NTNU 2002:31

Hybrid Intelligent Systems in Manufacturing Optimization

Application Methodologies of Computational Intelligence in Integrated Design and Manufacturing

Doctoral Thesis

By

Hirpa Lemu Gelgele

Submitted to the Norwegian University of Science and Technology (NTNU), Department of Production and Quality Engineering, in partial fulfillment of the requirements for the Norwegian degree of Doctor of Engineering (Dr.Ing.)

Trondheim, 2002

URN:NBN:no-2332

Printed by Tapir

ISBN 82-471-5425-0
ISSN 0809-103X

URN:NBN:no-2332

To my parents *Dessie Yadeta* and *Lemu Gelgele*

and most of all ……

to my brother *Tabsaye Jabeessa,* whom I missed a lot.

"*Tabsu, oto hawwi fi dandeetti guddaa qabdu,*
dheebu barumsa oto hin bahin darbuu keef itti yaadanno haa tatu!"

URN:NBN:no-2332

PREFACE

The work presented in this dissertation was carried out at the Department of Production and Quality Engineering, the Norwegian University of Science and Technology (NTNU) within the period February 1997 to January 2002. Professor Kesheng Wang, at the Department of Production and Quality Engineering, has been the supervisor of the work. The course work was funded by the Norwegian State Educational Loan Fund, while the major part of the study (research and write up) was carried out parallel with employed work as a Scientific Assistant at the Department of Production and Quality Engineering, NTNU.

This dissertation would not have been possible without the inspiration, support and encouragement of many people. With much pleasure, I would like to take the opportunity to thank all who stood beside me through my graduate studies.

First, I would like to thank my supervisor Professor Kesheng Wang for his invaluable guidance, advice and encouragement. His constant confidence in my works and his persistent availability, stimulating discussions and friendliness has been and will always be inspiring to me. I would like to express my gratitude for the Management and other Faculty members of the department who allowed me to work as a Scientific Assistant parallel with my doctoral studies. I highly appreciate the administrative and technical personnel of the department for their cooperation and willingness when requested. My special thanks are also due to the Norwegian Government for allowing me to study at this university and for the financial support during the initial phase of this study.

This work could not have been a reality without the help of personnel at the SINTEF Industrial Management Division and the workshop personnel of the department. Therefore, I would like to acknowledge the cooperation of Kristian L. Aas and Per Kristian Sannæs in acquisition of test data for the EDM process. The cooperation of Hans Bruvik as well as that of Jan T. Håkvåg and his co-workers at the workshop is also highly appreciated.

My family members, relatives and friends have always surrounded me with love and support. I would like to thank my wife Ayantu Geneti and our children Siddise and Fran'ol for their love, support, encouragement and patience during the whole period of my graduate studies. Most of, I am very grateful to Ayantu for taking care of our children and her limitless tolerance during my long office hours including weekends.

URN:NBN:no-2332

Finally yet importantly, I appreciate all my friends who directly or indirectly supported me during my graduate studies. I am particularly grateful to my friend Rabbira Garba and his family who stood beside me both in family matters and for the stimulating discussions we had related with dissertation writing. I am also grateful to the viewpoints and good ideas I used to harvest from the friendly discussions with Dr. Adunga Tolera. I would like also to use this opportunity to thank Galata Feyisa, File Eddosa and Wirtu Eddosa whose support I cannot express in words. I would like also to convey my special appreciation to all my Oromo friends around the globe who in one or another way encouraged me during this doctoral study.

URN:NBN:no-2332

SUMMARY

The main objective of the work reported in this thesis has been to study and develop methodologies that can improve the communication gap between design and manufacturing systems. The emphasis has been on searching for the possible means of modeling and optimizing processes in an integrated design and manufacturing system environment using the combined capabilities (hybrids) of computational intelligence tools particularly that of artificial neural networks and genetic algorithms.

Within the last two decades, a trend of interest towards use of computers has been observed in almost all business activities. This has forced the industrial business to undergo dynamic profound changes with automation through information and communication technology being on the forefront of business success. Business in manufacturing engineering is no exceptional to this trend. Several functions in the manufacturing field such as design, process planning and manufacturing have enjoyed the recent advances in information and communication technology. However, the earlier isolated automations in each function have created a significant hindrance to smooth flow of information particularly because there has been a very high system incompatibility among the computerized systems.

One of the most difficult problems in modern manufacturing is the inability of production systems to mimic the basic human capabilities such as adjusting appropriately to the ever-changing environment. From past studies, it has been possible to witness that advances in theory and application methodology of artificial intelligence techniques can overcome many of the obstacles existing in manufacturing discipline. Today, the emergence of advanced computational methods in the artificial intelligence world such as genetic algorithms and neural networks, both inspired by the natural evolutionary process, has created a new field of research and application referred to as *computational intelligence* (CI) approach.

Accordingly, the thesis focuses on the application of computational intelligence tools from two main perspectives. On the one hand, instead of the isolated automation of each manufacturing function, the CI techniques have been considered as powerful tools that allow all functions to operate within a fully integrated and intelligent manufacturing system. Particularly, since process

URN:NBN:no-2332

planning is the main linking element between design and manufacturing functions, an automated and optimized process planning function creates a much more powerful environment that leads to the optimization of the whole process. Particularly, being able to integrate feature recognition and operation sequence optimization is an important element in the manufacturing system chain that can highly contribute to the automation and flexibility of the integrated design and manufacturing system. On the other hand, the computational intelligence techniques themselves have certain weakness of their own in solving the complex manufacturing process as a stand-alone form. In a hybrid form, however, they can either support or complement each other.

To realize these two points, this thesis has focused on the development of theories and application methodologies of hybrid computational intelligence systems to model and optimize complex manufacturing processes. The aim is to exploit the strong side of one computational intelligence tool and support or complement the weakness of the other. To this effect, qualitative analysis and reasoning of computational intelligence based hybrid systems are comprehensively discussed. The developed theoretical backgrounds and methodologies are further used in key problem areas of the manufacturing system such as operation sequencing, machining economics analysis using multi-objective optimization approach and modeling and optimization of unstructured data collected from a non-conventional machining environment (electro-discharge machining). The results from the hybrid CI application to model and optimize the electro-discharge machining show that the methodology is important not only to the industrial activities using this technology, but also promotes further research and application in the discipline. Though the focus in this thesis has been on discrete part manufacturing industries, it is important to mention that the facts, the developed methodologies and the discussed issues in the study are applicable to other industrial businesses.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1. Overview

Manufacturing system management has been significantly influenced by the changes in market demands. In addition to the growing complexities in the manufacturing system itself, the demand for higher efficiency, greater flexibility, better product quality and lower cost have kept the manufacturing practices in a continuous change. Until the end of sixties, low costs were the driving forces for production. The demand for high quality products was increasingly important in the seventies and early eighties. The emphasis on manufacturing system flexibility together with the developments of information technology has been the characteristics of the late eighties and nineties. As part of this dynamic process, companies have been facing a continuous demand for decreased product life cycle and an increased product variety as well as shorter and more reliable delivery time.

With advances in information and communication technology (ICT), industries have also shown a trend of interest towards the use of computers in each phase of the manufacturing process. This puts the overall industrial activity to undergo profound changes with computerization being on the forefront of success in business. However, the application of computers in this area has been limited to specific tasks such as numerical analysis, process control, simulation and mechanical automation. On the other hand, a new understanding of the nature of manufacturing, namely that *manufacturing is in principle a system*, has appeared. With the aid of ICT, this system should operate not only as a flexibly automated system, but also be integrated and optimized.

In the search for a more computerized system that leads to faster product development, higher productivity and flexibility, lower costs and better quality products, two basic concepts, *integration* and *intelligence;* have emerged in the manufacturing vocabulary. For example, computer integrated manufacturing (CIM), integrated computer-aided design and manufacturing (CAD/CAM), intelligent manufacturing systems (IMS), intelligent machines, intelligent planning systems, intelligent sensors etc. are becoming common expressions in today’s, surely in tomorrow's manufacturing environment. Though it is not simple to find unique definitions for these concepts, the wide use of the concepts gives a general impression that most of the users seem to refer to a manufacturing system that operates seamlessly and untended or operates better than before.

Nowadays, having an integrated system is an important competitive factor in manufacturing industry. Thus, the use of integration concept in manufacturing implies a seamless linking of [manufacturing] systems that used to be automated as stand-alone units. This is because integrated manufacturing systems can improve product quality and competitiveness, have better flexibility and productivity and reduce production costs. In short, integration brings different functions of the manufacturing system together so that they function as a unified system allowing the tasks of initial concept generation to the realization of a finished product executed within one system.

The concept of manufacturing intelligence refers to the ability of the manufacturing system to act appropriately in an uncertain environment. This directly implies the automation of the manufacturing system using computers, particularly the emergence of the *artificial intelligence* (AI) technology and its applications. The following definition of AI is commonly used in the literature:

> *AI is a branch of computer science dealing with intelligence in human behavior, including reasoning, learning, self-improvement, goal seeking, self-maintenance, problem solving and adaptability.*

In a broader sense, AI is a discipline concerned with the development and application of symbolic and computational tools that mimic or are inspired by natural intelligence to execute tasks with performances similar to or better than those of natural systems. Thus, the focus on the two concepts, *integration* and *intelligence* in current research trend implies an attempt to utilize computer-based systems to intelligently linking different manufacturing functions that can work as integral units of the whole system and show flexibility in reacting to changes in their environment.

1.2. Manufacturing as a System

According to Merchant (1984), a *manufacturing system* can be conceptually thought of being an integrated whole of complex interacting subsystems organized in such a way to endeavor towards a common goal of transforming an idea to a saleable physical product. Each subsystem or activity handles different stages of the process and represents a unique discipline.

URN:NBN:no-2332

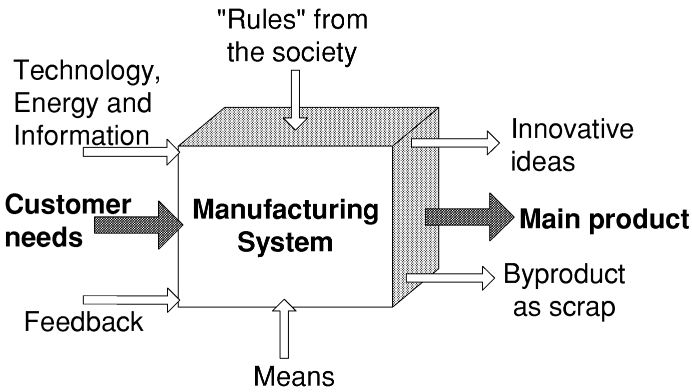


Figure 1.1: Manufacturing as a system

Figure 1.1 shows a typical functional representation of a manufacturing system. In this multi-objective goal seeking process, manufacturing system takes in the customer needs, feedbacks from different sources, the available technology, energy and other related information to transform them into products in an efficient way. The input energy constitutes raw materials, manpower, resources and common energy forms. From society point of view, the system has also to deal with waste disposals, recycling of scrap, personnel issues and governmental and environmental matters. Beside the main product, the process also results in new technologies that emerge from the innovative ideas put into action and further advance the manufacturing system.

Depending on the properties of the main product and the mechanisms used to produce them, a manufacturing system can generally be categorized into two groups: *discrete part manufacturing* and *continuous process manufacturing*. While the former refers to a process where a product undergoes a finite number of production and assembly processes, the latter indicates a manufacturing process where the product undergoes a continuous change such as transformation of raw materials into finished products through chemical reactions. Unless specifically stated, the term *manufacturing system* refers, in this thesis, only to the discrete parts manufacturing industry, more specifically, manufacturing by machining.

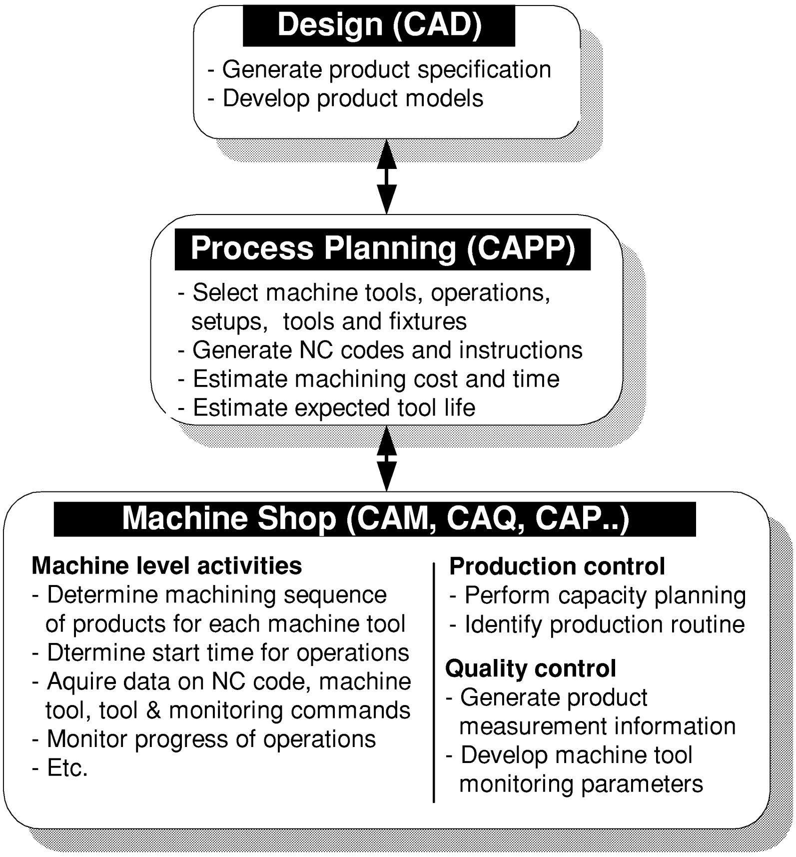


Figure 1.2: The discrete manufacturing system and its main functions

Figure 1.2 shows the representation of the discrete part manufacturing process in its simplest form with the main activities involved in it. The design level represents the initial phase of the process chain in the system where decisions are taken that influence not only the product geometry specifications, but also the product life-cycle costs, performances and the general layout of the manufacturing system. Studies show that 70 % of the production cost of a part is determined at the engineering stage where decisions concerning the selection of materials, dimensions, tolerances, surface qualities, etc. give the significant portion of the product's shape. Therefore, better understanding of the design phase has a direct positive impact on the effective performance of other downstream processes in the system.

To address the requirements of different design tasks, many kinds of software are nowadays available or under development. These design tools model the part based on a variety of product modeling methods. Because product modeling encompasses a very vast domain, no single design method can satisfy all needs of today's dynamic environment. Using one or a combination of the product modeling methods, CAD systems facilitate solid modeling, visualization, design analysis and drafting.

Process planning is a critical link between design and manufacturing. It determines the detailed manufacturing requirements to transform a raw material into a machined part according to the part design specification. In order to effectively link design and manufacturing, a *computer-aided process planning* system should handle the following tasks:

- Identification of part features
- Selection of operations, machine tools (hereafter referred to as machines), cutting tools (hereafter referred to as tools) and cutting parameters
- Sequencing of operations by considering various constraints
- Selection of jigs and fixtures and
- Determination of cutting conditions and tool paths

In its widest sense, computer-aided manufacturing encompasses all modern manufacturing technologies that use computers in a central role. This involves activities such as programming numerical controlled (NC) machines, material requirement planning, production planning and scheduling. The NC codes are often generated based on the part design created by the CAD system, and the process plan is generated by the CAPP system. The information obtained from these systems is essential for the efficiency of machining, assembly, quality control and other manufacturing functions.

The close relation between design, process planning and manufacturing shows that they have a natural dependence on each other that calls for their *integration*. The logical way out for this integration is to use sufficiently high-level product models that can communicate design information of a product to process planning and manufacturing.

1.3. Why Integration in Manufacturing?

In order to enhance the productivity of the conventional design and manufacturing activities, and hence to automate various stages of the product life cycle, CAD and CAM technologies have evolved over the last decades. Increasing market demands and the complex structure of manufacturing systems has necessitated these and other computer assisted systems to be the main tools of manufacturing system automation.

Regardless of the positive contribution made by each automated system, the communication gap between the systems is partially open due to incompatibility of equipment or software. Since both CAD and CAM systems were independently developed, each having its own method of representing product data, many of them still experience communication problems in terms of smooth flow of information between each other due to the following reasons:

1. Data format incompatibilities between systems
2. Lack of an integral database serving all engineering functions
3. Lack of standardization of CAD/CAM interfaces and
4. Insufficient automation of choice of tools, machining conditions and procedures

These issues particularly arise when systems developed by different vendors cannot understand each other. Sometimes, the required data are available in one of the systems' database with different format. For example, CAD data are frequently incompatible with the information required for process planning. In short, each of the developed systems formed a sort of '*islands of automation*' in manufacturing (Alting and Zang, 1989). Accordingly, the incompatibility of systems and software is one of the major difficulties of integration efforts in manufacturing.

1.4. Optimization of Manufacturing Processes

Optimization can be defined as a process of identifying objects or solutions that are better than the other alternatives. Using a certain measure of utility, often provided by a mathematical model and a method of calculating that measure, optimization techniques attempt to identify the best direction to move in search of the solution.

Starting from 1950s, the optimization of metal cutting has been an active area of research. Since then, many new concepts and optimization procedures have been developed that created the theoretical basis of machining operation optimization. In cost optimization of metal cutting, four cost components are important: *machining cost*, *tool cost*, *tool change cost* and its *setup cost* (Figure 1.3). Traditional optimization techniques evaluate all these cost elements against the cutting speed as the main parameter. However, manufacturing cost is affected by not only cutting speed, but also other multivariate control variables. Simplified approaches are traditionally used because the available techniques cannot treat the entire control variables and the multi-objectives simultaneously.

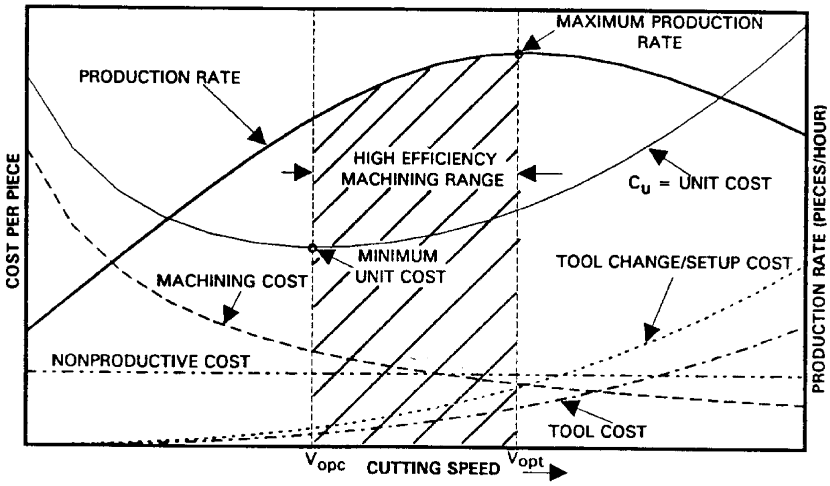


Figure 1.3: Plot of manufacturing cost elements[1]

The advent of computer technology has resulted in extensive research interests to develop computer-assisted systems for manufacturing system optimization. The initial efforts focused on computer-aided selection of limited machining parameters based on look-up tables and mathematical formulas. The optimization techniques were limited to more traditional ones such as *linear programming* (Ermer and Patel, 1974), *geometric programming* (Gopalakrishan and Faiz, 1991), *dynamic programming* (Shin and Joo, 1992), *graphical methods* (Kilic *et al.*, 1993) and *simplex method* (Agapiou, 1992).

The results from several optimization studies show that these conventional methods are very sensitive to small variations of parameter values. Compared with the conventional techniques, *genetic algorithms* (GAs) appear today as the best alternatives to get optimal or near optimal global values. Nowadays, several general-purpose GA programs are available that are aimed to solve industrial problems. However, the performance of these programs depends on how well a particular problem is setup, i.e. what representation is adopted.

1.5. Motivation and Goal of This Study

The initial objective of this study was to investigate how the integration of design and manufacturing can be improved by intelligently automating the process planning function. After analyzing the flow of information between

[1] Courtesy: (Stephenson and Agapiou, 1997)

design and manufacturing and critically reviewing how different tools can be implemented to solve the problems (Gelgele and Wang, 1998), it has been understood that the manufacturing system needs not only integration, but also *modeling* and *optimization* tools. Optimization enhances the integration effort and the benefits of integrated systems. Whereas the optimization of manufacturing processes presupposes existence of certain form of the process model, there are manufacturing processes for which it has not been possible to develop appropriate [mathematical] model yet.

In addition, it has been observed that one of the most difficult problems in modern manufacturing is the inability of production systems to mimic such basic human capabilities such as adjusting appropriately to the ever-changing environment. To support the intelligence required, both the hardware and the software used in manufacturing area should have the ability to adapt to the dynamic changes. To solve this problem, the current research direction has been towards wider application of AI techniques. In particular, recent developments show the powerfulness of CI technology to resolve these manufacturing problems of integration, modeling and optimization.

So far, there is no clear definition of the *computational intelligence* concept apart from the simple fact that it represents a category of techniques in AI that can be used for analyzing, designing and developing intelligent systems. According to the current understanding, this group consists of artificial neural networks, fuzzy logic systems and genetic algorithms (Figure 1.4 (b)).

Inspired by computation in biological systems, genetic algorithms and neural networks comprise the major part of the implementation of the CI technology. GAs are proven search or optimization techniques based on adaptive mechanism of biological systems (Holland, 1975). In accordance with the *Darwinian theory* of evolution, genetic algorithms emulate the biological process of genetic change and *survival of the fittest* concept to solve problems in engineering, science and other disciplines.

Neural networks are computational models of the human brain. In the biological nervous system, the neuron represents the fundamental element of the information-processing unit that receives electrochemical stimuli from multiple sources through its input paths and generates electrical impulses that are transmitted to other neurons through its output paths. Based on the performance of this nervous system and the mathematical theories of learning, artificial neural networks constitute a new approach to computation in the AI field. For example, the human capability of *learning-by-examples* is simulated by using an artificial model through adjustment of weights between the neurons.

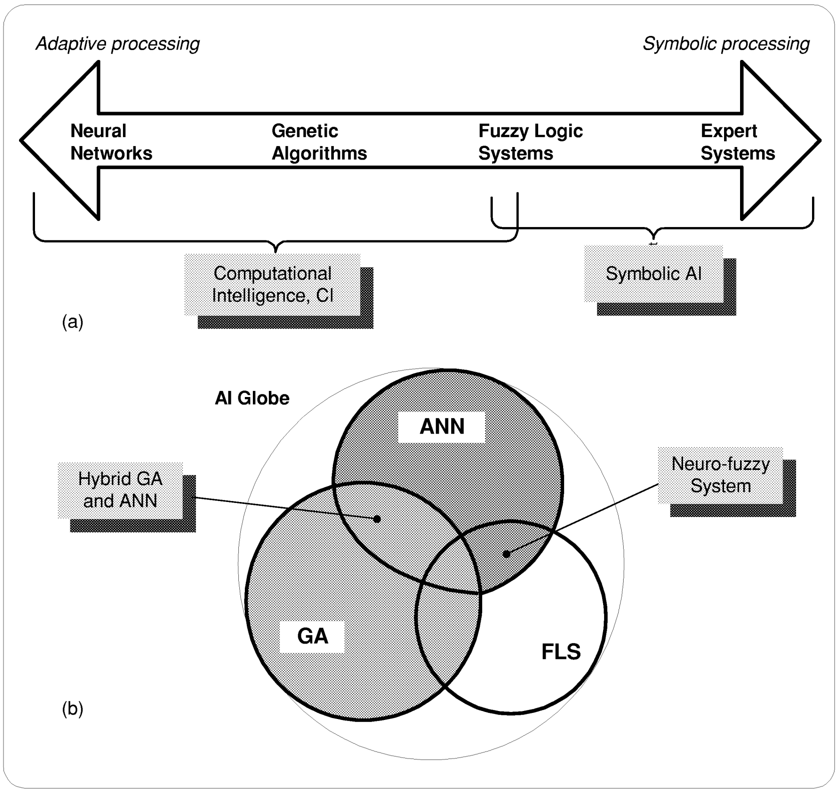


Figure 1.4: Categories of intelligent technologies for manufacturing

The versatility of CI tools has resulted in an extensive research interests both to get better functionality of the tools themselves and to solve practical problems in various fields. From functionality point of view, the CI tools have certain weaknesses to provide the required intelligence in manufacturing, and thus hybridizing one CI tool with the other(s) to improves the performance or to complements its functionality has been the recent focus of research. Hybridization can be considered at different levels including hardware, but in this thesis, it implies only to mating of software particularly that of intelligence technologies.

As shown in Figure 1.4 (a), the intelligent technologies for manufacturing problem solving include various techniques of artificial intelligence such as expert systems, fuzzy logic systems, genetic algorithms and neural networks. Putting these technologies on a continuum form, we observe from this figure that these intelligent technologies can be categorized as *symbolic processing* techniques on one extreme such as the expert systems and *adaptive processing* techniques on the other extreme such as neural networks. The latter forms the category of *computational intelligence* technology.

Figure 1.4 (b) also illustrates different forms of hybridizing the CI technologies to solve manufacturing problems. Within the last few years, research has focused on the application of the CI technologies in manufacturing systems particularly on their possible hybridization. For example, a survey report by Gen and Cheng (2000) shows that more than 200 Ph.D dissertations have been published only in the last decade on studies of GAs and their applications covering diverse disciplines ranging from biology to engineering optimization. In manufacturing field, for example, we find the following lists:

- Engineering control (Abido, 1997; Cheng, 1996; Memon, 1995)
- Engineering design (Gold, 1998; Joines, 1996; Tay, 1995; Palmer, 1994; Bowden, 1992)
- Engineering optimization (Mathew, 1998; Yokota, 1996; Pinon, 1995)
- Planning and scheduling (Li, 1999; Cheng, 1997; Wang, 1995; Wright, 1994)

Similar applications are reported in several journal articles (Kumar and Shanker, 2000; Ong and Khoo, 1999) and conference proceedings. The general view shows that the studies have focused on design and scheduling problems while the process planning problem, the most important bridging element in manufacturing system, has got insignificant attention. Further, some Ph.D dissertations (Chen, 1997; Arguelles, 1996; Jin, 1996; Hashimoto, 1995) and other publications (Monostori and Egresits, 1997; Shaffer *et al*., 1992) have been reported that attempted to combine genetic algorithms and neural networks, where most focus is given to explore how to evolve neural architectures with genetic algorithms.

From those research efforts, it is possible to conclude that (1) there has been high interest of research in application of CI tools during the last decade and (2) there are indications of active ongoing researches on application of these tools in diverse disciplines. Motivated by these observations and the existing problems in industry, the central theme of this dissertation is to study, analyze and develop methodologies how the CI technology can be implemented in integrated design and manufacturing environment. Particular focus is given to how the hybrid of genetic algorithms and neural networks can be implemented to model and optimize different manufacturing processes. This is because hybridizing enables us to utilize the combined capabilities (intelligence) of the CI tools to solve the complex engineering problems.

From the author's point of view, the contribution of this thesis can be seen from the following points. Combining the feature recognition and classification capability of neural networks to the powerful search and optimization aspect of genetic algorithms, the hybrid computational intelligence approach leads to better automation and optimization of the manufacturing system. The proposed methodology to map feature information to machining actions and optimization of the process plan task from direct interpretation of the CAD model truly establishes the smooth flow of product information from design to manufacturing.

In addition, regardless of their wider use, our knowledge about some manufacturing processes such as the *electro-discharge machining* (EDM) is very limited. Determining the optimum performance of these difficult-to-model processes and integrating them with other systems is very difficult. Introducing the hybrid CI approach to model and optimize the EDM process not only improves better performance of the process, but also attracts further researches in the area. Furthermore, the issues raised in analyzing the existing problems and the outlined methodologies contribute to better understanding of the bottlenecks in manufacturing system integration and the modern means of modeling and optimization of the process.

1.6. Outline of the Thesis

The remaining part of this thesis is classified into six chapters. Chapter 2 starts with discussion of the fundamental bridging elements for integration of design and manufacturing systems. Because features are the building blocks of modern CAD systems and the major information carriers for automation of Cam systems, this chapter first introduces the feature concept in product modeling perspective and explains the central issues in product modeling, feature recognition and feature-based design. The required level of manufacturing system automation cannot be fulfilled if one of its elements in the integrated system chain is not automated. The industrial practice also shows that process planning is both the key link as well as the weakest link in the process chain. Therefore, this chapter focuses on the underlying principles, techniques and challenges of computer-aided process planning with an objective of improving this linking element using hybrid CI systems.

Chapter 3 gives comprehensive analysis of the motivations, the principles and the applications of hybrid CI systems in manufacturing environment. Among others, the chapter discusses the operation principles and application aspects of the two main CI components – *genetic algorithms* and *neural networks*. As part of the intended hybrid CI implementation to automate the process planning function, the application of neural networks for feature recognition has been demonstrated. The last section of this chapter explains the development and implementation principles of different forms of hybridizing genetic algorithms and neural networks.

As stated earlier, the theme of the study, in brief, is being able to combine the CI tools, extracting their best capabilities and solving manufacturing problems. Integration, modeling and optimization are the forefront problem areas where enhancing manufacturing intelligence is required. Accordingly, Chapter 4 discusses methods of manufacturing system optimization using genetic algorithms and other CI tools. Adopting the travel salesman approach, this chapter discusses the developed methodology to solve the combinatorial optimization of operation sequencing problem.

Chapter 5 takes the optimization problem further in detail and demonstrates problem formulation for multi-objective optimization. Based on a particular face milling operation, the chapter discusses a methodology for economic analysis of machining and the result of the optimization.

Many relations in the manufacturing environment are not amenable for mathematical modeling approaches. For experimentally collected data in particular, graphical modeling techniques are often used to visualize the relationships between the control variables and the performance parameters of the processes. In most cases, trial-and-error methods are used based on certain recommendations from machine manufacturers. Electro-discharge machining (EDM) is one example of such processes where mathematical representation of the performances as some combination of the input variables is not simple. Chapter 6 discusses this problem and uses a hybrid CI approach to model and optimize the process based on experimental dataset.

Finally, Chapter 7 presents the concluding words and indications for further research and application in the area.

CHAPTER 2

THE BRIDGING ELEMENTS FOR DESIGN AND MANUFACTURING SYSTEMS

2.1. Introduction

As introduced in Figure 1.2, the manufacturing system can be thought of as an organization of many activities working together to function as an integrated unit. In this *integrated system*, the design phase represents the product idea in CAD models; the process planning task transforms the design information into its manufacturing counterparts, and the manufacturing phase realizes the initial design concept as a saleable physical object. The realities at the workshop floor of manufacturing systems show that full integration of design and manufacturing is not yet achieved. The earlier automation efforts could not address the problem because the capacity of the hardware and software was limited, and most research attentions were focused on stand-alone systems where design, process planning and manufacturing were automated as isolated entities.

Today, process planning is widely accepted as a potential linking element of design and manufacturing systems if the complete information of the product idea is embedded into part features. Part features allow the geometric representation of the part and at the same time carry the product model information that supports automation of the downstream processes.

For better understanding of this integration problem, it is important to define the *feature* concept and the important roles of the concept in product modeling (design). After discussing some solid modeling techniques and feature recognition problems, this chapter briefly presents the computer-aided process planning task and its challenges to solve the integrated design and manufacturing problem.

2.2. Part Features in Product Modeling

2.2.1. Feature definition

A feature is a very general term that often indicates certain non-unique shapes realized as a result of some manufacturing processes on a raw material. The term may mean different things at different contexts. In the engineering context, features represent the significance of the geometry of a part or a product including both simple geometrical shapes such as points, lines and curves; and rather complex analytical shapes such as holes, slots and pockets. The concept originates from research in process planning and has been extended into other engineering applications. Many researchers in the past have defined this concept in several ways. For example, we find the following different definitions in the literature:

Table 2.1: Examples of feature definition

Source	**Definition**
Shah, 1990; Shah, 1992	A feature is a carrier of product information that may aid design or communication between design and manufacturing or other engineering tasks.
Zhang and Alting, 1994	A feature is a region of interest in a part model.
Mazumder *et al*. 1995	A feature is one that represents a collection of entities in an intelligent form that match the way engineers think and hence provide information at a higher conceptual level than the purely geometrical representation like lines, arcs and texts.
Henderson and Prabhakar, 1992	A feature is a geometrical and topological pattern of interest in a part model that represents high-level entities useful in part analysis.

