach consists of two stages. In the first stage, a metamodel is built to map the relationship between forming parameters and process responses. In the second stage, GA is employed to obtain the optimal forming parameters of THP.

Briefly, the neural network maps the input-output observed data and defines the fitness function of the parameter selection by function approximation. Consequently, the genetic algorithm utilizes the fitness function to identify the optimal solution of the optimization problem of tube hydroforming process.

The proposed GA algorithm can be defined as follows:

Step 1. Set GA Parameters

There is no established method to find the optimal parameters of the GA. Then, we conducted trial-and-error experiments to determine the parameters of proposed GA approach. Parameters of the proposed GA to be determined are:

- Population size
- Crossover probability
- Mutation probability
- Generation number

Step 2. Representation of solution

Herein, the chromosome structure is used to represent the possible solution. The length of the chromosome is determined by the number of forming parameters under study. The genes in the chromosome can be binary or real integer number and can be represented either by the level of a forming parameter or by the real value of the parameter.

Step 3. Fitness Function

The fitted metamodels in input-output mapping are used as fitness functions of the proposed GA. To build the metamodel representing the relationship between forming parameters and process responses, we used two approaches: Response surface analysis and ANNs.

Since the objective of the process is getting a deformed tube with higher bulged height and less wall thickness reduction, the response of the process, y, may be expressed in three different ways as described in the previous section.

Step 4. Create New Population

To create the new population, genetic operators are employed.

Step 4.1. Selection: In the proposed algorithm, roulette wheel selection is used. This is a way of choosing members from the population of chromosomes in a way that is proportional to their fitness. It does not guarantee that the fittest member goes through to the next generation; however it has a very good chance of doing so. The algorithm of roulette-wheel selection is:

i) With respect to fitness functions, calculate the associated probability of selecting each chromosome.
ii) Compute cumulative distribution of the probabilities obtained in (i).
iii) Generate a random number from a uniform continuous distribution in [0 1]. Select the chromosome that the random number is less than associated cumulative probability.

Step 4.2. Crossover: The proposed genetic algorithm uses a simple crossover operator in which a random crossover point is determined and the second parts of the chromosomes are exchanged. The probability of selecting a chromosome for crossover is calculated by crossover probability and it is determined by trial and error in the range of 0.5 and 0.9. The algorithm of crossover operator is as follows:

i) Select two random chromosomes as described in Step 4.1.
ii) Generate a random number from uniform continuous distribution in [0 1]. If this random number is less than crossover probability, go to (iii). Otherwise, select two new chromosomes.

iii) Generate a random number from 1 to the length of the chromosome as the crossover point. Perform crossover.

iv) Do this until the number of created children is equal to initial population.

Step 4.3. Mutation: The probability of selecting a chromosome for mutation is calculated by mutation probability and it is determined by trial and error in the range of 0.001 and 0.05. The mutation operator is performed by the following algorithm:

i) Generate a random number from uniform continuous distribution in [0 1]. If this random number is less than mutation probability, go to (ii). Otherwise, select the next chromosome.

ii) Generate a random number from 1 to the length of the chromosome as the mutation point. Perform mutation and change the value of the gene within the bounds of the selected forming parameter.

Step 4. Reproduction

In order to select the chromosomes for the next generation, all the newly created chromosomes are to be evaluated. Selection of the chromosomes for the next generation is done by roulette-wheel selection as described above.

Now, the new population is obtained and the algorithm again starts with the "Selection" step. The algorithm is performed until the termination criteria; maximum number of generations is met.

5.4 Simulation Example

To illustrate the performance of proposed algorithm for tube hydroforming process, an illustrative example based on the experimental data from Li, Nye & Metzger (2006) in which Taguchi approach is used to find preferable forming parameters of THP. They applied Taguchi method to design an orthogonal array and the virtual experiments were analyzed by the use of FEM.

The process has eight forming parameters each with three levels as in Table 5.2 and two responses: bulge ratio and thinning ratio.

The Taguchi L-18 experimental design of these eight forming parameters and two responses is shown in Table 5.3. The aim of our study is to determine levels of these eight forming parameters in order to minimize thinning ratio while maximizing bulge ratio. Thus, the response of THP to be optimized can be expressed by (5.4)

Table 5.2 Level of forming parameters (from Li, Nye & Metzger, 2006)

Symbol	**Forming Parameter**	**Level 1**	**Level 2**	**Level 3**
A	Length of the tube	180	200	220
B	Thickness of the tube	1,35	1,5	1,65
C	Die entry radius	8	10	12
D	Bulge width	90	100	110
E	Hardening exponent	0,207	0,227	0,247
F	Internal pressure	36	40	44
G	Nominal stress ratio	0,2	0,4	0,6
H	Friction coefficient	0,02	0,06	0,1

5.4.1 ANN Optimization

In the following paragraph, we solve the problem considered applying the steps of the proposed ANN approach as described in Section 5.3.1.

Table 5.3 Experimental design (from Li, Nye & Metzger, 2006)

Run No	Forming Parameters								Response	
	A	B	C	D	E	F	G	H	Bulge Ratio	Thinning Ratio
1	1	1	1	1	1	1	1	1	1.448	0.284
2	1	1	2	2	2	2	2	2	1.982	0.497
3	1	1	3	3	3	3	3	3	1.923	0.477
4	1	2	1	1	2	2	3	3	1.596	0.407
5	1	2	2	2	3	3	1	1	2.029	0.559
6	1	2	3	3	1	1	2	2	1.449	0.304
7	1	3	1	2	1	3	2	3	1.691	0.483
8	1	3	2	3	2	1	3	1	1.439	0.315
9	1	3	3	1	3	2	1	2	1.59	0.429
10	2	1	1	3	3	2	2	1	1.678	0.386
11	2	1	2	1	1	3	3	2	1.719	0.423
12	2	1	3	2	2	1	1	3	1.64	0.385
13	2	2	1	2	3	1	3	2	1.498	0.345
14	2	2	2	3	1	2	1	3	1.639	0.417
15	2	2	3	1	2	3	1	3	1.853	0.53
16	2	3	1	3	2	3	2	1	1.744	0.493
17	2	3	2	1	3	1	2	3	1.492	0.384
18	2	3	3	2	1	2	3	1	1.597	0.416

5.4.1.1 Data Acquisition

The forming parameters given in Table 5.2 and experimental data given in Table5.3 are used as data set to train the network. The forming parameters are used in two different ways, with their original values and with their levels. This is considered as a parameter to be determined and by employing trial and error experiments, optimal representation of forming parameters is determined.

The percentage of cross validation data and testing data is set as 10% of the whole data, respectively. The rest, 80% of the data set is used for training. The target of the process response is set as zero and fed into the first network as the input.

5.4.1.2 Proposed Network Architecture

The first neural network is required as a procedure of parameter searching. For this network; the number of neurons in the input layer will be two, one for thinning ratio and one for bulge ratio. This network will have eight neurons in the output layer each for the forming parameters in Table 5.2. In order to find the number of hidden layers and the number of neurons in the hidden layer(s), trial-and-error experiments are performed.

The second neural network is used as a procedure of response estimating. Since the outputs of the first network are inputs of this network, this network will have eight neurons in the input layer. The outputs of this network are the process responses. Then the output layer has two neurons for process responses. In order to find the number of neurons in the hidden layer trial-and-error approach is used. Again, hidden layers are determined by performing trial and error experiments.

5.4.1.3 Network Training

A network with one hidden layer is selected as the starting network structure for both proposed networks. The initial values of ANN model parameters, learning rate and step size are set as the default values of the software used: 0.7 and 0.1, respectively. If the error of the trained network does not satisfy the given tolerance, values of these parameters are changed to decrease the network error.

4 networks are constructed with different parameters for the first network. The network configurations can be seen in Table 5.4.

Table 5.4 Neural network achitectures built for proposed model-1

Trial	Hidden Layers			Transfer Function of Output Layer	Training Function	Epoch	Learning Rate	Momentum
	# of Layers	# of Neurons	Transfer Function					
1	1	50	Sigmoid	Linear	Momentum	2000	Default	Default
2	2	50	Sigmoid	Linear	Momentum	2000	Default	Default
3	2	50	Sigmoid	Linear	Momentum	2000	Default	Default
4	1	50	Sigmoid	Linear	Variable learning rate	2000	Default	-

All networks are trained by the data set given in Table 5.3. The mean square error and correlation coefficient values for each network are as follows:

Table 5.5 Peformance comparison of neural network architectures-1

Trial	MSE_Testing	MSE_Validation	MSE_Training	r
1	0.6265	0.622	0.6223	0.3491
2	0.5672	0.5641	0.5623	0.4017
3	0.5659	0.5624	0.5613	0.4115
4*	0.5322	0.5311	0.5298	0.4551

Trial with an asterisk is the optimal network architecture since it has the minimum MSE and maximum r values than the others. According to the performance indicators, the optimal configuration of the first network is determined as “2-50-8” with one hidden layer employing logarithmic sigmoid transfer function. The best network performance is obtained when we applied the learning function, adaptive learning rate with the default value of learning rate as 0.7. Figure 5.3 displays the variation of the mean square error for the best configuration of the first network during training. Referring to this figure, it is observed that the trained network model is validated for its predictive capability based on its acceptable MSE.

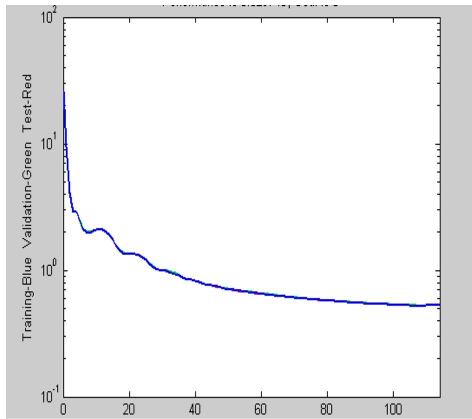


Figure 5.3 Variation of MSE for the first network configuration

Since the outputs of the first network will be the inputs of the second network, only one network is configured for the second network. The parameters of this network and network performance are given in Table 5.6 and Table 5.7, respectively. Since the MSE and r value of this network is acceptable for evaluation, this network is used to generate the outputs for the parameter values calculated by the first network. Then, the optimal network configuration is determined as "8-50-2" with one hidden layer employing logarithmic sigmoid transfer function. As in the first network, this network also applies adaptive learning rate with the default value of learning rate as 0.7.

Table 5.6 Neural network architectures built for proposed model-II

Network	**Hidden Layers**			**Transfer Function of Output Layer**	**Training Function**	**Epoch**	**Learning Rate**
	# of Layers	**# of Processing Elements**	**Transfer Function**				
1	1	50	Sigmoid	Linear	Variable learning rate	2000	default

Table 5.7 Performance measures of neural network architectures:

Trial	**MSE_Testing**	**MSE_Validation**	**MSE_Training**	**r**
1*	0.018	0.017	0.017	0.913

5.4.1.4 *Simulation*

In this step, the selected network configurations are simulated. The target value of the process response is inputted to the first network as zero to obtain the optimal parameter design. The optimal values of forming parameters obtained by the first network are given in columns 1-8 of Table 5.8 and these parameter values are now inputted to the second network to find the optimal response of tube hydroforming process considered.

Table 5.8 Results obtained by the proposed ANN model:

Forming Parameters								**Responses**	
A (1)	**B (2)**	**C (3)**	**D (4)**	**E (5)**	**F (6)**	**G (7)**	**H (8)**	**Bulge Ratio**	**Thinning Ratio**
0.4399	6.1468	3.7379	0.2312	0.1433	1.2340	1.2087	3.2309	0.7908	0.3783

5.4.2 Optimization Using the Proposed GA

As mentioned in the previous section, the proposed GA approach consists of two stages. In the first stage, a metamodel is built to map the relationship between forming parameters and process responses.

5.4.2.1 *Input-Output Mapping*

We used response surface analysis and artificial neural network methods for input-output mapping.

5.4.2.1.1. Response Surface Analysis: Using the 18 responses from Table 3, the adequate fit as the metamodel is obtained by using Minitab 15 Statistical Software®. The statistical analysis of model fit is given in Table 5.9. It can be concluded that the fitted model can explain approximately 100% of the variation in the response by the variable.

Table 5.9 Statistical Analysis of Model Fit

Response	R-square	R-square (adj.)
Bulge Ratio	99.90%	98.60%
Thinning Ratio	100.00%	99.90%

Then, the fitted regression model takes the form of:

Bulge Ratio= $1.8852-0.0171A-0.0771B+0.0461C+0.0015D+0.0556E+0.1644F-0.0172G-0.0123H-0.0341B^2-0.0761C^2-0.1104D^2-0.0595E^2-0.0216F^2-0.0602G^2+0.033H^2$

Thinning Ratio= $0.505098+0.000498789A+0.00424522B+0.0135077C-0.00700773D+0.0210833E+0.0777482F-0.0128971G+0.00401957H-0.0144274B^2-0.0221685C^2-0.0446685D^2-0.0264130E^2-0.0114185F^2-0.0229718G^2+0.0110022H^2$

5.4.2.1.2. Artificial Neural Networks: A BPN is trained to derive the relationship between input parameters and output responses. A three-layer backpropagation network is proposed for mapping the relationship between parameters and responses as in Figure 5.4. There are eight neurons in the input layer each corresponding to a forming parameter. There are two neurons in the output layer each corresponding to a response. The number of hidden layers can be determined by trial-and-error. The trained network is evaluated by mean square error and the network configuration with minimum network error is selected as the best.

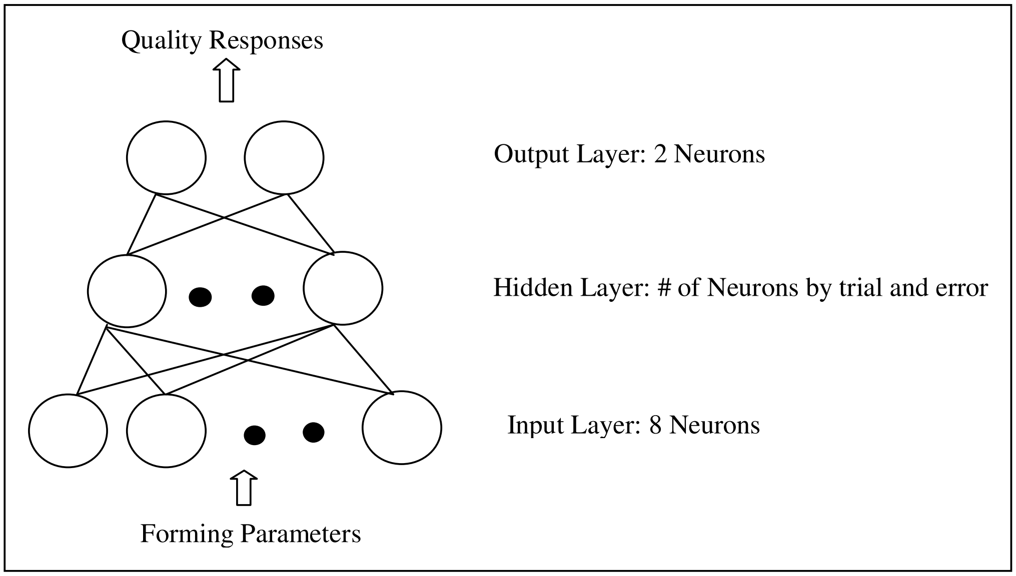


Figure 5.4 Architecture of proposed BPN

The parameters and performance of the selected network topology is as follows:

Table 5.10 Parameters of proposed BPN

Parameters	
# Hidden neurons	10
Transfer func of hidden neurons	Logarithmic Sigmoid
Transfer func of output neurons	Linear
Training function	Levenberg-Marquardt backpropagation
# Epochs	1.000
Performance	
MSE	0
Correlation Coefficient	0.97

The optimal neural network configuration has 10 hidden neurons which have logarithmic sigmoid transfer function. The output layer of the network has linear transfer function. The change of mean square error is shown in Figure 5.5.

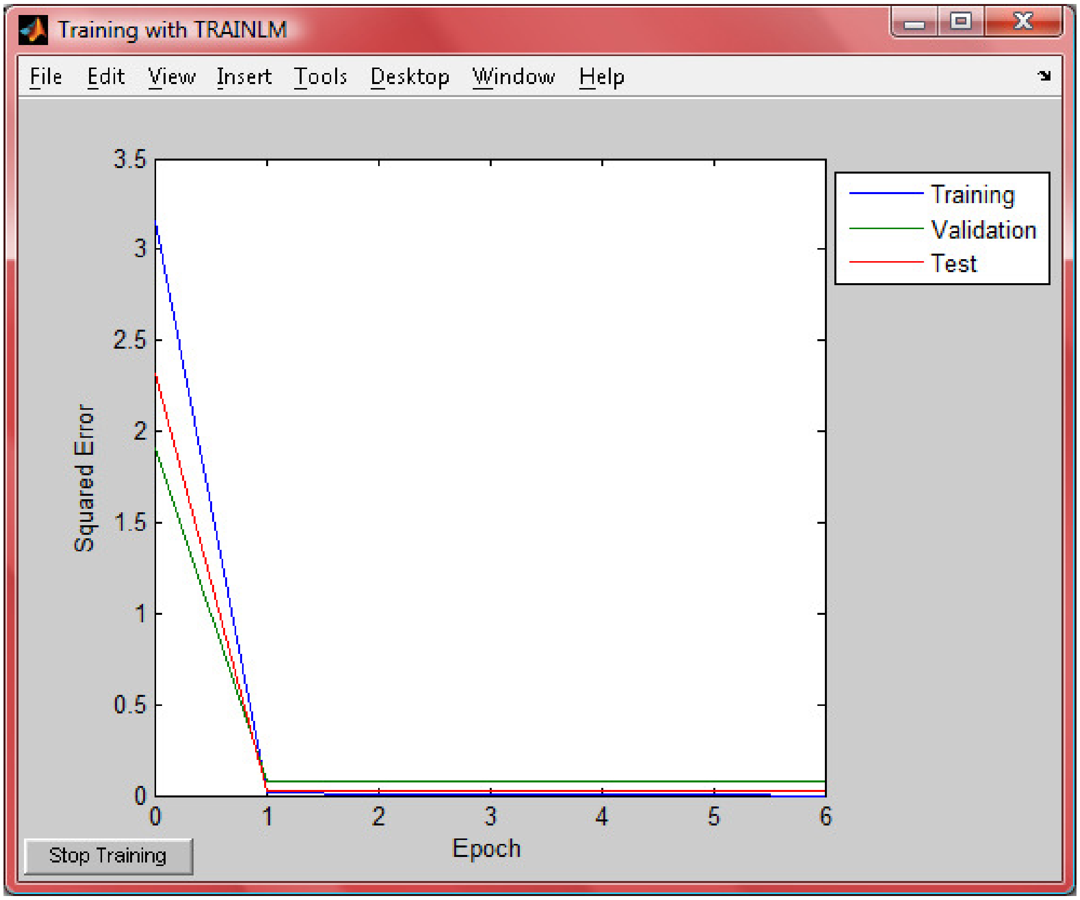


Figure 5.5 MSE of trained neural network

5.4.2.2 Optimization by Proposed GA approach

The application of the proposed GA optimization algorithm is described in the following paragraph:

Step 1. Set GA Parameters

The initial values of GA parameters are set as 20, 0.7, 0.05 and 100 for population size, crossover probability, mutation probability and generation number, respectively. During trial and error experiments different values for these parameters are examined.

Step 2. Representation of solution

The chromosome structure is used to represent the possible solution. The length of the chromosome is 8 in which each gene in the chromosome represents the level of a forming parameter in Table 5.2. The genes in the chromosome are represented by real-coded integer numbers by using the levels of the parameters that ranges between 1 and 3.

Step 3. Fitness Function

The fitted models by input-output mapping in the previous section are used as fitness functions separately. The process responses are calculated by either using the metamodel obtained by response surface analysis or inputting the parameters to the trained ANN.