The list of feature definitions in Table 2.1 shows that some of the definitions are quite general. For example, Shah's (1990) definition underlines the context dependence of a feature and specifies four requirements that it should fulfill.

- It has to be a physical part of a component.
- It ought be mappable to a generic shape.
- It should have engineering significance.
- It must have predictable properties.

From the above set of definitions, it is possible to observe that topology and non-geometry information are not considered as feature elements. To stress the implication of the topology and non-geometry information, the definition used in this thesis is that *a (design) feature is a geometrical or functional shape with some engineering significance or meaning*. According to this definition, any design attributes of a product or components of a product (including material specifications, surface finishes, and tolerances) are also regarded as features. For machined parts, however, a feature (manufacturing) represents the portion of the workpiece that is removed by means of certain machining operations.

The difference in the way designers and process planners perceive the feature concept (as a design feature and a manufacturing feature) is often debatable. This difference in view is known as the *multiple views problem* (Soenen and Olling, 1995). This multiple views problem can diminish if consensus is achieved on the feature and its attribute names. This could result in the definition of highly application dependent features that incorporate both design and manufacturing information.

2.2.2. Solid modeling

In mechanical design of parts using CAD systems, there are normally four different ways to represent 3D geometric models: wireframe models, surface models, constructive models, and boundary models, where the last two modeling techniques constitute the *solid modeling* concept. Solid modeling is computer representation of geometric objects (in CAD systems) that can provide an unambiguous representation of the part. The aim of such representation is also to cover all engineering functions from the initial concept to part manufacturing and reuse. Solid models provide a full 3D representation and a high-level geometric description of the object.

The constructive models and boundary models mentioned above are usually designated as *constructive solid geometry* (CSG) and *boundary representation* (B-rep) respectively. Though the two modeling techniques are used to describe the shape of a typical workpiece precisely, the hybrid of CSG, B-rep and other modeling techniques are often used in modern CAD systems.

Constructive solid geometry

In CSG, solids are described as combinations of simple primitives or other “super” solids that are constructed using the so-called *building block* approach and a series of *Boolean operations*[2]. Typical standard primitives include cone, cylinder, sphere, plane, block and wedge. Solid models built using CSG method hold no explicit information about the geometry of the part, but instead describe

[2] Contrary to applications in set theory, *union*, *intersection* and *difference* are often referred to as *Boolean operators* in solid modeling environment.

how to obtain the part from a series of Boolean operations performed on the geometrical entities. Therefore, CSG model is often referred to as an implicit part model.

Formally, application of Boolean operators in solid modeling field can be defined as follows:

- *Union*: set of points that belong either to the first or to the second solid (denoted $A \cup B$)
- *Intersection*: set of points that belong to both solids (denoted $A \cap B$)
- *Difference*: set of points that belong to the first solid but not to the second (denoted $A - B$)

Using these Boolean operations, a new solid is constructed from two intersecting solids. This technique is common in mechanical engineering since it gives a precise analytic description of the model. The modeling method is also popular because adding and subtracting elementary volumes simulates the natural design process, as well as the process of removing material volume by machining from the raw material (the initial feature). For instance, Figure 2.1 shows a simple CSG model with a Boolean subtraction of a cylinder from a block that is constructed using one of the available parametric feature-based CAD systems, Pro/Engineer from Parametric Technologies. In this construction, the subtraction of the cylinder from the block corresponds to a drilling operation.

Developing solid models using features, for example, as carried out in Pro/Engineer, is strictly sequential, and it adds features whose placement depend on the prior geometry. The creation of features such as *protrusions* and *cuts* correspond to the Boolean operations of *union* and *difference* respectively.

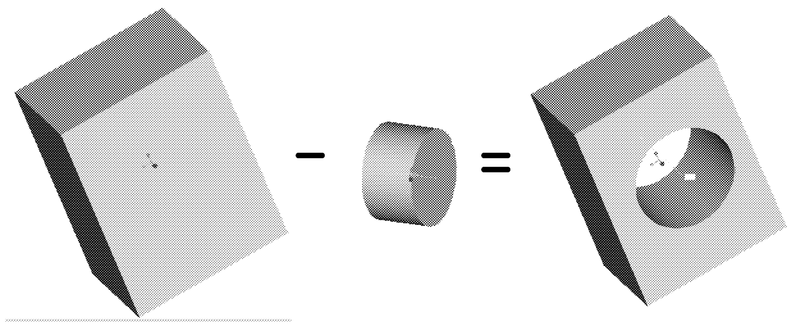


Figure 2.1: A simple CSG model – subtracting cylinder from a block

The design information in CSG modeling is stored in a tree structure with the primitives as *leaves* and the Boolean operators as *internal nodes*. The internal data structure of the tree is simple, and its data size is small because it contains not the actual object, but the instructions for how to make the part again. This makes easy not only the process of modeling, but also the modification of the solid. However, only limited operations are available to create and modify the solid. Generally, it is not easy to implement operations other than Boolean operations.

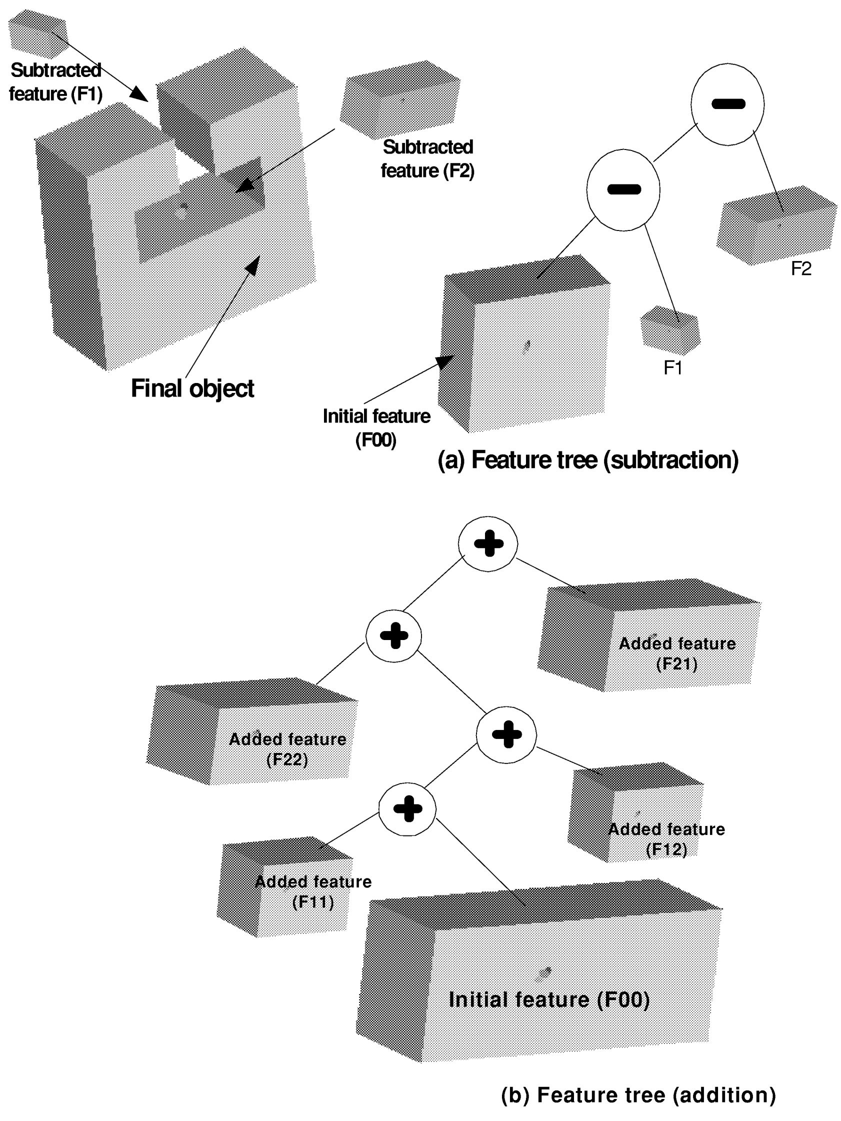


Figure 2.2: Principles of 3D object modeling using CSG

The other crucial problem with respect to current demands on solid models is the difficulty concerning *feature recognition.* As demonstrated in Figure 2.2, a number of CSG trees can represent a finite number of CSG models. Thus, the same object can be saved in several data structures resulting in a non-unique part representation. Such a tree stores the model information in an unevaluated and implicit form. Because there is no explicit geometric information in CSG models, this modeling technique is not so attractive when it comes to feature identification for process planning purpose.

Boundary representation (B-rep)

B-Rep models represent a solid by bounding surfaces that form a volume contained in a set of faces together with topological information that defines the relationships between the faces. Figure 2.3 shows a very simple B-rep model constructed using six faces. The faces, edges, vertices and the related geometric information form the basic components of the models. The *geometric information* contains the face and edge equations (or information to compute them) and vertex coordinates. The topology contains the information on the relation of the components, i.e. how the faces, edges and vertices are connected together. The boundary of the solid separates points inside from points outside of the solid.

The data structure in B-rep is simple and easy to implement. The model stores part data in an “evaluated” form, such that all vertices, edges and faces have an explicit representation. However, due to the complexity of the construction of the models, it is not trivial for a designer to build correct models directly. To implement this technique, the designer needs a sufficient collection of more well-situated and efficient solid description methods.

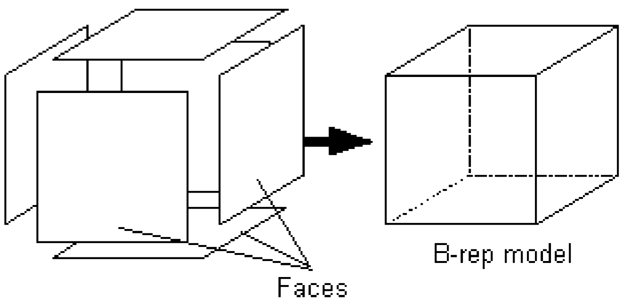


Figure 2.3: A simple B-rep model constructed using six faces

2.2.3. Part feature implementation issues

The part model represented in a conventional CAD system mainly contains design primitives that may not always correspond to manufacturing features. Manufacturing features, on the other hand, are those well suited for defining manufacturing methods systematically through geometric and technological information. Thus, the information retained in CAD systems should be interfaced or integrated with other systems in some way so as to enhance full automation of manufacturing.

Two main philosophies are generally used to develop such interfacing using features (Sakurai and Gossard, 1990; Shah and Rogers, 1988):

1. ***Feature-based modeling***: incorporating manufacturing information at the design stage that can be preserved and transferred to the manufacturing stage and
2. ***Feature recognition***: designing the part using solid modeling techniques with explicit specification of manufacturing information where the manufacturing information is retrieved from the solid model separately.

This implies that the information retained in the CAD system should either be initially built with manufacturing operations in mind or retrieved or interfaced in some way to suit the downstream processes. It is commonly accepted that both approaches are necessary and even complementary (Soenen and Olling, 1995).

2.2.4. Feature-based modeling and CAD/CAM integration

Advances in CAD/CAM related technologies have so far helped reduction of cost and lead-time of manufacturing parts to a certain extent. Particularly in 3D CAD systems, feature-based part modeling is an emerging technology that raises the level of abstraction of the primitives used to conceptually model a part and helps the system capture the design intent while modeling.

One implication of *feature-based design* is the reduction of the product development cycle by creating awareness of manufacturing processes at the design phase. Accordingly, the technology has currently getting another dimension in the field of integrated design and manufacturing research. The notion has also a positive impact on development of CAD systems because designers were forced to work at a higher abstraction levels than working with primitive shapes.

There are certain important drawbacks that the feature-based design approach suffers. Primarily, it may not always be possible to perform a one-to-one mapping of part design (form) features to their manufacturing counterparts. As suggested by many research reports, for example Arikan *et al*. (1992), unrealistic or ambiguous feature models may be generated since all manufacturing features are not well suited for design. Secondly, it is not a natural way to design shapes using manufacturing requirements because designers may not be familiar with manufacturing processes.

2.2.5. Recognition of features from CAD models

Why do we need to recognize features?

The initial idea to recognize features for machining can be traced back to the development of CAPP systems. Process planning, in general, involves a series of operations each associated with a set of machines, tools, fixtures and other resources. Feature recognition is an intermediate step that can be taken as a means to the end. The need for recognizing features from CAD models always arises with regard to the automation of process planning when retrieval of one or more of the following are required from the solid model database:

- Manufacturing features like holes, bosses, pockets, keyways etc.
- Topological information like adjacency and neighborhood to other features
- Technological information like tolerance (parallelism, perpendicularity, concentricity) and surface finish information
- Material specifications like size, type of material and its properties and
- Work holding and setup features.

Nowadays, certain CAD systems have inbuilt post processors that can directly convert the geometrical data to NC codes for direct machining. This form of integrating design and manufacturing by bypassing process planning is often designated as *integrated CADCAM system*. With this respect, one may question the reason why we need to go through the intermediate step of feature recognition and process plan generation.

There are many reasons that justify this need. Primarily, direct post processing is not appropriate technique for mass production. The downstream process in manufacturing is not limited to only machining. Manufacturing as a system also highly depends on production planning, quality control, maintenance etc., which also need the design model information. If design features are not recognized and process planning is bypassed, then the downstream processes in the manufacturing chain lack the necessary information. The machining operation done based on the direct generated NC-code is also not optimal and incompatibilities always make human intervention a necessity.

The major obstacle to completely automate the manufacturing process is the incompatibility and inconsistency of data representations in several of computer applications in manufacturing. This leads to a conceptual difference in design features and manufacturing features. Besides, a part may have different interpretations in different CAD and CAM applications; this also hinders the reasoning process of the CAD data for manufacturability.

Techniques of recognizing part features

Starting from early 1980's, many research works have been published in the field of feature recognition. Though significant milestones have been setup by those studies, the adopted techniques still suffer to solve several aspects of the demand to smoothly link CAD and CAM systems. Some of the adopted approaches will be shortly discussed.

Graph-based approach (Joshi and Chang, 1988; de Floriani, 1989):

> This approach uses the graph nature of a B-rep solid model to recognize features from a CAD model. The graph represents the boundary elements (face, edge and vertex) as nodes, and the topological relationship as arcs of a graph. The drawback of using this approach is however, the combinatorial explosion and existence of sub-graph *isomorphism* problem in recognizing interacting features. Extracting the sub-graphs from the complete feature and defining isolated features results in a large and computationally difficult search space that can be categorized as *NP-complete* problem (Garey and Johnson, 1979).

Volume decomposition approach (Kim, 1994; Sakurai, 1995)*:*

> This approach computes the removed volumes from the solid model and decomposes them into cells for machining purpose. The recognition method in this approach varies depending on the way the total volume is partitioned. The approach is effective in handling interacting features, but it involves very expensive computation and the recognized features are deficient in topological information.

Rule-based approach (Vandenbrande and Requicha, 1993)*:*

> In this approach, sets of rules, written in the form of *if-then-else*, describe the topological and geometric information of predefined features. If all conditions are satisfied, then the features satisfying the rules are

recognized. The rules represent a coded form of human knowledge in a knowledge base. However, it is impossible to define all rules for all features, and new sets of rules are required to define features with slight adjustments. The rules are also non-unique and exhaustive search is necessary (Lin *et al*., 1997).

Most recently, the application of artificial neural networks has been suggested to solve the feature recognition problem (Henderson and Prabhakar, 1992; Dagli, 1994). This application will be further discussed in Chapter 3.

2.3. Computer-aided Process Planning

2.3.1. Backgrounds of process planning

Society of Manufacturing Engineers has defined *process planning* as: *the systematic determination of the methods by which a product is manufactured economically and competitively* (Tulkoff, 1987). As an element located between design and manufacturing, process planning transforms design specifications into manufacturing processes. Computer-aided process planning (CAPP) uses computers to automate the decision-making tasks of process planning.

The process planning task for machining operations can be broken down into several subtasks as shown in Figure 2.4. The subtasks are performed based on an input data of the part design and involve several selections and decision-making processes including consideration of a number of alternatives. The first and foremost problem in creating smooth flow of information between design and manufacturing starts at extraction of the design data for process planning task. The selection of machining resources and determination of operation sequences under several constraints represents an optimization problem that decides the major cost of producing the part (Gelgele and Wang, 2000). Thus, a structured and well-developed process-planning tool obviously leads to a shorter product throughput time, lower costs and higher quality of products or services.

2.3.2. Computer-aided process planning methods

Traditionally, and still in most industries, process planning is performed manually where a skilled process planner, mostly a previous machinist, examines the part drawing and makes all necessary decisions needed to produce the part based on his/her knowledge about the process. Such manual process plans are mostly not elaborate. The quality of such process plans depends highly on the planner's knowledge about the manufacturing process environments.

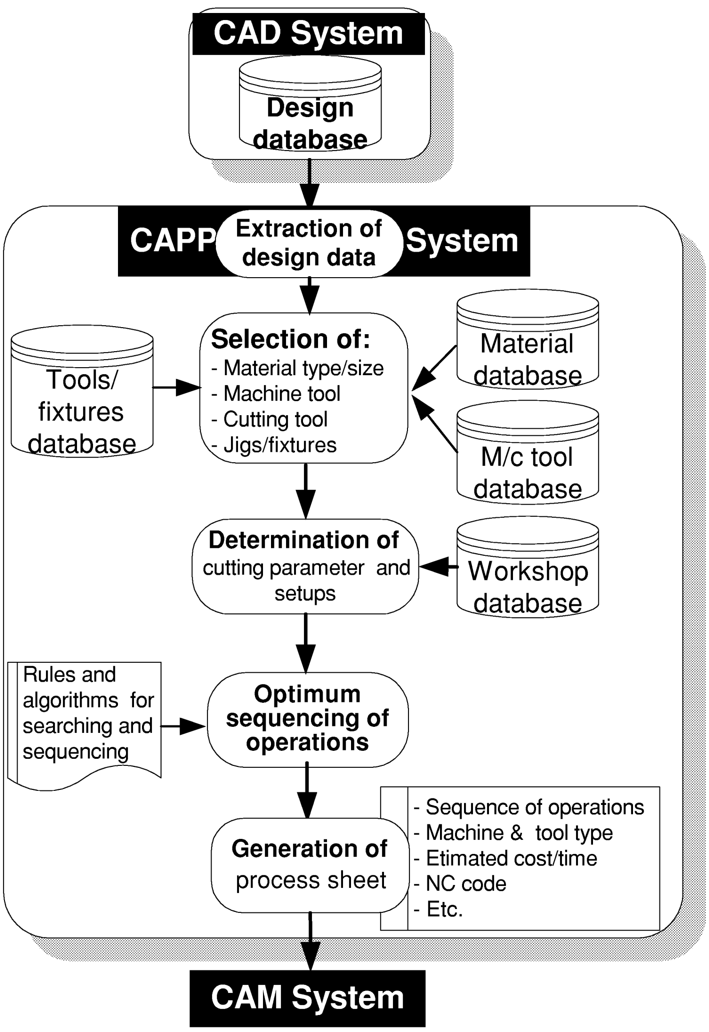


Figure 2.4: Process planning tasks

Subjective judgments of the process planner such as personal preferences and exposures to similar problems can also highly influence the generated plan. As a result, process plans generated by different planners can vary even for the same part. The planner's expertise is mostly not documented and retires with the planner himself.

With the advent of computers, however, computer-based assistance has been introduced in manufacturing area and several forms of automating the process planning function have been developed. Among those methods, the *variant* and the *generative CAPP* approaches are often mentioned as the main types.

The variant CAPP approach

In the variant CAPP approach, a process plan for a part is generated based on an existing process plan for a similar part in a Group Technology (GT) database. The method follows the principle that similar parts require similar plans. This can be considered as an advanced manual method where the search process in a computer replaces the human memory retrieval process. For a part having a similar model or matching GT code whose process plan is already generated, the part is first coded into part families based on its geometrical characteristics. A computer is then used to retrieve the operation sequence, tooling and other process planning parameters for that family from the database, a necessary modification is done for the particular geometry, and the plan is stored for further use.

In this approach, the major decision still depends on the knowledge or expertise of the planner. As a result, the effectiveness of the generated plan depends on the structuring and knowledge accumulation level of the database where the computer functions only to store data of the planner's selections and retrieve upon request.

The advantage of the variant approach is its simplicity. Compared with the manual method, it advances the process planning task with respect to the following aspects:

1. Uses existing manufacturing data and expertise consistently
2. Frees the process planner from routine clerical work
3. Uses shorter time to generate new process plans
4. Enables easy updating and modification and
5. Allows company data standardization

However, research and implementation shows that this approach suffers high inflexibility and inaccuracy because it assumes that a process plan for a new part is essentially a copy and a modification of an existing part. It also requires input from an experienced process planner for plan modifications and cannot perform planning for parts that have no matching GT code.

The generative CAPP approach

Contrary to the variant type, generative CAPP systems attempt to synthesis a process plan for a part based on information obtained directly from CAD database or a blueprint as input. The system uses rules and decision algorithms to capture process planning information. To perform the process planning tasks, the approach requires the development of sophisticated data analysis techniques that emulate human decision-making processes. Shop floor data must be represented in a form that allows decomposition of a part into elements corresponding to the necessary operations. The system must also be able to establish precedence among operations automatically based on the available geometrical and technological data and optimize if there are several alternative plans. This requires suitable optimization algorithms or search strategies with appropriate objective functions. Unfortunately, these tasks are found difficult to solve in previous researches due to computational complexities.

At its lowest form, generative process planning method reduces the time and effort required to prepare process plans in more or less consistent way. At its advanced form, it is expected to provide an industrial environment having a seamlessly automated interface between design and manufacturing, and in the process to achieve the complete integration within the manufacturing system. Compared with the variant approach, a generative CAPP system has better advantage that it does not require high expertise from human planner, and can produce plans for parts not belonging to existing part family. However, the number of parts that the system can handle is limited due to the vast knowledge requirements in its development. Particularly, representing the planning logic is a difficult task.

The advent of AI systems has made computers to mimic the logic of making process planning decisions. Most of the AI based systems developed so far are, however, rule-based expert systems, where the basic ideas of manufacturing process and its constraints are represented by rules of the form *IF* <condition> *then* <action>. This principle has been implemented in systems such as GARI (Descotte and Latombe, 1981) and PART (van Houten, 1990).

2.4. Challenges of Intelligent CAPP Systems to Integrate CAD/CAM

Process planning is still predominantly a labor-intensive activity highly based on experience, skill and intuition. As seen above, dependence on human intuition often precludes a thorough analysis and optimization of the whole process. Some of the previously developed CAPP systems had attempted to assist human planners to a certain degree in generating process routes. Today, the advent of AI technologies has given rise for *intelligent CAPP* (ICAPP) systems that are meant not only to assist the human planner, but also to replace some functions of the expert process planner.

Forced by the demands from industrial applications, integrating design and manufacturing though ICAPP is today one of the active research areas. However, most of the research efforts remained short of addressing the problem partly because people in the computer science field, whose research efforts lack accommodation of the important challenges in the manufacturing environment, develop these computer-based (intelligent) systems. The decision-making approaches considered in those efforts are mostly sequential whereas process planning requires a combinatorial problem solving approach. As a result, the developed systems are short of achieving an optimized goal.

To realize the intended integrated design and manufacturing system with a feature-based part model as input, an ICAPP system should be able to solve the following outstanding challenges simultaneously:

1. Map design features to their manufacturing counterparts
2. Select and optimally sequence operations
3. Select manufacturing resources (material, machines, tools, jigs, etc.) and
4. Optimize the utilization of manufacturing resources and data.

Numerous factors affect these tasks including: geometric shape, tolerance, surface finish, part size, material type, quantity and manufacturing method used.

2.4.1. Mapping from design model to manufacturing information

Bridging CAD systems to CAPP systems requires that the system should first be able to transfer the geometrical and technological information from the CAD model into a set of manufacturing actions. Though CAD systems are the natural and obvious source of data for process planning, they still cannot store and process all data needed by ICAPP systems. It has not yet been possible to recognize all technological data of necessary attributes including surface finish and tolerances from CAD systems. Moreover, attributes such as form, type and size of the raw material used for the part production as well as workshop capabilities and order information are not possible to extract from CAD systems.

One possible reason for this drawback can be the fact that the technological data are not recognized as features in existing feature-based design systems, but exist as text information for human understanding. The conventions used in drawings as a medium of design grammar are not always amenable for computerized communications. Certain research efforts attempted to overcome these limitations by investigating new methods of binding the technological data to the part model representation (Roy *et al*., 1989). Some even argue that the need for feature recognition can be bypassed if the design system itself uses standardized features for the construction, representation and storage of the parts (Lee *et al.*, 1993; Duan *et al.*, 1993). According to these solutions, the designer develops the part model in CAD system by selecting a blank, and specifying what operations (drilling, slot cutting etc.) should be performed on the workpiece to produce the final part. The problem with this type of solution is that it places work burden of process selection and sequencing on the designer by enforcing the designer to assume the role of both the design and manufacturing engineer.

In order to map design features to meaningful machining actions, computers need to mimic the way a human process planner thinks to do similar tasks. A process planner recognizes machining features by means of the specific actions needed to convert a raw material into the part specified by the shape, surface finish and/or size of the part. The actions such as face turning, stepping, taper turning, drilling, etc., can be interpreted in terms of the combination of machines, tools, setups and machining parameters. According to the complexity of the part, a set of either simple or combination of several actions can be employed where change of several parameters is involved.

The procedure of converting feature information into meaningful machining operations or process planning actions is carried out using the following steps:

1. Select all operations that match the geometrical and technological parameters of the feature.
2. For each selected operation, find all combinations of machine tools (MT) and cutting tools (CT) that can execute the operation.
3. For each (MT, CT) combination, determine all possible setups (SP) including cutting parameters and tool approach directions.

As demonstrated in Figure 2.5, this algorithm maps a feature to one or several sets of operations. The cylindrical part model shown is a simplest case to demonstrate the mapping process. Here, a cylindrical surface such as Ø1 is mapped to a turning operation that can be performed by several possible combinations of machines, tools and setups. For rather complex parts, several options exist from which we can select a suitable combination.

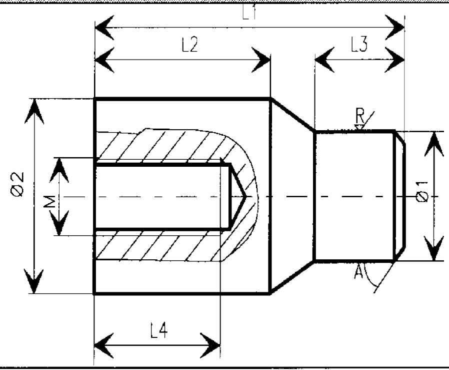
Part model	Operation	Operation states
	1: Facing 2: Turning Ø1 3: Turning Ø2 4: Taper turning 5: Drilling 6: Internal threading 7. Chamfering	(MT1, CT2, SP1) (MT1, CT1, SP2) (MT2, CT1, SP1) Etc.

Figure 2.5: Mapping from feature sets to machining operations

For example, a flat surface can be mapped to different operations such as planning, shaping, milling, etc. Each operation can be executed on different machines or each machine can utilize different tools. In short, an operation can be realized using different (MT, CT, SP) combinations and up to six normal directions of tool approach can be assumed to machine certain prismatic objects.

2.4.2. Operation selection and sequencing

For machined parts, *operation selection and sequencing* is the most critical activity of process planning. Its essence is to determine what operations possess the capability to produce the features of the part. Accordingly, the feature type, its geometry and technological requirements drive the selection process.

Operation sequencing involves determining what order to perform the selected operations so that the resulting order satisfies the precedence constraint of the operations. Maintaining the precedence relations and appropriately allocating the resources of the production plant makes the need to treat operation sequencing as an optimization problem. In the last decade, GAs have been applied to many combinatorial optimization problems including job shop scheduling (Biegel and Daveren, 1990) and other TSP type *NP-complete problems* (De Jong and Spears, 1989). Due to the similarity in complexity, it is possible to treat operation sequencing problem (OSP) using the *TSP approach* (refer Section 4.4).

2.4.3. Selection and optimization of manufacturing resources

Since the solution space of process planning problem involves a number of selections, decision-making tasks and constraint evaluations, there can exist many feasible solutions for a given planning problem, where finding the best plan among the alternatives requires definition of an evaluation criteria for each alternative. This is, in short, an optimization task.

The minimum processing time and the related cost are the most commonly used criteria for evaluating process plans. The later is mostly favored because accurate calculation of processing time at early stage of selecting operation methods and their sequences is relatively difficult. On the other hand, the machining cost can be estimated from machine usage, cutting tool usage and setup costs. This issue will be discussed in Chapter 5.

2.5. Chapter Summary

In this chapter, an overview of challenges within integrated design and manufacturing system has been presented. The need to identify the problems that hinder the effort of increasing the flexibility and intelligence of the manufacturing system has been the driving force behind the discussions of this chapter. Presently, CAD systems have automated the geometric modeling of product idea. Particularly, features are today accepted as the potential carriers of the product model information that lay the foundation of design and manufacturing automation. CAPP systems can also assist process planners to a certain level. However, getting the design information directly from CAD models and generating optimum operation sequences remains as one of the most important challenges in the area.

To facilitate the information flow between design and manufacturing using features, a mechanism of mapping feature sets to machining actions has been proposed. The issues highlighted in this chapter will be used as a foundation for developing an operation sequencing methodology in later chapters, where a special focus is first given to exploring the tools that can create a smooth flow of information.

CHAPTER 3

HYBRID INTELLIGENT SYSTEMS AND APPLICATIONS IN MANUFACTURING

The rapid expansion of the newer intelligent technologies has meant that many manufacturing fields are increasingly dependant on computational intelligence (CI) tools. The interest for these tools emanates from two angles. On one hand, instead of the isolated automation of each manufacturing function, the newer CI approaches will allow all to be incorporated within a fully integrated and intelligent manufacturing system. An integrated intelligent system creates a much more powerful manufacturing environment that is flexible and optimized. On the other hand, the CI tools have certain weakness of their own in solving the complex manufacturing problem as a stand-alone form. In a *hybridized* form, they can support each other to deliver the required intelligence for manufacturing systems.

This part of the thesis focuses on the operation principles and development of hybrid intelligent systems for manufacturing system application. After highlighting the motivations and challenges of using hybrid intelligence systems in manufacturing, the chapter focuses on the hybrid of the two CI tools – artificial neural networks and genetic algorithms.

3.1. Motivations for Hybrid Intelligence in Manufacturing

The progress in computer-aided manufacturing systems is evolving towards a new phase that can be designated as the phase of *intelligent manufacturing systems* (IMS). The computer aids at this phase are challenged to have capabilities to solve unprecedented and unforeseen problems on the basis of even incomplete and imprecise information. In short, the demands from the manufacturing environment that often challenge the research works include:

- Self-learning capability
- Capability to compute tasks in a short time
- Solving problems having incomplete and qualitative data
- Representing knowledge resulting from many years of experience
- Adapting to new situations such as new knowledge directly coming from production process or laboratory tests

At the same time, the industrial business is undergoing a profound change, with knowledge or intelligence being the forefront element of competitiveness. Thus, IMS itself must be able to manage a tremendous amount of knowledge to offer the intelligence required. Since the conventional programming tools cannot offer the required intelligence, the research studies in this field are progressively directed towards the use of tools and methods developed in the computational AI world: artificial neural networks, fuzzy logic systems, genetic algorithms and the like. AI-based systems model the adaptive and complex thinking processes of the human brain and formulate solutions to the manufacturing system where traditional approaches cannot be applied. Hybrid of CI tools that can be innovative, evolutionary and have self-learning capability are specially better attractive to solve the complex problem.

3.2. Forms of Hybrid Intelligent Systems

Since its inception in the 1950s, the field of AI has produced a variety of tools to solve manufacturing problems. Table 3.1 summarizes few key functions of AI techniques in different sectors of manufacturing. It is possible to see from this table that the capability of the AI techniques varies. At the same time, several of the techniques have more complementary nature than replacing each other, which attracts their hybrid form of application.