Step 4. Create New Population

To create the new population, genetic operators are employed. In the proposed algorithm, roulette wheel selection is used. A simple crossover operator which means that one splicing point is selected to create new individuals is employed by an initial crossover probability of 0.7. Finally, by applying the mutation operator and determining the chromosomes surviving for the new population, a generation is completed. The algorithm is performed until the termination criteria; maximum number of generations is met.

Table 5.11 Parameters of proposed GA

Parameters of GA		
Parameter	**RSM**	**NN**
Population size	35	35
Crossover rate	0.7	0.6
Mutation rate	0.05	0.05
Iteration number	1.000	1000

The parameters of the proposed GA-RSM and GA-NN integrations are shown in Table 5.11. These parameter values are based on our computational experiences and it is to be noted that the same parameter values are tried to be found for both metamodels during simulations. The evolutions of the fitness function with these parameters are shown in Figure 5.6 and Figure 5.7.

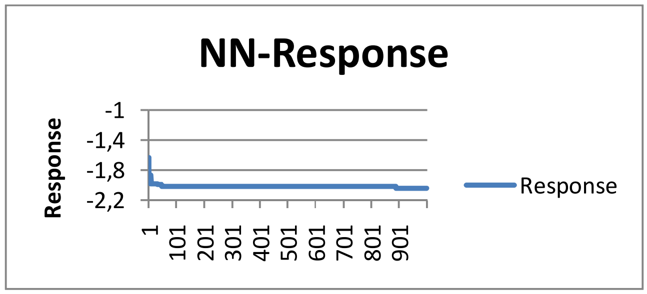


Figure 5.6 History of proposed GA approach using ANN-Metamodel

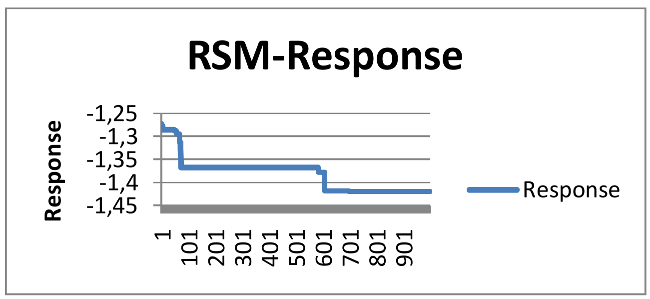


Figure 5.7 History of proposed GA approach using RSM-Metamodel

The optimal forming parameters and process response values of genetic algorithms for both fitness functions are as in Table 5.12.

5.4.3 Simulation Results

The forming parameters in THP were optimized using our proposed ANN and GA approaches. The proposed approaches are implemented by using MATLAB 5.0 computing environment. Table 5.13 shows the final process responses obtained by Li Li, Nye & Metzger (2006) using the Taguchi's method, and by our proposed two-stage ANN and GA approaches, respectively.

Table 5.12 Results of the proposed GA

Parameter	**GA-RSM**	**GA-NN**
A	1.4055	1.61
B	1.0230	2.586
C	1.0409	2.013
D	1.0178	2.7806
E	1.1152	2.9766
F	2.9697	2.5528
G	2.9602	1.4536
H	1.0354	1.2852
Thinning Ratio	**0.4095**	**0.0013**
Bulge Ratio	**1.8301**	**2.0479**

Table 5.13 Comparison of results

Approach	**Thinning Ratio (1)**	**Bulge Ratio (2)**	**Process Response (3)**	**Improvement of Proposed Approaches (4)**
GA-RSM	0.4095	1.8301	-1.421	18%
GA-NN	0.0013	2.0479	-2.047	70%
Two stage NN	0.3783	1.7908	-1.413	17%
Taguchi Approach	0.3910	1.5960	-1.205	-

In this table, the first and second columns show the thinning ratio and bulge ratio values, respectively, obtained by the solution approaches. Since the process response is expressed as in 5.4, the third column displays the process response that is optimized through our study. The fourth column indicates the % improvement and is calculated by:

$$\%\ Improvement\ of\ proposed\ integrated\ GA - RSM\ approach = \frac{Process\ Response(GA - RSM) - Process\ Response(Taguchi)}{Process\ Response\ (Taguchi)}$$

$$\%\,Improvement\ of\ proposed\ integrated\ GA-NN\ approach = \frac{Process\ Response(GA-NN) - Process\ Response(Taguchi)}{Process\ Response\ (Taguchi)}$$

$$\%\,Improvement\ of\ proposed\ two-stage\ ANN\ approach = \frac{Process\ Response(Two-stage\ ANN) - Process\ Response(Taguchi)}{Process\ Response\ (Taguchi)}$$

In terms of solution quality, both of the proposed approaches yielded better results compared to Taguchi method. The proposed two-stage ANN approach provided 17%, the proposed integrated GA-RSM approach provided 18% and the proposed integrated GA-NN approach provided 70% improvement in the THP under consideration. Consequently, both the bulge ratio and thinning ratio were improved by using the proposed approaches. Thus, it can be concluded that the hybridization of ANNs with GA are very promising in optimization of THP.

The main advantages of the proposed approaches are as follows:

- They can optimize the multiple process responses for a given THP simultaneously.

- They are simple to complement and adopt without modifying the existing model structure even if new forming parameters or new process performances are considered.

- The proposed approaches can be integrated with an intelligent manufacturing system for automated process planning. This will lead to reduction in production cost and production time, flexibility in selection of forming parameters and improvement of quality of the hydroformed part.

- They provide optimal results within a reasonable time span. By using the proposed approaches, forming parameters can be determined before starting the tube hydroforming processing.

- Failure happening can be avoided during hydroforming process by selecting the optimum parameters before starting hydroforming.

- The forming limit of the hydroformed metal increases as thinning ratio decreases. Thus, the reliability of THP increases and it results in higher forming quality.

- The higher bulge ratio indicates that the part is hydroformed with a maximum of material in the die cavity and necking is delayed as much as possible.

On the other hand the main strength of the proposed ANN model we observed during the current study is that the developed NN model provided strong mapping ability and high precision for THP and it has the result that was intended in replacing the time-consuming FEM simulations for mapping forming parameters-process responses (thinning ratio and bulge ratio) relationship in THP. The advantages of the proposed GA over the ANN model can be listed as below.

- It was easier to keep the optimum point within the experimental region defined by the search ranges of forming parameters as given in Table 5.2.

- It provided faster optimization of forming parameters that is a key problem in THP.

5.5 Conclusion

In this chapter the use of artificial neural networks for solving the process optimization problem for tube hydroforming processes with two process responses, thinning ratio and bulge ratio is studied. Focus of this study has been on demonstrating the optimization capabilities of the proposed ANN approaches by solving an example problem available in the literature, considered by Li, Nye & Metzger (2006). To analyze the performance of the proposed approaches, results are compared with the Taguchi solution method commonly used to solve the problem under study and also used by Li, Nye & Metzger (2006). The results demonstrated that the proposed approaches improved thinning ratio and bulge ratio of the THP under consideration. However, hybridization of ANN with GA provided better results. Thus, it can be concluded that the artificial neural networks are effective alternatives to Taguchi and other commonly used solution methods for THP such as FEM simulations. But they may need to be integrated by other solution methods to achieve more optimal results.

From the improvement of proposed approaches to tube hydroforming process, it can be concluded that besides the convergence to feasible and valid solutions, convergence of the proposed approaches to good quality solutions indicates their general applicability in also other metal forming optimization problems.

Future research should consider selecting the parameters of both ANN and GA automatically rather than choosing by trial and error, which may be a time-consuming task. Second, extension of the results to large size problems with different forming parameters and performance responses will be worthwhile.

CHAPTER SIX

OPTIMIZATION OF MACHINING PARAMETERS FOR METAL CUTTING PROCESS USING ARTIFICIAL NEURAL NETWORKS

6.1 Introduction

Cutting is a collection of material-working processes wherein material is brought to a specified desired geometry by removing excess material using various kinds of power-driven machine tools such as saws, lathes, milling machines and drill presses. The most common type of cutting used, machining, is a part of the manufacture of almost all metal products, and it is common for other materials, such as wood and plastic, to be machined. A classical example of machining process is shown in Figure 6.1.

The three principal machining processes are classified as turning, drilling and milling. Other operations falling into miscellaneous categories include shaping, planing, boring, broaching and sawing.

- Turning operations are operations that rotate the workpiece as the primary method of moving metal against the cutting tool. Lathes are the principal machine tool used in turning.

- Milling operations are operations in which the cutting tool rotates to bring cutting edges to bear against the workpiece. Milling machines are the principal machine tool used in milling.

- Drilling operations are operations in which round holes are created or refined by bringing a rotating cutter with cutting edges at the lower extremity into contact with the workpiece. Drilling operations are done primarily in drill presses but sometimes on lathes or mills.

Machining operations usually divide into two categories, distinguished by purpose and cutting conditions:

- Roughing cuts, and
- Finishing cuts

Roughing cuts are used to remove large amount of material from the starting work part as rapidly as possible, in order to produce a shape close to the desired form, but leaving some material on the piece for a subsequent finishing operation. Finishing cuts are used to complete the part and achieve the final dimension, tolerances, and surface finish. In production machining jobs, one or more roughing cuts are usually performed on the work, followed by one or two finishing cuts, referred as multi pass machining operations.

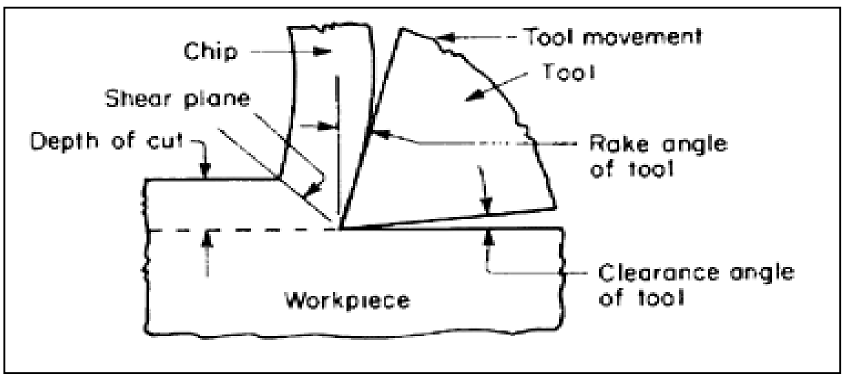


Figure 6.1 Metal cutting process (from Design for Manufacturability Handbook)

The machining task is basically a process plan that involves determination of appropriate machines, selection of tools and tool trajectory plan based on the required surface tolerance. Process planning is very important to ensure the quality of machining products, reduce the machining costs, and increase the machining effectiveness.

The study of metal cutting focuses, among others, on the features of tools, input work materials, and machine parameter settings influencing process efficiency and part quality characteristics (or responses). A significant improvement in process efficiency may be obtained by process parameter optimization that identifies and determines the regions of critical process control factors leading to desired outputs or responses with acceptable variations ensuring a lower cost of manufacturing (Sonmez, Baykasoglu, Dereli & Filiz, 1999):

- Due to the complex nature of the process, optimization of machining parameters is doubtlessly very difficult, since the following aspects are required;

- Knowledge of machining (i.e., turning or milling);

- Empirical equations relating the tool life, forces, power, surface finish etc., to develop realistic constraints;

- Specification of machine tool capabilities, (i.e., maximum power or maximum feed available from a machine tool);

- Development of an effective optimization criterion, (e.g., maximum production rate, minimum production cost, maximum profit or a combination of these);

- Knowledge of mathematical and numerical optimization techniques.

However, machining is a preferred manufacturing process to produce products with low cost and high quality. Hence machining economics is a very important consideration to achieve such an objective.

Economics of machining processes has been a trend of interest for many researchers. Earlier studies in the field of machining parameter optimization are limited to single-pass operations where total desired depth of cut is removed in just one pass. Ermer (1971) analyzed single and multi-pass machining operations by geometric programming. However one pass is rarely preferred in practice and multi-pass operations where the amount of stock to be removed during machining exceeds the maximum allowable depth of cut are used. The number of passes and subdivision of depth of cut are important parameters in multi-pass machining operations.

Various traditional optimization methods for determining and modeling of these parameters have been used in the literature. Iwata, Muratsu, Iwatsubo & Fujii (1972); Agapious (1992a,b & c); Shin & Joo (1992) used dynamic programming model to determine optimum number of passes and optimum cutting conditions. Sonmez, Baykasoglu, Dereli & Filiz (1999) used dynamic programming to obtain the optimum number of passes and geometric programming for obtaining the optimum cutting conditions. Gupta, Batra & Lal (1995) propose a methodology for selection of depth of cut for rough and finished passes in multi-pass turning operation to minimize total manufacturing cost by integer linear programming (ILP). Prasad, Rao & Rao (1997) combine LP and geometric programming to optimize the values of process parameters for a multipass turning operation. Al-Ahmari (2001) presented a non-linear mathematical model to solve the problem. Lee, Shin & Yang (1996) provide an interactive algorithm using both RSM and mathematical modelling to solve a parameter optimization problem in turning operation. Jeang, Li & Wang (2010) employed response surface methodology to predict cutting time and tool life.

However, the machining optimization problem becomes more complicated when a large number of constraints are included. The additional variables due to number of passes make the solution procedure more complicated (Wu, 2008). Considering the drawbacks of traditional optimization techniques, non-traditional optimization techniques have been commonly used to optimize the machining problem. Satishkumar, Asokan, & Kumanan (2006) investigated the use of different non-traditional optimization techniques like genetic algorithms, simulated annealing and ant colony optimization to solve the problem. Bhaskara, Shunmugam & Narendran (1998) and Shunmugan, Bhaskara & Narendran (2000) used genetic algorithm approach to optimize the subdivision of depth of cut and number of passes. Onwubolu & Kumalo (2001) propose a local search GA-based technique in multi-pass turning operation with mathematical formulation in line with work by Chen & Tsai (1996) with simulated annealing-based technique. Wang, Da, Balaji & Jawahir

(2002) apply GA-based technique for near-optimal cutting conditions for a two-and three-pass turning operation having multiple objectives. Cus & Balic (2003) use GA-based technique to determine the optimal cutting conditions in NC-lathe turning operation on steel blanks that minimize the unit production cost without violating any imposed cutting constraints. Savas & Ozay (2008) determined the minimum surface roughness at the process of tangential turn-milling by using genetic algorithms. Sankar, Asokan, Saravanan, Kumanan & Prabhaharan (2007) has proposed a modified genetic algorithm to solve the optimization problem of cutting parameters for constrained machining operations.

Chen & Tsai (1996) combine SA and Hooks-Jeeves pattern search technique for optimizing cutting conditions in complex machining (multi-pass turning operation) to minimize unit operation cost. For optimization of CNC turning process, Juan, Yu & Lee (2003) apply SA-based technique to attain optimal cutting conditions of high sped milling operation. Zain, Haron & Sharif (2010) showed the application of simulated annealing to estimate the optimal effect of the radial rake angle of the tool, combined with cutting speed and feed influencing the surface roughness result.

Vijayakumar, Prabhaharan, Asokan & Saravanan (2003); Baskar, Asokan, Saravanan & Prabhaharan (2005); Wang (2007) and Wu & Yao (2008) proposed an ant colony optimization procedure for determining the machining parameters of multi-pass operations. Saravanan, Asokan & Vijayakumar (2003), Karpat & Ozel (2006), Onwubolu (2006) and Srinivas, Giri &Yang (2009) implemented particle swarm optimization for finding optimum machining parameters.

The problem of optimization of machining parameters is a non-linear optimization problem with constraints, and it is difficult for the conventional optimization algorithms to solve this problem because of problems of convergence speed or accuracy (Cus & Zuperl, 2006). The success of Hopfield and Tank in finding an efficient method for obtaining optimal solutions motivated the researchers to apply neural networks to optimization problems. The motivation behind the Hopfield and Tank neural network model was to take the advantage of the great speed associated with the massively parallel computing capabilities of neural networks for fast solution of optimization problems.

In the last literature, several applications of ANN-based input-output relationship modeling of metal cutting processes are reported. Back propagation neural network, proposed by Rumelhart, Hinton & Williams (1986), have been successfully applied by Sathyanarayanan, Lin & Chen (1992); Jain, Jain & Kalra (1999) and Feng, Wang, & Yu (2002) for modeling a typical creep feed super alloy-grinding, prediction of material removal rate and surface finish parameter of a typical abrasive flow machining, and a honing operation of engine cylinder liners, respectively. Grzesik & Brol (2003) show the usefulness of ANN modeling for controlling surface finish characteristics in multistage machining processes. From the literature reviewed, it is seen that neural networks has been employed for modeling the relationship between machining parameters and process responses. To the best of our knowledge, there is no attempt to obtain optimal or near-optimal machining parameters by using ANNs as the optimization tool.

The purpose of this study is to exhibit the performance of neural networks for metal cutting process optimization. For this purpose, we employ a Hopfield-type dynamical network approach. After the appropriate energy function is constructed by using a penalty function approach, the dynamics are defined by steepest gradient descent on the energy function. The objective of the proposed approach is to minimize total production cost and surface roughness without violating cutting constraints. The machining parameters affecting the objective are assumed to be cutting speed, feed rate and depth of cut. The production model of Shin & Joo (1992) is adopted to illustrate the proposed model and to simplify the comparisons between different optimization methods using illustrative examples.

A general outline of this chapter is as follows. The list of notations and acronyms used in this section is given in 6.2. We give a brief explanation of the process and mathematical formulation of the optimization problem in 6.3. The proposed network is described in subsection 6.4. Subsection 6.5 provides the computational results, and the conclusions with future research directions are given in 6.6.

6.2 Notations and Acronyms

A_5	Cost of tool preparation
C_{ij}	Minimum cost corresponding to d_{ij} depth of cut, $/piece
C_0	Taylor's tool life constant
d_{ij}	j^{th} element of depth of cut series for i^{th} pass, mm
$d_{i,min}$	Minimum allowable depth of cut for i^{th} pass, mm
$d_{i,max}$	Maximum allowable depth of cut for i^{th} pass, mm
d_r	Depth of cut for rough pass, mm
d_s	Depth of cut for finish pass, mm
d_t	Total depth of cut, mm
f_r	Feed for rough pass, mm/rev
f_s	Feed for finish pass, mm/rev
f_{min}	Minimum allowable feed, mm/rev
f_{max}	Maximum allowable feed, mm/rev
F_{max}	Maximum cutting force, kgf
h_1,h_2	Constants pertaining to tool travel and approach/depart time
k_0	Overhead cost, $/min
k_1	Constant in cutting force equation
k_t	Cost of a cutting edge, $/cutting edge
L	Work piece length
n	Assumed maximum number of rough passes
p,q,r	Exponents of speed, feed rate and depth of cut range in the i^{th} pass
P_{max}	maximum available cutting power, hp
R	Nose radius of cutting tool, mm
R_r	Peak to valley height for surface roughness for rough passes
R_{max}	Peak to valley height for surface roughness for finish pass
t_e	Time required to exchange a tool, min/cutting edge
t_p	Preparation time, min/piece
T_{min}	Minimum tool life, min
T_{max}	Maximum tool life, min
T_p	Fixed interval of time after which tool bit is changed, min
T_s	Tool life in finish pass, min
U	Total production cost per unit, $/piece
Uc_r	Total cost for a rough pass, $/piece
Uc_s	Total cost for finish pass, $/piece
v_{min}	Minimum cutting speed, m/min
v_{max}	Maximum cutting speed, m/min
v_r	Cutting speed for rough pass, m/min
v_s	Cutting speed for finish pass, m/min
μ,υ	Exponents of feed rate and depth of cut in cutting power equation
η	Cutting power efficiency of machine tool
ξ	Constant

6.3 Problem Statement

The factors influencing the machining operation are the type of machining (turning milling etc.), machine tool parameters, cutting conditions, work piece characteristics and type of cutting tool. Undoubtedly, the cutting conditions are the most important factor. Basically, the optimization of machining conditions requires determining the economic process parameters according to a variety of economic criteria (objective function) without violating any imposed cutting conditions. These machining parameters must be selected in such a way that the machine is utilized to the maximum extent and the tool life as long as possible. These are two conflicting objectives and the purpose of optimization is to determine such a set of the cutting conditions that satisfies the limitation equations and balances the conflicting objectives.