Table 3.1: Key functions of AI techniques

AI techniques	Key functions	Manufacturing sector
Expert Systems	Advice, goal seeking, explanation, etc.	Process planning, scheduling, diagnosis
Fuzzy Logic Systems	Communication, uncertainty handling, classification	Control (quality, inventory), scheduling, planning
Genetic Algorithms	Optimization, generalization, global search, etc.	Design, planning, control
Neural Networks	Learning, knowledge acquisition, pattern recognition, optimization, classification, etc.	Diagnosis, monitoring, prediction, modeling, quality control, inspection, forecasting

Hybrid application of AI systems enhances manufacturing intelligence through automation, integration and optimization of the system from design to part production. For example, hybrid expert systems and neural networks are often recommended both in research and practice because they are often considered as two sides of the AI coin (Tafti, 1992). This is so because each technique can solve certain type of problem that the other one cannot. In other words, neural networks can provide most of the features that are not suitable for or supported by expert systems and vice-versa. This helps to extract the best features of one system and complement its weaknesses from the hybrid (mate).

As an essential element of the IMS, *intelligent CAPP systems* are one of the potential areas for fusion of AI tools (Medsker, 1994; Ming *et al*. 1999). Similarly, hybrid systems have been implemented in many fault diagnosis and monitoring problems (Senjen *et al*., 1993; Tsoukalas and Reyes-Jimenez, 1990).

In CI, mating neural networks with FLSs had been an intensive study to develop their hybrid offspring often referred to as *neuro-fuzzy system* (NFS). It seems that AI system hybridization research has highly focused in this direction. NFSs have mostly the architecture of fuzzy systems and use the neural learning technique. The hybrid of genetic algorithms and neural networks have been suggested in many developments related with the improvement of the performance of neural networks (Winter *et al*., 1995).

Within manufacturing, the field of diagnosis and monitoring is the most attractive area to exploit the potential of NFSs (Monostori and Egresits, 1997; Ozyurt and Kandel, 1996). The symbolic representation of membership condition in fuzzy systems and the powerful pattern recognition and classification capability of neural networks have favored this integrated (hybrid) application for this particular field.

The main reason for all hybridization efforts is the fact that each AI tool has certain weaknesses of its own. The best qualities of one tool may lack in the other. In most cases, one AI tool cannot completely replace the other; they are rather complementary. Therefore, hybrid application of the tools is obviously a necessity so that the manufacturing environment can enjoy the full capability of the tools. Shortly, to give success in manufacturing fields, intelligent systems in the future are expected to be hybrid, integrated and be modular in nature.

The remaining part of this chapter first briefly discusses the working principles and applications of two CI systems: artificial neural networks (ANN) and genetic algorithms (GA). Finally, two approaches of hybridizing the two CI tools for manufacturing system applications are elaborated.

3.3. Operation and Application Principles of Neural Networks

3.3.1. The biological system analogy

Artificial neural networks are inspired by the biological nervous system and mathematical theories of learning, information processing and control. Though our knowledge about the nervous system is still far from complete, neuroscientists and others have learned some important facts in the past few decades. In the biological nervous system, the *neuron* represents the fundamental element of the information-processing unit. The neuron is a small cell that receives electrochemical stimuli from multiple sources through its input paths (dendrites). Based on the strength of the combined signals, it generates electrical impulses that are transmitted to other neurons through its output path (axon), which splits up and connects to other neurons' input paths through a junction (referred to as a *synapse*). The synaptic strength of a junction, which is chemical in nature, determines the amount of signal that is transferred.

In short, an ANN can be considered as a computational model of the human brain that is designed to simulate the biological process of the brain in performing a particular task or function of interest using computers. Due to this performance analogy, ANNs can be categorized as a family of models that are based on a *learning-by-example* principle, where problem-solving knowledge is automatically generated according to actual examples presented to them.

Figure 3.1 shows the working principle of this artificial model (the fundamental element of ANNs).

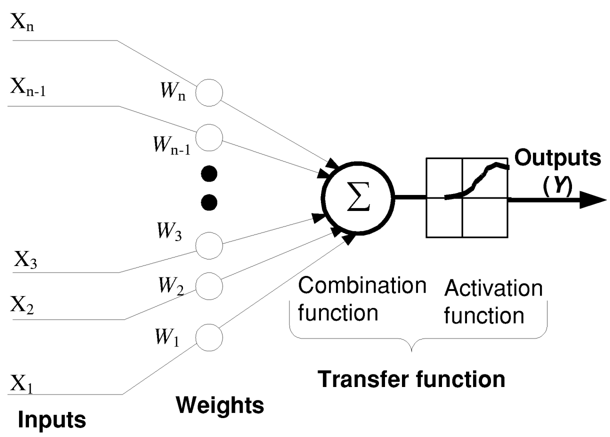


Figure 3.1: Model of an artificial neuron

Associated with each unit of the artificial neuron is a *transfer function* that determines how the neuron's value (or activation) is updated. Typically, the *combination function* computes the net input to the neuron usually as the weighted sum of all inputs. This function is a linear combination given by the following relation:

$$S_j = \left(\sum_{i=1}^{n} W_{ij} X_i - \theta_j \right) \quad (3.1)$$

where S_j is the activation level of the processing element PE_j, X_i is the input vector at input node i, W_{ij} represents the synaptic strength of the i to j interconnection (W_{ij} =0 for i=j), θ_j is the threshold value of PE_j or the bias term and *n* stands for the number of inputs.

Equation (3.1) involves three fundamental operations of the artificial model: input signal evaluation, summation and comparison with the threshold value. In this process, the sum of the weight inputs is computed and a very simple *activation function,* denoted by $f(S_j)$, is applied to the net input. The activation function transmits the output along links for further processing and limits the amplitude of the outputs to some finite value usually between 0 and 1. *Step function*, *sigmoid function* and *hyperbolic tangent function* represent the three main types of activation functions in use.

3.3.2. The structure and learning mechanism of neural networks

The functional structure of neural networks is based on our understanding of the biological nervous system. The neural network structure is built on a large number of simple and adaptable neurons. A single neuron functions rather slowly and is of little use, but collectively performs tasks at great speed when they are interconnected in a network. The way individual neurons are interconnected and the nature of the connections define the *structure* of neural networks.

In terms of their structures, neural networks are divided as *feedforward networks* and *recurrent networks*. The most popular feedforward neural network is the multi-layer perceptron (MLP) where all signals flow in a single direction from the input to the output of the network. Figure 3.2 shows a principle sketch of a MLP structure with its three distinct functional layers and the corresponding neurons: *input, output* and *hidden*. Input and output neurons form the nodes at which data enters or leaves the network, whereas hidden neurons are internal to the network. The neurons in the input layer receive data from outside of the network, whereas those in the output layer contain the network's results.

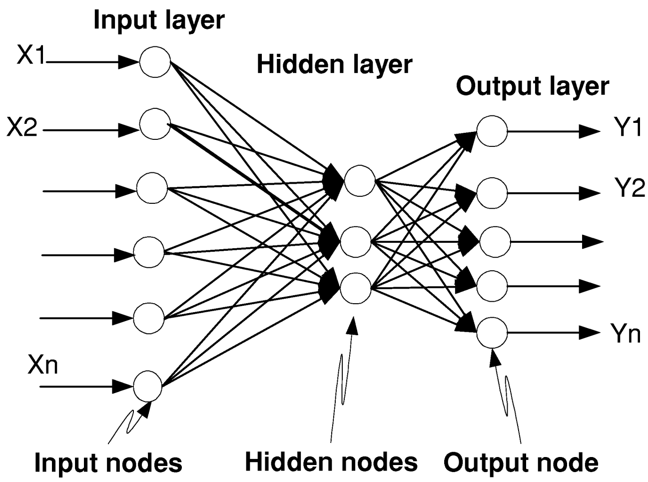


Figure 3.2: Principle sketch of MLP neural network structure

Except for the nodes at the input layer, the net input to each node is the sum of the weighted outputs of the nodes in the previous layer. These outputs are transmitted to the following layer through the connections that amplify or intensify the outputs through weight factors.

Recurrent networks are networks where the outputs of some neurons are fed back to the same neurons or to neurons in layers before them allowing signals to flow in both forward and backward directions. While feedforward networks perform a static mapping between input and output spaces, recurrent networks are said to have a dynamic memory. The *Hopfield network* is an example of recurrent networks.

The *learning algorithm* governs how the strength of the connections are adjusted or trained to achieve a desired state of the network. In order to function in a dynamic environment and react to input signals accordingly, *adaptive learning* capability is essential to neural networks. The network is as good as the data with which it is *trained* – process of putting knowledge into the network.

Using one or the combination of the above ways of learning, ANNs execute a change in the memory (weight matrix) that makes them to adapt to the nature of the input signals and solve our complex problems. Therefore, the structure and learning algorithm are the two main parameters used to categorize neural networks.

Some networks can be trained by feeding them with typical input patterns and expected output patterns. The error between the actual and expected outputs is then used to modify the weight of the connections between the neurons. This method is referred to as *supervised learning* approach. This emphasizes that the objective of training the network in supervised networks is to determine the correlation between a known input and output pattern. A good performance of the training process highly depends on the ability to provide all ranges of input variables that affect the outputs, which is, in most cases, very difficult to obtain.

There are also networks that do not need an external example for learning, but the system relies only upon local information and internal control strategies. Such learning mechanism is referred to as *unsupervised learning* approach. The response of the network in this approach is based on its ability to organize itself. The only available information to the network is the set of input patterns from which the network extracts knowledge and develops its own classification rules.

3.3.3. Application areas of neural networks in manufacturing systems

The application area of neural networks in manufacturing is surprisingly broad, covering nearly all fields from design to final product use and disposal. Table 3.2 summarizes the important applications in process control, quality control, industrial inspection, modeling and optimization.

The vast majority of these applications use supervised training approach, but obtaining the input patterns and the correct outputs is sometimes difficult. The data requirement of unsupervised training is, on the other hand, much easier and less costly to meet. However, the capability of unsupervised networks is significantly less than that of supervised training.

Table 3.2: Application areas of ANN in manufacturing

Process Control	Quality control/inspection	Modeling
- Adaptive control - Fault diagnosis - Steelmaking fusion - Furnace control - Molding operation - Petroleum distillation - Force prediction - Etc.	- Nondestructive testing - Damage identification - Fault tracing - Car troubleshooting - Welding inspection - Etc.	- Process modeling - Product design - Pulp manufacturing - Etc. **Optimization** - Production Schedule - Maintenance Schedule - Delivery of materials - Etc.

The useful properties that mostly attract the application of ANNs in all these fields include the following (Barschdroff and Monostori, 1991):

- High processing speed, due to massive parallelism
- Adaptability by means of efficient knowledge acquisition
- Capability to process unknown data
- Compactness for space- and power-constrained applications

It has been repeatedly stressed that the key to the success of manufacturing organizations is the effective integration of design to downstream applications. In the efforts done to recognize design features so far, generating a feature library for all kinds of features or writing rules every time a new feature appeared has proved to be complex and impossible. To overcome this problem, neural networks appeared recently as potential tools. The next section highlights the most important application of neural networks in an integrated design and manufacturing environment, i.e., the feature recognition problem.

3.3.4. Recognition of design features using neural networks

As a main component of CI, neural networks represent the most recent and promising approaches to *feature recognition* . Many research efforts are also reported concerning this approach with a promising result. For instance, Chan and Fisher (1996) proposed an artificial neural network methodology that can interpret the trained features directly from the geometric information instead of relying on expert-defined rules. Dagli (1994) presented a neural network approach for 2D features using *backpropagation* neural network. This approach uses a matrix of binary numbers as input to the net, which limited the applicability of the idea for general features. Henderson and Prabhakar (1992) proposed a five layer neural network using a *face adjacent matrix* as an input to the net. The face adjacency matrix used in the work could not capture all the geometrical information. Hence, the approach could not differentiate between features that have the same topology but different in dimension of faces. Similarly, Kumara *et al*. (1994) proposed a 3D interactive feature recognition system using graph and neural network approach, where a concept of *face-scores* is implemented to classify certain number of features.

The study of the literature shows that several ways of applying neural networks for feature recognition have been attempted. A number of Ph.D. dissertations have been dedicated merely to this problem (Chan, 1994; Hwang, 1991). The coverage in the literature is an indication of the importance of the feature recognition problem in manufacturing system automation and the good potential of the neural network tools to solve the problem. However, the diversity in the proposed solution approaches indicates that the search for the solution is not yet over. Regardless of the long training time that appears to be a common difficulty often encountered, the advantage of neural networks is very promising to overcome the conflicts and complexities in recognizing design features.

The literature study also shows that features considered in those approaches do not cover the range of currently implemented feature categories. As part of this challenge, an approach with new feature classes is proposed here using a multi-layer backpropagation network.

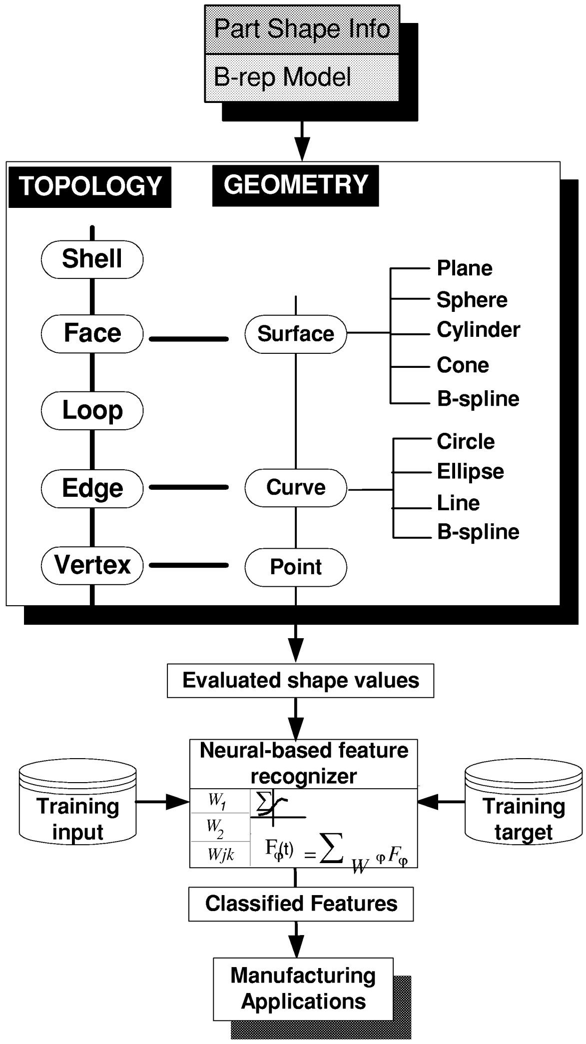


Figure 3.3: An architecture of neural network based feature classification

Figure 3.3 shows a block diagram representation of the architecture of the approach that converts a B-rep based traditional CAD model to a feature-based environment. According to this block diagram, feature data containing both geometry and topology information is evaluated to shape values or face-scores, whose calculation method is explained elsewhere (Dagli, 1994).

Based on experience on feature-based CAD modeling tools like ProEngineer, we can identify 10 commonly used feature classes that cover the possible range of features used in modern CAD models as listed below.

ID	Feature	ID	Feature
1.	Hole	6.	Pocket
2.	Round	7.	Protrusion
3.	Chamfer	8.	Rib
4.	Slot	9.	Pipe/shell
5.	Step	10.	Irregular shapes

Using the class IDs (I = 1, 2, ….., 10), the features are defined as input vectors in terms of edges, faces and vertices to present them for the neural network using the following relation:

$$\text{Feature}_j(t) = f(F_i, W) = \sum w_{ij}(t)F_i - \theta \tag{3.2}$$

where F_i is a B-rep based face-score, *w* is the weight vector of the net interconnections, *t* is the number of training patterns and θ is the threshold value.

The major problem that must be addressed in this approach is how to represent the solid model of a part so that it can be understood by the neural network. The network cannot be expected to perform any logical operations explicitly. Accordingly, the solid model has to be coded with identification of the elements and attributes required to recognize any feature. In depth study has been reported in the literature (Chan, 1994; Henderson, 1994) how the inputs of the network can be coded with numerical values as face-scores that describe the geometry and topology of the feature. The face-scores give the measure of complexity with respect to the *concavity* and *convexity* of the shape. They are calculated from *vertex scores* and give the definition of a face in terms of the face geometry and a set of boundary edges and vertices. Further, they are representations of a feature as a material removal process. A high concavity score for a face implies material removal from the face, while a high convexity score defines material addition to the face.

According to this approach, a feature is recognized if sets of faces that satisfy some predefined relationships and fulfill defined characteristics are found in the test part. For example, a cylindrical face with two adjacent plane faces having circular common edges, one concave and the other convex, can be recognized as a [blind] hole feature. After being trained on the input information and the 10 classified features as target, the neural network can be implemented on the input face-scores.

Features can also be classified using *confidence limits* that can be set to accept or reject levels. The accept level gives the minimum value that the output must reach to belong to a defined feature class, and the reject level gives the maximum value below which the output must be limited as not belonging to a defined feature class. For example, a confidence limit ranging between 0 and 1 can be defined such that an output close to 1 is recognized for a certain feature class while other limits close 0 are rejected. Accordingly, the task of the network becomes assigning the each case to one of the 10 feature classes.

The classified features are crucial for downstream manufacturing applications such as process planning, assembly planning and quality inspection. Having the classified features, operation sequencing of machining features can be presented in terms of relations among the features using the geometric and topological constraints that define precedence of operations.

3.4. Operation and Application Principles of Genetic Algorithms

The inspiration for *genetic algorithms* (GAs) comes from the principles of natural genetics and the theories of evolution. According to the *Darwinian theory of evolution*, only the most suited elements of the population are likely to survive and generate offspring, thus transmitting their biological heredity to new generations. A genetic algorithm emulates this biological process of genetic change and survival of the fittest concept to solve problems in many engineering, science and other areas by applying random, and yet structured parallel search technique.

Goldberg (1998) defines GAs as stochastic global search algorithms based on the mechanics of natural selection and natural genetics. Though the exact mechanisms of natural evolution are not well known, there exist some of its important aspects. For example, evolution works with *chromosomes* - organic information carriers containing the exact characteristics of a living being. Evolution works on these chromosomes instead of the living being they represent. GAs are essentially the software version of this evolutionary process where a solution to a problem is stored in a form of an artificial chromosome using strings of bits or numbers.

3.4.1. Operation principles of genetic algorithms

Figure 3.4 shows the basic implementation method of the genetic operators of GAs. As this technology is based upon the principles of Darwinian theory of evolution, the terminologies used here are drawn from the field of natural population genetics.

Having an objective (or *fitness)* function that can ideally represent a problem and is able to evaluate the merit of the system performance, the search for a solution starts from a *population* consisting of a number of randomly generated chromosomes. Selected chromosomes based on best-fitness create intermediate populations that are mate to form suitable couples. Genetic operators (crossover and mutation) are then applied to produce the next *generation*. *Crossover* recombines information currently available by using sections of each *parent* to create a *child* chromosome. Mutation introduces new information by altering the state of randomly selected individuals.

The general algorithmic implementation of these principles can be represented in a *canonical form* as follows:

1. Form an initial population
2. Evaluate the fitness of all solutions in the current population
3. While termination criteria is not met, repeat

   {

   - Select parents for crossover
   - Generate offspring by crossover
   - Mutate some of the members of the original population
   - Remove the least fit solutions from the parent population
   - Merge the new offspring (mutants) to the existing population

   }
4. Choose the best-fit solution and the corresponding layout.

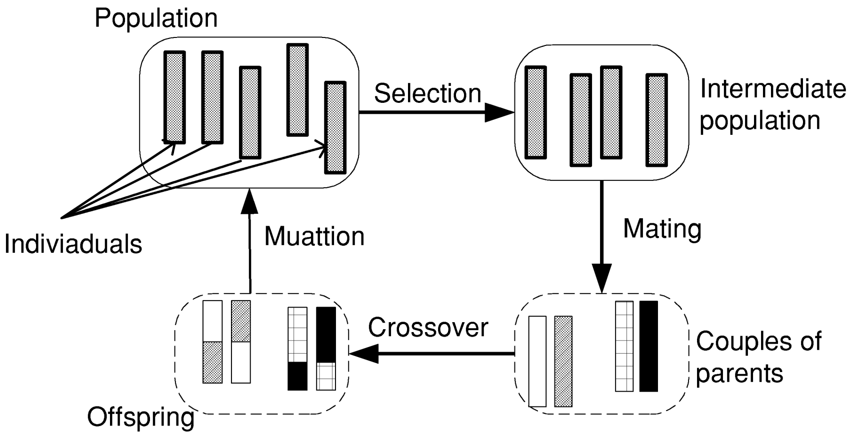


Figure 3.4: Four phases of genetic algorithm operation

It is possible to see from this canonic form that each GA application requires three main parameters:

- Methods of generating the initial population
- Defining the fitness function and
- Stopping the algorithm run

The initial population could be generated at random or from a problem specific procedure. A certain form of model that can represent the problem environment is defined as the objective(s) of optimization. The algorithm run continues by creating successive new generations and ends when a given *stop criterion* that allows enough number of generations is reached. Typically used stop criteria include specified maximum fitness, a given period of generations, a given number of total generations or a specified number of generations with unchanged fitness.

The reproduction operator

Genetic *reproduction* is a biased selection mechanism to determine which individuals will continue to the next generation to form a new population. Taking the encoded individual solution as input, a reproduction operator computes the value of that solution with respect to the objective function. Among several reproduction mechanisms (or policies) in use, the *proportionate method* (like *roulette wheel*) and the *elitism method* are mostly applied in practice.

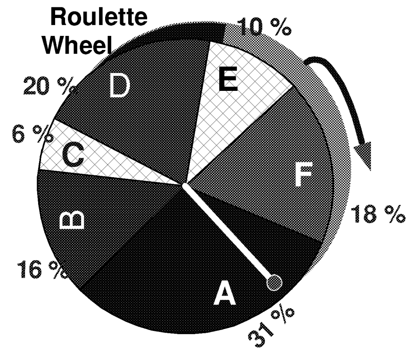


Figure 3.5: Working principle of the roulette wheel selection strategy

According to the proportionate method, individuals become candidates of the next generation depending on their fitness probability i.e., the probability of an individual to be selected is proportional to its fitness in the population. Figure 3.5 shows the working principle of the roulette wheel selection. This selection strategy is the most common proportionate technique in GA.

In the roulette wheel method, the selection probability (P) of an individual that occupies a portion α_i of the wheel and having a fitness of f_i can be expressed as:

$$P = \frac{\alpha_i}{2\pi} = \frac{f_i}{\sum_i f_i} \tag{3.3}$$

For instance, as shown on portion E of Figure 3.5, if a chromosome's fitness is 10% of the total fitness of the population, then it receives one tenth of the roulette wheel's area. The roulette wheel is spun for each free space in the new population. This space is assigned to the chromosome in the slot the ball lands. Thus, those chromosomes of greater fitness are expected to receive more spaces in the new population (such as A) than those with lower fitness (such as C). Accordingly, this ensures the selection chance of a chromosome be proportional to its fitness.

While the roulette wheel selection gives only a higher probability for the best-fit individuals, the elitism strategy selects individuals strictly based on their fitness. It keeps a copy of the sequence from the best individual to date thus keeping the fitness function non-increasing. Random selection is implemented if two or more sequences have the same best fitness. Thus, elitism always allows the best individuals to pass on to the next generation.

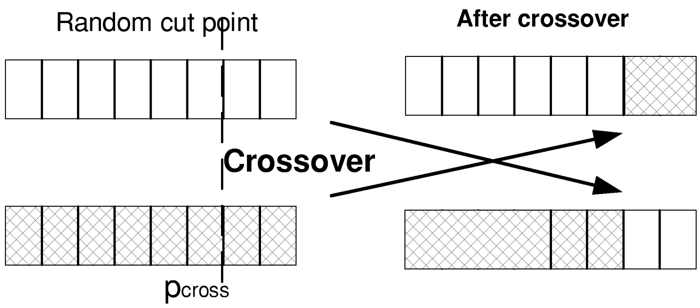


Figure 3.6: A fixed-length array, a single-point crossover

Genetic crossover operators

Crossover is the main operator of the GA approach and operates on two chromosomes at a time to produce new offspring from selected pairs of solutions in the current population. The simplest form of the operator is a single-point crossover where a random position is chosen and portions of the two parents are exchanged to form two offspring. A crossover probability p_{cross} is used to identify random-cut point. By generating a random number p between 0 and 1, a comparison is made between p and p_{cross}. If $p > p_{cross}$, then the random-cut produces two segments, head and tail, of the chromosome strings where the tail segment of parent 1 creates the tail segment of child 1. The head segments are then swapped over to produce two new full-length chromosomes. Figure 3.6 illustrates the operation principle of a single-point crossover.

Forced by the demands from real-world problem complexities, more advanced crossover techniques that fit the problem domain have been designed. Among these, the *partially mapped crossover* (PMX), *order crossover* (OX) and *cycle crossover* (CX) are mostly discussed (Goldberg, 1998). Goldberg and Lingle (1987) initially proposed PMX algorithm as a modification to the simple two-point crossover in their TSP solution. As with the single-point crossover, PMX occurs with a probability *p* that is greater than the *crossover probability of* p_c. In a similar way, the crossover point is selected randomly and the same points are identified in both parent chromosomes.

As shown in Figure 3.7, PMX has two implementation stages: *crossover* and *revalidation*. The first stage is identical to a simple double-point crossover. Illegal structures can appear at this stage because it involves multiple values in some cases while some solution elements are missing in other cases. Stage 2, on the other hand, reestablishes valid sequences in both children by swamping elements between the children. At this stage, the elements outside the crossover points that are similar to those inside the crossover points are identified, and then the repeated elements in child 1 are paired and exchanged on a piecewise basis with repeated elements in child 2 (shown by arrows in stage 2).

The PMX and OX operators are similar. The major difference is, however, that PMX, because of point-to-point mapping, tends to respect absolute gene positions, whereas sliding fill of empty spaces makes OX operator to respect relative gene positions. Further theoretical and empirical analysis of these operators is available in Goldberg (1989). According to the theoretical analysis of the article by Oliver and his colleagues (Oliver *et al*., 1987), the best crossover operator for problems such as the TSP is OX, followed by PMX and CX. This result could be expected because only adjacency of genes is important in TSP type problems. In a similar empirical study, Chan and Tansri (Chan and Tansari, 1994) have attempted to find out which operator best suites for facilities layout problem, and contrary to the TSP, they found the order of the better performing operators to be PMX, OX, and CX.

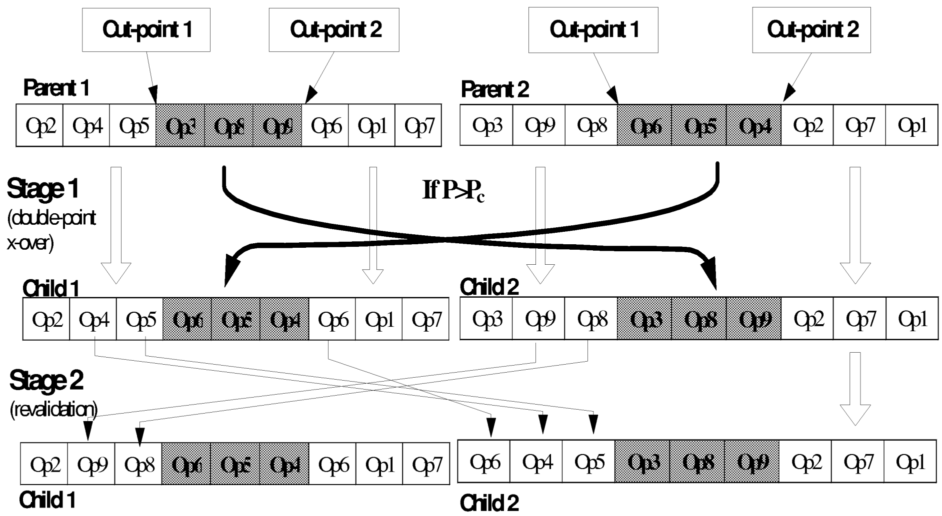


Figure 3.7: Principles of PMX operator

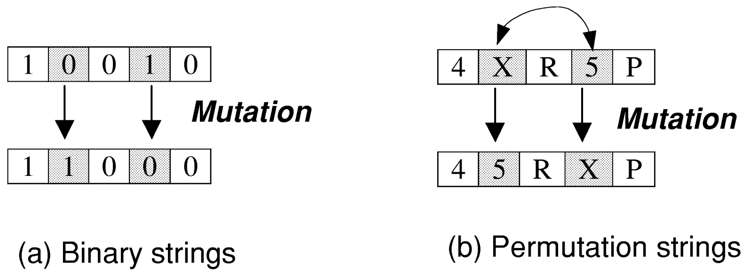


Figure 3.8: Principles of mutation genetic operator

Genetic mutation operators

Mutation is a random modification of one or more elements in the solution string. It is applied to each child after crossover and has a function of ensuring the possibility of exploring the space of solutions for any initial population. Mutation introduces noise in the population that helps to ensure that no point in the search space has a zero probability of being examined and serves the crucial role of either (a) replacing the genes lost from the population during the reproduction process or (b) providing the genes that were not present in the initial population (Gen and Cheng, 1997). Similar to crossover, mutation occurs at every cycle according to an assigned probability known as *mutation probability*, p_m. An element in the string making up the chromosome is randomly selected and changed when the probability $p > p_m$ is achieved.

The simplest form of mutation is the use of binary strings, where the mutation operator changes an element (i.e., 0 or 1) to the other elements. In the general case, however, permutation strings are designed to generate new strings that fulfill the requirement of the problem. Two elements in the string are randomly selected with a probability $p > p_m$, and are swapped with each other. Figure 3.9 shows examples of both mutation methods.

3.4.2. Applications of genetic algorithms in manufacturing systems

GAs are designated as one of the next generation tools for intelligent manufacturing (Gu and Norrie, 1995). Particularly since the 1960's, there has been an increasing tendency to imitate human beings to solve difficult optimization problems in science, research and engineering. Currently, the technology is under intensive research for several manufacturing system fields, mainly in scheduling and manufacturing system facilities layout (Syswerda, 1991; Uckun *et al*., 1993).

Compared with the traditional optimization algorithms, GAs have the following important differences:

- They work with coded version of parameters, instead of the parameters themselves. This allows simultaneous optimization of the whole parameter.
- They search from a population of points simultaneously, not from one single point. This helps to avoid many local hills.
- They use objective function information instead of derivatives or auxiliary knowledge. This contributes to their robust function in achieving an optimal solution.
- They use stochastic reproduction instead of deterministic rules. The probabilistic approach to solve problems introduces the intelligence capability to the system.

The scope of GA applications is, however, restricted to those problems where it is possible to encode the set of solutions as chromosomes and where a fitness function may be defined. Nowadays, many commercial products of GA programs are also available on the market. However, experience shows that applying a standard off-the-shelf GA technology to a particular manufacturing problem is often not possible. Therefore, a specially tailored GA for the specific problem is required. Accordingly, the next section of this chapter discusses the development of a hybridized genetic algorithm with neural networks for implementation in manufacturing system integration, modeling and optimization.

3.5. Developing a Hybrid Genetic Algorithm and Neural Network System

Evolution and learning are the two main processes that improve the adaptation capabilities of living creatures to a changing world. These two natural processes have inspired the development of non-traditional problem-solving tools within computational intelligence field, namely *genetic algorithms* and *neural networks*.

The development of *hybrid genetic algorithms and neural networks* (in short HyGANN), as the main CI tools, is necessitated due to the inherent strengths and weakness of both systems. In design and manufacturing environment, for example, many problems need the power of the two CI tools simultaneously. ANNs are known to have the capability to process problem solving through distributed and parallel search mechanisms that contribute to fast computations of the huge manufacturing data. They have shown good performance in feature classification, modeling and predication problems.

Genetic algorithms have also proved to be versatile and effective approaches in many engineering optimization problems, among others, due to their parallel and robust search capability for a global optimum. In most cases, however, GAs can only find near global optimal solutions because of their randomness characteristic. These solutions may be satisfactory for some applications, but not acceptable for others. In this sense, a search method using only GA is not powerful enough in some complex applications.

Intuitively, since a hybrid search algorithm based on GAs and some other search techniques can get a much better solution than GAs alone, various optimization techniques have been developed that combine GAs with other search mechanisms. One common way suggested in the literature is incorporating local heuristic search mechanisms into the basic loop of genetic algorithm.

3.5.1. Two forms of hybrid CI system applications

Figure 3.9 shows the two development approaches adopted for a hybrid CI system application to solve manufacturing system problems.

(a) *Hybrid form I* - enhancing the capability of one intelligent system tool by completely embedding the other(s).

(b) *Hybrid form II* - creating partially embedded or independent, but complementary applications by combining the strong side of each intelligent system tool.

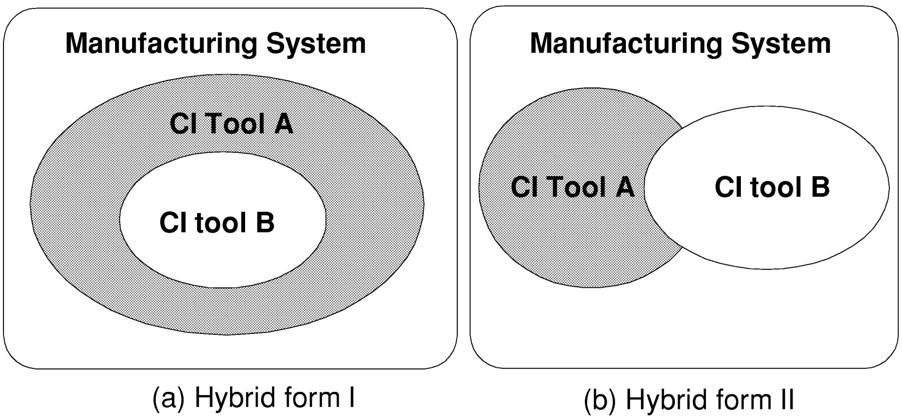


Figure 3.9: Forms of hybrid CI systems in manufacturing

Type I hybridization (Figure 3.9a) may not be surprising because a good deal of biological neural system architecture is determined genetically. This form of hybridization has been suggested in many research reports particularly with respect to construction of better ANN architecture. For example, genetic based evolution of the neural system has been reported within the following applications:

- To set the weights of a MLFF network architecture (Whitley and Devis, 1993)
- To learn neural network topologies for given applications – determining the number of nodes, the learning rates etc. (Murray, 1994)
- To construct optimal networks for given applications (Bornholt and Graudenz, 1992)
- To select training data and to interpret the output behavior of neural networks (Mitchell *et al*., 1993)

The real problem in most cases is finding an appropriate representation for the variables and translating into a constrained solution space that is tractable for a GA solution. Accordingly, there seems much work remaining before the full benefits of GAs are realized in supporting ANN development efforts.

Type II hybridization is a new trend of implementation where different CI tools support each other to solve a huge problem domain. The basic idea of implementing this form of hybridizing using genetic algorithms and neural networks is aimed to solve the overall problem in integrating design and manufacturing.

3.5.2. Application based on degree of hybridization

Depending on the level or degree of fusion of genetic algorithms into the function of neural networks or vice-versa, two further sub-divisions of the second hybridization form can be considered. Figure 3.10 shows these sub-divisions together with their potential applications in integrated design and manufacturing environment. In the first case, the degree of fusion is considered as “weak“ because the two CI tools are loosely coupled, maintain their identity and solve a sequence of problems by sharing data with one another. For instance, this fusion combines the *feature recognition* and *classification* capability of neural networks and the parallel and global *search* and *optimization* capability of genetic algorithms to enable automated feature recognition and operation sequence optimization for machining operations.

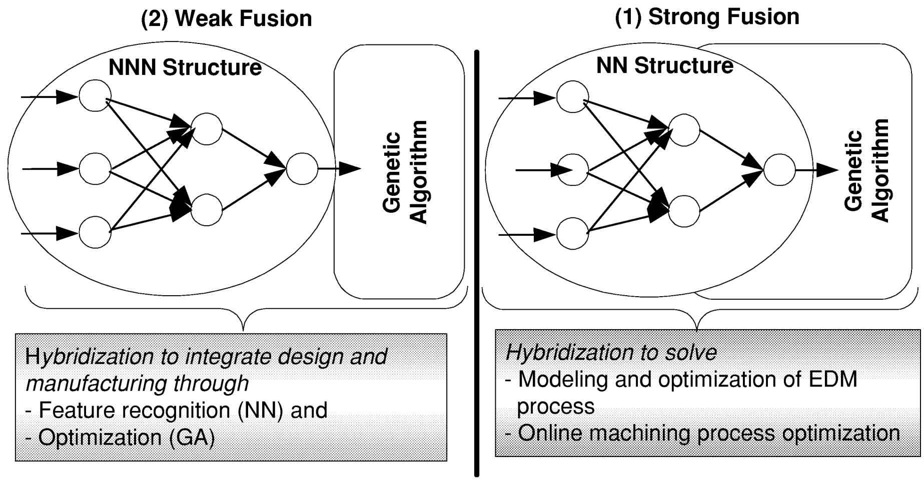


Figure 3.10: Hybridization levels of genetic algorithms and neural networks

In an integrated design and manufacturing system, feature recognition from design models and mapping to a process planning or manufacturing information stands as one of the most difficult challenges in the field. Having the feature information, the process planning task has to make selections among many alternative options under conflicting objectives and constraints. This task represents a combined combinatorial and multi-objective optimization problem demanding very powerful search and optimization techniques.

The second sub-division is designated as "strong" because part of the work of neural networks is assisted by the genetic algorithm's capability to search and generalize. This principle is primarily designed to solve problems in manufacturing systems where it is not possible to accurately define a mathematical model representing the performance of the processes. Using the power of neural networks without knowing what is going on inside the network structure is one of the main challenges facing implementation of hybrid CI tools. Therefore, defining the neural network structure through genetic evolution and finding the correlation between the input-output patterns has been proposed for those manufacturing problems having no optimization model. Accordingly, the model defines the *fitness function* that is further optimized using genetic algorithms. As demonstrated in Chapter 6, the electro-discharge machining process is such a particular problem area where it has not been possible to define an analytical optimization model and where this hybrid system can be very useful. Furthermore, this approach can be implemented for rather complex and computerized machining processes such as *NC machining centers* for online control and optimization of the process parameters.

3.6. Chapter Summary

In this chapter, the development and implementation of hybrid CI systems together with the working principles and application perspectives of genetic algorithms and neural networks for manufacturing system integration and optimization have been discussed. The proposed hybrid CI system of GAs and neural networks can cover the whole process of manufacturing a part - from product idea generation to the realization of the physical object. The forfront intention for developing this hybrid system is to combine pattern recognition and classification power of neural networks with the powerful capacity of genetic algorithms to global search and optimization in solving a common manufacturing problem. Having design features appropriately recognized, the hybrid system performs optimization tasks including, but not limited to design, operation sequencing, facility layout and production scheduling. In this process, the GA-based optimizer communicates with the CAD system, the feature library and the manufacturing resources database to realize not only the integration, but also the complete intelligence of the manufacturing system.

On the other extreme, there are manufacturing processes that are difficult to model for optimization. The proposed hybrid CI system is also aimed to solve such difficult-to-model processes using the neural networks' learning capability based on input-output patterns and further feeding of the networks results to genetic algorithm based optimization.

From these viewpoints, the study in this chapter concluded to classify the hybrids of GAs and neural networks into two - "weak" and "strong" hybrids. In the first case, both GA and neural network tools stand more or less as independent entities and solve the manufacturing problem – GA for optimization and neural network for feature recognition. In the second case, however, two strong interdependences of the tools are observable. On the one hand, genetic algorithms can promote the performance of neural networks to solve a given problem. On the other hand, neural networks can provide appropriate patterns for genetic algorithm application whereby the structure of the neural network is actively used to perform optimization tasks. The following chapters demonstrate the application of the hybrid system from both perspectives.

CHAPTER 4

OPTIMIZATION IN MANUFACTURING SYSTEMS

4.1. Introduction

Optimization can be defined as a process of identifying objects or solutions that are better than the other alternatives. To identify the best alternative(s), optimization techniques require a measure of the "goodness" or utility of an object and a method of calculating that measure. A mathematical model that takes a parametric description of the object and gives the perfection, effectiveness or functionality degree of the object - known as the objective function generally provides such a measure of utility.

Traditional optimization techniques use local information to identify the best direction in which to move such as in the case of *hill-climbing* problem. The hill-climbing man, after having taken a small step in a given direction, tries to get new local information to move on, and the process is repeated until no such improvements can be made. For this technique, there is no possible way to know whether the best possible solution has been identified or not. The solution may be better than those values that are local to it, but there is no guarantee that this solution is better than all other values. Such techniques are, therefore referred to as *local optimization* techniques.

One distinguishes between two types of optimization problems: *single objective problems* (SOP) and *multi-objective problems* (MOP). The first case treats only one scalar-valued criterion in its objective function, whereas the objective function for the second case involves more than one criterion that are treated simultaneously (Dev, 1995; Steuer, 1986). Real-world problems in industrial production often require optimization of more than one measure of performance at once. Multi-objective optimization approach is necessary in such areas because the measures may conflict with each other, and it can be unsatisfactory to combine them into a single optimization objective or reduce them in some way so that only one is optimized.

SOPs have been extensively studied within the last 50 years to solve our problems in manufacturing and other industrial functions. As almost every important real-world problem involves multiple and conflicting objectives, there has been an increasing interest to search for tools for MOPs since the 1960s. A number of scholars such as Pareto (1971), the most recognized pioneer in the field, have also made significant contributions to the problem. Among the developed global optimization tools for MOP, genetic algorithms have received a considerable attention as a novel approach creating a new direction of research and application known as *genetic multi-objective optimization* (Gen and Cheng, 2000).

4.2. Categories of Optimization Methods

There are many forms of categorizing optimization problems depending on the optimization variables, the objective function used, the optimization constraints and others. The optimization variables can take on *continuous* or *discrete* as well as *symbolic* values. The objective function can be continuous or discrete. It can also have *linear* or *nonlinear* forms. Constraints can also have linear or nonlinear forms or even may not exist. From application point of view in the manufacturing environment, we can focus on two categories of optimization methods:

1. Optimization using models and
2. Optimization without models

Among the *model-based* methods, mathematical models have been highly explored for nonlinear optimization problem researches using analytical, numerical, graphical and experimental techniques.

In manufacturing, some machining processes are too complicated to warrant appropriate analytical models and most of the time, the assumptions on which the analytical models are developed can contradict the reality. More importantly, it is sometimes difficult to adjust the parameters of the models according to the actual situation of the machining process. As a result, optimization as well as optimal control of some processes such as the EDM process (Chapter 6) is difficult to perform.

As shown in Figure 4.1, artificial neural network approach (Section 3.3) is one category of model-based optimization method. Such a model defines a pattern between an input vector x and an output pattern. Because most relations in manufacturing systems are not simple to model using mathematical modeling approaches, the neural network technology has recently attracted very wide application areas in modeling and optimization of manufacturing processes. This is also partly due to the capability of neural networks to map the input/output relationships using a sequence of training runs and massive parallel computing.

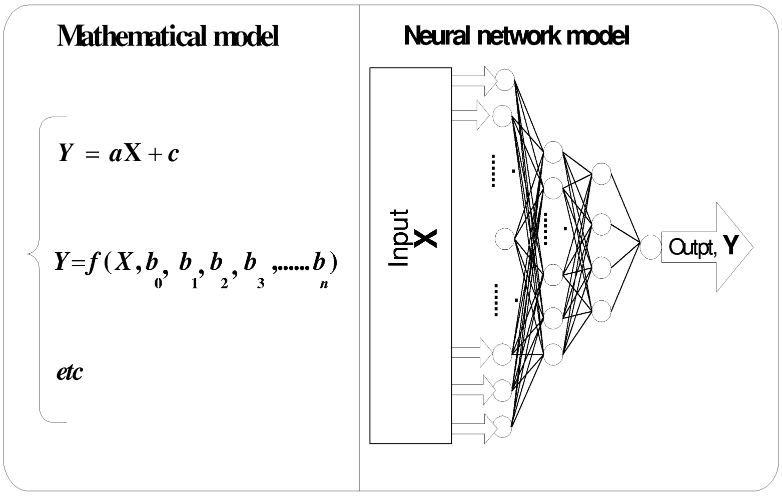


Figure 4.1: Example of model-based optimization methods

Optimization without models approach includes *one at a time*, *simplex method* and *genetic algorithms*. Among these, the one at a time method is the most widely used, but it is often criticized for being one of the worst optimization techniques. The implementation of this method, for example for a 2-variable problem, starts by changing one variable, say x. Keeping the other variable constant, the response obtained is recorded. Searching for the optimum value of the other variable, y, then starts by keeping the achieved optimum value of x constant. Unfortunately, this leads to small and local improvements of solutions, and the solution obtained extremely depends on initial start point. The actual optimum value can obviously be hidden away from the search path.

4.3. Multi-objective Optimization

Multi-objective optimization problems are common in manufacturing because most of real-world industrial problems involve two types of difficulties:

1. *Multiple and conflicting objectives* – where, instead of a single optimal solution, competing goals give rise to a set of compromise solutions, and none of the possible solutions can be said to be better than the others, and
2. *A highly complex search space* - where an overwhelmingly large and complex search space creates difficulties for traditional methods

Traditional methods of optimization deal with these problems by allocating weights to each of the objectives to indicate their importance in the problem.

Research shows that treating multi-objective problems in this way is very subjective, may over-simplify the behavior of the problem, and it is often hard to find weights that can accurately reflect the real-life situation.

Traditional methods also attempt to combine the multiple objectives into a SOP or reduce them in some way so that only one is optimized. For many of industrial problems, this is often unsatisfactory because some of them can have conflicting objectives. Keeping costs low and quality high in manufacturing; achieving high material removal rate and good surface quality in machining, etc. are examples of such conflicting objectives in manufacturing. Thus, in the search for efficient optimization strategies for MOPs, parallel optimization techniques such as genetic algorithms have been developed. In a complex manufacturing process optimization scenario, for example, a MOP solution has to minimize the total operation costs and maximize the production rate simultaneously. Depending on the application, further objectives may be included that can be either defined explicitly as separate optimization criteria or formulated as constraints that cannot be violated.

4.3.1. Basic concepts and terminologies

For optimization to be meaningful there must be an *objective function* to be optimized and more than one *feasible solution*, i.e., a solution that does not violate the *constraints* must exist.

Defining the multi-objective optimization concept

Formally, multi-objective optimization can be defined as follows:

> A general MOP includes a set of *n* parameters (decision variables), a set of *k* objective functions, and a set of *m* constraints where the objective functions and the constraints are functions of the decision variables.

Mathematically, this can be formulated as:

$$Max\ Y = f(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots, f_k(\mathbf{x})\}$$

$$Subject\ to{:}\ C_i(\mathbf{x}) = \{c_1(\mathbf{x}), c_2(\mathbf{x}),\ \ldots c_m(\mathbf{x})\} \leq 0$$

where f($\mathbf{x}$) is an objective function, $\mathbf{x} \in \mathbf{X}$ is a decision vector $\mathbf{x}$ in decision space $\mathbf{X}$ and $C_i(\mathbf{x})$ is an inequality constant of *m* functions that form an area of the feasible solution set.

This means that to achieve, for example, two objectives such as low cost (f_1) and high production rate (f_2) under given constraints (C_i) for a machining operation, an optimal solution might be one that achieves maximum performance at minimal cost and does not violate the ranges of the decision variables and other constraints of the machining environment. If such a solution exists, then it is a question of solving only a single-objective optimization problem since the optimal solution for either objective may be optimum for the other objective. On the other hand, what makes multi-objective problems difficult is the common situation when the individual optima corresponding to the distinct objective functions are sufficiently different. Then, the objectives are conflicting and cannot be optimized simultaneously. Multi-objective optimization approach attempts to find a compromise solution, which of course is at the sacrifice of certain goals.

The optimization search space

A *search space* specifies the ranges of variable assignments that are explored during the search process for optimal solutions. The search space may contain only feasible regions specified by constraints or may contain some infeasible regions as well. The optimization search space can also be finite or infinite. Optimization of engineering problems mostly involves *finite search space* because the search involves integral number of objects.

For continuous optimization problems in which variables can take on real values, the search space becomes *infinite*. In general, the size of the search space directly affects the computational complexity of the corresponding search algorithm.

4.3.2. Traditional approaches for MOP

The general method of solving for MOP using traditional methods involves either obtaining a compromised solution or identifying all near-optimal solutions. As discussed above, traditional methods such as *weighting* and the *constraint* methods attempt to combine the objectives into a single parameterized function. These methods rely on assumptions about the problem functions to make progress towards the optimal solution. For smooth nonlinear functions, they can compute derivatives or gradients that indicate the directions in which the functions are increasing or decreasing. For linear functions, they can move immediately to the extreme values of straight-line functions (as determined by other constraints) in a single step. The application principles of these two methods will be briefly discussed.

Weighting method

This method attempts to convert an original multi-objective problem into a single objective one by forming a function that is a *linear combination* (weighted sum) of the criteria. The difficulty involved in choosing appropriate weights for the criteria however makes this technique to be unsatisfactory to use in an area that is very sensitive to variation in weights. Using *weighting method*, a maximization problem, for example, can be expressed as:

$$Maximize\ \ Y = f(x) = \sum_{i=1}^{n} w_i f_i(x)\text{, subject to x} \in \mathbf{X} \qquad (4.1)$$

where $w_i > 0$ represents a normalized weight such that $\sum w_i = 1$ and $f_1(.), \ldots f_n(.)$ are *n* objective functions to be maximized.

Using such formulation, alternative feasible solutions are generated from which the best solution(s) can be chosen by parametrically varying the weights. It is obvious that the solution, in general, is not unique for not linearly dependent functions. With the introduction of the *Pareto* dominance concept, it is possible to divide any group of solutions into two subgroups: the *dominated* and the *non-dominated*. Solutions belonging to the second group are the “efficient” solutions, i.e. the ones for which it is not possible to improve any objective value without deteriorating the values of the remaining objectives. The main disadvantage of this technique is that it cannot generate all alternative optimal solution sets.

Constraint method

Another traditional optimization technique used to find multi-objective optimal solutions is the *constraint method*. The method arbitrarily chooses one objective function for SOP solution and transforms *k-1*of the *k* objectives into constraints. After the transformation, the whole optimization process is treated as a nonlinear optimization problem at the presence of equality and/or inequality constraints. Due to existence of complex constraints, nonlinear optimization problems are not easy to solve even though they are very important in practical use. In the last few years however, there has been a growing effort to apply genetic algorithms for nonlinear optimization problems (Richardson *et al.*, 1989).

Discussion on traditional methods

The attraction for the traditional optimization approaches and their popularity comes most probably from the fact that there exist well-studied algorithms that can solve single objective problems specially in areas of small search spaces or where analytical methods are available. Solving such simpler search spaces using analytical methods involves enumerating, evaluating the objective function and selecting the best feasible solution.

For large-scale problems, however, hardly any real multi-objective optimization techniques have been available. By contrast, wide ranges of heuristic methods have been known that can be used to solve single-objective optimization problems. Traditional optimization approaches assume also existence of a continuous function that is simple unimodal and differentiable. Real-world problems, on the other hand, involve discontinuous functions that are non-differentiable, complex, multi-modal or noisy. In other extreme cases, we may not have functions that describe the optimization problem.

The difficulty in terms of sensitivity to change of parameter values is one of the drawbacks of applying traditional optimization strategies in manufacturing. Because these methods cannot include all conflicting parameters, their application can be only to restricted areas. Moreover, all traditional methods require several optimization runs to obtain an approximate optimal solution. As the runs are performed independently from each other, combined action is usually not possible which, in turn, may cause high computation overheads depending on the application.

Recent research efforts have focused on finding alternatives to traditional methods through which (1) problems having large search spaces can be handled and (2) multiple alternative trade-offs can be generated in a single optimization run. Genetic algorithms are such computational intelligence tools that can be implemented in such a way that both of the above difficulties can be addressed.

4.3.3. Genetic multi-objective optimization problem

Genetic algorithm takes an alternative approach to the task of optimization based upon the power of *Darwinian evolution* to solve complex problems. The inherent characteristics of multiple directional and global search capability of GAs demonstrate why *genetic search* is well suited for MOPs. Particularly, domains that are traditionally difficult to optimize - discontinuous, multi-modal and noisy domains are good candidates for this technology. This is because rather than operating upon the objects themselves, the genetic algorithm approach operates upon the parametric description of the objects.

The fact that GAs perform optimization by searching from one population of solutions to another, rather than from one solution to another makes them also well suited to multi-objective optimization. The way the principle of the survival of the fittest is implemented is the key to the successful application of genetic algorithms for multi-objective optimization.

4.4. Operation Sequence Optimization Using Genetic Algorithms

The power of GAs has meant that many functions in manufacturing area are interested to solve optimization problems using this approach. In integrated design and manufacturing environment, optimization of the design process itself and the operation sequencing task in process planning are the most important areas where GAs can contribute immensely. The former is extensively studied in conjunction with the development of CAD system, while research on optimization of operation sequencing is still at its infant stage.

With respect to the automation of CAPP systems, there is still a question on how to formulate the operation sequencing problem. Some studies treat the problem as a *combinatorial* type (Korde *et al.*, 1992; Zhang *et al.*, 1997), while others treat it as *multi-objective* type (Fenton and Gagnon, 1993; El-Sayad and El-Gizawy, 1997). In some cases, the problem is purposely formulated very simplified to a solvable degree and thus remain short of solving the rather complex manufacturing system. The complexity nature of sequencing machining operations with all possible conditions and constraints has necessitated the use of trial-and-error and iteration based solutions.

In general, the formulation approaches, combinatorial and multi-objective, cannot be mutually exclusive. The objective (problem) at hand always dictates how to treat the problem. Primarily, operation sequencing has to do with precedence of operations in which finite number of solutions consisting of non-separable entities such as machines, tools, machine operators and other physical objects are involved. The combination of these objects defines a huge body of problems with different features and properties. Though the objects have different characteristics, the optimization problem can be characterized as determining the combination of the objects with constraints.

At a particular machine level, on the other hand, the objectives such as minimum machining cost and high surface quality can be conflicting because the first objective implies short machining time, whereas the later needs long machining time. Such problems can be formulated as multi-objective optimization problems.

Thus, the conclusion is that depending on the input parameters and the goals, the operation sequencing problem for process planning can be formulated either as a combinatorial or a multi-objective optimization problem or combination of the two with constraints. Generally, the two problem formulation approaches cannot be seen separately because the solutions to a given sequencing problem require both approaches at different levels.

4.4.1. Combinatorial optimization of operation sequencing problem

In general, optimization of engineering problems can be categorized as *discrete constrained optimization problems*. Based on their computational complexities, discrete optimization problems can be *polynomial* or *non-polynomial* (NP) type. The former has been extensively studied and a number of algorithmic solutions exist. However, many discrete optimization problems in real-world applications do not have such algorithmic solutions or are not solvable in polynomial-time. Such problem types are referred to as *NP-hard* optimization problems.

Discrete optimization of manufacturing problems where there is a need for efficient use of scarce resources is traditionally done using simple *enumeration* techniques. For large size problems, such as in machine scheduling, sequencing and balancing, cellular manufacturing design, etc., this is impossible because the number of feasible solutions to be enumerated can experience a combinatorial explosion. This defines an optimization problem type known as *combinatorial optimization problem* (COP) - the term *combinatorial* refers to the fact that only a finite number of alternative feasible solutions exist.

The COP concept also implies that there are many possible alternatives to consider and one overall goal determines which of these alternatives is best. Many engineering problems come under this category because the activities and the resources such as machines, tools, operators, etc. are discrete and indivisible items. In addition, some problems have only a finite number of alternatives and consequently can appropriately be formulated as combinatorial optimization problems.

The *travel salesman problem* (TSP) is one of the most studied problems in combinatorial optimization category. According to this problem, a travel salesman must visit a given number of cities once and only once and in so doing attempts to optimize the total distance traveled. TSP is not only combinatorial in nature, but also a problem type that computer scientists classify as *NP-hard* - problem type for which there, most likely, exists no algorithm that can consistently find the optimum solution in a polynomial time. The technique has been used as a benchmark model for many other problem areas such as assembly process optimization (Fu and Su, 2000) and vehicle scheduling (Karuno *et al.*, 1997; Malborg, 1996). Due to the difficulties involved in solving TSP type combinatorial optimization, genetic algorithms are now suggested to be better tools for this problem category.

Based on the feature mapping procedure outlined in Section 2.4.1, the following section explains the implementation of the TSP approach for *operation sequencing problem* (OSP).

4.4.2. The TSP scenario for operation sequencing

Many application areas are motivated in the work on the travel salesman problem because TSP provides an ideal platform for the study of general methods that can be applied to a wide range of discrete optimization problems. In previous researches, this modeling approach has been proposed for sequencing of machining operations. For example, Kim and Suh (1998) have reported a mathematical model for operation grouping and sequencing of multistage machining system from production scheduling perspective; Irani *et al.* (1995) have implemented TSP technique to generate alternate process plans and rank them in order of increasing cost.

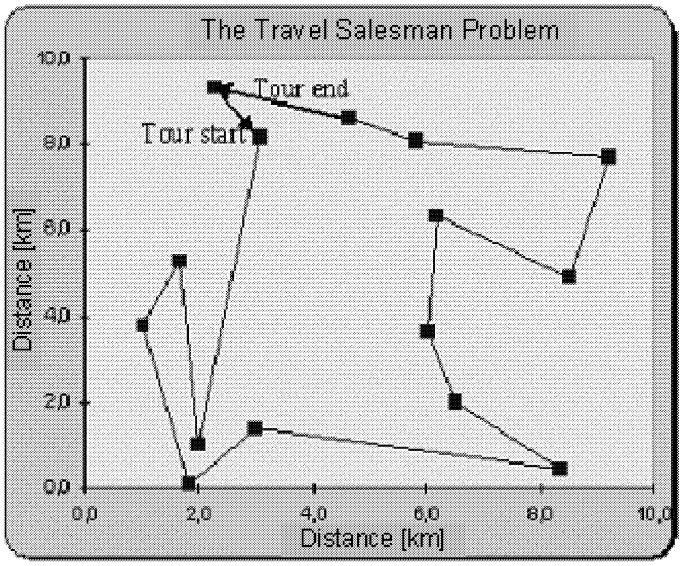


Figure 4.2: Example of a travel salesman tour of 15 cities

As illustrated in Figure 4.2 and Figure 4.3, there are certain similarities between OSP and TSP that motivate the adoption of the TSP approach. Thus, OSP can reasonably be treated as a combinatorial optimization problem and solved using the TSP approach together with the following assumptions:

1. Every operation is traversed once and only once in the machining cycle.
2. Machining states to be visited by the machined part are defined as the corresponding cities to be visited by the travel salesman.
3. Any change of machine or tool or setup or any combination of these actions establishes a new machining state.
4. The total cost incurred from start of operation i to start of operation $i+1$ corresponds to the distance between the cities.
5. A dummy operation transforms the goal state back to the initial state.

In machining operations, the part (raw material as an initial state) visits the machining states, and its destination is the goal state of the part as specified in the part design. The main objective in this formulation is reflected in the fourth condition where the cost of machining and changing machine, tool and setup are incorporated. At the same time, it is important to note some of the differences between the two problem types that hinder direct application of TSP model for operation sequencing. For the travel salesman, there are only two constraints i.e., each city is visited only once, and the length of the tour is minimized.

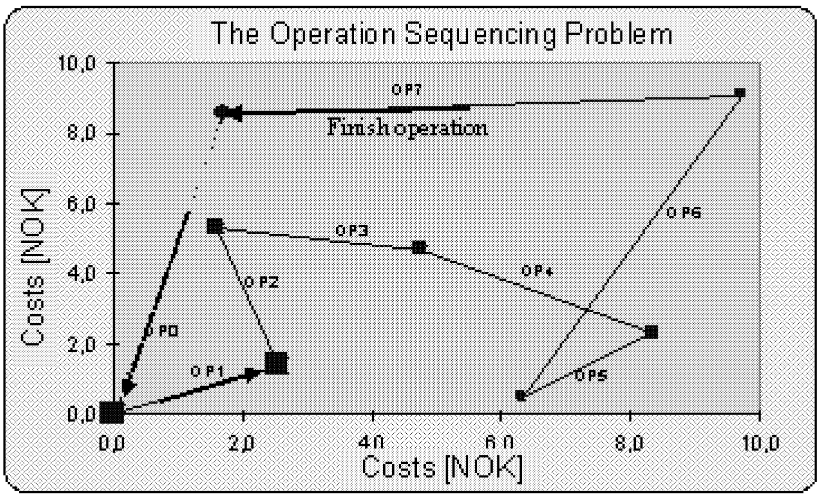


Figure 4.3: Model of OSP using the TSP approach

However, operation sequencing problem has many constraints that are even more complex. Primarily, the "distance" between the states is not often fixed. The cost incurred while changing the state of the combination (machine tool change, cutting tool change or setup change) plus machining cost of the given feature can vary even between two fixed machining states. Secondly, the travel salesman completes his tour by visiting all cities in the problem set and returns to his initial point, while the state of the part upon end of machining can be very different from its initial state. The geometrical and technological specifications of the part produced is obviously different from the raw material. To overcome this difference, a dummy operation, an operation with *zero cost* (time), is introduced that can close the loop. If the above-mentioned and other practical conditions are considered and suitable representations are devised, GAs can provide a valid option to solve this problem.

Problem formulation and representation

The major step in formulating an OSP for GA solution is to represent or encode the parameters of the problem in strings that are problem dependent. Two general categories of coding techniques are often implemented: *binary coding* and *permutation (real-valued) coding*. While binary coding is often used for function optimization problem, permutation coding is usually used for combinatorial optimization problems such as scheduling and the TSP. This is because permutation strings of a set of numerals are more natural to represent these and other manufacturing problems than the binary strings.

Figure 4.4 illustrates the two coding techniques – binary and real-valued. Strings consisting of the coded binary or numeral elements are referred to as *genotype*, and the solutions decoded from those strings are called *phenotype*. GAs search in the genotype world and the optimized solutions are obtained after decoding to the phenotype world. The binary strings of the genotype world, consisting of “0” and “1” are often decoded to the parameter value in integer, real number, and so on in the phenotype. On the other hand, permutation strings consist of numerals “1” to “n”. For a part having “n” distinct machining features, a string composed of “n” segments covering all the solution space of the part defines an n-operation problem for machining.

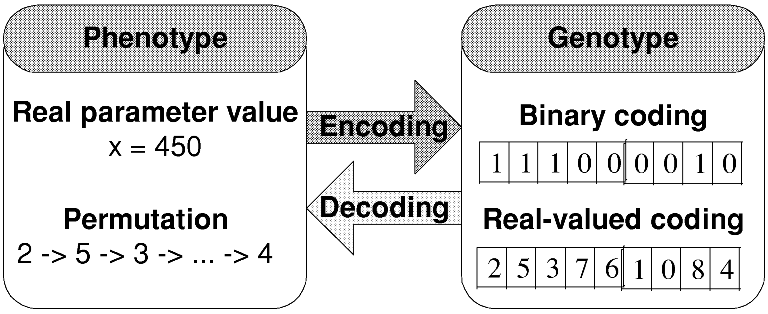


Figure 4.4: Encoding techniques for GA representation

Each operation set is accordingly assigned an ordinal number of “1” to “n” in the genotype world where each number corresponds to an operation in the sequencing problem. A population of an operation sequence for a part having seven distinct features is thus represented in a form of (2, 5, 3, 7, 6, 1, 4) to imply that feature number 2 is first machined and followed by feature number 5 and so on. The dummy operation is appended on this sequence. Thus, operations are processed according to their order in the string. In GA terms, the entire sequence forms the *chromosome* where a single operation (machining state) in the chromosome represents the *gene*. The number of the genes or individuals in the population represents the search points in the search space (Figure 4.5).

Putting all together, the entire population consisting of the sequence of operations represents the total operations done to get the finished part. A sequence that optimizes the total cost represents the solution of the problem.