6.3.1 Objective Equations

The entire development of planning of the machining processes is based on the optimization of the economic criteria by taking the technical and organizational limitations into account. In the machining operations, mostly used economic criteria are the costs and manufacturing time such as maximum production rate (minimum production time), minimum unit production cost or maximum profit rate. The first two approaches have received much more attention since the mid-1960s. The third approach is not commonly used due to lack of information and uncertainty during manufacturing. The formulation of the problem using minimum production cost or minimum production time as the objective function is quite different, but the basic concepts are identical (Al-Ahmari, 2001).

Shin & Joo (1992) proposed a model to minimize total production cost. The model takes the use of Taylor's tool life equation and is represented by the sum of cost for finish pass, and rough passes.

$$\min U = Uc_s + i = 1\sum_{i=1}^{n} Uc_{r_i} + A_5 \tag{6.1}$$

where A_5 is the cost of tool preparation and $A_5=k_0t_p$.

- The production cost of a finish pass is the sum of machining cost, machine idle cost, tool replacement cost and tool cost:

$$Uc_s = A_2 + A_1 f_s^{\left(\frac{q}{p}-1\right)} d_s^{r/p} \tag{6.2}$$

where
$$A_1 = \frac{\pi D L k_0}{1000 T_p}\left[T_s / C_0\right]^{1/p}\left[T_s + \xi\left(t_e + \frac{k_t}{k_0}\right)\right] \tag{6.3}$$

$$A_2 = k_0\left(h_1 L + h_2\right) \tag{6.4}$$

and the relationship between T_p and T_s is defined as $T_p=T_s/\xi$

- The production cost of a rough pass is similar to the cost for finish pass and is given by:

$$Uc_{r_i} = A_2 + A_1 f_{r_i}^{\left(\frac{q}{p}-1\right)} d_{r_i} \tag{6.5}$$

6.3.2 Limitations

The parameters of the machining economics problem are, usually the cutting speed, feed rate, and the depth of cut since they have the greatest effect on the success of machining operation. Moreover, in determining these parameters, special attention is usually given to the restrictions or constraints imposed on the particular operation(s) by the machine tool, the cutting tool and the work-piece (Amiolemhen & Ibhadode, 2004). There are several factors limiting the process parameters. Those factors usually originate from technical specifications and organizational considerations. Generally, the following limitations are taken into account (Cus & Balic, 2003):

1) Permissible range of cutting conditions: Due to the limitations on the machine and cutting tool, and due to the safety of machining, the cutting parameters are limited with the bottom and top permissible limits. Limitations originated from permissible range of cutting conditions are:

- Cutting speed:

$$\frac{c_0^{1/p}}{T_p^{1/p} v_{\max}} \le f^{q/p} d^{r/p} \le \frac{c_0^{1/p}}{T_p^{1/p} v_{\min}} \tag{6.6}$$

- Feed rate:

$$f_{\min} \le f \le f_{\max} \tag{6.7}$$

- Depth of cut:

$$d_{\min} \le d \le d_{\max} \tag{6.8}$$

2) Implied limitations issuing from the tool characteristics and the machine capacity: For the selected tool, the tool maker specifies the limitations of the cutting conditions. Tool characteristics and machine capacity limit the process by following parameters:

- Cutting force

$$F = k_1 f^{\mu} d^{\nu} \le F_{\max} \tag{6.9}$$

- Cutting power:

$$f^{\left(\mu - \frac{q}{p}\right)} d^{\left(\nu - \frac{r}{p}\right)} \le \frac{6120 \eta T_p^{1/p} P_{\max}}{k_1 c_0^{1/p}} \tag{6.10}$$

Surface roughness is mainly a result of process parameters such as tool geometry and cutting conditions:

$$f_s \le \left[8 r R_{\max}\right]^{1/2} \tag{6.11}$$

By using (6.7) and (6.11):

$$f_{\min} \le f \le \min\left(f_{\max}, \sqrt{8 R R_{\max}}\right) \tag{6.12}$$

It has been observed that decreasing feed rate helps obtain a good surface finish but increases cost due to machining time. High cutting speeds may help to reduce the surface roughness. But since tool life at high cutting speeds is just a few couple minutes, this solution may not be applicable.

Cutting constraints for both rough pass and finish pass are similar. The cost function of finish pass given by (6.2) can be optimized taking into account the constraints above. For rough passes, same constraints are used for optimization; however, R_r is used instead of R_{max} in (6.12).

In addition to the above constraints, the total stock to be removed is determined as follows (Al-Ahmari, 2001):

$$d_t = d_s + \sum_{i=1}^{n} d_{r_i} \tag{6.13}$$

The cutting parameters can be chosen to be different in each of the individual passes. However for simplification, all rough passes have been considered to be identical. Thus, there are two sets of cutting parameters, one for all rough passes and one for the finish pass.

6.3.3 Mathematical Model of the Problem

In this study, the mathematical model proposed by Shin & Joo (1992) is adopted. Objective function is the minimum unit production cost required to machine a unit piece (Shin & Joo, 1992; Al-Ahmari, 2001).

$$\min U = Uc_s + \sum_{i=1}^{n} Uc_{r_i} + A_5 \tag{6.14}$$

where $A_5 = k_0 t_p$.

The objective function is subject to:

$$v_{s,\min} \le v_s \le v_{s,\max} \tag{6.15}$$

$$v_{r_i,\min} \le v_{r_i} \le v_{r_i,\max} \tag{6.16}$$

$$f_{s,\min} \le f_s \le \min\left(f_{s,\max}, \sqrt{8RR_{\max}}\right) \tag{6.17}$$

$$f_{r_i,\min} \le f_{r_i} \le \min\left(f_{r_i,\max}, \sqrt{8RR_r}\right) \tag{6.18}$$

$$k_1 f_s^{\mu} d_s^{\nu} \le F_{\max} \tag{6.19}$$

$$k_1 f_{r_i}^{\mu} d_{r_i}^{\nu} \le F_{\max} \tag{6.20}$$

$$f_s^{\left(\mu-\frac{q}{p}\right)} d_s^{\left(\nu-\frac{r}{p}\right)} \le \frac{6120 T_p^{1/p} P_{\max}}{k_1 c_0^{1/p}} \tag{6.21}$$

$$f_{r_i}^{\left(\mu-\frac{q}{p}\right)} d_{r_i}^{\left(\nu-\frac{r}{p}\right)} \le \frac{6120 T_p^{1/p} P_{\max}}{k_1 c_0^{1/p}} \tag{6.22}$$

$$d_{s,\min} \le d_s \le d_{s,\max} \tag{6.23}$$

$$d_{r_i,\min} \le d_{r_i} \le d_{r_i,\max} \tag{6.24}$$

$$d_t = d_s + \sum_{i=1}^{n} d_{r_i} \tag{6.25}$$

6.3.4 Solution Methodology

Yellowley & Gunn (1989) proposed a two-stage optimization algorithm, called optimal subdivision of depth of cut, to solve the multi-pass machining problem described above. The authors divided the total production cost minimization problem into two sub-problems as mentioned below:

6.3.4.1 Stage-1

This phase consist of determining costs for individual finish or rough pass considering various fixed values of depth of cut. A series of depth of cut is defined between minimum and maximum allowable depth of cuts. Minimization of cost for the finish pass can be achieved by using the maximum permissible value of feed under the constraints. The following steps are used to find the minimum cost for a finish pass.

<u>*Step 1:*</u> The maximum feed values (f) satisfying Equations (6.6), (6.10), (6.11) and (6.12) are determined for a given depth of cut. The minimum feed rate satisfying these constraints is selected as the optimum value of feed, f_s*. If selected feed rate is smaller than minimum allowable feed rate, then minimum allowable feed rate is assigned as f_s*.

<u>Step2:</u> For the selected f_s* and given T_p, d_s, the optimal speed, v_s*, is calculated from tool life equation:

$$v_s^* = \frac{c_0^{1/p}}{T_p^{1/p} f_s^* d_s^{r/p}} \qquad (6.26)$$

Using optimal speed, v_s*, optimal feed rate, f_s*, and d_s, power is calculated. If calculated power exceeds the maximum power available at the spindle, optimal speed is calculated by using Equation (6.12).

<u>Step3:</u> Next, the minimum cost for the finish pass is obtained using the values v_s*, f_s* for a given depth of cut, d_s.

A similar procedure is adopted for finding minimum cost for a rough pass (Yellowley & Gunn, 1989; Gupta, Batra & Lal, 1995).

6.3.4.2 Stage-2

In stage 2, number of rough passes (n_i), optimal combinations of depths of cut d_s^* and d_{ri}^* for (n_i+1) (i.e. one finish pass plus n_i rough passes) and minimum total production cost are determined. The optimization of this sub-problem is achieved subject to the following constraints:

1. There should be n rough passes and one final finish pass.
2. The individual depths of cut for n rough passes and one finish pass should be in the range of allowable depth of cut for roughing and finishing operations.
3. The sum of individual depth of cuts for *n+1* passes should be equal to the total stock to be removed (Gupta, Batra & Lal, 1995).

As mentioned in the previous section, in the literature, different solution approaches have been proposed for the solution of the problem. Gupta, Batra & Lal (1995) proposed a mixed integer linear programming approach for minimization of total production cost. By this approach, optimal subdivisions of depth of cut for rough passes and finish pass, optimal number of passes and minimum total production cost are determined. The results of the proposed model and the method of Shin & Joo (1992) are compared. For all the depths of cut the proposed model results in reduced production cost and the optimal number of passes in the proposed model are also either lower or equal to the ones found by Shin & Joo (1992). The proposed model of Gupta, Batra & Lal (1995) is also used as a reference for the proposed neural network approach in the study. Satishkumar, Asokan & Kumanan (2006) proposed different nontraditional optimization techniques- genetic algorithm, simulated annealing and ant colony optimization - for optimizing the depth of cut in multi pass machining problem given by Shin & Joo (1992), and compared the results with those obtained by Gupta, Batra & Lal (1995) and Shin & Joo (1992). Based on these comparisons they concluded that, the proposed nontraditional techniques give better results. Al-Ahmari (2001) proposed a direct non-linear mathematical model to solve the second stage for both finishing and rough cutting in a single run. They compared the proposed model to the model of Gupta, Batra & Lal (1995), and concluded that there is no significant difference between the results of two approaches in terms of optimum number of passes and cost.

6.4 Design of the Proposed Hopfield-Type Network

In this section, we describe how dynamical gradient networks can be used to solve the considered problem presented in the previous section. The proposed approach is an extension of the original formulation given in Hopfield (1985). The proposed network is based on the mixed integer linear programming model of Gupta, Batra & Lal (1995) and by this approach it is possible to solve the optimization problems which can be transformed into a linear programming model.

Gupta, Batra & Lal (1995) formulated the machining parameter optimization problem as a mixed integer linear programming model as follows:

$i = 0$ implies finish pass

$i = 1,2,\ldots,n$ implies i^{th} rough pass

$j = 1,2,\ldots,m_i$ implies correspondence to j^{th} depth of cut

$$X_{ij} = \begin{cases} 1 & \textit{if } d_j \textit{ value of depth of cut is selected in the } i^{th} \textit{ pass} \\ 0 & \textit{if } d_j \textit{ value of depth of cut is not selected in the } i^{th} \textit{ pas} \end{cases}$$

n = The maximum number of rough passes required

Then, the integer programming representation of the problem is as follows (Gupta, Batra & Lal, 1995):

$$\min U = \sum_{i=0}^{n} \sum_{j=0}^{m_i} C_{ij} X_{ij} \tag{6.27}$$

$$s.t. \qquad \sum_{j=0}^{m_0} X_{0j} = 1 \tag{6.28}$$

$$\sum_{j=0}^{m_i} X_{ij} \leq 1 \qquad \textit{for } i = 1,2,...,n \tag{6.29}$$

$$\sum_{j=0}^{m_i} \sum_{i=0}^{n} d_{ij} X_i = d_t \tag{6.30}$$

$$X_{0j} \in \{0,1\} \tag{6.31}$$

$$X_{ij} \in \{0,1\} \qquad \textit{for } i = 1,2,...,n \tag{6.32}$$

$$X_{0j}, X_{ij} \geq 0 \qquad \textit{for } i = 1,2,...,n \tag{6.33}$$

The first constraint (6.28) implies that there is only one depth of cut selection for the finish pass and the finish pass must always be selected. The second constraint (6.29) means that there is only one depth of cut selection in case a rough pass is selected. The last constraint (6.30) indicates that the sum of individual depths of cut is equal to the total depth of stock removal (Gupta, Batra and Lal, 1995).

To construct a dynamical gradient based network representation of the model above, firstly the network architecture is explained, and then derivation of the energy function representing the proposed network, and dynamics and proof of the convergence of the proposed network are given. Finally, the proposed approach is illustrated with an example.

6.4.1 The Network Architecture

The proposed gradient network has two interconnected networks, a maximum network and a continuous network. The maximum network (X_{0j} network) is used to assign a depth of cut for the finish pass (X_{0j}) and the continuous network (X_{ij} network, $i>0$) is used to determine optimal number of rough passes and to assign depth of cut(s) for rough pass(es).

The input-output scheme for each of the neural networks is shown in Figure 6.1. UX_{ij} symbolizes the input to the neuron for i^{th} rough pass and j^{th} depth of cut. UX_{0j} denotes the input to the neuron for depth of cut of finish pass. The dynamics of the proposed network will be defined in terms of input variables. VX_{ij} demonstrates the output of neuron for i^{th} rough pass and j^{th} depth of cut. This neuron will be activated if d_j value of depth of cut is selected in the i^{th} pass. Otherwise, the state of the neuron will be set as zero to indicate that the neuron is not activated. VX_{0j} depicts the depth of cut for finish pass. Similarly, this neuron will be activated if d_j value of depth of cut is selected in the finish pass. Neurons with sigmoidal nonlinearity are used to represent discrete variables, X_{ij}, so the activation function for discrete neurons can take any sigmoidal form with slopes λ.

Since all variables of the second network are binary-values, the outputs of the neurons are converged to discrete values by using hard limit transfer function.

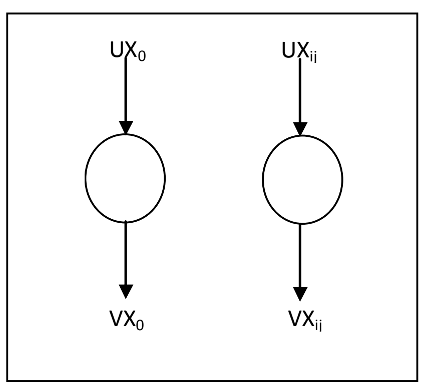


Figure 6.1 The input-output scheme for the neurons representing each unit

6.4.2 The Energy Function

Instead of using linear programming or the *k-out-of-N* rules to develop the energy function, we directly formulate the cost function according to the constraints term by term. The energy function for this network is constructed using a penalty function approach. That is, the energy function E consists of the objective function plus a penalty function to enforce the constraints. The penalty function involves the sum of the penalty terms each of which corresponds to each constraint of the problem. For the problem considered, the penalty function will include three penalty terms: *P1, P2* and *P3* corresponding to each constraint.

To prevent the selection of more than one depth of cut for the finish pass, the first penalty term, $P1 = \left(\sum_{j=0}^{m_0} X_{0j} - 1 \right)^2$, which will add a positive penalty if the solution does not satisfy the equality constraint given in (6.16), is included in the energy function. This penalty term will yield zero when these equality constraint is satisfied.

The second penalty term, *P2*, will add a positive penalty if the solution does not satisfy the inequality constraint given in (17). In accordance with this constraint, *P2* will take the following form, $P2 = \sum_{i=1}^{n} \nu \left(\sum_{j=0}^{m_i} X_{ij} - 1 \right)$, where υ represents the penalty function. υ (ε)= ε^2 for all ε>0 and υ (ε)= ε for all ε<=0. And the functional form of this function is shown in Figure 6.2.

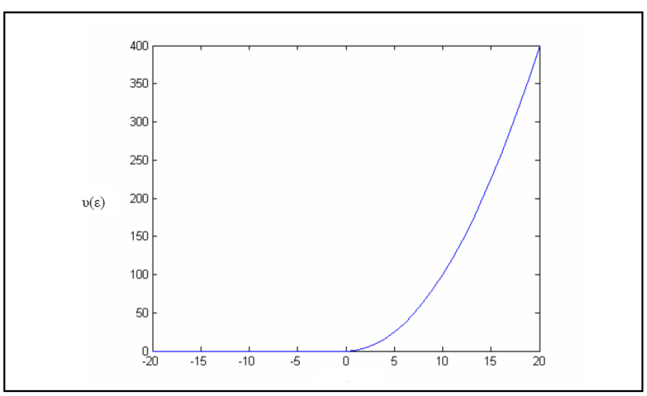


Figure 6.2 Penalty function for enforcing inequality constraints

The third term *P3* will add a positive penalty if the equality constraint given in (6.18) is violated. Therefore, *P3* should be defined by

$$P3 = \left(\sum_{j=0}^{m_i} \sum_{i=0}^{n} d_{ij} X_{ij} - d_r \right)^2 \quad 6.34)$$

We require that X_{0j} and X_{ij} $\epsilon\{0,1\}$. These constraints will be captured by the fourth and the fifth terms, *P4* and *P5*, which will add a positive penalty if the binary constraints given in (6.19) and (6.20) are violated. Hence,

$$P4 = \sum_{i=1}^{n} \sum_{j=0}^{m_i} X_{ij} (1 - X_{ij}) \quad (6.35)$$

$$P5 = \sum_{j=0}^{m_0} X_{0j} (1 - X_{0j}) \quad (6.36)$$

In Figure 6.3 the functional form of this penalty term is shown.

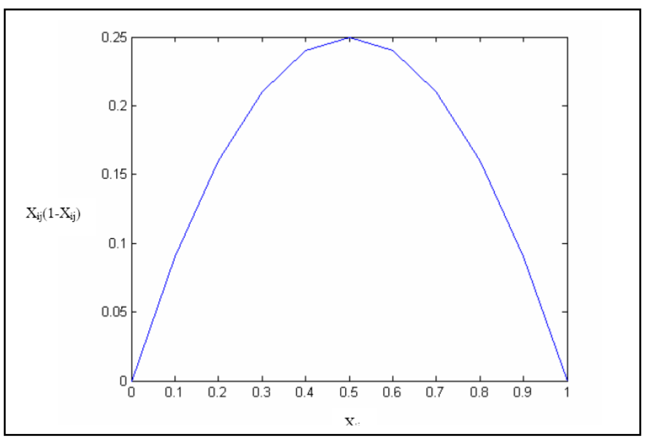


Figure 6.3. Penalty function for enforcing the 0,1 constraints

The non-negativity constraints given in (6.21) are not added to the energy function as penalty terms since these constraints will be captured by using an input-output function, *g,* where g(ε)= ε for all ε>0 and g (ε)= 0 for all ε<=0. Its functional form is given in Figure 6.4. In other words, for zero and positive input values, the activation function will be linear, and so the outputs will be equal to the inputs of the neurons, and for the negative values the output values will be zero.