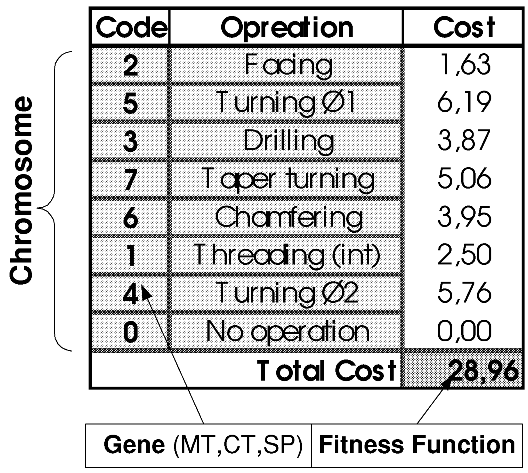


Figure 4.5: Representation example of operation sets as a chromosome

Generation of initial population

The initial population is composed of certain number of genetic chromosomes, in which each element or operation state represents an operation. The initial population of sequences can be generated randomly from the feasible solution set. The feasible solution set itself is a feasible sequence but not necessarily optimal. This feasible sequence is deemed to be one that does not violate any of the feasibility constraints. These constraints are processed sequentially to generate the *precedence relation* (PR) matrix. An initial PR matrix that does not violate the physical constraints of the machining environment is first established in a square matrix form. The sequence of operations in this matrix can be read from top to bottom (on vertical axis) and from left to right (on horizontal axis). The algorithm to generate this initial population looks as follows:

1. Look for an operation having a column sum of zero i.e. there is no predecessor (several options can exist).
2. Select an operation at random among those having no predecessors and append it to the end of the sequence.
3. Remove the constraint of this operation by deleting the row corresponding to this operation from the PR matrix and go to step 1 until no operation with column sum = 0 remains.
4. Update the column sum and go to step 1.

For example, Table 4.1 illustrates the above algorithms for a seven-operation task in a PR matrix form. In this matrix, the cell value (4,6) = 1 implies that operation 4 is constrained by operation 6, i.e., operation 6 has to be executed first.

By implementing this procedure for the above example, we get operation 2 preceding all operations, and thus operation 2 can be taken as the first operation. If the constraint of this operation is removed from the table, we find operation 5 and then operation 3 to be the next possible operations consecutively. At the third stage, i.e. after removing the constraints of operation 5 and then that of operation 3, the updated PR matrix looks as shown in Table 4.1 (b). It is possible to observe in this table that two or more operations can be potential candidates for the next operation where a random selection is often implemented.

Table 4.1: Application principles of PR matrix

Operations, y-axis ↑ \ Operations, x-axis →	1	2	3	4	5	6	7
1	x			1			
2	1	x			1	1	1
3	1		x			1	1
4				x			
5			1		x	1	1
6	1			1		x	
7						1	x
Sum1	3	0	1	2	1	4	3

(a)

Operations, y-axis ↑ \ Operations, x-axis →	1	2	3	4	5	6	7
1	x			1			
2	1	x			1	1	1
3	1		x			1	1
4				x			
5			1		x	1	1
6	1			1		x	
7						1	x
Sum2	2	-	1	2	0	3	2
Sum3	2	-	0	2	-	2	1

(b)

Genetic operators

After implementation of elitism and roulette wheel reproduction strategies, the two alternative sequences are mate to produce new offspring that represent the genetic structure. The advanced crossover operator PMX has been adopted because this operator is often recommended for combinatorial optimization problems (Goldberg, 1989).

To solve this particular problem, three *mutation operators*: machine tool, cutting tool and setup mutations are devised. Development of these operators is found necessary due to existence of alternative machines, tools and setups that can generate alternative solutions for the same operation. For example, the algorithm for machine mutation can be outlined as follows:

1. Select an operation (a position in the string) randomly and determine if machine mutation is necessary using mutation rate of P_m^m.
2. Choose a machine tool randomly from all the alternatives that can replace the current assigned machine.
3. Identify all other operations in the string that have the same machine as the current assigned machine.
4. Assign the selected alternative machine as current machine for all those operations.

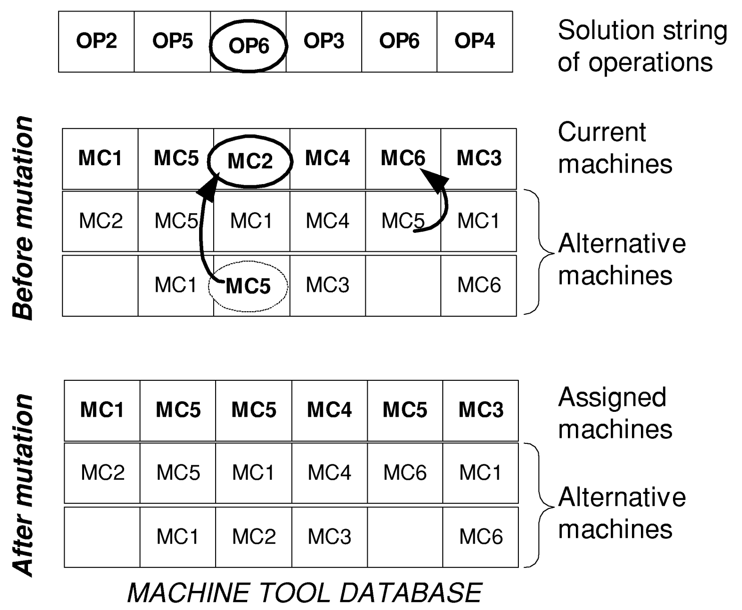


Figure 4.6: Working principle of machine mutation

It is also necessary to use certain heuristic rules in implementing this algorithm. For instance, as cost of changing a machine is higher than that of changing a tool, machine mutation is first exhaustively applied before tool mutation. Similarly, tool mutation is implemented before setup mutation. Figure 4.6 illustrates how the proposed machine mutation works. The implementation mechanisms of both tool and setup mutation are similar to that of machine mutation as given by the above algorithm.

The fitness function

Generating feasible sequences that do not violate the constraints is the primary step to apply an optimization algorithm to operation sequencing. Both geometric and technological constraints require that certain operations be performed before or after other operations. For instance, rough machining goes before fine finishes; first drill, then tape a thread; first bore, then ream; etc. Quantitatively, minimization of the total cost (C_p) that includes the machining cost (C_m) and other costs (C_{jkl}) due to the necessary changes of machine tool (MT), cutting tool (CT) and setup (SP) can be formulated as follows:

$$Min \quad (C_p) \;=\; (\sum_i C_{mi} + \sum_{jkl} C_{jkl}(MT, CT, SP)) \qquad (4.2)$$

where

i = 1 – n, for n number of features or operations
j = 0 – q, for q alternative machines (j = 0 for no machine change)
k = 0 – r, for r alternative tools (k = 0 for no tool change)
l = 0 – s, for s alternative setups (l = 0 for no setup change)

Among others, the following conditions should be fulfilled for the optimization:

1. The natural precedence of operations is not violated.
2. The number of machines, tools and setups are minimized
3. The allowable cutting speed, depth of cut, cutting force, etc. are not violated
4. All other physical constraints are not violated

According to this model, the total production cost of all sequences is to be optimized with the transition of operation states reflected in changes done on machine, tool or setup including other cutting parameters. For example, the following are possible transition one can encounter:

- (MT_1, CT_1, SP_1) ➔ (MT_1, CT_2, SP_1) – change of cutting tool for machining on the same machine and using the same setup (j = l = 0)
- (MT_1, CT_2, SP_1) ➔ (MT_2, CT_3, SP_2) – change of machine with corresponding change of tool and setup

For a total number of q, r and s alternative machines, tools and setups respectively, it is possible to observe the computational burden and the necessity to utilize the global and parallel search and optimization capacity of GAs.

Sequencing constraints

Knowing the constraints that should not be violated is very important for the evaluation of the optimal results. From machining perspective, we can roughly categorize the sequencing constraints into two: *dynamic* and *static* sequencing constraints. Dynamic constraints are those constraints varying with time or some other variable within the machining environment. This category includes constraints on cutting speed, force or power, depth of cut, feed, etc. these constraints depend on, for example, the raw material property, the cutting tool type, the coolant used and other factors in the machining environment. How these constraints are formulated for a particular face milling operation has been elaborated in Chapter 5. The static constraints, on the other hand, refer to those constraints that are not affected by the above parameters. Most of the static constraints are related to the feature being machined and their representation as equality or inequality form is not straightforward.

The following are examples of static constraints that are important to consider in operation sequencing:

- Location reference
- Accessibility
- Non-destruction
- Strict operation precedence
- Geometric tolerance

The location constraint involves examination of reference faces that can be used to locate and fixture the part while machining each feature. This reference identifies the necessity of machining the locating surface before the associated feature. At the same time, machining one feature first may cause problem to fixture the other feature.

To be machined a feature must be accessible. The accessibility constraint thus evaluates each feature's accessibility based on the feature type and its location relative to other features. This is particularly the case when machining secondary features such as threads and grooves on primary features like diameters, tapers and flat surfaces. As a result, it makes no sense to make an external thread before the correct diameter is formed or to make an internal thread before the appropriate hole is drilled.

By considering the non-destructive constraint, it is possible to ensure that a subsequent operation does not destroy the properties of the features that are machined in the prior operations. The natural precedence strictly determined by the feature type and properties is also equally important. This constraint is of course considered in almost all operation sequencing optimization cases. For example, first rough then finish cut, first bore (drill) then ream, first mill then grind, etc. are some such strict precedence constraints.

4.5. Chapter Summary

In this chapter, a methodology has been illustrated for the application of genetic algorithms to optimization of operation sequencing problem. This problem for machining operations is today one of the bottlenecks in the effort to automate the process planning function and integrate design and manufacturing systems. The operation sequencing problem is first defined as a combinatorial problem involving the combination of several entities such as machines, tools, setups and machine operators. Based on certain assumptions, the travel salesman problem approach has been adopted to formulate the operation sequencing problem that optimizes the total cost of production including machining cost and the cost of changing machine, tool and setup.

Since problem formulation is an extremely important part of problem solving, different mechanisms have been devised to represent the optimization objective, to define the constraints and other genetic operators from genetic algorithm application perspective. Combined with the multi-objective optimization of particular operations, as demonstrated in the next chapter, the developed and illustrated methodology to operation sequencing significantly contributes in the advancement of manufacturing system integration and optimization.

CHAPTER 5

GENETIC ALGORITHMS FOR OPTIMIZATION OF ECONOMICS OF METAL CUTTING

5.1. Introduction

Economic consideration is important in design and manufacture of components primarily because metal cutting is often a wasteful operation involving the removal of large quantities of material. Although there is no reliable figure to support this contention, it is possible to mention that only about 70% of the raw material purchased is converted to a finished product. Moreover, there are generally more than one alternative approaches to produce a part each alternative having an associated cost, degree of productivity and part quality. Accordingly, finding an efficient methodology that can help to identify an optimum approach to produce the part is necessary. This involves identifying the combination of best machines, tools, setups and cutting conditions for each approach and determining the best process(s). The machine tool operator cannot easily take care of all variables simultaneously to reach at optimum conditions of machining.

Machining process optimization appears in two basic forms:

1. Determination of a combination of optimum operation parameters and
2. Utilization of optimum resources

The most common method of determining operation parameters such as the cutting speed, feed rate and depth of cut is use of current practices in the form of rules of thumb, handbooks and other published guides. The recommended operation parameters are often given in *machining handbooks* and user manuals. These recommendations provide a set of cutting conditions that are mostly conservative and apply only to a particular machining situation. Their scope is to define the feasible range of applications, and thus they do not indicate which value of each parameter gives the best combination with respect to the prevailing cutting condition.

With the advances in automation of process planning, economic analysis of machining processes appears today as an important process plan evaluation criterion. Many new concepts and optimization procedures are thus developed that helped the evaluation of process planning results through theoretical analysis of machining operation optimization. Those studies used two optimization criteria:

1. Minimum manufacturing costs and
2. Maximum production rates

The minimum manufacturing costs objective alone often leads to a poor productivity due to the longer production time than the optimum, while maximum production rate alone results in higher manufacturing costs. Because of this conflict, subsequent investigations have focused on maximum profit rate hoping that it yields compromise results (Saravanan *et al*., 2001). Unfortunately, the variables involved are not amenable for traditional optimization approaches. These early studies were also limited to problems without constraints and treated only *single-pass operations*, whereas *multi-pass operations* are often preferred from economic point of view (Gupta *et al*., 1994; Agapiou, 1992). Optimization of multi-pass operation problems involve determination of optimum cutting speed, feed rate, depth of cut and number of passes simultaneously for a given depth of cut.

As a part of the OSP treated in the previous chapter using combinatorial optimization approach, this chapter focuses on the optimization of a particular operation – *face milling*. Face milling represents one of the most complex conventional material removal processes. The parameters involved at the tool-workpiece interface do not allow effective application of analytical modeling approaches. Accordingly, the relation between the control variables and the performance of the process are often determined experimentally. Putting the existing empirical relations together, the optimization problem of face milling operation is formulated as a multi-objective optimization problem and its implementation is demonstrated using GeneHunter – a genetic algorithm tool.

5.2. Optimization Model for Economics of Machining

The steps involved in formulating an optimization model for economics of machining for genetic algorithm solution include:

1. Formulating the objective function(s)
2. Determining the control variables (chromosomes) for the optimization
3. Defining all constraints applicable to the machining environment
4. Minimizing/maximizing the objective functions subject to the constraints

The overall formulation requires specification of expressions that represent the economic and physical parameters of the machining process and the entire system involving machine, tool and workpiece. The mathematical models are normally obtained through tests, previous production runs or existing machining experiences.

5.2.1. Formulating the objective function

Economic analysis of machining normally involves the cost elements, the material removal rate and the tool life. Two main objectives are considered in this analysis:

1. *Minimizing the total machining cost*: this identifies the cutting conditions that best balance the metal removal rate and tool life for the lowest cost
2. *Achieving a maximum productivity in terms of the maximum possible material removal rate*: this objective identifies the cutting conditions that best balance the material removal rate and the tool life to produce the highest possible output

Combining these objectives into a SOP is often not possible due to their conflicting goals. Thus, a *multi-objective genetic algorithm* (MOGA) approach is proposed.

For most machining processes, the machining cost depends on the machining time. Both the machining cost and time can be represented in a similar form as a function of other machining parameters. Their representation however varies in many reported studies in the literature. Based on a production batch of Nb that can be produced with a single setup (setup time = T_s), the production process of a part on a machine can be classified into the following major time components:

1. Setup time for tools, jigs and fixtures ($t_s = T_s/Nb$)
2. Part loading and unloading time (t_l)
3. Machining time for the specific feature (t_m)
4. Tool change time (t_{ct})

Accordingly, the total time to produce the part (T_p) is:

$$T_p = t_s + t_l + t_m + t_{ct} \tag{5.1}$$

Including the cost of tools and the necessary jigs and fixtures, the objective function in terms of the production cost becomes as follows:

$$C_p = C_{jf} + C_o\left(t_s + t_l + t_m + t_{ct}\frac{t_m}{T}\right) + C_t\frac{t_m}{T} \tag{5.2}$$

where,

C_p = production cost (all costs in NOK)
C_{jf} = cost of jigs and fixtures
C_o = overhead cost (including machine and operator rates)
T = tool life [min]
C_t = tool cost

In this equation (Equation (5.2)), machining time (t_m) represents the major cost element. This is particularly the case for operations that have very small material removal rate such as in grinding operations and non-conventional machining methods like EDM. This machining time, the actual time during which the cutting tool is actively removing material, can be expressed as a function of other machining parameters like cutting speed and feed rate. For example, machining time is generally expressed by Equation (5.3) for a single cut turning, boring or drilling operation.

$$t_m = \frac{\pi D_0 L}{1000vf} \tag{5.3}$$

where,

D_0 = cutting diameter [mm]
L = cutting length [mm]
v = cutting speed [m/min]
f = feed rate [mm/rev]

Similarly, Equation 5.4 gives the general expression of machining time for milling operations.

$$t_m = \frac{L + \sqrt{w(D - w)}}{f} \tag{5.4}$$

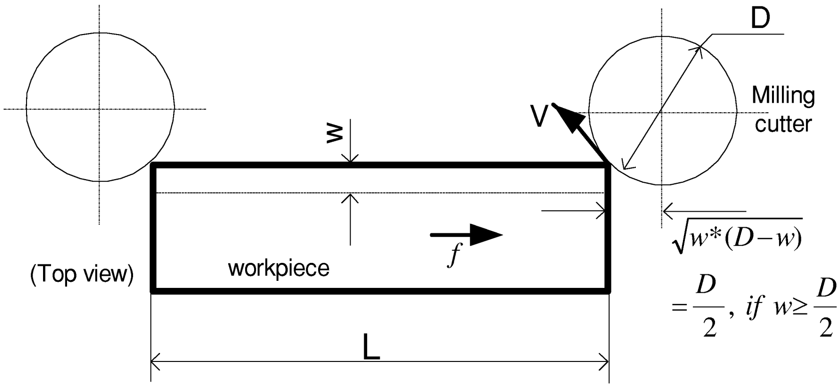


Figure 5.1: Designations in milling operation

where,

L = length of cut [mm]

w = width of cut (often referred to as feed engagement) [mm]

D = milling cutter diameter [mm]

f = feed speed of workpiece [mm/min]

The cutting tool of milling operations is in contact with the workpiece for only a portion of the machining time. Therefore, it is necessary to correct for the proportion Q of the machining time during which the cutting edge is engaged with the workpiece.

Including this correction factor into the above equations, it is possible to derive the following generalizing equation for the machining time:

$$t_m = \frac{K_m}{Qvf}, \text{ where } Q = \frac{1}{\pi}\arcsin\frac{w}{D} \tag{5.5}$$

K_m is an operation constant that depends on the geometry of the part or feature being machined. In general, it corresponds to the total machining length for the feature.

The tool life equation is the central focus of any material removal optimization problem. In cases where accurate tool life data is available, predictive tool life models and other machinability models can be developed. In this case, Taylor's tool life relation for minimum cost, according to *Kronenberg's extended cutting* *speed law*, is considered. This relation is expressed as follows (Kronenberg, 1966):

$$T = \frac{K_T}{v^{\frac{1}{n}} A^{\frac{z}{n}}} = \frac{K_T}{v^{\frac{1}{n}} f^{\frac{p}{n}} d^{\frac{q}{n}}} \tag{5.6}$$

where K_T is a constant as given in (5.7), and z, p and n are constants as described in Table 5.1.

Compared with the simple Taylor's life equation, which reflects the dominance of the cutting speed, this extended tool life equation accounts for the smaller but significant effect of feed and depth of cut. This dominance depends on the magnitude of the constants q and p whose value is always less than unity. The clear observation from Equation (5.6) is that the cutting speed must be decreased in order to increase either the feed rate or the depth of cut so that the tool life is not deteriorated. If the smaller effect of the depth of cut is neglected, Equation (5.6) can be simplified as follows:

$$T = \frac{K_T}{v^{\alpha} f^{\beta}}, \quad \text{where} \quad K_T = \frac{60 * (3.28 C_v)^n}{25.4^{\frac{p}{n}}} \quad \text{and} \quad \alpha = \frac{1}{n}, \; \beta = \frac{p}{n} \tag{5.7}$$

Table 5.1 shows the description of the constants used in the above equations and their recommended values.

Table 5.1: Recommended values of tool life constants[3]

Symbol	Description		Material: Steel	Material: Cast iron
z	Exponent of chip cross sectional area (A)		0.28	0.20
p	Exponent of feed rate		0.42	0.30
n	Tool life constant	Carbide tools	0.30	0.25
		HSS tools	0.15	0.25
C_v	Cutting speed constant for a cutting speed of 1m/min	Carbide tools	280	240
		HSS tools	85	50

[3] Courtesy: (Kronenberg, 1966)

Setting the relations for machining time and the tool life from Equation (5.5) and (5.6) into Equation (5.2), the expression for the production cost of a single-tool operation becomes as follows.

$$C_p = C_{jf} + C_o\left(t_s + t_l\right) + C_o \frac{K_m}{vf}\left(1 + t_{ct}\frac{v^{\alpha} f^{\beta}}{K_T}\right) + C_t \frac{K_m}{K_T} v^{\alpha-1} f^{\beta-1} \qquad (5.8)$$

This implies that machining cost is directly related to the cutting time that, in turn, depends on the cutting speed and the size of feed. According to this equation, large feeds and high cutting speeds reduce the machining costs, but at the expense of the tool life.

Therefore, the minimization of the total costs optimizes the production costs for a given setup. Equation (5.8) assumes a single-tool operation where tool-changing time is a function of the ratio of machining time to tool life. For multi-tool operations on a part or a feature, tool-changing time for each tool used must be considered regardless of this ratio (Tolouei-Rad and Bidehendi, 1997). As discussed in the previous chapter, this is treated as a separate operation with different machine, tool and setup combination.

The cutting speed and the area of chip removed (A) are the two main controllable parameters during milling operations. They can be controlled by setting the depth of cut (d), the width of cut (w) and the feed rate (f). The product of the area of the chip removed and the feed rate gives the volume of the material removed (defined as the *material removal rate*, MRR). This size highly affects the productivity of the machine and thus finding a means to maximize the rate of chip removal is of great practical importance in optimization of machining processes.

$$MRR = f\ A = f\ d\ w \qquad (5.9)$$

This expression shows that MRR of milling operations is directly proportional to the feed speed of the workpiece and the chip size. However, technical constraints impose limitations on both the feed speed and the size of the chip cross-section. The specifications of the machine, the tool and the workpiece are the sources of these restrictions on the optimization parameters.

To summarize, Equation (5.8) and (5.9) define the optimization objectives (fitness functions) that balance the material removal rate and the tool life for low cost production and high productivity respectively. The former, as the main objective function, is minimized simultaneous with the maximization of the latter (the sub-goal) using a multi-objective optimization approach.

5.2.2. Defining machining constraints

The optimization constraints for machining include the physical constraints imposed by the machine-tool-workpiece system and the cutting process as well as the geometric constraints imposed by the natural requirements for operation precedence. In the previous chapter, these constraints have been categorized as dynamic and static constraints. The physical constraints should be considered in the optimization formulation based on experience from the machining environment.

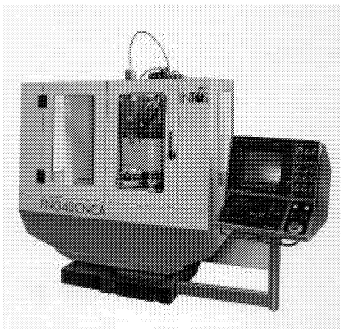
Table 5.2 gives the formulation of the optimization objectives together with definition of the ranges of the control variables and some of the constraints.

Table 5.2: Ranges of variable values and constraints

Variables	**Cutting speed and feed**	**v, f**
Objective	Minimize machining cost	$Min.\ f_{\cos t}\ (v, f)$
	Maximize productivity	$Max.\ f_{MRR}\ (v, f)$
Range of variables	Cutting speed limits	$v \leq v_{\max} = \Pi DN_{\max} * 10^{-3}$ $v \geq v_{\min} = \Pi DN_{\min} * 10^{-3}$ D [mm], N [s^{-1}]
	Feed speed limits	$f_{\min} \leq f \leq f_{\max}$
Constraints	Depth of cut	$d \leq d_{max}$
	Tool life	$T \geq n_p t_m$, for n_p products
	Machine power limits	$P = P_{max}$
	Cutting force limits	$F \leq F_{max}$

Table 5.3: Technical specifications of a milling machine (INTOS)

Parameter	Unit	Value
Range of speeds (N)	rpm	63 - 3150
Range of feeds (Axis X and Y) (f)	mm/min	12 - 3000
Motor output power at 1400 rpm	kW	4
Maximum motor power, P_{emax}	kW	8
Max allowable torque on spindle (M_{max})	Nm	500
Longitudinal feed of table (axis X)	mm	600
Horizontal feed of milling head (axis Y)	mm	400
Vertical feed of table (axis Z)	mm	400



Using the above-derived relations, formulating an optimization model for a face milling operation requires knowing values of the constraints that are specific to a given machine. Particularly, the maximum and/or minimum limits for cutting speed, feed rate, power and torque as specified by the machine manufacturer should be known. In this model, data for a multipurpose milling machine from INTOS is used, whose limit values are given in Table 5.3.

Cutting speed and feed are used as the main control variables in modeling the optimization problem. In GA terminology, these variables are defined as chromosomes. The range of these chromosomes is practically constrained by the tool life, surface finish requirements, depth of cut and the maximum cutting force allowed. For best economy, parameters that enable the maximum power utilization are often selected. The power used depends on the cutting conditions and should not exceed the maximum available power of the machine tool. For the given maximum allowable torque on spindle (Table 5.3), the constraints on the cutting power and force can be expressed as in Equation (5.10) and (5.11) respectively.

$$P = \frac{p_s * MRR}{60} \leq P_{\max} = \frac{2*\pi}{60} * M_{\max} * N \tag{5.10}$$

$$F_c = \frac{P}{v} \leq F_{c\max} = \frac{2M_{\max}}{D} \tag{5.11}$$

where p_s [W.s/mm^3] is a material dependent specific energy, MRR [mm^3/min] is the material removal rate, F_c and $F_{c_{max}}$ [N] are the actual and the maximum cutting force, P and P_{max} [W] are the actual and the maximum allowed power respectively and N [rpm] is spindle speed of the milling cutter.

The highest practically possible depth of cut is often used, mainly in rough machining, with favorable compromise between the tool life and the material removal rate. Studies show that a 50% increase in the depth of cut produces only 15% reduction in tool life when the depth of cut exceeds ten times the feed rate (Stephenson and Agapiou, 1997). The surface finish level of milling operation depends on the feed rate. Surface finish as an optimization constraint is important only for finish operations where a specific degree of surface quality is required. For roughing operations, this constraint can be neglected.

5.3. Problem Definition and Implementation in GeneHunter

The genetic optimization model is then established in GeneHunter, a tool that solves problems based on the genetic theory of evolution. GeneHunter is a genetic algorithm tool that mainly works in Microsoft Excel environment as an Add-in element. After formulating the problem with the fitness functions, the chromosomes and constraints appropriately defined in the Excel spreadsheet, GenHunter is then activated to work on the evolution process. Given the necessary information including specification of genetic operators, the system first creates a population of possible solutions to the problem.

5.3.1. Defining the fitness function

As shown in Figure 5.2, the main parameter definitions involved are the following:

1. Location of the *fitness function* cell
2. Objective of the optimization (max, min or finding a given value)
3. Location of the cells containing chromosomes
4. Chromosome type and range of their search space
5. Other constraints, sub-goals and genetic operators

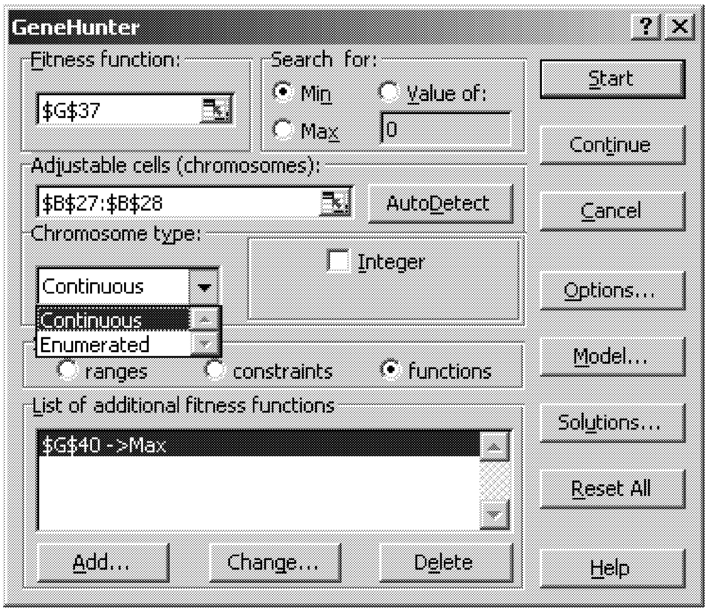


Figure 5.2: Parameter definitions in GeneHunter

The fitness function tells the location of the cell that contains the formula for the measure of goodness. For rather complex problems that can be formulated mathematically, the fitness relations can be created as *Visual Basic* functions. In cases where there is no appropriate mathematical formula to describe the problem, the fitness function can be created in neural network structure and integrated to the genetic algorithm tool.

5.3.2. Defining the chromosomes

To perform the genetic evolution, the genetic algorithm tool needs information about the *chromosomes* the variables whose values are adjusted in the process of solving the problem. The value of the chromosomes is related in some way to the fitness function. Two types of chromosomes can be identified: *continuous* and *enumerated*. Though many engineering problems are characterized as discrete optimization problems with an integral search space, the chromosomes of this case are defined as continuous type because selection of continuous cutting speeds and feed rate values are possible. On the other hand, enumerated chromosomes are used when the problem involves finding an optimal combination of tasks, resources, duties etc. where only integer values should be used as adjustable variables such as in combinatorial optimization problems.

5.3.3. Specifying the constraints

Some constraints of constrained optimization problems are often defined in such a way that they must be satisfied by the solutions. These categories of constraints are referred to as *hard constraints*. As shown in Figure 5.3, GeneHunter provides an easy way to define the range of the search space for these hard constraints and the restrictions on other sub-goals that should be fulfilled simultaneously.

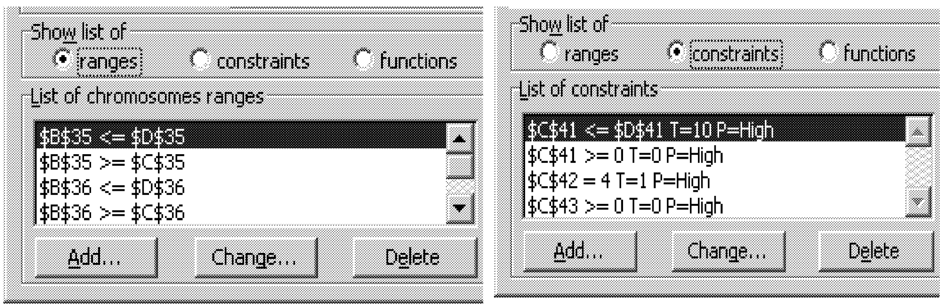


Figure 5.3: Definition of ranges and constraints in GeneHunter

The system attempts to find solutions that meet these constraints and the sub-goals referred as *soft constraints* while optimizing the main fitness function. Accordingly, multi-objective optimization tasks can be easily defined and solved using either intervals or functions.

Compared with traditional optimization approaches, the advantage of genetic algorithms can be clearly seen here that

- Complex problems having several variables can be solved with less difficulty. This is hardly possible in traditional optimization techniques where problems more than two variables often encounter complex computation.
- Multi-objective variables with as many objectives as necessary can be treated simultaneously.

5.3.4. Criteria to terminate genetic evolution

Some optimization problems can be solved within a short time. Unfortunately, genetic optimization is often used to solve complex engineering problems that demand long computational time. In certain cases, the quality of the optimal result depends on the amount of time available for evolution. Hence, method of terminating the evolution is among the important genetic parameters to be defined in GA application. Depending on the problem complexity, one of the following three methods can stop the evolution: time elapsed for the whole evolution, specific limit for the number of generations or total number of best generations unchanged.

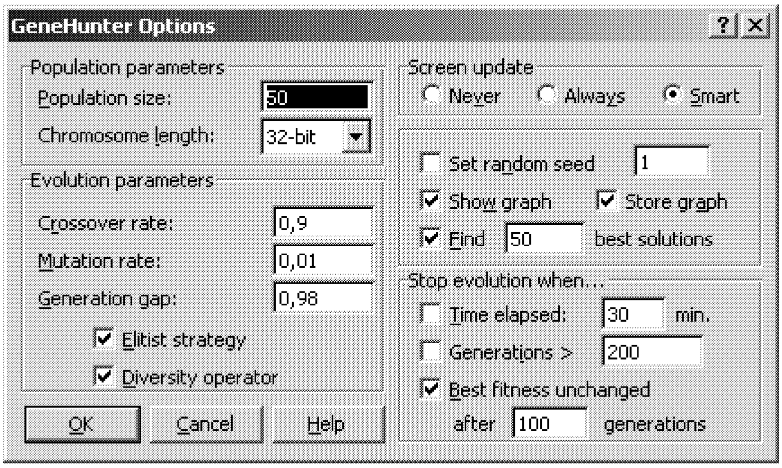


Figure 5.4: Defining genetic operators and stopping criteria

5.4. Discussion on Optimization Results

5.4.1. The population size

The population size is the number of individuals in the genetic breeding pool. In principle, a sufficiently large population size is favored to represent adequate members of the solution sets. However, a large population size leads to larger computation costs in terms of memory requirement and evaluation time. For too many individuals in the population, a good solution takes far too long to find because the fitness function must be calculated for every individual in every generation.

Too low population size, on the other hand, does not allow enough individuals to be involved in solving the problem. Similar to the behavior of traditional search techniques, the system can also be trapped at local optima. Generally, the population size depends on the number of variables involved in the chromosome and thus, the appropriate population size for a particular problem should be experimentally decided.

Figure 5.5 shows the variation of the population size against time of evolution. All the population sizes except the first (pop_size = 10) gave the optimum value. Considering the computation time required, however, a population size of 50 – 80 has been selected as a reasonable size for this problem.

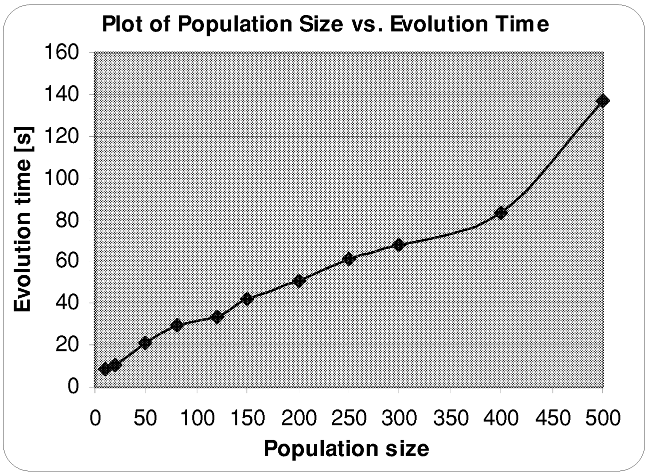


Figure 5.5: Plot of population size vs. evolution time

5.4.2. The initial population

The quality of the final solution of traditional methods is very dependent upon the position of the starting point of optimization in the search space. This is because the choice of the initial population plays a significant role especially for problems with a large number of local optima. As depicted in Figure 5.6, genetic algorithms do not suffer as much from this drawback.

Table 5.4 shows a comparison of different initial chromosome values. For the three cases considered i.e. two cases outside the range of the chromosomes and one within the range, the final optimum values justify that genetic algorithm based optimization does not depend on the initial values.

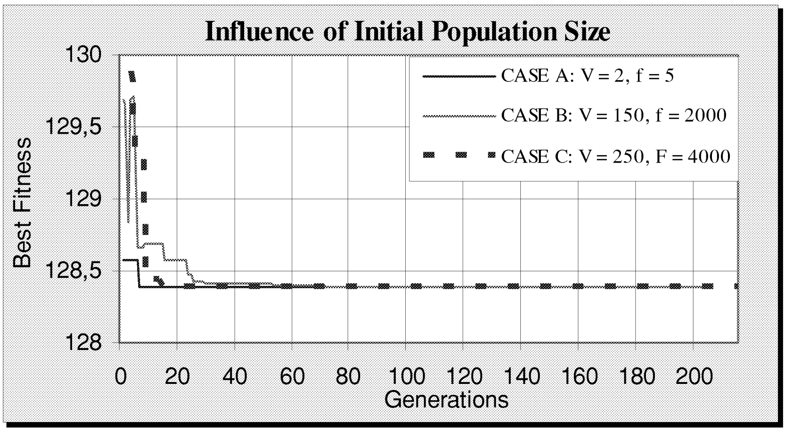


Figure 5.6: Plot of influence of initial population size

Table 5.4: Analysis of the influence of initial population

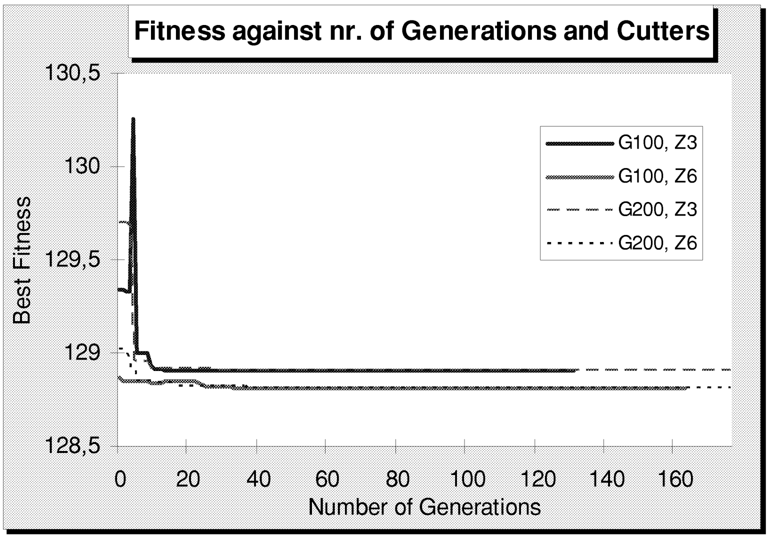
	Case A	Case B	Case C
v [m/min], 3,96 ≤ v ≤ 198,00	2,00	150,00	250,00
f [mm/min], 12 ≤ f ≤ 3000	5,00	2000,00	4000,00
Initial cost [NOK]	1655,00	129,38	140,93
Initial MRR [mm^3/min]	200,00	80000,00	160000,00
Time of generation [s]	52	67	70
Total number of generations	175	204	216
Optimum cost [NOK]	128,39	128,39	128,39
Optimum MRR [mm^3/min]	93889,28	93889,28	93889,28

Though the generation time and the total number of generations run to get the optimum values seem to have a trend with the initial values, repeated experiments show no such trend. Because the evolution process does not always follow the same path, it is not possible to expect equal evolution time or number of generations even when the same initial values are evaluated several times. The important thing is that the path of search does not influence the result of the final goal.

5.4.3. Number of generations and elitism based evolution

As shown in Figure 5.7, using large number of generations for the best fitness as a stop criterion has no significant influence on the optimization result for *elitism-based evolution* (reproduction). This is because elitism strategy takes the elites into the next generation and thus leads the evolution to the optimum direction very quickly. In principle, using large number of generations for the best-fit values increases the accuracy of the result, but at the expense of evolution time.

In addition to the number of generations, Figure 5.7 shows the influence of number of milling cutters used for machining. The production cost using six cutters is lower than that of using three cutters. This is obvious because using multi-point cutters increases the tool life. At the same time, the productivity increases as the number of cutters on the same cutter size increases with respect to the material removed per unit time.



G = number of generations, z = number of cutter

Figure 5.7: Plot of fitness against number of generations

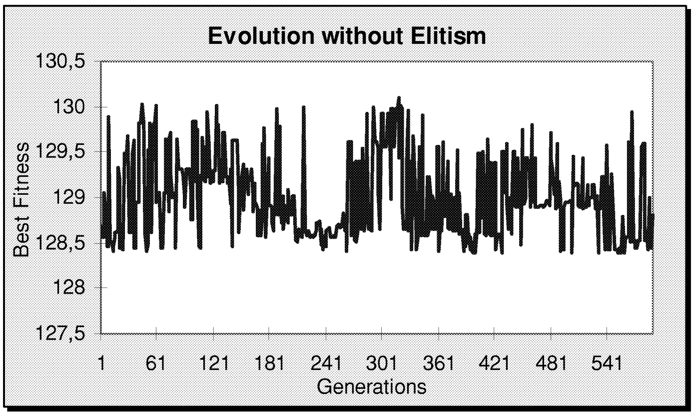


Figure 5.8: Evolution without elitism

On the other hand, evolution without elitism needs use of either *elapsed time* for evolution or specifying the total number of generations that satisfy the requirements. It is hardly possible to get a satisfactory number of unchanged best fitness values in the absence of elitism criterion because the evolution always starts from scratch and takes a very long time to get such a result. As depicted in Figure 5.8, the system did not manage even to show a certain trend towards optimum value even after running more than 500 generations that elapsed four minutes

5.4.4. Optimization results

From the optimization results shown in Table 5.5, it is possible to conclude that the results have an acceptable agreement both with manufacturers' specifications and practically accepted values. As given in Table 5.3, the motor output power at 1400 rpm is 4 kW, which of course does not necessarily correspond to an optimum operation point, but recommended for general-purpose operations. The optimization results show a power demand of 4,69 kW at a spindle speed of 1510 rpm.

Because the material removal rate in face milling is directly proportional to the feed speed of the table, the optimum cost of machining with an additional objective of maximized productivity tends towards the highest feed speed. This is contrary to the traditional understanding that cutting speed dominates the performance of machining operations. As expected, the optimum values tend towards operating at a cutting speed that allows utilization of the highest possible cutting power

Table 5.5: Results of the multi-objective optimization

COST ELEMENTS				TIME ELEMENTS		
Name	Designation	Value		Name	Designation	Value
Jigs and fixtures	Cjf [NOK]	50,00		Machining	tm [min]	0,217
Overhead costs	Co [NOK/min	15,00		Setup+loa	ts [min]	5,000
Tool	Ct [NOK]	50,00		Tool chang	tct [min]	3,000
Batch size	Nb				Total [min]	8,217
Chromosomes		Range of Values		Spindle Speed		
Parameter	Value	min	max	N[rpm]	Nmin[rpm]	Nmax[rpm]
V [m/min]	94,940	3,958	197,920	1511	63	3150
f [mm/min]	2347,232	12,000	3000,000			
fz [mm/tooth]	0,518					
Constraints and Operation Parameters			Limit	Production Costs [NOK]		
Name	Designation	Value	Values	Cost of jigs and fixtures		50,00
Torque on spindle	M [Nm]	29,67	500	Setup+load unload cost		75,00
Cutting force	Fc [N]	2,97		Mchining costs		3,26
Cutting power	P [kW]	4,69	6	Tool change costs		0,06
Tool life	T [min]	162,41		Tool costs		0,07
Depth of cut	d [mm]	4,0			**Total cost**	**128,39**
Width of cut	w [mm]	10,0			(Main fitness function)	
Tool diameter	D [mm]	20,0		**MRR [mm3/min]**		
Length of cut	L [mm]	500,0			Sub-goal	**93889,28**
Number of teeth	z	3,0				

5.5. Chapter Summary

As a continuation of the operation sequencing optimization problem of Section 4.4, an application methodology of MOGA has been developed and demonstrated in this chapter. Face milling was selected to demonstrate this problem because it is one of the conventional machining techniques known to have complex relationships among the variables at the machine-tool-workpiece interface. Due to the complexities of the material removal process, less in depth research has been reported for this machining operation compared with other conventional machining methods such as turning.

With respect to the number of control variables considered here, this example sounds simple for genetic algorithm application. However, traditional methods cannot be recommended for the problem due to the following main reasons. Primarily, this is not the end solution, but it is part and parcel of the rather complex combinatorial optimization problem of operation sequencing (process planning). For n set of operations to be sequenced, (n-1)!/2 alternative machining options have to be analyzed using this MOGA approach. Secondly, the conflicting objectives do not lend themselves for traditional methods. Thirdly, there are other variables such as the cutting temperature, tool geometry, etc. that are not considered in this example, but those parameters can highly influence the machining process depending on the workpiece material, tool material and machine loading conditions. Traditional methods fall short of solving such complex interaction of variables together with the conflicting multi-objectives.

The MOGA methodology implemented in this chapter is based on empirical relations put together from different sources. Those relations are mathematical models developed from experimental data with certain assumptions and simplifications. Today, the application of hybrid CI technology can create a good platform to overcome the error committed due to these assumptions and simplifications. Training neural networks on known input-output patterns from the machining environment and using the network structure for optimization of the parameters is a potential beneficial application area for hybrid CI systems.

CHAPTER 6

MODELING AND OPTIMIZATION OF EDM PROCESS USING HYBRID CI APPROACH

6.1. EDM Process Technology

Among known *non-conventional machining* methods that nowadays have a wide range of applications on the production floor, *electro-discharge machining* (EDM) is the most extensively used technique for die-making, precision machining and manufacturing of prototypes. EDM is an *electro thermal* process where the material removal mechanism is achieved by the erosive effects from repetitive electrical sparks generated between tool (the anode) and work material (the cathode) with constant electric field emerged in dielectric fluid.

Figure 6.1 shows a simplified diagram of the working principle of EDM process. In this process, the feed motion of the tool is controlled by a servo-controller that maintains a spark gap in the range of 0.025 – 0.05 mm (Bendict, 1987). The workpiece is placed in the dielectric fluid that circulates through a hole or holes in the tool electrode under pump pressure.

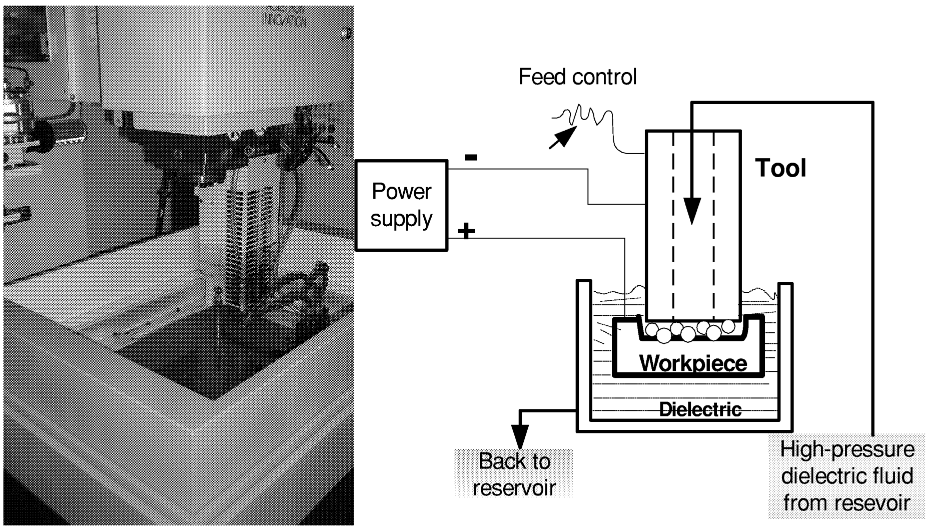


Figure 6.1: EDM machine and its working principle

Both the material removal from the workpiece and the wear of the tool electrode take place without any mechanical interaction between them. A spark is generated by a pulsating (DC) power supply is connected to the tool and the workpiece that becomes responsible for this effect by melting and vaporizing material from both the electrode and the workpiece.

Based on the type of the electrode and the dielectric used, EDM application is classified into two: *die-sinking* and *wire cutting*. Die-sinking has been the major application so far and much research and development has been concentrated to advance this technique. Particularly, the process has been made to perform better, accurate and dependable through the development of pulse generators, advanced control mechanisms and solid-state circuits that enabled control of the pulse Ontime and Offtime. This advance has to be accompanied by continuing investigations of the process and the development of new methods for economic production. This includes ensuring process stability and repeatability, and optimized process performance i.e., maximum possible material removal combined with best possible surface quality and minimized electrode wear.

EDM is also an alternative method of serial or batch production of *difficult-to-cut* parts when use of conventional techniques such as milling and grinding are not possible. Particularly in aerospace industries, the combined effect of poor machinability of super-alloys used in turbine components and the intricacy of the geometry makes EDM a natural choice of production method. However, the machining efficiency is much lower compared to conventional machining processes.

6.2. The Challenges of EDM Performance Improvement

Although up-to-date computer technologies have been used to control the EDM process, the variables for process efficiency and part accuracy are still obtained using empirical methods. The process is also one of the expertise-demanding processes in the industry, and the mechanism of metal erosion during sparking is still debatable due to the complex thermal conduction behaviors in the machining vicinity. This may explain the reason why it is hard to establish models that accurately correlate the process variables and performances.

This lack of accurate explanation of the thermal process at the tool tip contributes highly to the complexity of modeling the material removal process. Though some efforts are going on to be able to explain what takes place at the tool tip or the spark gap (Van Dijck and Dutrè, 1974; Dibitonto *et al.*, 1989), a complete model that gives the physical process could not yet been described in detail. Improving the MRR and surface quality are still the challenging problems that constrain the expanded application of the technology (Mohri *et al*., 1997; Kuneide *et al*. 1991).

Existing practices have difficulties to identify as which parameters need to be changed or how they should be modified for best process performance. The developed models have also helped only to provide approximate conclusions about the influence of different input variables on performance parameters. When new and advanced materials appear in the field, it has not been possible to use existing models and hence experimental investigations are always required, which manufacturers are sometimes hesitant to use statically designed experiments because of the purposefully taking the process out of or to the limits of control. Making frequent tests or many experimental runs is also not economically justified.

Accordingly, the task of this part of the thesis is twofold. Primarily, developing a methodology that optimizes the EDM process and other related non-conventional machining methods based on an appropriate process model is extremely important for both industry and research. Such a methodology is particularly necessary to integrate the system with other manufacturing systems and functions for online control and operation. Secondly, the overall behavior of the machining process for certain tool-workpiece combinations is not well defined. Thus, it is not possible today to find the best parameter recommendations for some newly developed materials. Graphite tool on nickel-base alloy represents one of such combinations. Thus, analyzing the behavior of this combination based on experimental dataset is one of the objectives of the thesis.

6.3. Description of Parameters

The developers of EDM machines normally suggest parameter settings for optimum performance of the machine with respect to speed, electrode wear and surface roughness (Aas *et al*., 2001). Since getting such results for materials of special interest is often difficult, the search for parameter combinations that perform better is still far from over. Even many of the suggested parameters are available only for tools made of steel. Determining optimum parameters for other materials is, thus based on trial and error.

Achieving a high machining productivity in terms of the *material removal rate* with a desired accuracy and *surface finish* are the most wanted performance parameters that must be optimized with respect to the input variables. Even a highly skilled operator can rarely achieve these required optimum results due to the large number of variables and the stochastic nature of the process. Particularly, determining the relation between the controllable input variables and the performance parameters using suitable mathematical models is not simple.

Many independent parameters influence the performance of EDM processes including Ontime, Offtime, peak current, voltage, compression, gain, dielectric fluid and electrode material. Table 6.1 shows the designations of some of the input variables. This is important because many EDM machine manufacturers designate some of the parameters very differently.

Table 6.1: Definition of input parameters

Parameter	Symbol	Unit	Definition
Ontime	T	μs	Duration of each spark
Offtime	P	μs	Pause time between two sparks
Peak current	I	A	Maximum current during spark
Voltage	U	V	Voltage between gap just before spark
Compression	COMP	mm	Distance between electrode and
Gain	GAIN	-	Servo sensitivity to changes in spark

In addition, the main performance parameters of interest while machining on EDM are defined as follows:

1. *Material removal rate, MRR [mm^3/min]*: MRR is defined as the ratio of the volume of material removed from the workpiece and the time required for the removal. Thus, MRR was calculated from measured depth of cut, measured time of machining and the contact area of the tool tip exposed to the erosion process. Maximum MRR is the optimum value searched.

2. *Surface roughness, SR [Ra]*: this is the measure of the surface quality on the machined part. This parameter can be measured using the average roughness (Ra) indicator tool. According to this indicator, the lower Ra value gives the better surface quality.

3. *Tool wear [%, mm]*: two parameters, relative wear (RWR) and corner wear (CWR) are used to measure the wear resistance of the tool. Relative wear or wear ratio gives the relative amount of tool lost compared with the volume of material removed. This was obtained by measuring the size of the tool before and after machining.

Regardless of the tool-workpiece combination, the general phenomenon of EDM process shows that the duration of the Ontime influences the MRR, the tool wear and the gap between the electrode and the workpiece. The longer the Ontime, the higher the wear of the electrode, the MRR drops and the roughness increases. Lower Ontime has the reverse effect on the machining process. Similarly, the higher voltage results in larger distance between the tool and the workpiece and improves the flushing condition as a result. A higher current intensity causes rougher workpiece surface. Ontime, Offtime and compression are the primary parameters to optimize the process. The Offtime stabilizes the erosion, but has no significant effect on the roughness. Longer duration of Offtime, however, reduces the material removal rate. Compression allows varying the distance of the working area. Reduced compression increases the gap, but EDM is less efficient.

In general, high MRR produces a very rough surface finish because of the molten and re-solidified surface structure. On the other hand, MRR and surface finish increase with increasing current density and decreasing frequency of spark. A better surface accuracy can be achieved only at a sacrifice of the machine productivity. Therefore, establishing a correlation between the input variables and the performance parameters has been a challenge in the past for which different techniques have been tried to develop models of the process.

6.4. Modeling EDM for Graphite Tool Material

6.4.1. Tool material (electrode) properties

Similar to conventional machining processes, an EDM tool wears under the machining process. Therefore, *wear resistance* is one of the primary requirements of the tool material. Two parameters are often used to measure the wear resistance of the electrode tool: *front wear* and *corner wear* (Figure 6.2).

The corner wear is often measured in terms of the radius. The relation between the front wear and the depth of material removed gives the *relative wear* of the tool or the wear ratio. Studies show that the wear ratio, ratio of the volume of metal lost from the tool to the material removal rate, depends on the tool-workpiece combination. For example, the wear ratio of brass tool with different workpiece materials looks as listed in Table 6.2.

Table 6.2: Examples of wear ratio values for brass tool

Tool material	Workpiece material	Wear ratio
Brass	Brass	0.5
Brass	Hardened carbon	1.0
Brass	Tungsten carbide	3.0

Compared with other materials such as copper and copper-tungsten electrodes, graphite suffers the highest wear on both front and corner. In addition, a tool material should have properties of good *electrical conductivity* and *machinability*. Graphite and copper-graphite are by far the most commonly used tool materials. The former is preferred due to its machinability. Adding copper to graphite produces *copper-graphite* that increases the conductivity of graphite (Boothroyd and Knight, 1989).

Good machinability makes graphite one of the most commonly used tool material for EDM. Nonetheless, insignificant information exists about the implementation technology of this material for different workpieces and cutting conditions. Because commercial EDM machines are optimized mainly for steel tools, industrial application of a new material often requires an experimental development of a completely new technology for the specific material. However, the time to make such tests in the production environment is limited, posing the demand for a mechanism that enables the modeling and optimization of parameters within the limits of technology and economy.

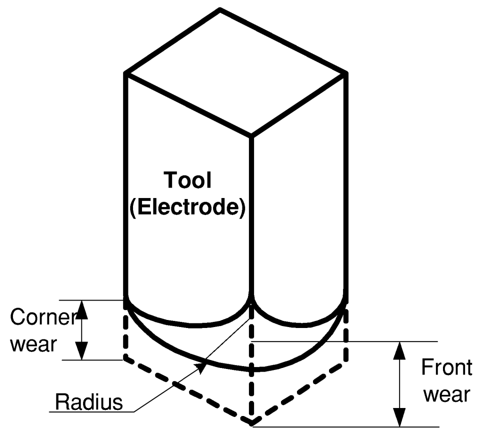


Figure 6.2: Designations of electrode wear

6.4.2. Description of experimental data

Table 6.3 shows the values of the input parameters for the test done on graphite electrode (size 2,9 x 9,8 mm^2) with nickel-base alloy workpiece using AGIE INNOVATION EDM machine. The variable combinations were generated using the AGIE TECTRON that generates recommended combinations for good performance of the process for steel material. Since there is no such data for graphite tool material on nickel-base alloy, the recommended combinations for steel were used. Most of the performance parameters are however not possible to measure directly. Appendix A shows the measured values and the calculated performance parameters. It is observable from the data that the tool wear values registered were very insignificant and investigating for these parameters for such a short time machining tests gives inconclusive results.

Table 6.3: Experimental dataset for graphite tool on nickel-base alloy

	Input parameters						Performance parameters			
Nr	T	P	I	U	COMP	GAIN	MRR	SR	CWR	RWR
1	100	10	21	100	42,3	15	400	8,87	0,10	0,00
2	56	10	21	100	42,3	15	380	7,87	0,15	-0,06
3	42	10	21	100	42,3	15	682	6,37	0,20	-0,03
4	32	10	21	100	42,3	15	1039	5,44	0,20	0,02
5	24	10	21	100	42,3	15	846	5,54	0,30	0,05
6	49	56	21	100	35,3	15	628	7,14	0,20	0,02
7	49	37	21	100	35,3	15	578	7,00	0,20	0,01
8	49	10	21	100	35,3	15	583	6,87	0,05	0,02
9	49	37	39	100	35,3	15	1345	8,06	0,20	-0,01
10	49	37	29	100	35,3	15	1018	7,48	0,10	0,04
11	49	37	21	100	35,3	15	487	7,88	0,15	-0,02
12	49	37	17	100	35,3	15	334	6,73	0,10	0,01
13	49	37	10	100	35,3	15	179	4,95	0,20	0,02
14	49	37	21	100	30,2	15	536	7,54	0,20	0,01
15	49	37	21	100	20,0	15	536	7,02	0,20	-0,01
16	49	37	21	100	35,3	20	594	6,51	0,18	0,02
17	49	37	21	100	35,3	15	526	6,93	0,20	0,02
18	49	37	21	100	35,3	12	503	6,55	0,15	0,02
19	49	37	21	100	35,3	10	484	7,29	0,10	0,00
20	49	37	21	80	35,3	15	604	6,58	0,20	0,01
21	49	37	21	60	35,3	15	501	6,42	0,20	0,03

6.5. The Hybrid CI Approach to Modeling and Optimization

Under the umbrella of hybrid CI approach, the integration of genetic algorithms and neural networks is seen as a fruitful area of research, where many new and exciting ways of merging the technologies are emerging. The background for this interest is that neural networks can find patterns in training sets of data, learn these patterns, and develop the ability to correctly classify new patterns or to make forecasting or prediction models. They excel at problems where modeling by mapping input patterns to output results is important and precise computational answers are not required.

In addition, genetic algorithms seek to solve optimization problems using evolution methods. In typical optimization problems, formulas or algorithms that combine a number of control variables are used to fully model processes. Then, the problem boils down to finding the values of the variables that optimize the model in some way. If the model is a formula and very few variables are involved, then the maximum or minimum value of the formula can be sought using conventional optimization approaches, which can optimize problems of this nature for fairly "well behaved" problems. These traditional methods tend to break down when the problem is not so well behaved.

Modeling EDM process for optimum operation represents a particular problem type in manufacturing environment where defining the optimization objective function using a smooth, continuous mathematical formula is not possible. By combining the capability of the two CI tools, a methodology is developed to solve both the modeling and optimization problem in a hybridized form. By using an input-output pattern of experimental data, a hybrid GA and neural network (HyGANN) approach is implemented for this methodology development. Figure 6.3 shows the structure of this hybrid system.

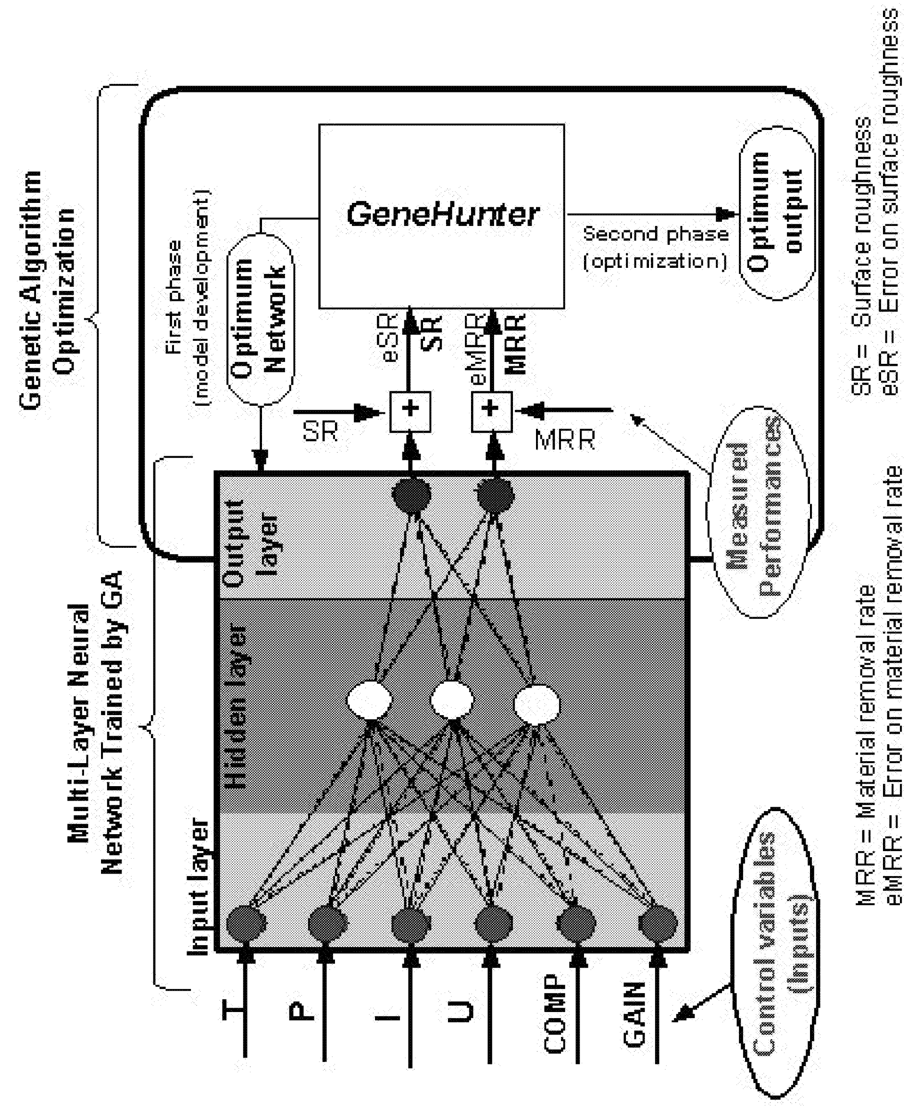


Figure 6.3: HyGANN system for EDM process modeling and optimization

In this hybrid system, the capability of neural networks to model and predict ill-structured data is exploited together with the power of GAs for optimization. The fundamental optimization problem for this hybrid system can be expressed as follows:

$$\text{Optimize } Y = f(X, W) \tag{6.1}$$

where, Y represents the [output] performance parameters such as the MRR that is to be maximized and the tool wear and the surface roughness that are to be minimized during the optimization; X is a vector of the input variables to the neural network and W is the weight matrix that is evaluated in the network training process. *f*(.) represents the model of the process that is to be built through neural network training. While training, the neural network adjusts the weights and establishes the correlation between the input variables and the output parameters. Then, new X values are encoded as chromosomes for genetic evolution using the weight matrix of the trained network to determine the optimum Y values.

To achieve the goal of the task, a two-phase hybrid system of genetic algorithm and neural network has been implemented. These two phases can be categorized as *modeling phase* and *optimization phase*.

6.5.1. Hybridization at the modeling phase

The first phase involves the establishment of the model using multi-layer feedforward neural network architecture. Instead of the standard backpropagation error minimization approach, GA is implemented to find the optimum values of the weights that minimize the error between the measured and the evaluated (network output) performance parameters. Therefore, genetic optimization establishes a strong intercommunication between the neural network pattern identification (modeling) and the genetic algorithm optimization tasks.

In this phase of hybridization, the weight matrices between input and hidden nodes; and hidden and output nodes were coded as *chromosomes*. Then, the following relation was used to combine the inputs of the network at the nodes of the hidden layer.

$$H_j = \sum_i v_{ij}.X_i \tag{6.2}$$

where,

H_j = the combined input to hidden node j from the nodes in the input layer

v_{ij} = the weight between input node i (i = 1, 2. .x) and hidden node j (j = 1, 2, .. h) for x nodes in the input layer and h nodes in the hidden layer respectively and

X_i = the input value at input node i.

The outputs of the hidden layer are again combined at the output nodes in a similar way.

$$O_k = \sum_j w_{jk}.Z_j \qquad (6.3)$$

where,

O_k = the combined input to the output node k from the nodes in the hidden layer

w_{jk} = the weight between node j of the hidden layer and the output node k (k = 1, 2, … z) for z number of nodes in the output layer

The outputs of both the hidden ($Z_j = f(H_j)$) and the output layer ($Y_k = f(O_k)$) can be calculated by an arbitrary transfer function. For the sake of an experiment, *sigmoid function* of the form given in Equation (6.4) has been adopted for both layers because of its well-known use as a transfer function for many applications.

$$Z_j = f(H_j) = \frac{c_1}{1+e^{-\sum_i v_{ij}.x_i}} \text{ and } Y_k = f(O_k) = \frac{c_2}{1+e^{-\sum_j w_{jk}.Z_j}} \qquad (6.4)$$

where, c_1 and c_2 are arbitrary constants.