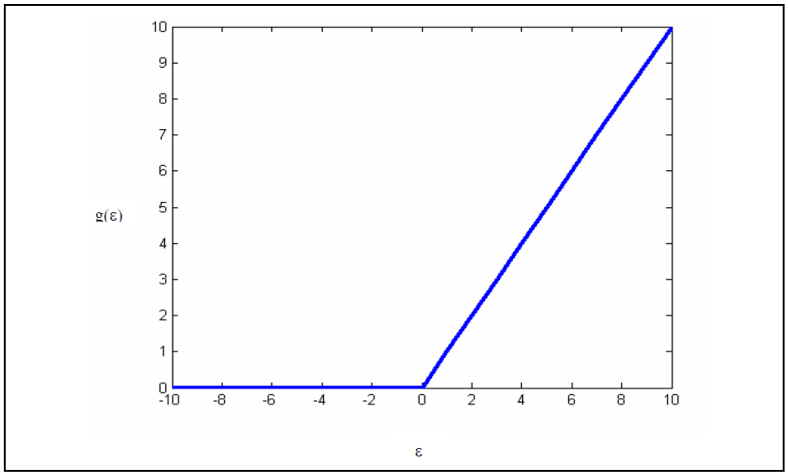


Figure 6.4 Activation function for continuous neurons of E, T and X Networks

Therefore, the penalty function for the coupled gradient network can be written as follows:

$$P = B\left(\sum_{j=0}^{m_0} X_{0j} - 1\right)^2 + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} X_{ij} - 1\right) + D\left(\sum_{j=0}^{m_i}\sum_{i=0}^{n} d_{ij} X_{ij} - d_r\right)^2$$
$$+ E\sum_{i=1}^{n}\sum_{j=0}^{m_{ii}} X_{ij}(1 - X_{ij}) + F\sum_{j=0}^{m_0} X_{0j}(1 - X_{0j}) \tag{6.37}$$

If we sum the objective function given in (6.27) and the penalty function, we will have the following energy function to be minimized:

$$A\sum_{i=0}^{n}\sum_{j=0}^{m_0} C_{ij} X_{ij} + B\left(\sum_{j=0}^{m_0} X_{0j} - 1\right)^2 + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} X_{ij} - 1\right) + D\left(\sum_{j=0}^{m_i}\sum_{i=0}^{n} d_{ij} X_{ij} - d_r\right)^2$$
$$+ E\sum_{i=1}^{n}\sum_{j=0}^{m_{ii}} X_{ij}(1 - X_{ij}) + F\sum_{j=0}^{m_0} X_{0j}(1 - X_{0j}) \tag{6.38}$$

where A,B,C,D,E and F are positive penalty coefficients.

If we rewrite the energy function in terms of the output variables, we may obtain:

$$E = A\sum_{i=0}^{n}\sum_{j=0}^{m_0} C_{ij} VX_{ij} + B\left(\sum_{j=0}^{m_0} VX_{0j} - 1\right)^2 + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} VX_{ij} - 1\right) + D\left(\sum_{j=0}^{m_i}\sum_{i=0}^{n} d_{ij} VX_{ij} - d_r\right)^2$$
$$+ E\sum_{i=1}^{n}\sum_{j=0}^{m_{ii}} VX_{ij}(1 - VX_{ij}) + F\sum_{j=0}^{m_0} VX_{0j}(1 - VX_{0j}) \tag{6.39}$$

Applying a winner take all (WTA) mechanism to the X_{0j} network, the energy terms for the first constraint (with weighting factor B) can be omitted from the energy function. The *WTA* learning rule guarantees the satisfaction of (6.28), that is only one depth of cut is assigned for the finish pass. In addition, it ensures the binary constraint. The energy term for this constraint (with weighting factor F) is also dropped from the energy function. By this way, these energy terms will be handled explicitly. Therefore, the energy function takes the following form:

$$E = A\sum_{i=0}^{n}\sum_{j=0}^{m_0} C_{ij} VX_{ij} + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} VX_{ij} - 1\right) + D\left(\sum_{j=0}^{m_i}\sum_{i=0}^{n} d_{ij} VX_{ij} - d_r\right)^2 + E\sum_{i=1}^{n}\sum_{j=0}^{m_{ii}} VX_{ij}(1 - VX_{ij}) \tag{6.40}$$

The penalty term, $E\sum_{i=1}^{n}\sum_{j=0}^{m_{ii}} VX_{ij}(1 - VX_{ij})$ can also be eliminated from the energy function because by applying hard limit transfer function for continuous neurons, the outputs may take values of either 0 or 1. Thus, the binary constraint X_{ij} *ϵ {0,1}* is satisfied for all variables and the energy term for this constraint can be dropped from the energy function. Final form of the energy function can be written as follows.

$$E = \min\ A\sum_{i=0}^{n}\sum_{j=0}^{m_0} C_{ij} VX_{ij} + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} VX_{ij} - 1\right) + D\left(\sum_{j=0}^{m_i}\sum_{i=0}^{n} d_{ij} VX_{ij} - d_r\right)^2 \tag{6.41}$$

Although the original energy function given in Eq 6.39 includes many penalty terms to be minimized using a difficult trial-and error procedure, by imposing a competitive WTA rule for the updating of the neurons, we get rid of the trouble of determining the proper values for some of the weighting factors. We can see from the above equation that except the weighting factor of the original objective function, the resulting energy function includes only 2 penalty parameters to be determined.

6.4.3 The Dynamics

Once the energy function is determined, it is necessary to consider the equation of motion of the neuron input. The dynamics of the proposed network are obtained by gradient descent on energy function. The equations of motion are obtained as follows:

For the X_{0j} network:

$$\frac{dUX_{0j}}{dt} = -\frac{\partial E}{\partial VX_{0j}}$$
$$= -AC_{ij} - 2Dd_{0j}\left[(VX_{0j}d_{0j} + \sum_{j=o}^{m_i} d_{ij}VX_{ij}) - d_r\right] \tag{6.42}$$

For the X_{ij} network:

$$\frac{dUX_{ij}}{dt} = -\frac{\partial E}{\partial VX_{ij}}$$
$$= -AC_{ij} - 2Cv'\left[\sum_{l=o}^{m_i} VX_{il} - 1\right] - 2Dd_{ij}\left[\left[(VX_{0j}d_{0j} + \sum_{j=o}^{m_i} d_{ij}VX_{ij}) - d_r\right]\right] \tag{6.43}$$

where v' is the derivative of the penalty function v and v'(ε)=2 ε for all ε>0 and v'(ε)=0 for all ε≤0.

The states of neurons at iteration k are updated at iteration k by using the first-order Euler method as follows:

$$UX_{0j}^{k} = UX_{0j}^{k-1} + \eta_{X_{ij}} \frac{dUX_{0j}}{dt}$$
$$UX_{ij}^{k} = UX_{ij}^{k-1} + \eta_{X_{ij}} \frac{dUX_{ij}}{dt} \tag{6.44}$$

where $\eta_{X_{ij}}$ and $\eta_{X_{ij}}$ are positive coefficients which will be used to scale the dynamics of the two networks.

Since the computation is performed in all neurons at the same time, the network operates in a fully parallel mode. Neuron outputs are calculated by $V=g\ (U)$, where g (.) is the activation function, U is the input and V is the output of a neuron. As mentioned before, the activation function, g, for the continuous neurons X_{ij} will take the usual sigmoidal form as displayed in Figure 6.5. Thus, the outputs of the neuron will take values between 0 and 1. However, the variables are binary that a hard limit transfer function is applied to convert continuous neurons to discrete neurons as in Fig 6.6.

$$VX_{ij} = hard\lim\left(g\left(\lambda * UX_{ij}\right)\right) \tag{6.45}$$

where λ is the slope of the activation function.

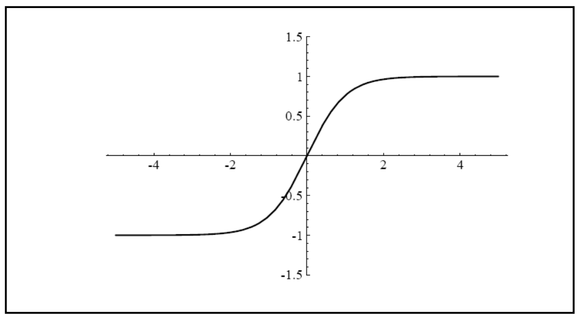


Figure 6.5 Activation function for neurons of X_{0j} network.

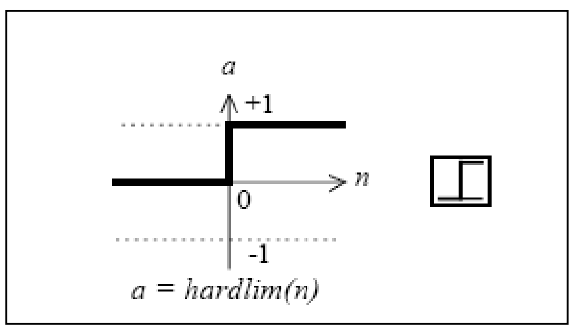


Figure 6.6 Hard limit transfer function

The neuron outputs of the WTA network are updated by the maximum neuron model of Takefuji, Lee, & Aiso (1992) as below and its functional form is given in Figure 6.7.

$$VX_{0j} = \begin{cases} 1 & if\ UX_{0j} = \max(UX_{01}, UX_{02}, ..., UX_{0m}) \\ 0 & otherwise \end{cases} \tag{6.46}$$

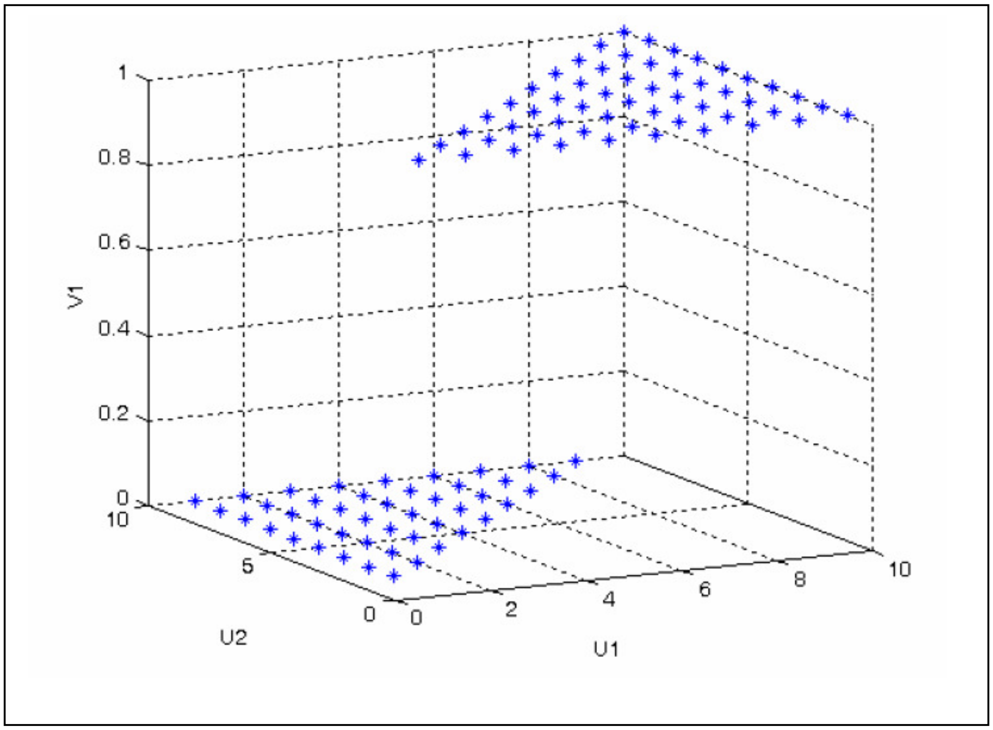


Figure 6.7 Activation function for neurons of the max network when there are two inputs

6.4.4 Convergence

In order to use the proposed Hopfield-like network for solution of the problem, the convergence of the network must be proved. This can be done by showing:

a. Energy does not increase along the trajectories
b. Energy is bounded below
c. Solutions are bounded below
d. Time derivative of the energy is equal to zero only at equilibria.

If we want to consider the derivative of a function at an endpoint of the interval over which it is defined, we need to use a one-sided derivative because the function is not defined beyond the endpoint. Thus, in the proposed network VX_{0j}'s are not differentiable functions of time t and they have right-hand derivatives.

To prove the convergence of the proposed network, an extension of the La Salle's invariance principle can be used (Sengor, Cakir, Guzelis, Pekergin, & Morgul, 1999). The Lemma below, which is needed for taking the time derivative of the energy, states that the chain rule is valid also for the right derivative.

Definition: The right derivative of a function $x(.) : R \to R^n$ is defined as $\frac{dx(t)}{dt^+} = \lim_{\Delta \to 0} \frac{x(t+\Delta) - x(t)}{\Delta}$ where $\Delta \to 0^+$ means that Δ approaches zero throughout positive values only (Sengor, Cakir, Guzelis, Pekergin, & Morgul, 1999).

Lemma: Consider the functions $\psi(.) : D_\psi \subset [0, \infty) \to D_g \subset R^n$ and $g(.) : D_g \to R$. Let $t \in Int(D_\psi)$ with Int stands for the set of interior points. Assume that g(.) is continuously differentiable at $\psi(t)$, and $\psi(.)$ is right differentiable at t. Then, g o ψ is right differentiable at t and $\frac{d(go\psi((t)}{dt^+} = [\nabla_\psi g(\psi)] \frac{d_\psi(t)}{dt^+}$ (Sengor et al., 1999).

Using the Lemma given above, the time derivative of the energy function E can be found as follows:

$$
\begin{aligned}
\frac{dE}{dt^+} &= \sum_{i=1}^{n} \sum_{j=0}^{m_i} \frac{\partial E}{\partial VX_{ij}} \frac{dVX_{ij}}{dt} + \sum_{j=o}^{m_0} \frac{\partial E}{\partial VX_{0j}} \frac{dVX_{0j}}{dt} \\
&= \sum_{i=1}^{n} \sum_{j=0}^{m_i} - \frac{dUX_{ij}}{dt} \frac{dVX_j}{dt} + \sum_{j=0}^{m_0} - \frac{dUX_{0j}}{dt} \frac{dVX_{0j}}{dt} \\
&= \sum_{i=1}^{n} \sum_{j=0}^{m_i} - \frac{dUX_{ij}}{dVX_{ij}} \frac{dVX_{ij}}{dt} \frac{dVX_{ij}}{dt} + \sum_{j=0}^{m_0} - \frac{dUX_{0j}}{dt} \frac{dVX_{0j}}{dUX_{0j}^+} \frac{dUX_{0j}}{dt} \\
&= \sum_{i=1}^{n} \sum_{j=0}^{m_i} - \frac{dUX_{ij}}{dVX_{ij}} \left(\frac{dVX_{ij}}{dt} \right)^2 - \sum_{j=0}^{m_0} \left(\frac{dUX_{0j}}{dt} \right)^2 \frac{dVX_{0j}}{dUX_{oj}^+}
\end{aligned}
\qquad (6.47)
$$

where $\frac{\partial E}{\partial V}$ is replaced by $-\frac{dU}{dt}$.

Since the tangent sigmoid function is strictly increasing, $\frac{dUX_{ij}}{dVX_{ij}} = [g^{-1}(VX_{ij})]' \geq 0$ for this function. Thus the first term of (6.47) will be obviously negative.

Although the output of X_{0j} network is not differentiable function of time, it has right-derivative and for the neurons of the maximum neural network X_{0j}, the right-derivative of the energy function with respect to time can be written as:

$$\begin{aligned} \frac{dE}{dt^{+}} &= \sum_{j=0}^{m_0} -\frac{dUX_{0j}}{dt}\frac{dVX_{0j}}{dt} \\ &= \sum_{j=0}^{m_0} -\frac{dUX_{0j}}{dt}\frac{\partial VX_{0j}}{\partial UX_{0j}^{+}}\frac{dUX_{0j}}{dt} \end{aligned} \tag{6.48}$$

Since VX_{0j}'s are piecewise constant functions of UX_{0j}'s, $\frac{\partial VX_{0j}}{\partial UX_{0j}^{+}} = 0$. Therefore, the second term in (6.47) will be zero.

Because the energy E is bounded (since the cost is always greater than zero), we conclude that the energy does not increase along trajectories, so we can write $\frac{dE}{dt} \leq 0$.

$\frac{dE}{dt} = 0$ implies that $\frac{dVX_{ij}}{dt} = 0$ *for all* i, j and $\frac{dVX_{oj}}{dt} = 0$. In other words, if the points are equilibrium points then it can be seen that $\frac{dE}{dt} = 0$. All trajectories go to the points where $\frac{dE}{dt} = 0$ and energy eventually becomes constant thus any trajectory reaches an equilibrium point.

Because VX_{0j}s and VX_{ij} are binary, they will be bounded. Combining this fact with the fact that the cost is always greater than zero, implying the energy E is bounded below; we can conclude that the time evolution of the network is a motion in space tends to that minimum point as t goes to infinity.

6.4.5 Selection of Parameters

In order to simulate the proposed network, values for the following parameters must be chosen:

- **i.** The penalty coefficients: A, C and D.
- **ii.** The scaling factor, η_x and activation slope, λ.
- **iii.** Initial conditions (States of neurons, UX_{ij}).

Because there is no theoretically established method for choosing the values of the penalty coefficients for an arbitrary optimization problem, the appropriate values for these coefficients can be determined empirically. That is simulation runs are conducted, and optimality and/or feasibility of the resulting equilibrium points of the system are observed. The network can be initialized to small random values, and then synchronous or asynchronous updating of the network will allow a minimum energy state to be attained. In order to ensure smooth convergence, step size must be selected carefully (Watta & Hassoun, 1996).

The dynamics of the proposed Hopfield-like gradient network will converge to local minima of the energy function E. Since the energy function includes three terms, competing to be minimized, there may be local minima and a tradeoff among the terms. An infeasible solution may be obtained when at least one of the constraint penalty terms is non-zero. In this case, the objective function term will generally be quite small but the solution will not be feasible. Alternatively, a local minimum, which causes a feasible but not a good solution, may be encountered even if all the constraints are satisfied. In order to satisfy the each penalty term, its associated penalty parameter can be increased but this results an increase in other penalty terms and a tradeoff occurs. The penalty parameters that result feasible and good solutions, which minimize the objective function, should be found.

Determining the appropriate values of the penalty parameters, network parameters and initial states are so critical issues associated with gradient type networks that by adjusting the parameters, the convergence performance to valid solutions can be improved. It is clear that solving process planning problems represented by many constraints will cause a tradeoff among the penalty terms to be minimized.

Due to the problems of Hopfield like NNs in solving optimization problems, various modifications are proposed to improve the convergence of the Hopfield network. Here, we propose to use time varying penalty parameters, proposed by Dogan & Guzelis (2006), that take zero values as a starting value and then are increased in a linear fashion in a stepwise manner to reduce the feasible region and also by updating all the neurons synchronously, better simulation results are obtained.

The proposed gradient network algorithm can be summarized by the following pseudo-code.

Step 1. Construct an energy function for the considered problem using a penalty function approach.

Step 2. Initialize all neuron states to random values.

Step 3. Select the slope of the activation function (λ) and step sizes (η) and determine the penalty parameters evolving with time.

Step 4. Compute the motion equations by (6.42) and (6.43). Update neuron inputs, U by the first-order Euler method which is explained through (6.44) and then update the neuron output V using equations (6.45) and (6.46).

Step 5. Repeat the iterations n times and check the cost terms of the energy function penalized. If the required criterion is met proceed to Step 6, otherwise go back to Step 3 to pass to other phase of the simulation. If the work is in the part of the simulation where all the constraints are taken into consideration, check whether the energy has converged to a local minimum. If yes, proceed to step 6 otherwise go back to Step 5.

Step 6. If the energy has converged to local minimum, examine the final solution to determine feasibility and optimality.

Step 7. Adjust parameters *A, C, D* if necessary to obtain a satisfactory solution, reinitialize neuron states and repeat from step 5.