Combining Equations (6.2) and (6.3), we get the following relation for the output of the network.

$$Y_k = f(O_k) = f(\sum_j w_{jk}.Z_j) = f(\sum_j w_{jk}.(\sum_i v_{ij}.X_i)) \qquad (6.5)$$

Clearly, for non-linear function *f(.),* Y_k is a no-linear function of the input vector X and the network weights W = (*v,w*), i.e., Y = f(X,W) as given in (6.1). Figure 6.4 shows the above representations and the working principle of the approach.

In application of commercial neural network products, this task of defining the correlation between the inputs and the outputs is left for the network itself. Considering the neural network as a "black box", our task could be determining how the output of the network can be extracted as a single measure of performance (fitness function) for the genetic algorithm part, where the major challenge of hybridizing the two systems lays.

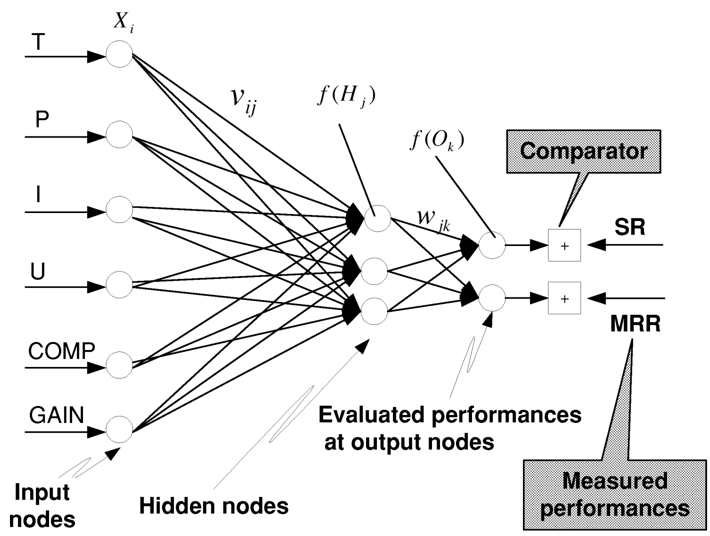


Figure 6.4: The structure of a genetic learning based neural network

At the end, the output of the network is compared with the measured performance of the process using a simple mean square error as follows.

$$E_k = \sqrt{\sum_{k=1}^{z} (Y_k - Q_k)^2} \tag{6.6}$$

where, E_k is the mean squared error between the output (Y_k) and the target performance (Q_k) of node k.

Thus, the optimization problem of this phase, for example for the material removal rate, can be expressed as follows:

$$\text{Minimize} \left(E_k = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2} \right)_{MRR} \tag{6.7}$$

Such that

1. $0 \le v_{ij} \le 1$ (Range of chromosomes for weights from input node i to hidden node j.
2. $0 \le w_{jk} \le 1$ (Range of chromosomes for weights from hidden node j to output node k.

3. $\sum_{i=1}^{6} v_{ij} = 1 \pm 0{,}1$ (Constraint on contribution of each input variable i to hidden node j.

4. $0 \leq f(H_j) \leq 1$ (Constraint on each hidden node's activation).

5. $0 \leq f(O_k) \leq 1$ (Constraint on each output node's activation).

6. Minimize $\left(E_k = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2} \right)_{SR}$ (Sub-goal of optimization).

Because of the squashing of the network output to an interval [0, 1], a linear transformation was used to scale the performance target values within a similar range [0, 1].

Further, to optimize the error in the neural network structure and define the correlation, GeneHunter was used with the following genetic parameter definitions.

Population size	= 80
Chromosome length	= 32 - bits
Crossover rate	= 0,90
Mutation rate	= 0,01
Elitism strategy	Yes, with generation gap of 0,98
Stopping criteria	200 generations with unchanged fitness

Knowing the relative importance of the input variables for each performance is one of the important goals in this study. For online control purposes, in particular, it is important to identify as which variable highly creates deviations on the performance of the process so that system parameter adjustments can be done as quickly as possible. To make this comparison, three *error optimization* cases were considered:

(a) Optimizing using the error between the output of the node for MRR and its target value with $E_{MRR} = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2}$ as a fitness function

(b) Optimizing using the error between the output node of surface roughness and its target value with $E_{SR} = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2}$ as a fitness function and

(c) Optimizing using a MOP approach with the error for MRR ($E_{MRR} = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2}$) as the main fitness function and that of the surface roughness ($E_{SR} = \sqrt{\sum_{k=1}^{21} (Y_k - Q_k)^2}$) as the sub-goal.

This *relative importance* concept has been used here to establish a simple measure of how significant each input variable is to predict the process using the model. For this objective, the range of the chromosomes (weights) was defined between 0 and 1 so that higher values are associated with more important variables. Further, the sum of the weights over all input variables was constrained to 1±0,1 so that the relative importance values may be thought as the percent contribution of each respective variable to the model performance. It does not mean however that a variable having 40% relative importance is twice as important as that having 20% relative importance. On the other hand, it may imply that it is possible to omit a variable if its relative contribution in defining the model structure tends to zero.

Table 6.4 shows the values of the weights resulted from these optimizations and Figure 6.5 shows the relative importance of the variables for the three cases. The rest of the results together with the network structure is given in Appendix B, C and D for the above three cases respectively.

Table 6.4: Weight values (v_{ij}) for the three cases of error optimization

(a) Weight values for material removal rate error optimization case							
	T	*P*	*I*	*U*	*COMP*	*GAIN*	*Sum*
Node 1	0,2084	0,0989	0,1276	0,1992	0,1634	0,1333	0,9307
Node 2	0,2021	0,1322	0,2519	0,0067	0,3171	3E-08	0,9100
Node 3	0,3382	0,092	0,1246	0,0419	0,1049	0,174	0,8756
Sum	0,7487	0,3231	0,504	0,2477	0,5855	0,3073	2,7163
(b) Weight values for roughness error optimization case							
	T	*P*	*I*	*U*	*COMP*	*GAIN*	*Sum*
Node 1	0,1472	0,1229	0,4164	0,0003	0,1298	0,1276	0,9442
Node 2	0,0742	0,1101	0,142	0,1104	0,3597	0,1939	0,9903
Node 3	0,1258	0,2155	0,1002	0,4225	0,0622	0,0385	0,9648
Sum	0,3472	0,4485	0,6587	0,5331	0,5517	0,3601	2,8993

(c) Weight values for multi-objective error optimization case							
	T	**P**	**I**	**U**	**COMP**	**GAIN**	**Sum**
Node 1	0,6077	0,0196	0,1038	0,0923	0,0077	0,0508	0,8818
Node 2	0,1132	0,0038	0,364	0,0135	0,0186	0,016	0,5292
Node 3	0,1454	0,0875	0,0152	0,0356	0,0008	0,213	0,4974
Sum	0,8662	0,1109	0,483	0,1414	0,027	0,2798	1,9084

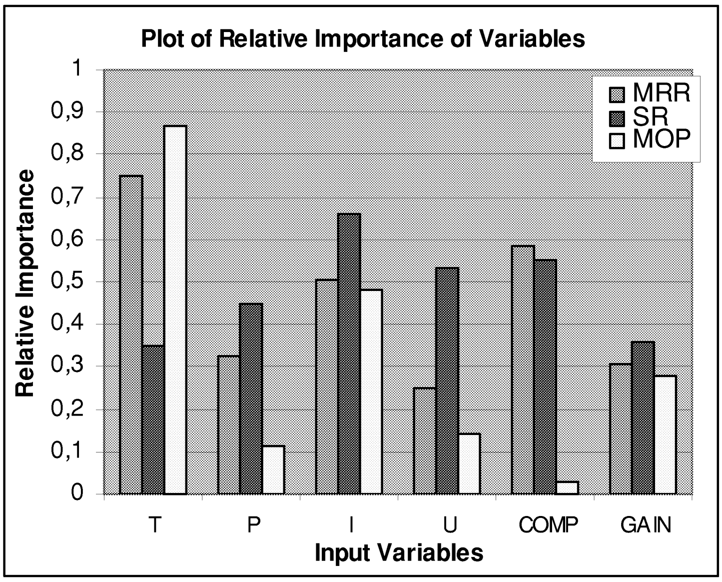


Figure 6.5: The relative importance of the variables for three test cases

The result for all cases shows an interesting relationship that the current intensity (I) has a relatively uniform high relative importance for the process performance. At the same time, the pulse Ontime (T) highly influences the optimization of the material removal rate as well as both performances simultaneously and the compression has the least relative importance for this simultaneous case.

The network error for all cases sounds reasonable to accept the network for further application. In the multi-objective case, for example, the network optimized error for MRR was 5,60 % while that of the surface roughness was 4,98 % (Appendix D).

6.5.2. Hybridization at the optimization phase

In the second phase, i.e. after defining the model, the input parameters to the neural network structure were coded as *chromosomes* for genetic evolution. Since the major part of the problem has been already solved in the previous phase, this is of course the characteristic of most neural network applications; the implementation phase in this part is straightforward. Figure 6.6 shows the general structure and principles used in this implementation.

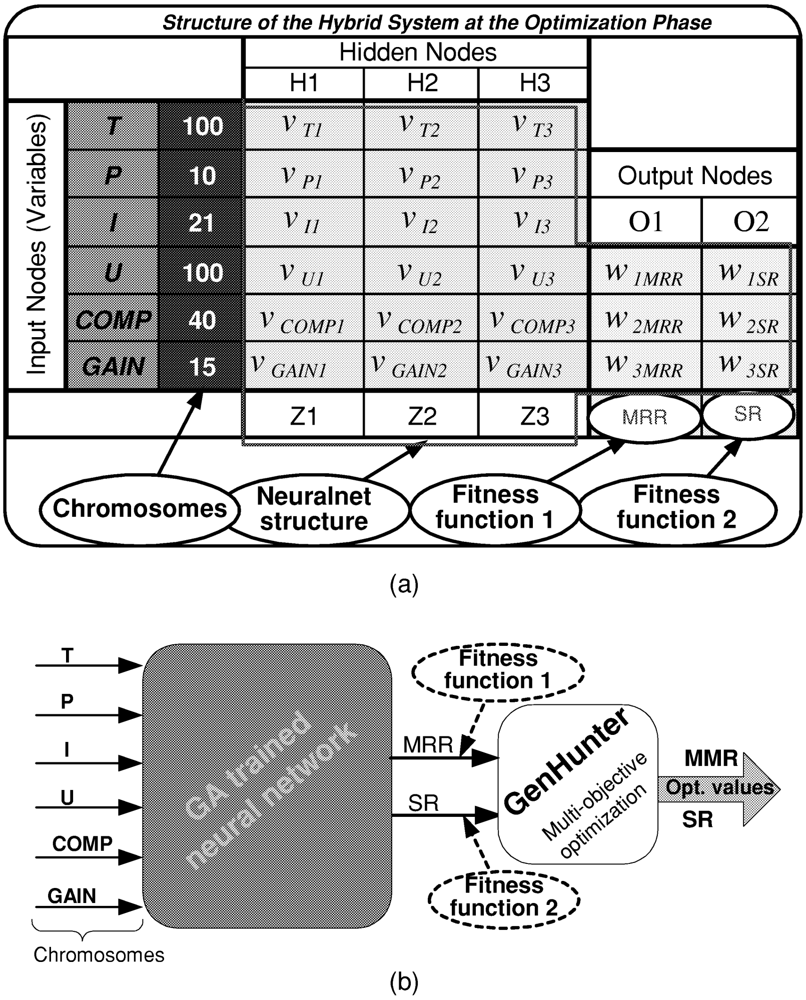


Figure 6.6: Structure of the hybrid system at the optimization phase

Table 6.5: Sample of optimization results

Variables	*T*	*P*	*I*	*U*	*COMP*	*GAIN*
Units	[μs]	[μs]	[A]	[V]	[mm]	-
Opt. values	25,49	13,73	9,80	33,33	37,76	28,04

Outputs	Units	Opt. Results	Target
MRR	[mm^3/min]	1038,58	Max
Roughness	[Ra]	6,97	Min

In this case, the structure of the neural network is seen as a "black box" (Figure 6.6 (b)) for the user whereas, contrary to use of other commercial neural networks for such hybridization, the genetic algorithm knows what is going on inside the structure and uses the structure in the evolution process very actively. This is because the outputs from the neural network are used as the *fitness functions* for the genetic algorithm based optimization.

The multi-objective optimization was then run using GeneHunter and the results after several test runs look as shown in Table 6.5. These results are of course only for demonstration of the methodology, and thus it is important to note that the numerical values need further verification. This includes the refinement of the network structure in terms of the number of hidden layers and the number of nodes in each layer as well as running the model with more experimental data.

6.6. Predicting the Model Performance Using Neural Networks

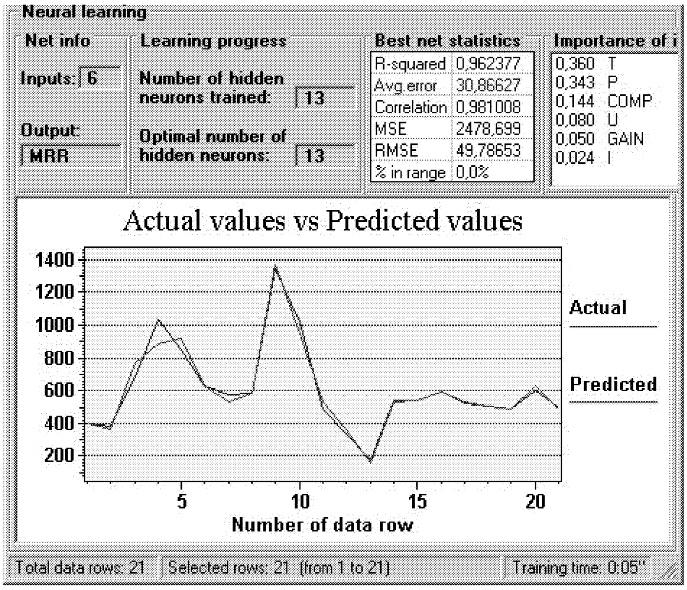
Beside the modeling and optimization problem, predicting the performance of EDM processes for some tool-workiece combinations is also very important. Among existing prediction tools, neural networks appear today as powerful tools to handle complex datasets. There are also many commercial, but mainly research-oriented, neural network tools that were developed to solve some of our problems. The disadvantage of such tools with respect to our hybridization goal is that their internal activities are often "closed" to the user since it is not possible to extract the internal structure of the created correlation. Thus, it is difficult to find the exact combination of contributions of each of the variables that creates the best predictions. This nature of neural networks makes the major part of the difficulty to merge the work of the networks with other AI tools.

6.6.1. Neural network and genetic algorithm based predicator

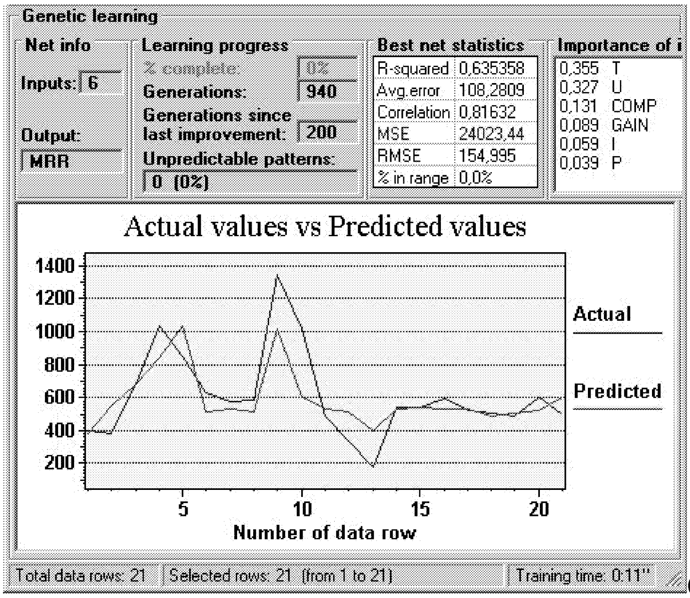
In cases where only analyzing performances is required, neural networks are often used as powerful tools to predict the performance of processes. To analyze our dataset for graphite tool on nickel-based alloy workpiece, NeuroShell® Predictor was implemented. Two alternative strategies of training input-output data patterns were implemented. The first uses standard *backpropagation learning* approach that dynamically grows the hidden neurons to build a model and to find optimum network structure for best predictions. In a way, this is a form of *multiple regression analysis* because a neural network with no or small number of hidden neurons behaves like or close to a regression analysis. On the other hand, using a high number of hidden neurons leads to a risk of over-training or *over-fitting* the model. With a possible maximum number of hidden neuron set to 20, 40 and 80, the network for this model shows that only 13 hidden neurons are necessary for best predictions (optimum results).

The second strategy employs genetic algorithm evolution combined with *statistical estimators* to develop the model. Statistical estimators such as *maximum correlation*, *minimum total error* and *root mean squared error* are included in the package. The parameter that is very important to determine while using this approach is the *number of generations* that the system runs with no improvement in the network's best performance. Since a low number of generations give premature predictions of the model, appropriate number should be experimentally determined based on the data size, problem complexity and other factors. To be able to make a rough comparison with the developed model using the hybrid system (Section 6.5), the same number of generations with unchanged fitness of 200 was used in this genetic learning process.

URN:NBN:no-2332



(a)

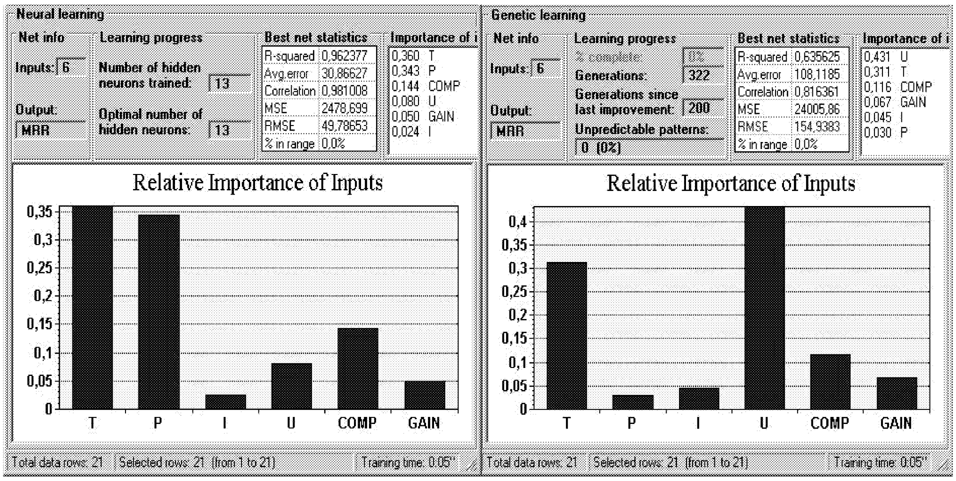


(b)

Figure 6.7: Actual vs. predicted values using neural (a) and genetic (b) learning strategies

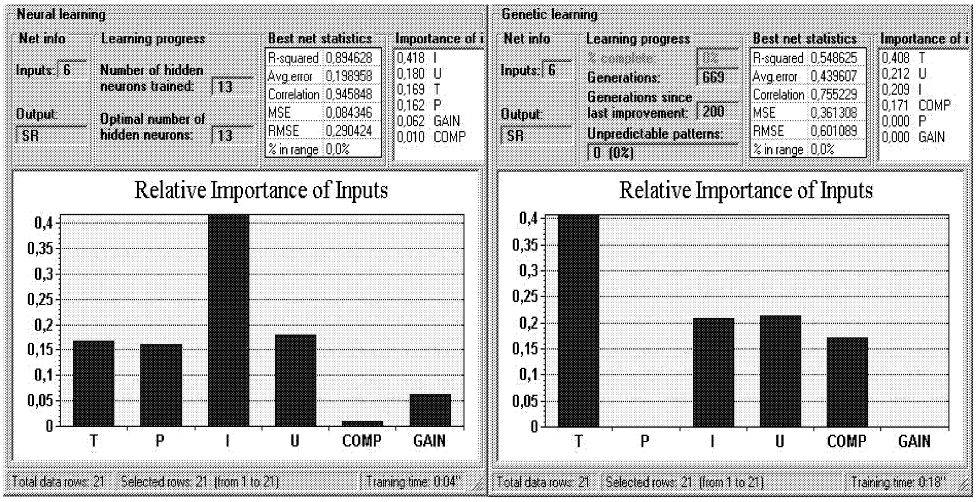
The network was trained using these two strategies and the plot of actual vs. predicted values of the material removal rate looks as shown in Figure 6.7. Contrary to the NeuroShell® Predictor developers' indication (NeuroShell, 1998), the neural training strategy shows better correlation between the actual and the predicted values of the model than the genetic based learning strategy. The lower correlation of the genetic based learning strategy can be attributed to the fact that this strategy cannot predict values that are outside the range of the training dataset. The plots for both cases show that the material removal rate for dataset number 9 (Table 6.3) is the highest though the magnitude is lower in genetic learning case. Compared with the optimization result of the hybrid system in the previous section, an interesting correlation can be seen concerning the size of the maximum material removal rate. There are two peak values of the material removal rate (1015,58 and 1038,68 mm^3/min) for the genetic based prediction that are very close to the optimum value achieved by the hybrid model (1038,68 mm^3/min).

As shown Figure 6.8 the neural network based prediction shows that Ontime has the highest relative importance to control the material removal process; while the peak current and the gain show very low relative importance. For the genetic based model, however, the voltage is the dominating variable while Offtime and peak current have insignificant role for the process. Comparing these results with the hybrid system is difficult because contrary to these predictions, the hybrid system shows a higher relative importance of the peak current for the material removal rate.



(a) Neural learning (b) Genetic learning

Figure 6.8: Relative importance of variables for material removal rate



(a) Neural learning (b) Genetic learning

Figure 6.9: Relative importance of variables for surface roughness

Figure 6.9 shows, on the other hand, higher relative importance of the peak current and the pulse Ontime for the surface roughness as predicted by the neural based and the genetic based predicators respectively. Compared with the hybrid system we find here certain resemblances because, as shown in Figure 6.5 shows, the peak current has the highest relative importance for the surface roughness.

6.6.2. Practical implementation issues

In practice, processes are often operated at recommended best performance values than optimum values. This can be because either it is not always simple to define an optimum point of operation or sometimes it may not be necessary to find such a point accurately as far as a reasonable range of best performance is known. The best performance values can be considered as near optimal values defined within certain plus or minus ranges from the optimum point. To define such a range, it is important to know the influential variable(s) as addressed in the above discussions and a method of visualizing the performances.

According to EDM Technical Handbook (1994), the common method of visualizing the influence of the control variables on EDM performance parameters is using a diagram showing the parameters as a function of the Ontime. The results of the above-discussed model show also that this parameter is highly influential in most of the cases.

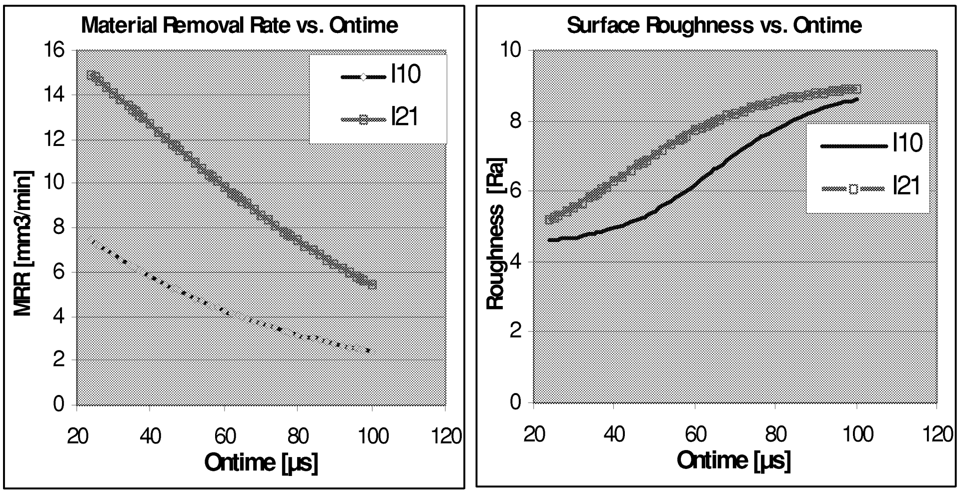


Figure 6.10: Variation of performance parameters with Ontime

Based on this conclusion and knowing the neural network's optimum structure of 13 hidden neurons, another *backpropagation network* was modeled using NeuFrame from Neurosciences because NeuFrame allows both training and testing datasets. The trained model was then tested using a randomly generated test data within the range of the training dataset. Two peak current values were selected for comparison and the results are shown in Figure 6.10.

Reports for some materials such as polycrystalline diamond (PCD) show that the material removal rate increases with Ontime to a certain extent and then decreases (Kozak, *et al.*, 2001). According to the report, the MRR for both water and oil dielectric fluid is attained at Ontime of 60 µs. In this analysis, however it shows a decreasing tendency with very small variations towards higher Ontime values (Figure 6.10). This result requires further verification because it somehow contradicts with existing results for other materials. The surface roughness, on the other hand, agrees with accepted trends with the fact that higher peak current contributes to a rougher surface. This implies that the longer the pulse Ontime, the higher is the roughness because of a large quantity of energy transfer to the material that can produce large craters. Using short pulses, no significant differences were observed in the roughness for different discharge currents. However, increased pulse duration tends to increase the roughness in the high peak current region. This is an indication that good surface quality can be achieved only at lose of productivity.

6.7. Chapter Summary

This chapter has addressed one of the most difficult problems in modern manufacturing industry particularly the modeling and optimization problem of EDM process. A mechanism that combines the important capabilities of neural networks and genetic algorithms has been developed as an implementation of the hybrid CI methodology outlined in Chapter 3. This hybrid CI approach is aimed to find an integrated solution to the existing problem of modeling and optimization of manufacturing processes for which formulating an optimization model is not straightforward. EDM represents a category of manufacturing processes called *non-conventional machining* techniques where defining an appropriate analytical model is often difficult.

The methodology development was divided into two main phases. In the first phase, a genetic algorithm based learning strategy was used instead of the standard backpropagation learning to develop a neural network model that can identify the input-output patterns. This was found necessary primarily because developing a hybrid system using standard or commercial neural network tools is very difficult due to the "black box" nature of neural networks. Moreover, standard backpropagation learning is often criticized to be trapped at local minimum for relatively large problem domains. Coding the network weights as chromosomes and the mean square error as the fitness function(s), the network structure was defined through genetic learning strategy that minimizes the error.

Having the network structure, the second phase was very simple and straightforward. Defining the input variable as chromosomes, the network output was used as the fitness function for GA optimization. Finally, different methods of verifications were implemented to see if the model performance is reasonable. Comparisons with other predicator tools show that the developed model performance is satisfactory and most of all, the hybridization model shows a good potential to achieving a multi-objective optimization of the overall process.

A relative importance approach was used to see which input variables influence the output performances so that operation and control of the process can target those most dominating variables. In good agreement with research results for other tool-workpiece combinations, the result of this investigation shows that pulse Ontime and the peak current are the most important variables to control both the material removal rate and the surface roughness.

In short, the benefits of this hybrid CI system for modeling and optimization of EDM process is twofold. Primarily, the developed methodology and the solutions achieved from the investigation using the methodology establish better knowledge about the interaction between the tool (graphite) and the workpiece (nickel-based alloy) for the process. The most beneficiaries of this result include aircraft, automobile and tool-making industries. Secondly, the solution approach implemented in this chapter opens a new direction for research so that other manufacturing processes particularly those known to be difficult to model using existing modeling techniques can be solved using similar approach.

CHAPTER 7

CONCLUSIONS AND FURTHER WORKS

After going through the critical review and synthesis of previous and ongoing works in the area, this study focused on the two process planning tasks, namely mapping design information to manufacturing actions and optimal sequencing of operations since they are bottlenecks of integrating design and manufacturing. Today, the industry has a problem of directly transferring design information of modern CAD models to manufacturing actions because of the lack of common "language" among the different functions in the manufacturing field. The type of information needed by CAM systems and other operation, technology and resource related complexities in the machining environment have also contributed to the difficulty of optimizing operation sequencing. Accordingly, the study concluded that the countless alternative options involving resources and technology used to produce a part together with several constraints represent a combinatorial optimization task demanding powerful tools that can accomplish multi-objective solutions simultaneously.

The problem of an integrated design and manufacturing is not only optimization. To use existing optimization tools, we need also appropriate modeling of the process. Very important theoretical foundations and mathematical relations are available for some conventional machining processes such as metal cutting using a lathe machine. For more complex processes such as face milling operations however either (a) the existing relations need to be appropriately formulated so that the existing optimization tools can be used or (b) there exists no mathematical model that fully represent the process. EDM process represents an example of the last case, where both modeling and solving for optimum operation is one objective of this research.

To achieve the goals of the study, some theoretical foundations and solution approaches (methodologies) have been developed that can enhance the required manufacturing intelligence using several forms of hybrid computational intelligence approach. The study done in genetic algorithm and neural network technologies created the background for the proposed hybridization approaches and their implementation issues. Motivated by the good capability of neural networks to recognize features through appropriate training and the global and robust search capability of genetic algorithms, the study developed the idea of *hybrid genetic algorithm and neural network* (HyGANN) whose implementation in manufacturing is treated from two perspectives.

1. Hybrid CI approach has been proposed whereby the design information, having a manufacturing meaning, is recognized by neural networks and further used as input for operation sequencing optimization using genetic algorithms. Due to the complexity of the problem, the study concluded that operation sequencing can be treated, at different levels, as a combination of combinatorial and multi-objective optimization problem with constraints. Particularly, an optimization methodology using the TSP approach has been adopted with certain modifications to the original principle. As a mechanism of evaluation criterion for process planning, a multi-objective optimization model for machining economics has been developed and a solution approach has been demonstrated using a genetic algorithm tool – GeneHunter.

2. Optimization using genetic algorithms needs the figure of merit in certain form of mathematical model. Though the relation between various variables and performance parameters in manufacturing can be found through direct measurements, establishing a formula that represents the relation among the variables and the parameters is often difficult. Thus, direct application of genetic algorithms to optimization of such problems is impossible. To overcome this bottleneck, the study proposed a hybrid CI approach where neural networks, supported by genetic algorithms, are trained to define the structure of the network that minimizes the error between the target samples and the network output. The optimized network structure has been used as the fitness function of genetic algorithm based optimization. The approach has been demonstrated using experimental data from an electro-discharge machining, a non-conventional machining process that is characterized as difficult-to-model.

Putting all together, the contribution of the thesis to the scientific works in the field can be summarized as follows:

- The theories and the proposed methodologies of hybrid CI system to map design information into their manufacturing counterparts for optimization can advance the existing attempts to bridge design and manufacturing.
- The optimization methodology of operation sequencing and economics of machining using genetic algorithm approach advances the ongoing research to find a compromised near optimal operation of manufacturing processes.
- The new approach of using hybrid CI to model and optimize complex machining environments such as the EDM process not only benefits those

industrial activities specializing in the field, but also opens a new field of research for other similar non-conventional operations.

- The highlighted problem areas in integration and optimization of design and manufacturing can advance better understanding of existing challenges for further works.
- Finally yet importantly, the facts, solution methodologies and issues discussed in this thesis can be appropriately applied in other industrial business.

In general, the thesis has focused on development and demonstration of methodologies whereby hybrid CI systems are treated merely as application tools. The central objective has been elevating manufacturing system intelligence through integration of manufacturing entities and optimization of processes. This is because an integrated and optimized manufacturing system is expected to be better flexible and profitable.

It is also natural that the discussed methodologies and the achieved results require further verifications. For example, the proposed hybridization methodologies may require improvement in terms of the genetic parameters and structure of the neural network. More importantly, the EDM modeling and optimization problem was based on few experimental dataset due to the specific characteristic of the non-conventional machining environment. The test data was also collected in accordance with the recommended combinations of parameters for a particular workpiece (steel). Therefore, the obtained results are subject to further verification in terms of tests that are more systematic or in implementation.

APPENDICES

APPENDIX A

The table shows the measured values used to calculate the surface roughness, the material removal rate and the tool wears.

	Surface roughness tests						Material removal rate				Tool hiegth			Tool wear	
Test	Test numbers					Av.SR	Time	Depth	Feed r.	MRR	Before	After	Diff	CWR	RWR
nr	1	2	3	4	5	Ra	[s]	[mm]	[mm/s]	[mm3/min]	[mm]	[mm]	[mm]	[mm]	%
1	8,53	8,41	9,89	7,66	9,85	8,87	4,23	0,98	0,23	400,11	100,00	100,00	0,00	0,10	0,00
2	7,57	9,00	7,12	7,90	7,76	7,87	4,11	0,91	0,22	380,33	100,09	100,14	-0,05	0,15	-0,06
3	6,22	5,95	6,52	6,18	7,00	6,37	2,34	0,93	0,40	681,71	99,78	99,81	-0,03	0,20	-0,03
4	6,49	5,00	5,10	5,26	5,36	5,44	1,50	0,91	0,61	1038,68	99,84	99,83	0,01	0,20	0,02
5	5,67	6,39	5,73	4,48	5,43	5,54	1,90	0,90	0,50	846,31	99,83	99,79	0,04	0,30	0,05
6	8,00	6,77	5,36	7,22	8,34	7,14	2,60	0,95	0,37	627,90	100,32	100,29	0,02	0,20	0,02
7	7,42	7,22	5,85	6,96	7,55	7,00	2,81	0,93	0,34	577,78	99,48	99,47	0,01	0,20	0,01
8	6,65	6,56	6,90	7,35	6,89	6,87	2,89	0,95	0,34	582,68	99,42	99,40	0,02	0,05	0,02
9	6,85	8,99	7,92	8,71	7,83	8,06	1,23	0,97	0,79	1344,52	99,54	99,54	-0,01	0,20	-0,01
10	7,67	7,85	7,66	6,96	7,27	7,48	1,44	0,86	0,60	1017,52	99,55	99,52	0,03	0,10	0,04
11	7,47	7,64	7,16	8,54	8,58	7,88	3,09	0,87	0,29	486,63	99,54	99,55	-0,01	0,15	-0,02
12	6,57	6,58	5,57	7,78	7,16	6,73	4,62	0,90	0,20	334,35	99,46	99,45	0,01	0,10	0,01
13	4,38	4,63	5,69	4,67	5,40	4,95	8,17	0,85	0,11	179,12	99,45	99,43	0,02	0,20	0,02
14	7,47	7,69	7,90	7,04	7,60	7,54	3,42	1,08	0,31	536,12	98,90	98,90	0,01	0,20	0,01
15	6,76	6,61	7,23	7,67	6,84	7,02	3,37	1,05	0,31	536,40	99,76	99,76	-0,01	0,20	-0,01
16	6,19	7,12	6,23	5,64	7,38	6,51	2,63	0,91	0,35	593,54	98,97	98,95	0,02	0,18	0,02
17	6,06	6,85	7,10	8,02	6,63	6,93	2,35	0,72	0,31	526,27	98,90	98,89	0,02	0,20	0,02
18	5,74	6,45	6,55	7,04	6,95	6,55	2,42	0,71	0,30	503,13	98,84	98,82	0,02	0,15	0,02
19	7,40	6,06	7,68	7,89	7,41	7,29	3,28	0,91	0,28	484,10	98,81	98,81	0,00	0,10	0,00
20	5,84	5,91	7,94	6,83	6,40	6,58	2,67	0,93	0,35	603,71	98,81	98,79	0,01	0,20	0,01
21	6,91	6,44	5,97	6,95	5,81	6,42	3,24	0,95	0,29	500,53	98,89	98,86	0,02	0,20	0,03

APPENDIX B

The table shows the network structure with values of nodes in the input layer, the hidden layer and the error comparison of the material removal rate at the output layer.

Test	INPUT LAYER						Nodes of hidden layer			Output	**Compa-**	Target MRR	
nr.	*T*	*P*	*I*	*U*	*COMP*	*GAIN*	N1	N2	N3	MRR	**rator**	Scaled	Raw data
1	100	10	21	100	42,3	15	0,9952	0,9835	0,9959	0,5	0,0963	0,190	400,110
2	56	10	21	100	42,3	15	0,9881	0,9609	0,9821	0,5	0,1072	0,173	380,326
3	42	10	21	100	42,3	15	0,9841	0,9487	0,9716	0,5	0,0047	0,431	681,713
4	32	10	21	100	42,3	15	0,9805	0,938	0,9606	0,5	0,0564	0,738	1038,684
5	24	10	21	100	42,3	15	0,977	0,9279	0,949	0,5	0,0053	0,572	846,307
6	49	56	21	100	35,3	15	0,9902	0,9691	0,995	0,5	0,0132	0,385	627,899
7	49	37	21	100	35,3	15	0,9881	0,9606	0,9906	0,5	0,0249	0,342	577,783
8	49	10	21	100	35,3	15	0,9846	0,9447	0,9768	0,5	0,0236	0,346	582,683
9	49	37	39	100	35,3	15	0,9906	0,9746	0,992	0,5	0,2500	1,000	1344,521
10	49	37	29	100	35,3	15	0,9893	0,9676	0,9912	0,5	0,0481	0,719	1017,516
11	49	37	21	100	35,3	15	0,9881	0,9606	0,9906	0,5	0,0558	0,264	486,635
12	49	37	17	100	35,3	15	0,9875	0,9566	0,9902	0,5	0,1345	0,133	334,349
13	49	37	10	100	35,3	15	0,9864	0,9487	0,9896	0,5	0,2500	0,000	179,120
14	49	37	21	100	30,2	15	0,9871	0,954	0,9904	0,5	0,0375	0,306	536,118
15	49	37	21	100	20	15	0,9848	0,9376	0,9899	0,5	0,0374	0,307	536,400
16	49	37	21	100	35,3	20	0,9889	0,9606	0,991	0,5	0,0209	0,356	593,536
17	49	37	21	100	35,3	15	0,9881	0,9606	0,9906	0,5	0,0409	0,298	526,266
18	49	37	21	100	35,3	12	0,9877	0,9606	0,9903	0,5	0,0493	0,278	503,130
19	49	37	21	100	35,3	10	0,9873	0,9606	0,9901	0,5	0,0568	0,262	484,101
20	49	37	21	80	35,3	15	0,9825	0,9601	0,9879	0,5	0,0184	0,364	603,707
21	49	37	21	60	35,3	15	0,9741	0,9596	0,9846	0,5	0,0503	0,276	500,531
Scaled MRR		=	Xi - Min								0,0560	Max	1344,521
			Max - Min							Fitness function		Min	179,120

APPENDIX C

The table shows the network structure with values of nodes in the input layer, the hidden layer and the error comparison of the surface roughness at the output layer.

Test	INPUT LAYER						Nodes of hidden layer			Output	Compara	Target SR	
nr.	*T*	*P*	*I*	*U*	*COMP*	*GAIN*	N1	N2	N3	SR	(Error)	Scaled	Raw data
1	100	10	21	100	42,3	15	0,9613	0,9832	0,9912	0,5027	0,2473	1,000	8,868
2	56	10	21	100	42,3	15	0,9286	0,9768	0,9847	0,5026	0,0588	0,745	7,870
3	42	10	21	100	42,3	15	0,9136	0,9743	0,9818	0,5026	0,0195	0,363	6,374
4	32	10	21	100	42,3	15	0,9013	0,9724	0,9794	0,5025	0,1428	0,125	5,442
5	24	10	21	100	42,3	15	0,8903	0,9708	0,9773	0,5025	0,1245	0,150	5,540
6	49	56	21	100	35,3	15	0,9496	0,981	0,9874	0,5027	0,0031	0,558	7,138
7	49	37	21	100	35,3	15	0,9372	0,9767	0,984	0,5026	0,0004	0,523	7,000
8	49	10	21	100	35,3	15	0,9146	0,9688	0,9777	0,5026	0,0002	0,490	6,870
9	49	37	39	100	35,3	15	0,9693	0,9818	0,9891	0,5027	0,0846	0,794	8,060
10	49	37	29	100	35,3	15	0,9542	0,9791	0,9865	0,5027	0,0205	0,646	7,482
11	49	37	21	100	35,3	15	0,9372	0,9767	0,984	0,5026	0,0597	0,747	7,878
12	49	37	17	100	35,3	15	0,9266	0,9753	0,9826	0,5026	0,0023	0,454	6,732
13	49	37	10	100	35,3	15	0,9042	0,9728	0,9799	0,5025	0,2526	0,000	4,954
14	49	37	21	100	30,2	15	0,9332	0,9721	0,9803	0,5026	0,0250	0,661	7,540
15	49	37	21	100	20	15	0,9244	0,9602	0,97	0,5026	0,0007	0,528	7,022
16	49	37	21	100	35,3	20	0,9408	0,9788	0,9845	0,5026	0,0109	0,398	6,512
17	49	37	21	100	35,3	15	0,9372	0,9767	0,984	0,5026	0,0000	0,505	6,932
18	49	37	21	100	35,3	12	0,9349	0,9753	0,9837	0,5026	0,0092	0,407	6,546
19	49	37	21	100	35,3	10	0,9333	0,9743	0,9835	0,5026	0,0088	0,596	7,288
20	49	37	21	80	35,3	15	0,9372	0,971	0,9806	0,5026	0,0074	0,416	6,584
21	49	37	21	60	35,3	15	0,9371	0,9642	0,9764	0,5026	0,0167	0,374	6,416
Scaled SR		=	Xi - Min								0,04983	Max	8,868
			Max - Min							Fitness function		Min	4,954

APPENDIX D

The table shows the network structure with the multi-objective error optimization - the material removal rate and the surface roughness at the output layer.

Test	INPUT LAYER	Nodes of hidden laye			Output	Compar	Target MRR		Output	Comp	Target SR	
nr.		N1	N2	N3	MRR	(Error)	Scaled	Raw data	SR	(Error)	Scaled	Raw
1		1	0,89	0,89	0,5	0,0963	0,190	400,110	0,507	0,243	1,000	8,868
2		0,99	0,84	0,81	0,5	0,1072	0,173	380,326	0,507	0,057	0,745	7,870
3		0,98	0,81	0,78	0,5	0,0047	0,431	681,713	0,507	0,021	0,363	6,374
4		0,96	0,8	0,75	0,5	0,0564	0,738	1038,684	0,507	0,146	0,125	5,442
5		0,94	0,78	0,73	0,5	0,0053	0,572	846,307	0,506	0,127	0,150	5,540
6		0,99	0,83	0,88	0,5	0,0132	0,385	627,899	0,507	0,003	0,558	7,138
7	The same inputs as in the previous Appendices	0,99	0,83	0,85	0,5	0,0249	0,342	577,783	0,507	0,000	0,523	7,000
8		0,99	0,82	0,79	0,5	0,0236	0,346	582,683	0,507	0,000	0,490	6,870
9		0,99	0,9	0,87	0,5	0,2500	1,000	1344,521	0,507	0,082	0,794	8,060
10		0,99	0,86	0,86	0,5	0,0481	0,719	1017,516	0,507	0,019	0,646	7,482
11		0,99	0,83	0,85	0,5	0,0558	0,264	486,635	0,507	0,058	0,747	7,878
12		0,99	0,8	0,84	0,5	0,1345	0,133	334,349	0,507	0,003	0,454	6,732
13		0,98	0,76	0,83	0,5	0,2500	0,000	179,120	0,507	0,257	0,000	4,954
14		0,99	0,82	0,84	0,5	0,0375	0,306	536,118	0,507	0,024	0,661	7,540
15		0,99	0,82	0,84	0,5	0,0374	0,307	536,400	0,507	0,000	0,528	7,022
16		0,99	0,83	0,85	0,5	0,0209	0,356	593,536	0,507	0,012	0,398	6,512
17		0,99	0,83	0,85	0,5	0,0409	0,298	526,266	0,507	0,000	0,505	6,932
18		0,99	0,83	0,85	0,5	0,0493	0,278	503,130	0,507	0,010	0,407	6,546
19		0,99	0,82	0,85	0,5	0,0568	0,262	484,101	0,507	0,008	0,596	7,288
20		0,98	0,82	0,84	0,5	0,0184	0,364	603,707	0,507	0,008	0,416	6,584
21		0,98	0,82	0,84	0,5	0,0503	0,276	500,531	0,507	0,018	0,374	6,416
						0,056	Max	1344,521		0,05	Max	8,87
					Fitness function		Min	179,120	Sub-goal		Min	4,954

REFERENCES

1. Aas, L.M., P.K. Sannæs, H.L. Gelgele, F.O. Rasch, and K. Wang (2001), Methodology to increase process-efficiency in EDM-machining of jet engine parts in a nickel-base alloy, *Proc. of the 13th Int. Symp. for Electromachining, ISEM XIII*, Bilabo, Spain, Vol II, 649-660.
2. Abido, M.A. (1997), Intelligent techniques approach to power system identification and control, Ph.D. Dissertation, University of Petroleum and Minerals, Saudi Arabia.
3. Agapiou, J.S. (1992), The optimization of machining operations based on a combined criterion, Part 2: Multi-pass operations, *ASME Transactions: Journal of Engineering Industry*, **114**, 508-513.
4. Alting, L. and H. Zhang (1989), Computer-aided process planning: A state-of-the-art survey, *Int. Journal of Production Research*, **27**, 553-585.
5. Arguelles, D. (1996), Hybrid artificial neural network/genetic algorithm approach to the on-line optimization of electrical power systems, Ph.D. Dissertation, George Washington University, USA.
6. Arikan, M., A. Sahir and O.H. Totuk (1992), Designing using machining operations. *Annals of CIRP*, **41**, 185-188.
7. Barschdoff, D. and L. Monostori (1991), Neural networks: Their applications and perspectives in intelligent machining, *Computers in Industry*, **17**, 101-119.
8. Bendict, G. (1987), *Nontraditional Manufacturing Processes*, Marcel Dekker Inc, NY.
9. Biegel, J.E. and J. Davern (1990), Genetic algorithm and job scheduling, *Computers in Industrial Engineering*, **19**, 81-91.
10. Boothroyd, G. and W.A. Knight (1989), *Fundamentals of Machining and Machine Tools*, Marcel Dekker Inc., NY.
11. Bornholt, S. and D. Graudenz (1992), General asymmetric neural networks and structure design by genetic algorithms, *Neural Networks*, **5**, 327-334.
12. Bowden, R.O. (1992), Genetic algorithm-based machine learning applied to the dynamic routing of discrete parts, Ph.D. Dissertation, Mississippi State University, USA.
13. Chan, C.C. and G.W. Fischer (1996), Automatic feature recognition using a neural network technique, *Int. Sym. on Information Storage and Processing Systems*; *ASME, Manufacturing Engineering Division*, MED, **4**, 731-744.

14. Chan, C.C.H. (1994), Artificial neural network based feature recognition and grammar based feature extraction to integrate design and manufacturing, Ph.D. Dissertation, University of Iowa, IA, USA.

15. Chan, K.C., and H. Tansari (1994), A study of genetic crossover operations on the facilities layout problem, *Computers in Industrial. Engineering*, **26**(3), 537-550.

16. Chang, C.A. and V. Angkasith (2001), Using Hopfield neural networks for operational sequencing for prismatic parts on NC machines, *Engineering Applications of Artificial Intelligence*, **14**, 357-368.

17. Chen, T. (1997), Hybrid intelligent system for process modeling and control using neural network and genetic algorithm, Ph.D. Dissertation, The University of Iowa, USA.

18. Cheng, M.Y. (1996), Control design and robustness measurement for biped locomotion, Ph.D. Dissertation, University of Missouri-Columbia, USA.

19. Cheng, R. (1997), Study on genetic algorithm-based optimal scheduling techniques, Ph.D. Dissertation, Tokyo Institute of Technology, Japan.

20. Dagli, C.H. (1994), *Artificial Neural Networks for Intelligent Manufacturing*, Chapman and Hall, UK.

21. de Floriani, L. (1989), Feature extraction from boundary models of three-dimensional objects. *IEEE Transaction of Pattern Analysis and Machine Intelligence,* **8**, 785-798.

22. De Jong, K.A. and W.M. Spears (1989), Using genetic algorithms to solve NP-complete problems, in *Proc. of 3rd Int. Conf. on Genetic Algorithms*, J.D. Schaffer Eds. San Mateo, CA: Morgen Kaufmann, 124-132.

23. Descotte, Y. and J.C. Latombe (1981), GARI: a problem solver that plans how to machine parts. *Proc. of the Seventh Int. Joint Conf. on Artificial Intelligence,*Vol II, 772-776.

24. Dev, K. (1995), *Optimization for Engineering Design: Algorithms and Examples*, Prentice-Hall, New Delhi.

25. Dibitonto, D.D., P.T. Eubank, M.R. Patel and M.A. Barufet (1989), Theoretical models of the electrical discharge machining process, I. A simple cathode erosion model, *Journal of Applied Physics*, **66**(9), 4095-4103.

26. Duan, W., J. Zhou and K. Lai (1993), FSMT: A feature solid-modeling tool for feature-based design and manufacture, *Computer Aided Design*, **25**(1), 29-38.

27. EDM Technical Manual (1994), Poco Graphite, Inc.

28. El-Sayad, J.S. and El-Gizawy (1997), Nonlinear multi-objective optimization of machining process parameters**,** *Int. Conf. on Computer Aided Optimum Design of Structures, OPTI,* 141-149.

29. Ermer, D.S. and D. C. Patel (1974), Maximization of production rate with constraints by linear programming and sensitivity analysis. *Proc. of Second North American Metalworking Research Conference*, WI, USA.

30. Fenton, R.G. and M.F.J Gagnon (1993), Computer-aided tool material selection for metal-cutting operations**,** *CIRP Annals,* **42**(1), 565-568.

31. Fu, H.P. and C.T. Su (2000), A comparison of search techniques for minimizing assembly time in printed wiring assembly, *Int. Journal of Production Economics,* **63**(1), 83-98.

32. Garey, M.R. and D.S. Johnson (1979), *Computers and Interactability - A Guide to the Theory of NP-Completeness,* W. H. Freeman, NY.

33. Gelgele, H.L. and K. Wang (1998), Part feature applications to integrate CAD/CAM systems, State-of-the-art Survey Report, No. NTNU 98022, NTNU, Norway.

34. Gelgele, H.L. and K. Wang (2000), Application of genetic algorithms for optimal sequencing of machining operations, To appear in *Proc. of the Nordic-Baltic and Northwestern Russia Summer School*, In *Application of Computational Intelligence in Engineering*.

35. Gen, M. and R. Cheng (1997), *Genetic Algorithms and Engineering Design*, John Wiley & Sons Nic., NY, USA.

36. Gen, M. and R. Cheng (2000), *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons Nic., NY, USA.

37. Gold, F.M. (1998), Application of genetic algorithm to the kinematic design of turbine blade fixtures, Ph.D. Dissertation, Wercester Polytechnic Institute, USA.

38. Goldberg, D. E. (1989, 1998), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.

39. Goldberg, D. E. and R. Lingle (1987), Alleles, loci and the travel salesman, In *Proc. of the First Int. Conf. on Genetic Algorithms*, Lawrence ErlbaumAssociates, Hillsdale, N.J., USA.

40. Gopalakrishan, B. and A-K. Faiz (1991), Machining parameter selection for turning with constraints: An analytical approach based on geometric programming, *Int. Journal of Production Research*, **29**(9) 1897-1908.

41. Gu, P. and D.H. Norrie (1995), *Intelligent Manufacturing Planning*, Chapman & Hall, London, UK.

URN:NBN:no-2332

42. Gupta, R., J.L. Batra and G.K. Lal (1994), Determination of optimal subdivision of depth of cut in multi-pass turning with constraints, *Int. Journal of Production Research*, **33**(9), 2555-2565.

43. Hashimoto, Y. (1995), Study on neural network and genetic algorithms on application to production planning problems, Ph.D. Dissertation, Ritsumeikan University, Japan.

44. Henderson, M. R. (1994), Manufacturing feature identification, In *Artificial Neural Networks for Intelligent Manufacturing* eds. by Dagli, C.H. Chapman and Hall, UK.

45. Henderson, M. R. and S. Prabhakar (1992), Automatic form-feature recognition using neural network based technique on boundary representations of solid models, *Computer-aided Design*, **7**, 381-393.

46. Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press.

47. Hwang, J.L. (1991), Applying the perceptron to 3D feature recognition, Ph.D. Dissertation, Arizona State University, USA.

48. Irani, S.A. et al. (1995), Feature-based operation sequence generation in CAPP. *Int. Journal of Production Research*, **33**(1) 17-39.

49. Jin, K. (1996), Study on optimal neural network design and its application using genetic algorithms, Ph.D. Dissertation, Hokkaido University, Japan.

50. Joines, J.A. (1996), Hybrid genetic search for manufacturing cell design, Ph.D. Dissertation, North Carolina State University, USA.

51. Joshi, S., and T.C. Chang (1988), Graph-based heuristics for recognition of machined features from a 3-D solid model, *Computer-aided Design*, **20**(2), 58-66.

52. Karuno, Y., H. Nagamochi and T. Ibaraki (1997), Vehicle scheduling on a tree with release and handling times, *Annals of Operations Research*, **69**, 193-207.

53. Kilic, S. E., C. Cogun and T. Sen (1993), A computer-aided graphical technique for the optimization of machining conditions, *Computers in Industry*, **22**, 319-326.

54. Kim, I.-T. and H.-W. Suh (1998), Optimal operation grouping and sequencing technique for multistage machining systems. *Int. Journal of Production Research*, **36**(8), 2061-2081.

55. Kim, Y.S. (1994), Volumetric feature recognition using convex decomposition. In *Advances in Feature-based Manufacturing*, J.J. Shah, M. Mantyla and D.S. Nau (eds) (Amestredam: Elsevier), 39-63.

56. Korde, U.P., B.C. Bora, K.A. Stelsen and D.R. Rely (1992), Computer-aided process planning for turned parts using fundamental and heuristic principles, *Journal of Engineering for Industry*. **114**, 31-40.

57. Kozak, J., K.P. Rajurkar and M. Ravi (2001), Neural network prediction of abrasive electrode discharge grinding process performance, *Proc. of the 13th Int. Symp. for Electromachining, ISEM XIII*, Bilabo, Spain, Vol I, 405-420.

58. Kronenberg, M. (1966), *Machining Science and Application: Theory and Application for Operation and Development of Machining processes*, Pergamon Press Ltd., Oxford, London, UK.

59. Kumar, N. and K. Shanker (2000), A genetic algorithm for FMS part type selection and machine loading, *Int. Journal of Production Research*, **38**(16), 3861-3887.

60. Kumara, S.R.T, C.Y. Kao, M.G. Gallagher and R. Kasturi (1994), 3D interacting manufacturing feature recognition, *Annals of the CIRP*, **43**(1), 133-136.

61. Kuniede, M., S. Furuoya and N. Taniguchi (1991), Improvement of EDM efficiency by supplying oxygen gas into gap, *Annals of CIRP*, **40**(1), 215-218.

62. Lee, I.B.H., B.S. Lim, and A.Y.C. Nee (1993), Knowledge-based process planning systems for the manufacture of progressive dies, *Int. Journal of Production Research*, **32**(2), 251-278.

63. Li, Y.Z. (1999), Study on hybridized genetic algorithms for production distribution planning problems, Ph.D. Dissertation, Ashikaga Institute of Technology, Japan.

64. Lin, A.C., S.-Y. Lin and S.-B. Cheng (1997), Extraction of manufacturing features from a feature-based design model, *Int. Journal of Production Research*, **35**(12) 3249-3288.

65. Malborg, C.J. (1996), A genetic algorithm for service level based vehicle scheduling, *European Journal of Operational Research*, **93**(1), 121-134

66. Mathew, R.M. (1998), Applications and limitations of genetic algorithms for the optimization of multivariate calibration techniques, Ph.D. Dissertation, Clemson University, USA.

67. Mazumder, S.K., M.K. Muju and K. Shanker (1995), An integrated feature recognition, operation sequencing and NC code generating for complex axis-symmetric components. *The First World Congress on Intelligent Manufacturing: Processes and Systems, Proceedings*, Vol. II, Puerto Rico, Feb. 13-17.

68. Medsker, L.R. (1994), *Hybrid Neural Network and Expert Systems*, Kluwer Academic, MA.

69. Memon, G.Q. (1995), Optimization methods for real-time traffic control, Ph.D. Dissertation, University of Pittsburg, USA.

70. Merchant, M.E. (1984), Analysis of existing technological forecasts pertinent to the utilization of artificial intelligence and pattern recognition techniques in manufacturing engineering, *16th CIRP Int. Seminar on Manufacturing Systems*, **14**(1), 11-16.

71. Ming, X.G., K.L. Mak and J.Q. Jan (1999), A hybrid intelligent inference model for computer aided process planning, *Integrated Manufacturing Systems*, **10**(6), 343-353.

72. Mitchell, R.J., J.M. Bishop, and W. Low (1993), Using genetic algorithm to find the rules of a neural network, *IEEE Proc. of the Int. Conf. on Artificial Neural Networks and Genetic Algorithms*, Innsbruk, Springer-Verrag, Wien, 664 – 669.

73. Mohri, N., N. Saito and M. Higashi (1995), A new process of finish machining on free surface by EDM methods, *Annals of CIRP*, **40**(1), 207-210.

74. Monostori, L. and Cs. Egresits (1997), On hybrid learning and its application in intelligent manufacturing, *Computers in Industry,* **33**, 111-117.

75. Murray, D. (1994), Tuning neural networks with genetic algorithm, *AI Expert*, 22-32.

76. NeuroShell (1998), NeuroShell User Manual for NeuroPredicator®.

77. Oliver, I.M., D.J. Smith and J.R.C. Holland (1987), A study of permutation crossover operations on the travel salesman problem. *Proc. of the 2nd Int. Conf. on Genetic Algorithms*, 224-230.

78. Ong, N.-S and L.P. Khaoo (1999), Genetic algorithm approach in PCB assembly, *Integrated Manufacturing Systems*, **10**(5), 256 -265.

79. Ozyurt, B., A. Kandel (1996), Hybrid hierarchical neural network-fuzzy expert system approach to chemical process fault diagnosis**,** *Fuzzy Sets and Systems,* **83**(1), 11-25.

80. Palmer, C.C. (1994), Approach to a problem in network design using genetic algorithms, Ph.D. Dissertation, Polytechnic University, USA.

81. Pareto, V. (1971), *Manual of Political Economy*, translated from Italian by A.S. Schweir, Macmillan, NY, USA.

82. Pinon, E. (1995), Investigation of the applicability of genetic algorithms to spacecraft trajectory optimization, Ph.D. Dissertation, University of Texas-Austin, USA.

83. Richardson, J., M. Palmer, G. Liepins and M. Hilliard (1989), Some guidelines for genetic algorithms with penalty functions, in *Proc. of the 3rd Int. Conf. on Genetic Algorithms,* Morgan Kanfmann Publishers, San Fransisco, USA.

84. Roy, U., M.D. Pollard, K. Mantooth and C.R. Liu (1989), Tolerance representation scheme in solid model: Part I, *ASME 15th Design Automation Conf.*, Montreal, Canada, 1-10.

85. Sakurai, H. (1995), Volume decomposition and feature recognition – Part I: Polyhedral objects, *Computer-aided Design*, **11**, 1269-1285.

86. Sakurai, H. and D.C. Gossard (1990), Recognizing shape features in solid models, *IEEE Computer Graphics and Applications.*

87. Saravanan, R., P. Asokan and M. Sachithanandam (2001), Comparative analysis of conventional and non-conventional optimization techniques for CNC turning process, *Int. Journal of Advanced Manufacturing Technology*, **17**, 471-476.

88. Senjen, R., M. de Beler, C. Leckie, and C. Rowles (1993), Hybrid expert systems for monitoring and fault diagnosis, *Proc. of the Conf. on Artificial Intelligence Applications*, 235-241.

89. Shaffer, J.D., D. Whitley and L. Esherman (1992), Combination of genetic algorithms and neural networks: State-of-the-art, *Proc. of Combination of Genetic Algorithms and Neural Networks*, IEEE Computer Society.

90. Shah, J. and M.T. Rogers (1988), Expert form feature modeling shell, *Computer Aided Design*, **20**(9).

91. Shah, J.J. (1990), An assessment of feature technology. *Proc. of the CAM-I Feature Symposium*, P-90-PM-02, 55-77.

92. Shah, J.J. (1992), Features in design and manufacturing. *Intelligent Design and Manufacturing* (John Wiley and Sons), 39-71.

93. Shin, Y. C. and Y. S. Joo (1992), Optimization of machining conditions with practical constraints, *Int. Journal of Production Research*, **30**, 2907-2919.

94. Soenen, R. and G. Olling (1995), *Advanced CAD/CAM Systems: State-of-the-art and Future Trends in Feature Technology*, Hartnolls Ltd. Bodmin, Cornwall, UK.

95. Stephenson, D.A. and J.S. Agapiou (1997), *Metal Cutting Theory and Practice,* NY: Marcel Dekker.

96. Steuer, R.E. (1986), *Multiple Criteria Optimization: Theory, Computation and Application*, Wiley, NY, USA.

97. Syswerda, G. (1991), Schedule optimization using GAs. In *Handbook of GAs* (eds. L. Davis) Van Nostrand, NY, 332-349.

98. Tafti, M.H.A. (1992), Neural networks: A new dimension in expert systems applications, *Data Base*, 51-54.

99. Tay, E.H. (1995), Automated generation and analysis of dynamic system designs, Ph.D. Dissertation, Massachusetts Institute of Technology, USA.

100. Tolouei-Rad, M. and I.M. Bidhendi (1997), On the optimization of machining parameters for milling operations, *Int. Journal of Machine Tools and Manufacture*, **37**, 1-16.

101. Tsoukalas, L. and J. Reyes-Jimenez (1990), Hybrid expert system-neural network methodology for nuclear plant monitoring and diagnostics, *Proc. of SPIE - The Int. Society for Optical Engineering*, **1293**(2), 1024-1030.

102. Tulkoff, J. (1987), Process planning: A historical review and future perspectives. *Proc. of the 19th CIRP Int. Seminar on Manufacturing System Computer-aided Process Planning*, PA, 207-210.

103. Uckun, S., S. Bagchc and K. Kawamura (1993), Managing genetic search in job shop scheduling, *IEEE Expert,* **8**(5), 15-24.

104. Van Dijck, F.S. and W.L. Dutrè (1974), Heat conduction model for the calculation of the volume of molten metal in electric discharges, *Journal of Physics*, **7**, 899-910.

105. van Houten, F.J.A.M. (1990), *PART: A Computer-aided Process Planning System*, FEBODRUK Enschede, The Netherlands.

106. Vandenbrande , J. H. and A.A.G. Requicha (1993), Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions Pattern Analysis and Machine Intelligence*, **15**, 1269-1285.

107. Wang, M.Y. (1995), Scheduling in flowshops and pallet requirements, Ph.D. Dissertation, University of Toronto, Canada.

108. Whitley, J.R. and J.F. Davis (1993), Qualitative interpretation of sensor patterns, *IEEE Expert*, 53-63.

109. Winter, G., J. Periaux, M. Galan and P. Cuesta (1995), *Genetic Algorithms in Engineering and Computer Science*, John Wiley & Sons Ltd., UK.

110. Wright, T. (1994), Genetic algorithm approach to scheduling resources for a space power system, Ph.D. Dissertation, Case Western Reserve University, USA.

111. Yokota, D. (1996), Study on solving system reliability optimization problems with interval data by genetic algorithms, Ph.D. Dissertation, Meiji University, Japan.

112. Zhang, F., Y.F. Zhang and A.Y.C. Nee (1997), Using genetic algorithms in process planning for job shop machining, *IEEE Transactions on Evolutionary Computation*, **1**(4).

113. Zhang, H.C. and L. Alting (1994), In *Computerized Manufacturing Process Planning System,* Chapman & Hall, London, UK.

INDEX

E

F

G

H