6.4.6 Simulation Results

In this section, a simulation experiment was conducted In order to evaluate the performance of the proposed gradient network in terms of solution quality. The example given by Shin & Joo (1992) and considered by Gupta, Batra & Lal (1995) is used to validate the performance of the proposed approach. The following data are used for the example problem:

$D = 50\,\text{mm}$	$L = 300\,\text{mm}$	$d_t = 6\ \text{mm}$
$d_{min} = 1{\cdot}0\,\text{mm}$	$d_{max} = 3{\cdot}0\,\text{mm}$	$f_{min} = 0{\cdot}1\ \text{mm rev}^{-1}$
$f_{max} = 5\,\text{m min}^{-1}$	$V_{max} = 500\,\text{m min}^{-1}$	$C_0 = 6 \times 10^{11}$
$p = 5$	$q = 1{\cdot}75$	$r = 0{\cdot}75$
$T_{min} = 25\,\text{min}$	$T_{max} = 45\,\text{min}$	$F_{max} = 200\ \text{kgf}$
$P_{max} = 5\,\text{kW}$	$k_1 = 108$	$\mu = 0{\cdot}75$
$v = 0{\cdot}95$	$R = 1{\cdot}2\,\text{mm}$	$R_{max} = 10\ \mu\text{m}$
$R_r = 100\ \mu\text{m}$	$h_1 = 7 \times 10^{-4}$	$h_2 = 0{\cdot}3$
$k_0 = 0{\cdot}5\ \$\ \text{min}^{-1}$	$k_t = 2{\cdot}5\ \$\ \text{edge}^{-1}$	$t_p = 0{\cdot}75\ \text{min piece}^{-1}$
$t_e = 1{\cdot}5\ \text{mm/edge}$		

For the above data, *A1, A2* and *A5* are calculated as:

$A1$=0.249; assuming Tp=25 min

$A2$= 0.255

$A3$= 0.375

The trial number of rough passes (n_i) are calculated based on the maximum depth of cut allowed in the roughing operation and the depth of cut for a finish pass within its range. Thus, the trial number of rough passes is assumed to be 3 up to 10 mm. Thus $i=0,1,2$ and 3 where $i=0$ implies the finish pass.

The value of m_i for $i=0,1,2,3$ is taken as 20 for generation of depth of cut series: $j=1,2,\ldots,20$. Therefore, $d_{i0}= 1.0\ mm$, $d_{i1}= 1.1\ mm$ and so on up to $d_{i20}=3.0\ mm$.

The proposed algorithm is applied for different stocks to be removed range between 6 mm and 10 mm. Table 6.1 shows the machining cost of a single pass corresponding to each depth of cut obtained by Gupta, Batra & Lal (1995).

The proposed procedure was implemented in Matlab language (Version 6.5) and the initial conditions of the network were chosen randomly from uniform distribution on the interval [0,1]. A penalty parameter method is proposed to be used during simulation experiments.

6.4.6.1 Example-1

In the first example, for both rough pass and finish pass, allowable depth of cut is assumed to be equal and ranges between 1 mm and 3 mm with a step size of 0.1.

In the following paragraph, we solve the problem applying the steps of the proposed approach given in the previous section. The problem is solved for total stock to be removed is 6.0 mm.

Table 6.1 Cost of a single and rough pass for different values of depth of cut

Depth of cut (d_{ij})	**Cost of single rough pass (c_{ij})**	**Cost of single finish pass (c_{0j})**
1.00	0.522	0.788
1.10	0.525	0.796
1.20	0.529	0.803
1.30	0.532	0.809
1.40	0.535	0.816
1.50	0.538	0.822
1.60	0.541	0.827
1.70	0.544	0.832
1.80	0.546	0.837
1.90	0.549	0.842
2.00	0.551	0.847
2.10	0.555	0.855
2.20	0.569	0.855
2.30	0.583	0.859
2.40	0.597	0.863
2.50	0.611	0.867
2.60	0.625	0.870
2.70	0.639	0.874
2.80	0.653	0.877
2.90	0.667	0.880
3.00	0.681	0.884

Step 1. For the problem considered, the following energy function is obtained by using a penalty approach.

$$E = \min A\sum_{i=0}^{n}\sum_{j=0}^{m_0} C_{ij}VX_{ij} + C\sum_{i=1}^{n} v\left(\sum_{j=0}^{m_i} VX_{ij} - 1\right) + D\left(\sum_{j=0}^{m_0} d_{0j}VX_{0j} + \sum_{i=1}^{n}\sum_{j=0}^{m_i} d_{ij}VX_{ij} - 6\right)^2 \quad (6.49)$$

Step 2. All the neuron inputs are randomly chosen from uniform distribution on the interval [0,1] and the initial values of the neuron outputs are fixed by activation functions.

Step 3. For the first phase of the simulation, Activation slope for continuous neurons is set as λ=100 and step sizes of both networks are selected as η=0,0005.

There are three penalty parameters in the energy function. But, since the penalty parameter A belongs to the original objective function, here, we will only deal with the satisfaction of the constraints and try to determine the values of the penalty parameters enforcing the constraints which will guarantee a feasible solution.

It is decided to penalize the equality and inequality constraints by using its associated parameter *C* and *D*. Thus, the initial values for penalty parameters *C* and *D* are set to deterministic values as 0 and 100, respectively.

Step 4. Start iteration

Step 4.1. The equations of motion are calculated by using equations (6.27) and (6.28).

Step 4.2. Neuron inputs are updated by Euler approximation by using (6.29).

Step 4.3. Then neuron outputs are updated using activation functions defined in (6.30) and (6.31).

Step 5. After performing trial-and-error experiments, it is seen that the best value of C and D is found as 120 and 20, respectively. In the second stage of the simulation experiment, it is decided to see the impact of the predetermined values of *C* and *D* on the results. Smaller step sizes are used for updating the neurons of the maximum network and a larger step size value is used for updating the continuous network, which are determined empirically. In this part of the simulation experiment, the activation slope and the step sizes are chosen as λ=100, $\eta_{X_{0j}} = 0.0001$, $\eta_{X_{ij}} = 0.0005$.

Step 6. We examine the final solution to determine the feasibility and optimality. Since all the constraints are satisfied, this can be interpreted as the optimum solution and the value of the energy function referred as the value of the objective function of the problem.

This algorithm is repeated for 2000 iterations and the evolution of the objective function during the simulation is given in Figure 6.8.

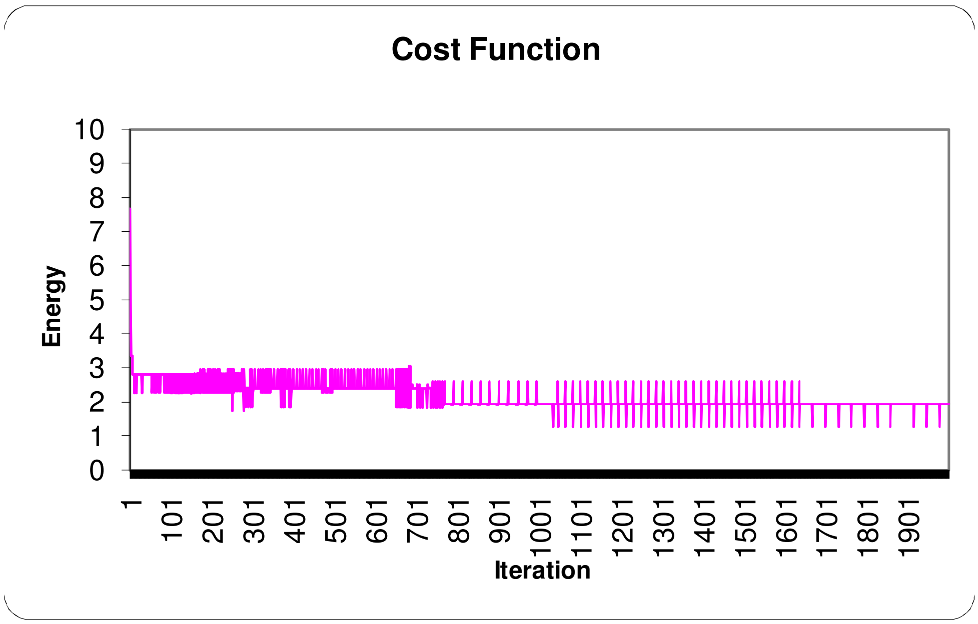


Figure 6.8 Energy evolution of network during simulation

The problem is solved for total stock to be removed from 6.0 mm to 10. mm. The optimal set of parameters found for the proposed networks are given in Table 6.2.

Table 6.2 Parameters used in the simulation

Stock To be Removed / Parameters	6.0 mm	8.0 mm	8.5 mm	9.0 mm	9.5 mm	10.0 mm
A	1	1	1	1	1	1
C	120	550	550	550	680	550
D	20	40	50	40	58	55
λ	100	100	100	100	100	100
$\eta_{x_{0j}}$	0.0001	0.0005	0.0001	0.0001	0.0001	0.0001
$\eta_{x_{ij}}$	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005

In Tables 6.3-6.8, the solutions obtained by the gradient network using the determined parameters are compared with other solutions in the literature.

6.4.6.2 Example-2

In the above example, the depth of cuts for both finish and rough passes were taken to be in the range of 1.0 mm to 3.0 mm. However, from a practical point of view this depth of cut range for the finish pass seems to be on the higher side and the range is taken as 0.4 mm to 1.2 mm (Gupta, Batra & Lal, 1995). The depth of cut range for rough passes and all other data are the same as in Example-1.

Table 6.3 Results for example-1 (Stock to be removed=6.0 mm)

Stock to be removed=6.0 mm	Proposed Model	Shin and Joo (1992)	Gupta et al. (1995)	Al-Ahmari (2001)
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	1	2	1	1
Finish pass (mm)	3.0	1.0	3.0	3.0
Rough pass_1	3.0	2.5	3.0	3.0
Rough pass_2	-	2.5	-	-
Rough pass_3	-	-	-	-
Unit Cost	1.94	2.39	1.94	1.94

Table 6.4 Results for example-1 (Stock to be removed=8.0 mm)

Stock to be removed=8.0 mm	Proposed Model	Shin and Joo (1992)	Gupta et al. (1995)	Al-Ahmari (2001)
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	2	3	2	2
Finish pass (mm)	3.0	1.0	3.0	3.0
Rough pass_1	2.5	2.33	2.1	2.079
Rough pass_2	2.5	2.33	2.9	2.921
Rough pass_3	-	2.33	-	-
Unit Cost	2.48	2.93	2.48	2.48

Table 6.5 Results for example-1 (Stock to be removed=8.5 mm)

Stock to be removed=8.5 mm	Proposed Model	Shin and Joo (1992)	Gupta et al. (1995)	Al-Ahmari (2001)
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	2	3	2	2
Finish pass (mm)	3.0	1.0	3.0	3.0
Rough pass_1	2.5	2.5	2.5	2.5
Rough pass_2	3.0	2.5	3.0	3.0
Rough pass_3	-	2.5	-	-
Unit Cost	2.55	3.00	2.55	2.55

Table 6.6 Results for example-1 (Stock to be removed=9.0 mm)

Stock to be removed=9.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	2	3	2	2
Finish pass (mm)	3.0	1.0	3.0	3.0
Rough pass_1	3.0	2.67	3.0	3.0
Rough pass_2	3.0	2.67	3.0	3.0
Rough pass_3	-	2.67	-	-
Unit Cost	2.61	3.07	2.62	2.62

Table 6.7 Results for example-1 (Stock to be removed=9.5 mm)

Stock to be removed=9.5 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	3	3	3
Finish pass (mm)	3.0	1.0	2.9	3.0
Rough pass_1	2.2	2.83	1.9	2.342
Rough pass_2	1.9	2.83	2.8	2.079
Rough pass_3	2.4	2.83	1.9	2.079
Unit Cost	2.97	3.13	3.01	2.95

Table 6.8 Results for example-1 (Stock to be removed=10.0 mm)

Stock to be removed=10.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	3	3	3
Finish pass (mm)	3.0	1.0	3.0	3.0
Rough pass_1	2.2	3.0	2.1	2.842
Rough pass_2	2.5	3.0	2.8	2.079
Rough pass_3	2.3	3.0	2.1	2.079
Unit Cost	3.02	3.21	3.02	3.02

The machining cost of a single finish pass corresponding to each depth of cut for Example-2 is obtained by following the steps in Section 6.2.4.1, as in Table 6.9.

Table 6.9 Cost of a single finish pass for different values of depth of cut (Example-2)

Depth of cut (d_{ij})	**Cost of single finish pass (c_{0j})**
0.4	0.720
0.5	0.735
0.6	0.749
0.7	0.760
0.8	0.771
0.9	0.780
1.0	0.788
1.1	0.796
1.2	0.803

Again, the proposed algorithm is applied for different stocks to be removed range between 6 mm and 10 mm. The optimal set of parameters found for the proposed networks are given in Table 6.10.

Table 6.10 Parameters used in the simulation

Parameters \ Stock To be Removed	**6.0 mm**	**8.0 mm**	**8.5 mm**	**9.0 mm**	**9.5 mm**	**10.0 mm**
A	1	1	1	1	1	1
C	120	550	550	550	680	550
D	20	40	50	40	58	55
λ	100	100	100	100	100	100
$\eta_{x_{0j}}$	0.0001	0.0005	0.0001	0.0001	0.0001	0.0001
$\eta_{x_{ij}}$	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005

In Tables 6.11-6.16, the solutions obtained by the gradient network using the determined parameters are compared with other solutions in the literature.

Table 6.11 Results for example-2 (Stock to be removed=6.0 mm)

Stock to be removed=6.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	2	2	2	2
Finish pass (mm)	1.2	0.4	1.2	1.2
Rough pass_1	2.5	2.8	2.7	2.72
Rough pass_2	2.3	2.8	2.1	2.08
Rough pass_3	-	-	-	-
Unit Cost	2.37	2.40	2.37	2.37

Table 6.12 Results for example-2 (Stock to be removed=8.0 mm)

Stock to be removed=8.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	3	3	3
Finish pass (mm)	1.2	0.4	1.0	0.4
Rough pass_1	2.8	2.53	2.0	2.52
Rough pass_2	2.2	2.53	3.0	2.08
Rough pass_3	2.0	2.53	2.0	3.00
Unit Cost	2.92	2.94	2.95	2.94

Table 6.13 Results for example-2 (Stock to be removed=8.5 mm)

Stock to be removed=8.5 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	3	3	3
Finish pass (mm)	1.2	0.4	1.0	1.2
Rough pass_1	2.8	2.7	3.0	2.22
Rough pass_2	2.2	2.7	2.0	2.08
Rough pass_3	2.3	2.7	2.5	3.0
Unit Cost	2.98	3.01	3.01	2.98

Table 6.14 Results for example-2 (Stock to be removed=9.0 mm)

Stock to be removed=9.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	3	3	3
Finish pass (mm)	1.2	0.4	1.0	1.2
Rough pass_1	2.6	2.87	3.0	2.72
Rough pass_2	2.9	2.87	3.0	2.08
Rough pass_3	2.3	2.87	2.0	3.0
Unit Cost	3.05	3.08	3.08	3.05

Table 6.15 Results for example-2 (Stock to be removed=9.5 mm)

Stock to be removed=9.5 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	4	3	3
Finish pass (mm)	1.2	0.4	1.2	1.2
Rough pass_1	2.9	2.275	3.0	3.0
Rough pass_2	2.5	2.275	3.0	2.3
Rough pass_3	2.9	2.275	2.3	3.0
Rough pass_4	-	2.275	-	-
Unit Cost	3.12	3.42	3.12	3.12

Table 6.16 Results for example-2 (Stock to be removed=10.0 mm)

Stock to be removed=10.0 mm	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**
# of passes	3	4	3	3
Finish pass (mm)	1.2	0.4	1.2	1.2
Rough pass_1	2.8	2.4	3.0	2.8
Rough pass_2	3.0	2.4	3.0	3.0
Rough pass_3	3.0	2.4	2.8	3.0
Rough pass_4	-	2.4	-	-
Unit Cost	3.19	3.48	3.19	3.19

The solution quality of the proposed approaches in terms of % deviations is given in Table 6.17 and Table 6.18. In these tables the columns (6), (7) and (8) represent the % deviation of the proposed gradient network solution from dynamic programming, integer programming and non-linear programming solutions, respectively. The % deviations are given by:

$$\% \, Deviation\, from\, Dynamic\, Programming = \frac{Unit\, cost(ANN) - Unit\, cost(Dymamic\, Programming)}{Unit\, cost\, (Dynamic\, Programming)} * 100$$

$$\% \, Deviation\, from\, Integer\, Programming = \frac{Unit\, cost(ANN) - Unit\, cost(Integer\, Programming)}{Unit\, cost\, (Integer\, Programming)} * 100$$

$$\% \, Deviation\, from\, Non-Linear\, Programming = \frac{Unit\, cost(ANN) - Unit\, cost(Non-linear\, Programming)}{Unit\, cost\, (Non-Linear\, Programming)} * 100$$

The last column of the table displays the percentage of simulation runs that resulted in a feasible solution.

Table 6.17 Comparison of results for Example-1

Total Stock To Be Removed	**Proposed Model**	**Shin and Joo (1992)**	**Gupta et al. (1995)**	**Al-Ahmari (2001)**	**Percent Deviation from Dynamic Prog. (6)**	**Percent Deviation from Integer Prog. (7)**	**Percent Deviation from Non-Linear Prog. (8)**	**Percent Feasibility of Computed Solution (9)**
	Artificial Neural Networks	**Dynamic Prog.**	**Integer Prog.**	**Non-Linear Prog.**				
6	1.94	2.39	1.94	1.94	-18.83%	0.00%	0.00%	100.00%
8	2.48	2.93	2.48	2.48	-15.36%	0.00%	0.00%	100.00%
8,5	2.55	3.00	2.55	2.55	-15.00%	0.00%	0.00%	100.00%
9	2.62	3.07	2.62	2.62	-14.66%	0.00%	0.00%	100.00%
9,5	2.97	3.13	3.01	2.95	-5.11%	-1.33%	0.68%	100.00%
10	3.02	3.21	3.02	3.02	-5.92%	0.00%	0.00%	100.00%

Table 6.18 Comparison of results for Example-2

Total Stock To Be Removed	Proposed Model	Shin and Joo (1992)	Gupta et al. (1995)	Al-Ahmari (2001)	Percent Deviation from Dynamic Prog. (6)	Percent Deviation from Integer Prog. (7)	Percent Deviation from Non-Linear Prog. (8)	Percent Feasibility of Computed Solution (9)
	Artificial Neural Networks	Dynamic Prog.	Integer Prog.	Non-Linear Prog.				
6	2.37	2.40	2.37	2.37	-1.25%	0.00%	0.00%	100.00%
8	2.92	2.94	2.95	2.94	-0.68%	-1.02%	-0.68%	100.00%
8,5	2.98	3.01	3.01	2.98	-1.00%	-1.00%	0.00%	100.00%
9	3.05	3.08	3.08	3.05	-0.97%	-0.97%	0.00%	100.00%
9,5	3.12	3.42	3.12	3.12	-8.77%	0.00%	0.00%	100.00%
10	3.19	3.48	3.19	3.19	-8.33%	0.00%	0.00%	100.00%

As our primary goal was to compare the proposed network solution with other solution methods in terms of solution quality, the CPU times required for solving each data set are not given. But from the simulation experiments, it is seen that, the proposed network could converge to valid solutions in reasonable times in approximately 30 seconds. Obviously, by utilizing the parallel computing, significant reduction can be obtained in computational time required to obtain optimal results.

To interpret the findings in Table 6.17 and Table 6.18, total stock to be removed is considered as 6.0 mm in Table 6.17. For all the simulation runs the proposed network resulted in a feasible solution, hence percent feasibility is 100%. The result of dynamic programming, 2.39 is 18.83% more costly than the result of the proposed approach, 1.94. The unit cost provided by the proposed approach is equal to the costs obtained by integer programming and non-linear programming. Thus, percent deviation from integer programming and non-linear programming is 0%.

In all the simulations carried out to show the performance of the network, convergence to valid schedules is achieved and better or at least the same results are obtained for all stock sizes to be removed. If all the test cases are considered, the proposed network is, on average, able to produce a solution with a unit cost, which is 8 % less than the dynamic programming results. Compared to integer programming, the proposed approach provides 1% lower cost. The results indicate that approximately equal optimal results are obtained by using ANN approach and non-linear programming. By tuning the penalty coefficients for each problem, it is possible to improve the convergence and the optimality of the solutions. On the other hand, besides its convergence to valid schedules, convergence to good quality solutions of the proposed network points out its general applicability in other manufacturing processes.

The main advantages of the proposed NN model over dynamic, integer and non-linear programming are listed below.

- It is not restricted with assumptions such as independency.
- It provides optimal results within a reasonable time span. By using the proposed approach, cutting conditions can be determined before starting the machining process.
- It reduces the time span for process development before machining and reducing the time span, gives the opportunity to increase efficiency.
- The proposed algorithm can also be extended for the first stage of the problem.
- It can be used even if the relationship between machining parameters and machining cost cannot be represented analytically.
- It can be used for all machining operations such as turning, milling, drilling etc.
- It can be adopted easily for other objective functions such as maximum production rate and for different total depth of cut values without any modification.

Besides the advantages above, the proposed algorithm utilizes the advantage of neural networks such as:

- The proposed network has the parallel implementation property and can obtain solutions extremely fast by a dedicated hardware.

- It is possible and simple to implement the existing algorithm and structure without any modification even if new inputs parameters are added.

- It is not problem specific; it can be employed with different objective functions and constraints and can be extended for other machining parameter optimization problems.

- It provides a good representation of non-linear relationship between inputs and outputs.
- It does not employ statistics to find analytical relationships between machining parameters and does not require statistical background.

6.5 Conclusion

In this chapter, a dynamical gradient network was presented for solving the process planning optimization problem for metal cutting operations with the unit cost criterion. Focus of this chapter has been on demonstrating the optimization capabilities of the proposed network by solving an example problem available in the literature, considered by Shin & Joo (1992) and Gupta, Batra & Lal (1995). To analyze the performance of the network, it is compared with the solution methods commonly used to solve the problem under study in terms of the solution quality. The simulation experiments demonstrated that the proposed network generated feasible solutions in all the cases, and in some of the cases it found smaller unit cost compared to dynamic programming and integer programming. In general, for all the instances, the average deviation percentage of the proposed network is 8% and 1% above the dynamic programming and integer programming, respectively. Thus, it can be concluded that the proposed approach provided an analytical alternative to integer programming and dynamic programming which are often limited by strict assumptions of linearity, variable independence etc. By conducting several simulation experiments, the influence of different initializations schemes was investigated on the solutions of the problem considered. The analysis results showed that the percent error of the network is very sensitive to the selection of the starting points and the choice of the parameters used in simulation.

The proposed approach can give the optimal solution in an extremely large solution space within a reasonable computation time. It is completely generalized and problem independent so that it can be easily modified to optimize machining economics problem under various economic criteria and practical constraints. From the results obtained, it is seen that besides the convergence to feasible and valid solutions, convergence of the proposed network to good quality solutions indicates its general applicability in also other machining parameter optimization problems such as milling and drilling operations.

The contribution of this study is twofold. We propose to use a novel penalty method that guarantees feasible and near optimal solutions for solving the Process planning optimization problem with the unit cost criterion. Although a large body of literature exists for solving these problems with the unit cost minimization criterion, to the best of our knowledge, there is no previously published article that tried to optimize metal cutting process by using neural networks. Therefore, this study will also make a contribution to the process optimization literature.

Several issues are worthy of future investigations. First, further studies will be focused on selecting the parameters of the network automatically rather than choosing by trial and error, which is one of the drawbacks of neural networks. Second, extension of the results to large size problems will be worthwhile. Finally, extension of the results to different manufacturing processes is important for industrial applications, and implementation of the network in hardware can make progress in computational efficiency.

CHAPTER SEVEN
CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this concluding chapter, we summarize what has been accomplished in this thesis, and describe some potential future work to extend the present results for the discussed problems.

7.1 Conclusions

The fundamental activity in the optimization of a manufacturing process is deciding the values of the process parameters. Selection of the optimal process parameters leads to improvement in the process and part quality. In the last decades, different solution methods such as operational research, design of experiments, simulation and artificial intelligence have been proposed for modeling and solution of process optimization problems. Due to the enormous complexity of many processes and the high number of influencing parameters, artificial neural networks are efficient techniques to solve manufacturing process optimization problems.

In the thesis, first, an overview of manufacturing processes together with the methods used in planning of manufacturing processes was given. A review of artificial neural network and genetic algorithm applications in optimization of manufacturing processes was provided in the following chapters. Then, we developed neural networks to deal with two problems of optimization of manufacturing processes. Our objective was two fold; to test the performance of ANNs by solving these problems and to compare performance of ANNs with that of other techniques'. Although neural network approach has been admitted as a promising alternative to solving a variety of combinatorial optimization problems, few works relate neural network to applications of optimization in manufacturing processes. To illustrate the use of artificial neural networks for optimization of manufacturing processes, two processes are selected: Tube hydroforming process and metal cutting process. From the literature reviewed, we can conclude that artificial neural networks are commonly used for capturing the complex relationship between the input and output variables of the considered manufacturing process. To the best of our knowledge, this study will be the first attempt to solve the considered industrial optimization problems using neural networks.

For the solution of the first problem, a tube hydroforming problem with optimization of two conflicting objectives: minimization of thinning ratio and maximization of bulge ratio, we employed a two-stage backpropagation network. The reverse direction concept of Hsieh & Tong (2001) is explored to hold communication between these networks. The first network is built for parameter searching while the second network is used for response estimating. Since the application of backpropagation networks in the literature indicates that they may be insufficient to obtain optimal results, a hybrid approach combining ANN with GA is proposed. For the considered problem, the relationship between forming parameters and process responses is unknown. Then, the proposed approach is divided into two stages. In the first stage a metamodel is built to capture the relationship between parameters and responses. We employed artificial neural networks and response surface analysis to build the metamodel. Then, by using the proposed GA procedure and applying genetic operators optimal forming parameters are obtained. The results of the simulation experiments indicated that applying the proposed methods generated good results comparable with that of Taguchi approach, commonly used for optimization of tube hydroforming process. The proposed two-stage ANN approach provided 17%, and the proposed integrated GA-NN approach provided 70% improvement in the manufacturing process under consideration. Then, it can be concluded that, hybridization of ANNs with GA improves the efficiency of artificial neural networks in solving manufacturing process optimization problems.

The second problem was the optimization of process parameters of a metal cutting process with unit cost minimization. Earlier studies on metal cutting were limited to single-pass operations. However, one pass is rarely preferred in practice and multi-pass operations are used. In the relevant literature some of the researches assumed that each pass has equal depth of cut that is not a practical way to apply during machining operations. The motivation behind this study was to find optimal subdivision of depth of cut for each pass in order to minimize unit cost of the process that includes machine idle cost, tool replacement cost and tool cost in addition to actual machining cost. For the solution of this problem, we proposed a Hopfield-type dynamical gradient network. The mixed integer formulation of the problem proposed by Gupta, Batra & Lal (1995) was used for constructing the energy function. Then, an interconnected neural network was developed to solve the problem. The proposed network is composed of one maximum network and one log-sigmoid network interacting with each other.

The motivation for using maximum networks was to reduce the network complexity by incorporating competitive mechanism into the network and to obtain a simplified energy function. Additionally, applying a hard-limit transfer function to the outputs of the log-sigmoid network helped us to get rid of some of the binary constraints. After the appropriate energy function was constructed by using a penalty function approach, convergence of the proposed network was analyzed and the dynamics were defined by steepest gradient descent on the energy function. The proposed approach was tested on a metal cutting optimization problem. An optimal solution which may be promising for the applications of large size problems was obtained.

In general, we can say that the results obtained using the proposed neural network models were acceptable in terms of solution quality. However, with the implementation of parallel processing, full benefits of the neural network approach can be explored and assessed. The main benefit one can expect from using the neural networks in performing task optimization is the additional efficiency gained from implementation of parallel neural processing. Parallel processing and parallel computation has been well accepted as a legitimate and effective way for speed improvement in solving many combinatorial optimization problems. However, a challenge with the parallel approaches is that many tasks cannot be easily or possibly broken down into a parallel structure so that the parallel processing can be performed. Because of the neural network's inherent parallel nature of processing units and network structure, once a problem is formulated into a neural network model, it will be in a ready mode to realize parallel processing. In other words, the neural network can be viewed as a natural vehicle to convert a problem into a parallel format. For full exploration of the neural network's potential in optimization, we need to firstly formulate a problem into a neural network model and then implement the neural network algorithms on a multiple processor machine or on a parallel-computing platform. Since the neural network computation in our experiments works also in serial mode, the experimental results reported do not reflect the potential of the true benefit of the neural network approach. With fast advance of high technology, parallel processing facilities will become inevitably more popular and easy to access. To this extent, we can expect a great improvement in computation time using the neural network approach.

7.2 Future Research

The followings which are possible extensions of this study are suggested for future research.

- One of the major shortcomings of the proposed artificial neural network approaches is the determination of the parameters required for the simulation of the proposed approaches by trial and error, such as the penalty parameters, the slope of the activation functions, the number of iterations and the step sizes of neural networks. This is a tedious process, and the parameter values obtained might not be the optimal values for this study. The methodology for obtaining appropriate parameters for the development of proposed NN models that will yield more precise results should be considered in a future study.

- One of the other issues for future research may be to introduce evolution to adjust the topology and the parameters of ANNs automatically or to search for the ways of developing automatic parameter controlling methods to overcome the need of tuning the parameters by a trial and error.

- As one of the drawbacks of backpropagation neural networks, they are not able to be involved directly in an optimization problem. Then, this type of stand-alone artificial neural networks cannot be remarked as an optimization tool. To overcome this drawback, many researchers have already applied backpropagation networks in conjunction with other optimization techniques to solve the optimization of manufacturing processes. It is also required to combine ANNs with other optimization techniques for the optimization of process parameters and in this thesis, we combined ANNs with GA for the optimization of tube hydroforming process. For a future research, a number of different techniques and metaheuristics rather than genetic algorithms can be combined with neural networks to tackle the problem considered.

- The performances of the proposed neural network models depend on the choice of the initial states. Another area on which future research has to focus may be to propose new models that are less sensitive to the initial states.

- Other objectives for manufacturing processes, especially for the second application, production rate or profit maximization or idle time minimization can be studied.

- For the proposed Hopfield- type dynamical network, dynamical gradient networks suffer from serious problems of getting stuck at local minima, having high sensitivity to parametric changes and tradeoff problem among these parameters. To overcome the local minima, stochastic methods such as simulated annealing can be integrated with this network. By introducing a probability for the acceptance of a new state, the network occasionally accepts transitions to states with higher energy and thus can escape from local minima. Replacing sigmoidal activation function with a stochastic decision type activation function, adding noise to the weights of the network or to the biases of the network are some of the main methods used to embed stochasticity into the Hopfield network (Smith, 1999).

REFERENCES

Adebrabbo, N., Worswick, M., Mayer, R. & Riemsdijk, I.V. (2008). Optimization methods for the tube hydroforming process applied to advanced high-strength steels with experimental verification. *Journal of Materials Processing Technology, 209* (1), 110-123.

Agapious, J.S. (1992a). An optimization of multi-stage machining system, part I: Mathematical solution part II: The algorithm and application. *Transaction ASME, Journal of Engineering for Industry, 114,* 524-538.

Agapious, J.S. (1992b). The optimization of machining operations based on a combined criterion, Part-I: The use of combined objectives in single-pass operations. *Transactions of ASME, Journal of Engineering for Industry, 114,* 500-507.

Agapious, J.S. (1992c). The optimization of machining operations based on a combined criterion, Part-II: Multi-pass operations. *Transactions ASME, Journal of Engineering for Industry, 114,* 508-513.

Aiyer, S.V.B., Niranjan, M. & Fallside, F. (1990). A theoretical investigation into the performance of the Hopfield model. *IEEE Transactions on Neural Networks, 1*, 204–215.

Akyol, D.E. (2006). *Neural networks based optimization in production scheduling*. PhD Thesis, Izmir: Dokuz Eylül University.

Al-Ahmari, A.M.A. (2001). Mathematical model for determining machining parameters in multipass turning operations with constraints. *International Journal of Production Research, 39* (15), 3367-3376.

Al-Aomar, R. & Al-Okaily, A. (2006). A GA-based parameter design for single machine turning process with high-volume production. *Computers & Industrial Engineering, 50,* 317-337.

Amiolemhen, P.E. & Ibhadode, A.O.A. (2004). Application of genetic algorithms-determination of the optimal machining parameters in the conversion of a cylindrical bar stock into a continuous finished profile. *International Journal of Machine Tools & Manufacture, 44,* 1403-1412.

An, L. & Chen, M. (2003). On optimization of machining parameters. *4th International Conference on Control and Automation.*

Anderson, K., Cook, G.E., Karsai, G. & Ramaswamy, K. (1990). Artificial neural Networks applied to arc welding process modeling and control. *IEEE Transactions on Industry Applications, 26,* 824-830.

Anderson, K., Cook, G.E., Springfield, J.F. & Barnett, R.J. (1991). Applications of artificial neural networks for arc welding. In *Intelligent Engineering Systems Through Artificial Neural Networks* (717-728). New York: ASME Press.

Anjum, M.F., Tasadduq, I. & Khaled, A.S. (1997). Response surface methodology: A neural network approach. *European Journal of Operational Research, 101,* 65-73.

Antony, J. (2001). Simultaneous optimisation of multiple quality characteristics in manufacturing processes using Taguchi's quality loss function. *International Journal of Advanced Manufacturing Technology, 17,* 134-138.

Antony, J., Anand, R.B., Kumar, M & Tiwari, M.K. (2006). Multiple response optimization using Taguchi methodology and neuro-fuzzy based model. *Journal of Manufacturing Technology Management, 17* (7), *908-925.*

Antony, J., Knowles, G. & Taner, T. (2001). 10 Steps to optimal production. *Quality,* 45-49.

Aue-U-Lan, Y., Ngaile, G. & Altan, T. (2004). Optimizing tube hydroforming using process simulation and experimental verification. *Journal of Materials Processing Technology, 146,* 137-143.

Aydemir, A., Vree, J.H.P. de, Brekelmans, W.A.M., Geers, M.G.D., Sillekens, W.H. & Werkhoven, R.J. (2005). An adaptive simulation approach designed for tube hydroforming process. *Journal of Materials Processing Technology, 159,* 303-310.

Baskar, N., Asokan, P., Saravanan, R. & Prabhaharan, G. (2005). Optimization of machining parameters for milling operations using non-conventional methods. *International Journal of Advanced Manufacturing Technology, 25,* 1078-1088.

Baskar, N., Asokan, P., Saravanan, R. & Prabhaharan, G. (2006). Selection of optimal machining parameters for multi-tool milling operations using a memetic algorithm. *Journal of Materials Processing Technology, 174,* 239-249.

Baykasoglu, A. & Dereli, T. (2002). Novel algorithmic approach to generate the “number of passes” and “depth of cuts” for the optimization routines of multipass machining. *International Journal of Production Research, 40* (7), 1549-1565.

Belhadj, T., Abbassi, F., Mistou, S. & Zghal, A. (2010). Numerical analyses of tube hydroforming problem using artificial neural networks. *International Journal of Material Forming, 3* (1), 295-298.

Bernardos, P.G. & Vosniakos, G.C. (2002). Prediction of surface roughness in CNC face milling using neural networks and Taguchi’s design of experiments. *Robotics and Computer Integrated Manufacturing, 18,* 343-354.

Bhaskara, R.S.V., Shunmugam, M.S. & Narendran, T.T. (1998). Optimal sub-division of the depth of cut to achieve minimum production cost in multi pass turning using a genetic algorithm. *Journal of Materials Processing Technology, 79,* 101-108.

Bisht, H., Gupta, J., Pal, S.K. & Chakraborty, D. (2005). Artificial neural network based prediction of flank wear in turning. *International Journal of Materials and Product Technology, 22* (4), 328-338.

Bralla, J. (1998). *Design for Manufacturability Handbook* (2nd ed.). NY: McGraw Hill.

Brandt, R.D., Wang, Y., Laub, A.J., & Mitra, S.K. (1988). Alternative Networks for Solving the Travelling Salesman Problem and the List-Matching Problem. *Proceedings of the International Conference on Neural Networks*, *2*, 333-340.

Castillo, E. D. (2007). *Process optimization: A statistical approach.* New York: Springer.

Chang, H.-H. (2006). Dynamic multi-response experiments by backpropagation networks and desirability functions. *Journal of the Chinese Institute of Industrial Engineers,23* (4), 280-288.

Chang, C.A. & Su, C.T. (1995). A comparison of statistical regression and neural-network methods in modeling measurement errors for computer vision inspection systems. *Computers and Industrial Engineering, 28* (3), 593-603.

Chen, J.H., Lee, Y.S. & Fang, S.C. (1998). Optimal cutter selection and machining plane determination for process planning and NC machining of complex surface. *Journal of Manufacturing Systems, 17* (5), 371-388.

Chen, M.C. & Chen, K.Y. (2003). Optimization of multipass turning operations with genetic algorithms: a note. *International Journal of Production Research, 41* (4), 3385-3388.

Chen, M.C. & Tsai, D.M. (1996). A simulated annealing approach for optimization of multi-pass turning operations. *International Journal of Production Research, 34,* 2803-2825.

Chen, R.S., Lee, H.H. & Yu, C.Y. (1998). Application of Taguchi's method on the optimal process design of an injection molded PC/PBT automobile bumper. *Composite Structure, 39,* 209-214.

Chiu, C.C., Su, C.T., Yang, G.H., Huang, J.S., Chen, S.C. & Cheng, N.T. (1997). Selection of optimal parameters in gas-assisted injection moulding using a neural network model and the Taguchi method. *International Journal of Quality Science, 2,* 106-120.

Chryssolouris, G. (2006). *Manufacturing systems: Theory and practice* (2nd ed.). USA:Springer.

Chowdhury, K., Pratihar, D.K. & Pal, D.K. (2002). Multi-objective optimization in turning-using GA algorithm. *Journal of the Institute of Engineers (India), 82,* 37-44.

Coit, D.W., Jackson, B.T. & Smith, A.E. (1998). Static neural network process models: considerations and case studies. *International Journal of Production Research, 36* (11), 2953-2967.

Cook, D.F., Ragsdale, C.T. & Major, R.L. (2000). Combining a neural network with a genetic algorithm for process parameter optimization. *Engineering Applications of Artificial Intelligence, 13,* 391-396.

Correia, D.S., Gonçalves, C.V., Cunha, S.S. & Ferraresi, V.A. (2005). Comparison between genetic algorithms and response surface methodology in GMAW welding optimization. *Journal of Materials Processing Technology, 160,* 70-76.

Cus, F. & Balic, J. (2000). Selection of cutting conditions and tool flow in flexible manufacturing system. *International Journal of Manufacturing Science and Technology, 2,* 101-106.

Cus, F. & Balic, J. (2003). Optimization of cutting process by GA approach. *Robotics and Computer Integrated manufacturing, 19,* 113-121.

Cus, F. & Zuperl, U. (2006). Approach to optimization of cutting conditions by using artificial neural networks. *Journal of Materials Processing Technology, 173,* 281-290.

Cus, F., Zuperl, U. & Milfelner, M. (2006). Dynamic neural network approach for tool cutting force modeling of end milling operations. *International Journal of General Systems, 35* (5), 603-618.

Daskin, M., Jones, P.C. & Lowe, T.J. (1990). Rationalizing tool selection in a flexible manufacturing system for sheet-metal products. *Operations Research, 38* (6), 1104-1115.

Dereli, T., Filiz, I.H. & Baykasoglu, A. (2001). Optimizing cutting parameters in process planning of prismatic parts by using genetic algorithms. *International Journal of Production Research, 9,* 3303-3328.

Dogan, H. & Guzelis, C. (2006). Robust and fuzzy spherical clustering by a penalty parameter approach. *IEEE Transactions on Circuit and Systems-II, 53* (8), 637-641.

El Baradie, M.A. (1993). Surface roughness model for turning grey cast iron (154BHN). *Proceeding of Institution of Mechanical Engineering Part B, Journal of Engineering Manufacture, 207,* 43-54.

El-Axir, M.H. (2002). A method of modeling residual stress distribution in turning for different materials. *International Journal of Machine Tools & Manufacture, 42,* 1055-1063.

Ermer, D.S. (1971). Optimisation of the constrained machining economics problem by geometric programming. *Transactions of ASME, Journal of Engineering for Industry, 93, 1067-1072.*

Fann, K.-J. & Hsiao, P.-Y. (2003). Optimization of loading conditions for tube hydroforming. *Journal of Materials Processing Technology, 140,* 520-524.

Fausett, L. (1994). *Fundamentals of neural networks: An architectures, algorithms and applications.* Prentice Hall.

Feng, C., Wang, X. & Yu, Z. (2002). Neural networks modeling of honing surface roughness parameter defined by ISO 13565. *SIAM Journal of Manufacturing Systems, 21* (8), 1-35.

Filipic, B. & Laitinen, E. (2005). Model-based tuning of process parameters for steady-state steel casting. *Informatica, 29,* 491-496.

Fox, R.T. & Lee, D. (1990). Optimization of metal injection moulding: Experimental design. *The International Journal of Powder Metallurgy, 26* (3), 233-243.

Foo, Y.P.S., & Takefuji, Y. (1988a). Stochastic neural networks for solving job-shop scheduling: Part 1, problem presentation. *Proceedings of Joint International Conference on Neural Networks*, *2*, 275-282.

Foo, Y.P.S., & Takefuji, Y. (1988b). Stochastic neural networks for solving job-shop scheduling: Part 2, architecture and simulations. *Proceedings of Joint International Conference on Neural Networks*, *2*, 283–290.

Foo, Y.P.S., & Takefuji,Y. (1988c). Integer linear programming neural networks for job-shop scheduling. *Proceedings of Joint International Conference on Neural Networks*, *2*, 341–348.

Fourment, L. & Chenot, J.L. (1996). Optimal design for non-steady state metal forming processes-I. Shape optimization method. *International Journal for Numerical Methods in Engineering, 39,* 33-50.

Fuh, K. & Chang, H. (1997). An accuracy model for the peripheral milling of aluminium alloys using response surface design. *Journal of Materials Processing Technology, 72,* 42-47.

Fukushima, K. (1975). Cognitrion: A self-organizing multilayered neural network. *Biological Cybernetics, 20*, 121-136.

Ghouati, O. & Gelin, J.C. (1999). Sensitivity analysis in forming processes. *International Journal of Forming Processes, 1,* 297-322.

Goldberg, D.E. (1989). *Genetic algorithms in search, optimization and machine learning*. MA: Addision-Wesley.

GopalaKrisnan, B. & Al-Khayyal, F. (1991). “Machine parameter selection for turning with constraints”, An analytical approach based on geometric programming, *International Journal of Production Research,* 1897-1908.

Gotoh, M. & Ishise, F. (1978). A finite element analysis of rigid-plastic deformation of the flange in a deep-drawing process based on a fourth-degree yield function. *International Journal of Mechanical Science, 20,* 423-435.

Gövekar, E., Grabec, I. & Peklenik, J. (1989). Monitoring of a drilling process by a neural network. *The 21st CIRP International Seminar on Manufacturing Systems, Stockholm, Sweden, June 5-6.*

Grossberg, S. (1972). Neural expectation: Cerebellar and retinal analogs of cells fired by learnable or unlearned pattern classes. *Kybernetik, 10*, 49-57.

Grossberg, S. (1976a). Adaptive pattern classification and universal recording: I. Parallel development and coding of neural detectors. *Biological Cybernetics, 23*, 121-134.

Grossberg, S. (1976b). Adaptive pattern classification and universal recording: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics, 23*, 187-202.

Grzesik, W. & Brol, S. (2003). Hybrid approach to surface roughness evaluation in multistage machining processes. *Journal of Materials Processing Technology, 134,* 265-272.

Guillot, M. & El Quafi, A. (1991). On-line identification of tool breakage in metal cutting processes by use of neural networks. In Dagli, C.H., Kumara, S.R.T., Shin, Y.C. (Eds.). *Intelligent Engineering Systems Through Artificial Neural Networks* (701-710). New York: ASME Press.

Gupta, R., Batra, J.L. & Lal, G.K. (1995). Determination of optimal subdivision of depth of cut in multi-pass turning with constraints. *International Journal of Production Research, 33,* 2555-2565.

Hayers, G.M. & Davis, R.P. (1979). A discrete variable approach to machine parameter optimization. *AIIE Transactions, 11* (2), 155-159.

Hedge, S., Sweet, J., & Levy, W. (1988). Determination of parameters in a Hopfield/Tank computational network. *Proceedings of the IEEE International Conference on Neural Networks*, *2*, 291-298.

Heider, D., Piovoso, M.J. & Gillespie, J.W.Jr (2002). Application of a neural network to improve an automated thermoplastic tow-placement process. *Journal of Process Control, 12,* 101-111.

Hewidy, M.S., El-Taweel, T.A. & El-Safty, M.F. (2005). Modeling the machining parameters of wire electrical discharge machining of Inconel 601 using RSM. *Journal of Materials Processing Technology, 169,* 328-336.

Hewidy, M.S. & Fattouh, M. (1989). Electrochemical cutting using tubular cathodes: response surface approach. *International Journal of Production Research, 27,* 953-963.

Holland, J.H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence.* MA: MIT Press.

Holland, J.H. (1992). Genetic algorithms. *Scientific American, 267* (1), 66-72.

Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *National Academy of Science, 79,* 2554-2558.

Hopfield, J. (1984). Neurons with graded response have collective computational properties like of two-state neurons. *Proceedings of the National Academy of Sciences of the USA*, *81*, 3088-3092.

Hopfield, J.J. & Tank, D.W. (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics, 52,* 141-152.

Hou, T.-H. & Lin, L. (1993). Manufacturing process monitoring using neural networks. *Computers and Electrical Engineering, 19,* 129-141.

Hou, T.-H., Su, C.-H. & Liu, W.-L. (2007). Parameters optimization of a nano-particle wet milling process using the Taguchi method, response surface method and genetic algorithm. *Powder Technology, 173,* 153-162.

Hsieh, K.-L. (2006). Parameter optimization of a multi-response process for lead frame manufacturing by employing artificial neural networks. *International Journal of Advanced Manufacturing Technology, 28,* 584-591.

Hsieh, K.-L. & Tong, L.-I. (2001). Optimization of multiple quality responses involving qualitative and quantitative characteristics in IC manufacturing using neural networks. *Computers in Industry, 46,* 1-12.

Hsu, C.M. (2004). An integrated approach to enhance the optical performance of couplers based on neural networks, desirability functions and tabu search. *International Journal of Production Economics, 92,* 241-254.

Hsu, C.M., Su, C.T. & Liao, D. (2004). Simultaneous of the broadband tap coupler optical performance based on neural works and exponential desirability functions. *International Journal of Advanced Manufacturing Technology, 23,* 896-902.

Hsu, H.M., Tsai, S.P., Wu, M.C. & Tzuang, C.K. (1999). A genetic algorithm for the optimal design of microwave filters. *International Journal of Industrial Engineering, 6* (4), 282-288.

Huang, C.-C. & Tang, T.-T. (2006). Parameter optimization in melt spinning by neural networks and generic algorithms. *International Journal of Advanced Manufacturing Technology, 27,* 1113-1118.

Huang, S.H. & Zhang, H.-C. (1994). Artificial neural networks in manufacturing: Concepts, applications and perspectives. *IEEE Transactions on Components, Packaging and Manufacturing Technology-Part A, 17* (2), 212-228.

Huh, H. & Kim, S.-H. (2001). Optimum process design in sheet-metal forminh with finite element analysis. *Transactions of the ASME, 123,* 476-481.

Hyun, B.S. & Cho H.S. (1994). Prediction of forming pressure curve for hydroforming process using artificial neural network. *Proceedings of the Institution of Mechanical Engineers Part I, Journal of Systems and Control Engineering, 208,* 109-121.

Imaninejad, M., Subhash, G. & Loukus, A. (2005). Loading path optimization of tube hydroforming process. *International Journal of Machine Tools & Manufacture, 45,* 1504, 1514.

Iwata, K., Muratsu, Y., Iwatsubo, T. & Fujii, S. (1972). A probabilistic approach to the determination of the optimum cutting conditions. *Journal of Engineering for Industry, 94,* 1099-1107.

Iwata, K., Muratsu, Y. & Oba, F. (1977). Optimisation of cutting conditions for multipass operations considering probabilistic nature in machining processes. *Journal of Engineering for Industry, Transactions of the ASME, February,* 210-217.

Jain, R.K., Jain, V.K. & Kalra, P.K. (1999). Modelling of abrasive flow machining process: A neural network approach. *Wear, 231,* 242-248.

Jansson, M., Andersson, A. & Nilsson, L. (2005). Optimization of draw-in for an automotive sheet metal part: an evaluation using surrogate models and response surfaces. *Journal of Materials Processing Technology, 159,* 426-434.

Jansson, M., Nilsson, L. & Simonsson, K. (2007). On process parameter estimation for the tube hydroforming process. *Journal of Materials Processing Technology, 190,* 1-11.

Jawahir, I.S. & Wang, X. (2007). Development of hybrid predictive models and optimization techniques for machining operations. *Journal of Materials Processing Technology, 185,* 46-59.

Jeang, A., Li, H.-C. & Wang, Y.-C. (2010). A computational simulation approach for optimizing process parameters in cutting operations. *International Journal of Computer Integrated Manufacturing, 23* (4), 325-340.

Jeyapaul, R., Shahabudeen, P. & Krishnaiah, K. (2006). Simultaneous optimization of multi-response problems in the Taguchi method using genetic algorithm. *International Journal of Advanced Manufacturing Technology, 30* (9), 870-878.

Jimenez-Marquez, S.A., Lacroix, C. & Thibault, J. (2003). Impact of modelling parameters on the prediction of cheese moisture using neural networks. *Computers and Chemical Engineering, 27,* 631-646.

Johnson, K., Nguyen, B., Davies, R., Grant, G. & Khaleel, M. (2004). A numerical process control method for circular tube hydroforming prediction. *International Journal of Plasticity, 20,* 1111-1137.

Juan, H., Yu, S.F. & Lee, B.Y. (2003). The optimal cutting parameter selection of production cost in HSM for SKD61 tool steels. *International Journal of Machine Tools & Manufacture, 43,* 679-686.

Kamgar-Parsi, B., & Kamgar-Parsi, B. (1992). Dynamical Stability and Parameter Selection in Neural Optimization. *Proceedings of International Joint Conference on Neural Networks*, *4*, 566-571.

Kang, S.-J., Kim, H.-K. & Kang, B.-S. (2005). Tube size effect on hydroforming formability. *Journal of Materials Processing Technology, 160,* 24-33.

Karkoub, M.A. (2006). Prediction of hydroforming characteristics using random neural network neuronale. *Journal of Intelligent Manufacturing, 17,* 321-330.

Karpat, Y. & Ozel, T. (2006). Swarm-intelligent neural network system (SINNS) based multi-objective optimization of hard-turning. *Transactions of NAMRI/SME, 34, 1-8.*

Karunakar, D.B. & Datta, G.L. (2007). Controlling green sand mould properties using artificial neural networks and genetic algorithms – A comparison. *Applied Clay Science, 37,* 58-66.

Knapp, G.M. & Wang, H.-P. (1992). Neural networks in acquisition of manufacturing knowledge. In *Intelligent Design and Manufacturing* (723-744). New York: John Wiley & Sons.

Knapp, G.M. & Wang, H.-P. (1992). Acquiring, storing and utilizing process planning knowledge using neural networks. *Journal of Intelligent Manufacturing, 3* (5), 333-344.

Knight, D. *The operation and applications of genetic algorithms,* from www.ivoryresearch.com.

Ko, D.C., Kim, D.H. & Kim, B.M. (1999). Application of artificial neural network and Taguchi method to preform design in metal forming considering workability. *International Journal of Machine Tools & Manufacture, 39,* 771-785.

Ko, D.C., Kim, D.H., Kim, B.M. & Choi, J.C. (1998). Methodology of perform design considering workability in metal forming by the artificial neural network and Taguchi method. *Journal of Materials Processing Technology, 80-81,* 487-492.

Koc, M. & Altan, T. (2002). Prediction of forming limits and parameters in the tube hydroforming process. *International Journal of Machine Tools & Manufacture, 42,* 123-138.

Kovavic, M., Uratnik, P., Brezocnik, M. & Turk, R. (2007). Prediction of the bending capability of rolled metal sheet by genetic programming. *Materials and Manufacturing Processes, 22* (6), 634-640.

Krimpenis, A. & Vosniakos, G.C. (2002). Optimization of multiple tool NC rough machining of a hemisphere as a genetic algorithm paradigm application. *International Journal of Advanced Manufacturing Technology, 20,* 727-734.

Kurtaran, H. & Erzurumlu, T. (2006). Efficient warpage optimization of thin shell plastic using response surface methodology and genetic algorithm. *International Journal of Advanced Manufacturing Technology, 27,* 468-472.

Kusiak, A. (1985). Integer programming approach to process planning. *International Journal of Advanced Manufacturing Technology,* 73-83.

Lakshminarayanan, A.K. & Balasubramanian, V. (2008). Process parameters optimization for friction stir welding of RDE-40 aluminium alloy using Taguchi technique. *Transaction of Nonferrous Metal Society of China, 18,* 548-554.

Lee, T.S. & Lin, Y.J. (2000). A 3D predictive cutting-force model for end milling of parts having sculptured surfaces. *International Journal of Advanced Manufacturing Technology, 16,* 773-783.

Lee, Y.H., Shin, H.M. & Yang, B.H. (1996). An approach for multi-criteria simulation optimization with application to turning operation. *Computers and Industrial Engineering, 30* (3), 375-386.

Li, H.-X. & Da, X.L. (2000). A neural network representation of linear programming. *European Journal of Operational Research, 124,* 224-234.

Li, B., Nye, T.J. & Metzger, D.R. (2006). Multi-objective optimization of forming parameters for tube hydroforming process based on the Taguchi method. *International Journal of Advanced Manufacturing Technology, 28,* 23-30.

Li, T.S., Su, C.T. & Chiang, T.L. (2003). Applying robust multi-response quality engineering for parameter selection using a novel neural-genetic algorithm, *Computers in Industry, 50,* 113-122.

Liang, M., Mgwatu, M. & Zuo, M. (2001). Integration of cutting parameter selection and tool adjustment decision for multi-pass turning. *International Journal of Advanced Manufacturing Technology, 17,* 861-869.

Liao, H.-C. (2005). Using N-D method to solve multi-response problem in Taguchi. *Journal of Intelligent Manufacturing, 16,* 331-347.

Lippmann, R.P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine,* 4-22.

Lin, T.R. (2002). Optimization techniques for face milling stainless steel with multiple performance characteristics. *International Journal of Advanced Manufacturing Technology, 19,* 330-335.

Liu, Q. & Altintas, Y. (1999). On-line monitoring of flank wear in turning with multi-layered feed-forward neural network. *International Journal of Machine Tools & Manufacture, 39,* 1945-1959.

Liu, Y. & Wang, C. (1999). Neural network based adaptive control and optimization in the milling process. *International Journal of Advanced Manufacturing Technology, 15,* 791-795.

Lu, H.-S., Chen, J.-Y. & Chung, Ch.-T. (2008). Theoptimal cutting parameter design of rough cutting process in side milling. *Journal of Achievements in Materials and Manufacturing Engineering, 29* (2), 183-186.

Mahanty, B., Agrawal, R.K., Shrin,S. & Chakravarty, S. (2007). Hybrid approach to optimal packing using genetic algorithm and coulomb potential algorithm. *Materials and Manufacturing Processes, 22* (6), 668-677.

Manabe, K.-I. & Amino, M. (2002). Effects of process parameters and material properties on deformation process in tube hydroforming. *Journal of Materials Processing Technology, 123,* 285-291.

Manabe, K.-I., Suetake, M., Koyama, H. & Yang, M. (2006). Hydroforming process optimization of aluminum alloy tube using intelligent control technique. *International Journal of Machine Tools & Manufacture, 46,* 1207-1211.

McCulloch, W.S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5, 115-133.

Miettinen, K., Makela, M.M. & Mannikko, T. (1998). Optimal control of continuous casting by nondifferentiable multiobjective optimization. *Computational Optimization and Applications, 11,* 177-194.

Min, H.S., Yih, Y., & Kim, C.O. (1998). A competitive neural network approach to multi-objective FMS scheduling. *International Journal of Production Research, 36* (7), 1749-1765.

Monostori, L. & Nacsa, J. (1990). On the application of neural nets in real time monitoring of machining processes. *The 22nd CIRP International Seminar on Manufacturing Systems, Enschede, Netherlands, June 11-12*.

Montgomery, D.C. (2005). *Design and analysis of experiments* (6th ed.). New York: John Wiley.

Mukherjee, I. & Ray, P.K. (2006). A review of optimization techniques in metal cutting processes. *Computers & Industrial Engineering, 50,* 15-34.

Naceur, H., Guo, Y.Q., Batoz, J.L. & Knopf-Lenoir, C. (2001). Optimization of draw bead restraining forces and drawbead design in sheet metal forming process, *International Journal of Mechanical Sciences, 43,* 2407-2434.

Nacsa, J. & Monostori, L. (1990). Real-time monitoring of machining processes. *Proceedings of International Conference on automatic Supervision, Monitoring and Adaptive Control in Manufacturing,* 197-233.

Nakhjavani, O.B. & Ghoreishi, M. (2006). Multi criteria optimization of laser percussion drilling process using artificial neural network model combined with genetic algorithm. *Materials and Manufacturing Processes, 21,* 11-18.

Ohdar, R.K. & Pasha, S. (2003). Prediction of the process parameters of metal powder preform forging using artificial neural network (ANN). *Journal of Materials Processing Technology, 132*, 227-234.

Oktem, H., Erzurumlu, T. & Col, M. (2006). A study of the Taguchi optimization method for surface roughness in finish milling of mold surfaces. *International Journal of Advanced Manufacturing Technology, 28,* 694-700.

Onwubolu, G.C. (2006). Performance-based optimization of multi-pass face milling operations using tribes. *International Journal of Machine Tools & Manufacture, 46,* 717-727.

Onwubolu, G.C. & Kumalo, T. (2002). Multi-pass turning optimisation based on genetic algorithms. *International Journal of Production Research, 39* (16), 3727-3745.

Osman, I.H. (2002). Preface, Focused issue on applied meta-heuristics. *Computers and Industrial Engineering*, 205-207.

Ozcelik, B. & Erzurumlu, T. (2005). Determination of effecting dimensional parameters on warpage of thin shell plastic parts using integrated response surface method and genetic algorithm. *International Communications in Heat and Mass Transfer, 32,* 1085-1094.

Ozcelik, B. & Erzurumlu, T. (2006). Comparison of the warpage optimization in the plastic injection molding, using ANOVA, neural network model and genetic algorithm. *Journal of Materials Processing Technology, 171*, 437-445

Ozel, T. & Karpat, Y. (2007). Identification of constitutive material model parameters for high-strain rate metal cutting conditions using evolutionary computational algorithms. *Materials and Manufacturing Processes, 22* (6), 659-667.

Palanisamy, P., Rajendran, I. & Shanmugasundaram, S. (2007). Optimization of machining parameters using genetic algorithm and experimental validation for end-milling operations. *International Journal of Advanced Manufacturing Technology, 32,* 644-655.

Parker, D.B. (1985). *Learning logic: Casting the cortex of the human brain in silicon. Technical Report,* TR-47. Center for Computational Research in Economics and Management Science, Cambridge, MA: MIT Press.

Patton, W.J. (1970). *Modern manufacturing: Processes and engineering.* New Jersey: Prentice-Hall.

Pignatiello, J.J. Jr (1998). An overview of the strategy and tactics of Taguchi. *IEEE Transactions, 20* (3), 247-254.

Ponthot, J.-P. & Kleinermann, J.-P. (2006). A cascade optimization methodology for automatic parameter identification and shape/process optimization in metal forming simulation. *Computer Methods in applied Mechanics and Engineering, 195,* 5472-5508.

Potvin, J.Y., & Smith, K.A. (2003). Artificial Neural Networks for Combinatorial Optimization. In: F. Glover, & G. Kochenberger, *Handbook of Metaheuristics* (429-455). Boston: Kluwer Academic Publishers.

Prasad, A.V.S, Rao, R.K. & Rao, V.K.S. (1997). Optimal selection of process parameters for turning operations in a CAPP system. *International Journal of Production Research, 35* (6), 1495-1522.

Raj. K.H., Sharma, R.S., Strivastava, S. & Patvardhan, C. (2000). Modeling of manufacturing processes with ANNs for intelligent manufacturing. *International Journal of Machine Tools & Manufacture, 40,* 851-868.

Rangwala, S.S. & Dornfeld, D.A. (1989). Learning and optimization of machining operations using computing abilities of neural networks. *IEEE Transactions on Systems, Man and Cybernetics, 19* (2), 299-314.

Rao, R.V. (2011). *Advanced modeling and optimization of manufacturing processes.* London: Springer-Verlag.

Reddy, P.B.S., Nishina, K. & Babu, S.A. (1998). Taguchi's methodology for multi-response optimization: A case study in the Indian plastics industry. *International Journal of Quality & Reliability Management, 15* (6), 646-668.

Risbood, K.A., Dixit, U.S. & Sahasrabudhe, A.D. (2003). Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process. *Journal of Materials Processing Technology, 132,* 203-214.

Rowlands, H., Antony, J. & Knowles, G. (2000). An application of experimental design for process optimization. *The TQM Magazine, 12* (2), 646-668.

Rowlands, H., Packianather, M.S. & Oztemel, E. (1996). Using artificial neural networks for experimental design in off-line quality control. *Journal of Systems Engineering, 6,* 46-59.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986). *Learning internal representations by error propagation.* Cambridge, MA: MIT Press.

Sadeghi, B.H.M. (2000). A BP-neural network predictor model for plastic injection molding process. *Journal of Materials Processing Technology, 103* (3), 411-416.

Sankar, R.S., Asokan, P., Saravanan, R., Kumanan, S. & Prabhaharan, G. (2007). Selection of machining parameters for constrained machining problem using evolutionary computation. *International Journal of Advanced Manufacturing Technology, 32,* 892-901.

Santos, A.D., Duarte, J.F., Reis, A., Rocha, B., Neto, R. & Paiva, R. (2001). The use of finite element simulation for optimization of metal forming and tool design. *Journal of Materials Processing Technology, 119,* 152-157.

Saravanan, R., Asokan, P. & Vijayakumar, K. (2003). Machining parameters optimization for turning cylindrical stock into a continuous finished profile using genetic algorithm and simulated annealing. *International Journal of Advanced Manufacturing Technology, 40,* 1-9.

Sardinas, R.Q., Santana, M.R. & Brindis, E.A. (2006). Genetic algorithm-based multi-objective optimization of cutting parameters in turning processes. *Engineering Applications of Artificial Intelligence, 19,* 127-133.

Sathyanarayanan, G., Lin, I.J. & Chen, M.-K. (1992). Neural network modelling and multiobjective optimization of creep feeding grinding of superalloys. *International Journal of Production Research, 30,* 2421-2438.

Satishkumar, S., Asokan, P. & Kumanan, S. (2006). Optimization of depth of cut in multi-pass turning using nontraditional optimization techniques. *International Journal of Advanced Manufacturing Technology, 29,* 230-238.

Savas, V. & Ozay, C. (2008). The optimization of the surface roughness in the process of tangential turn-milling using genetic algorithm. *International Journal of Advanced Manufacturing Technology, 37,* 335-340.

Scallan, P. (2003). *Process Planning: The design/manufacture interface.* Butterworth-Heinemann.

Schrader, R.S. (2003). *Optimization of cost and cycle time for multi-pass turning operation using GA.* Genetic algorithm at the University of Illinois Fall 2002. Report No: 2003010.

Sekhon, G.S. (1982). Application of dynamic programming to multi-stage batch machining. *Computer-Aided Design, 14* (3), 157-159.

Sen, M., & Shan, H.S. (2006). Optimal selection of machining conditions in the electrojet drilling process using hybrid NN-DF-GA approach. *Materials and Manufacturing Processes, 21,* 349-356.

Sengor, N.S., Cakir, Y., Guzelis, C., Pekergin, F., & Morgul, O. (1999). An analysis of maximum clique formulations and saturated linear dynamical network. *ARI*, 51, 268-276.

Shen, C., Wang, L. & Li, Q. (2007). Optimization of injection molding process parameters using combination of artificial neural network and genetic algorithm method. *Journal of Materials Processing Technology, 183,* 412-418.

Shengdun, Z., Yong, Z., Zhiyuan, Z. & Chengwei, Z. (2007). Optimization for the process parameters of hydroforming using genetic algorithm in combination with FE codes. *IEEE 3rd International Conference on Natural Computation.*

Shin, Y.C. & Joo, Y.S. (1992). Optimisation of machining conditions with practical constraints. *International Journal of Production Research, 30* (12), 2907-2919.

Shunmugam, M.S., Bhaskara Redd, S.V. & Narendran, T.T. (2000). Selection of optimal conditions in multi-pass face-milling using a genetic algorithm. *International Journal of Machine Tools & Manufacture, 40,* 401-414.

Sick, B. (2002). On-Line and indirect tool wear monitoring in turning with artificial neural networks: A review of more than a decade of research. *Mechanical Systems and Signal Processing, 16,* 487-546.

Singh, S., Shan, H.S. & Praddep, K. (2002). Parametric optimization of magnetic field assisted abrasive flow machining by the Taguchi method. *Quality and Reliability Engineering International, 18,* 273-283.

Smartt, H.B., Johnson, J.A., Einerson, C.J. & Cordes, G.A. (1991). Develeopment of a connectionist fuzzy logic system for control of gas metal arc welding. In Dagli, C.H., Kumara, S.R.T., Shin, Y.C. (Eds.). *Intelligent Engineering Systems Through Artificial Neural Networks* (711-716). New York: ASME Press.

Smith, A.E. (1993). Predicting product quality with backpropagation: A thermoplastic injection moulding case study. *International Journal of Advanced Manufacturing Technology, 8,* 252-257.

Smith, K.A. (1998). Neural techniques for combinatorial optimization with applications. *IEEE Transactions on Neural Networks, 9* (6), 1301-1317.

Smith, K.A. (1999). Neural networks for combinatorial optimization: A review of more than a decade of research. *INFOMS Journal on Computing, 11* (1), 15-34.

Smith, K.A. & Gupta, J.N.D. (2000). Neural networks in business: Techniques and applications for the operations researcher. *Computers & operations Research, 27,* 1023-1044.

Solimanpur, M. & Ranjdoostfard, F. (2008). Optimisation of cutting parameters using a multi-objective genetic algorithm. *International Journal of Production Research,*1-18.

Solimanpur, M., Vrat, P. & Shankar, R. (2004). A multi-objective genetic algorithm approach to the design of cellular manufacturing systems. *International Journal of Production Research, 42* (7), 1419-1441.

Sonmez, A.I., Baykasoglu, A., Dereli, T. & Filiz, I.H. (1999). Dynamic optimization of multipass milling operations via geometric programming. *International Journal of Machine Tools & Manufacture, 39* (2), 297-320.

Sreeram, S., Kumar, A.S., Rahman, M. & Zaman, M.T. (2006). Optimization of cutting parameters in micro end milling operations under dry cutting conditions using genetic algorithms. *International Journal of Advanced Manufacturing Technology, 30,* 1030-1039.

Srinivas, J., Giri, R. &Yang, S.H. (2009).Optimization of multi-pass turning using particle swarm intelligence. *International Journal of Advanced Manufacturing Technology, 40,* 56-66.

Strano, M., Jirathearanat, S., Shr, S.-G. & Altan, T. (2004). Virtual process development in tube hydroforming. *Journal of Materials Processing Technology, 146,* 130-136.

Su, C.-T., & Chen, M.-C. (1999). Computer aided optimisation of multi-pass turning operations for continuous forms on CNC lathes. *IEEE Transactions, 31,* 583-596.

Su, C.-T. & Chang, H.-H. (2000). Optimization of parameter design: an intelligent approach using neural network and simulated annealing. *International Journal of Systems Science, 31* (12), 1543-1549.

Su, C.-T. & Chiang, T.-L. (2003). Optimizing the IC wire bonding process using a neural networks/genetic algorithms approach. *Journal of Intelligent Manufacturing, 14,* 229-238.

Su, C.-T. & Hsieh, K.-L. (1998). Applying neural network approach to achieve robust design for dynamic quality characteristics. *International Journal of Quality & Reliability Management, 15* (5), 509-519.

Su, C.-T., Shiu, C.C. & Chamg, H.H. (2000). Parameter design optimization via neural network and genetic algorithm. *Journal of Materials Processing Technology, 7* (3), 224-231.

Su, C.-T & Wu, C.-L. (2001). Intelligent approach to determining burn-in time and cost for electronic products. *International Journal of Quality & Reliability Management, 18* (5), 549-559.

Sukthomya, W. & Tannock, J.D.T. (2005). Taguchi experimental design for manufacturing process optimisation using historical data and a neural network process model. *International Journal of Quality & Reliability Management, 22* (5), 485-502.

Suresh, P.V., Venkateswara Rao, P. & Deshmukh, S.G. (2002). A genetic algorithmic approach for optimization of surface prediction model. *International Journal of Machine Tools & Manufacture, 42, 675-680.*

Syrcos, G.P. (2003). Die casting process optimization using Taguchi methods. *Journal of Materials Processing Technology, 135,* 68-74.

Taguchi, G. (1986). Introduction to quality engineering: Designing quality into products and processes. *Asian Productivity Organization, Japan.*

Takefuji, Y., Lee, K-C., & Aiso, H. (1992). An artificial maximum neural network: a winner-take-all neuron model forcing the state of the system in a solution domain. *Biological Cybernetics*, *67* (3), 243-251.

Tan, F.P. & Creese, R.C. (1995). A generalised multi-pass machining model for machining parameter selection in turning. *International Journal of Production Research, 33,* 1467-1487.

Tang, S.C., Chu, E. & Samantha, S.K. (1982). Finite element prediction of the deformed shape of an automotive body panel during preformed stage. In *NUMIFORM'82* /629-640). Pineridge Press, Taramen, K. (1974). Multi-machining output –multi independent variable turning research by response surface methodology. *International Journal of Production Research, 12* (2), 233-245.

Tarng, Y.S., Yang, W.H. & Juang, S.C. (1998). Application of the Taguchi method to the optimization of the submerged arc welding process. *Materials and Manufacturing Processes, 13* (3), 455-467.

Tarng, Y.S., Yang, W.H. & Juang, S.C. (2000). The use of fuzzy logic in the Taguchi method for the optimisation of the submerged arc welding process. *International Journal of Advanced Manufacturing Technology, 16,* 688-694.

Tekkaya, A.E. (2000). State-of-the-art of simulation of sheet metal forming. *Journal of Materials Processing Technology, 103,* 14-22.

Toh, C.H. & Kobayashi, S. (1983). Finite element process modelling of sheet metal forming of general shapes. In *Grundlagen der Umformtechnik I* (39-56), Berlin.

Tsui, K.-L. (1999). Robust design optimization for multiple characteristics problems. *International Journal of Production Research, 37* (2), 433-445.

Udo, G.J. (1992). Neural networks applications in manufacturing processes. *Computers and Industrial Engineering, 23,* 97-100.

Ulrich, K.T. & Eppinger, S.D. (2003). *Product design and development* (3rd ed.). McGraw Hill.

Van Den Bout, D.E., & Miller, T.K. (1988). A Traveling Salesman Objective Function that Works. *Proceedings of IEEE International Conference on Neural Networks*, *2*, 299-303.

Van Hulle, M.M. (1991). A goal programming network for mixed integer linear programming: A case study for the job shop scheduling problem. *International Journal of Neural Systems, 2* (3), 201-209.

Vijayakumar, K., Prabhaharan, G., Asokan, P. & Saravanan, R. (2003). Optimization of multi-pass turning operations using ant colony system. *International Journal of Machine Tools & Manufacture, 43,* 1633-1639.

Vijian P. & Arunachalam, V.P. (2007). Modelling and multi objective optimization of LM24 aluminium alloy squeeze cast process parameters using genetic algorithm. *Journal of Materials Processing Technology, 186,* 82-86.

Von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik, 15*, 85-100.

Walsh, M.P., Flynn, M.E. & O'Malley, M.J. (1999). Augmented Hopfield network for mixed-integer programming. *IEEE Transactions on Neural Networks, 10* (2), 456-458.

Wang, Y.C. (2007). A note on "Optimization of multi-pass turning operations using ant colony system". *International Journal of Machine Tools & Manufacture, 47,* 2057-2059.

Wang, N.-M. & Budiansky, B. (1978). Analysis of sheet metal stamping by a finite-element method. *Transactions of ASME, Journal of applied Mechanics, 45,* 73-82.

Wang, X., Da, Z.J., Balaji, A.K. & Jawahir, I.S. (2002). Performance-based optimal selection of cutting conditions and cutting tools in multipass turning operations using genetic algorithms. *International Journal of Production Research, 40* (9), 2053-2065.

Wang, J., Kuriyagawa, T., Wei, X.P. & Guo, D.M. (2002). Optimization of cutting conditions for single pass turning operations using a deterministic approach. *International Journal of Machine Tools & Manufacture, 42,* 1023-1033.

Wang, Q., Sun, X., Golden, B.L., DeSilets, L., Wasil, E.A., Luco, S. & Peck, A. (1993). A neural network model for the wire bonding process. *Computers and Operations Research, 20,* 879-888.

Wang, Z.G., Wong, Y.S. & Rahman, M. (2004). Optimization of multi-pass milling using genetic algorithm and genetic simulated annealing. *International Journal of Advanced Manufacturing Technology, 24* (9-10), 727-732.

Watta, P.B. (1996). A coupled gradient network approach for static and temporal mixed-integer optimization. *IEEE Transactions on Neural Networks, 7* (3), 578-593.

Watta, P.B. & Hassoun, M. (1996). A coupled gradient network approach for static and temporal mixed-integer optimization. *IEEE Transactions on Neural Networks, 7* (3), 578-593.

Wei, L. & Yuying, Y. (2008). Multi-objective optimization of sheet metal forming process using Pareto-based genetic algorithm. *Journal of Materials Processing Technology, 208,* 499-506.

Werbos, P.J. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* Ph.D. thesis, Cambridge, MA: Harvard University.

Wifi, A.S. (1976). An incremental complete solution to the stretch-forming and deep-drawing of a circular blank using a hemispherical punch. *International Journal of Mechanical Science, 18,* 23-31.

Willshaw, D.J., & von der Malsburg, C. (1976). How patterned neural connections can be set up by self-organization. *Proceedings of the Royal Society of London*, *B194*, 431-445.

Wong, B.K., Lai, V.S. & Lam, J. (2000). A bibliography of neural network business applications research: 1994-1998. *Computers & Operations Research, 27,* 1045-1076.

Wu, D.H. & Chang, M.S. (2004). Use of Taguchi method to develop a robust design for the magnesium alloy die casting process. *Materials Science and Engineering, 379,* 366-371.

Wu, J. & yao, Y. (2008). A modified ant colony system for the selection of machining parameters. 2008 7th International Conference on Grid and Cooperative Computing, 24-26 October, China, 89-93.

Wu, H.-J., Liou, C.-S. & Pi, H.-H. (1991). Fault diagnosis of processing damage in injection molding via neural network approach. In Dagli, C.H., Kumara, S.R.T., Shin, Y.C. (Eds.). *Intelligent Engineering Systems Through Artificial Neural Networks* (645-650). New York: ASME Press.

Yang, Y., Cheng, Y., Zhao, R. & Govind, R. (1994). Process optimization using neural networks. *IEEE Conference on Neural Networks,* 4635-4639.

Yang, J.-B., Jeon, B.-H. & Oh, S.-I. (2001). Design sensitivity analysis and optimization of the hydroforming process. *Journal of Materials Processing Technology, 113,* 666-672.

Yang, Y., Lin, H.-C. & Chen, M.-L. (2006) Metamodeling approach in solving the machine parameters optimization problem using neural network and genetic algorithms: A case study. *Robotics and Computer-Integrated Manufacturing, 22,* 322-331.

Yang, B., Zhang, W.G. & Li, S.H. (2006). Analysis and finite element simulation of the tube hydroforming process. *International Journal of Advanced Manufacturing Technology, 29,* 453-458.

Yarlagadda, P.K.D.V. & Chiang, E.C.W. (1999). A neural network system for the prediction of process parameters in pressure die casting. *Journal of Materials Processing Technology, 89-90,* 583-590.

Yehuda, R.B., Feldman, J.A., Pinter, R.Y. & Wimer, S. (1989). Depth-first-search and dynamic programming algorithms for efficient CMOS cell generation. *IEEE Transactions on Computer-Aided Design, 8* (7), 737-743.

Yellowley, I. & Gunn, E.A. (1999). The optimal subdivision of cut in multi-pass machining operations. *International Journal of Production Research, 27* (9), 1573-1588.

Yong, Z., Chan, L.C., Chunguang, W. & Pei, W. (2009). Optimization for loading paths of tube hydroforming using a hybrid method. *Materials and Manufacturing Processes, 24,* 700-708.

Youssef, A.Y., Beauchamp, Y. & Thomas, M. (1994). Comparison of a full factorial experimemt to fractional and Taguchi designs in a lathe dry turning operation. *Computer and Industrial Engineering, 27* (1), 59-62.

Zafar, N. (2002). *Optimization of tube hydroforming process.* Thesis, East Lansing: Michigan State University.

Zain, A.M., Haron, H. & Sharif, S. (2010). Simulated annealing to estimate the optimal cutting conditions for minimizing surface roughness in end milling Ti-6AI-4V. *Machine Science and Technology, 14,* 43-62.

Zhang, H.-C. & Huang, S.H. (1995). Applications of neural networks in manufacturing: A state-of-the art survey. *International Journal of Production Research, 33,* 705-728.

Zhang, H.W., Xu, W.L., Di, S.L. & Thomson, P.F. (2002). Quadratic programming method in numerical simulation of metal forming process. *Computer Methods in Applied Mechanics and Engineering, 191,* 5555-5578.

Zhou, D.N., Cherkassy, V., Baldwin, T.R., & Olson, D.E. (1991). A Neural Network Approach to Job-Shop Scheduling. *IEEE Transactions on Neural Networks, 2*, 175-179.

Zuperl, U. & Cus, F. (2003). Optimization of cutting conditions during cutting by using neural networks. *Robotics and Computer Integrated Manufacturing, 19,* 189-199.