1997

An object-oriented fuzzy expert system for the design of rotating operations.

Xin. Yuan

University of Windsor

AN OBJECT-ORIENTED FUZZY EXPERT SYSTEM

FOR

THE DESIGN OF ROTATING OPERATIONS

by

Xin Yuan

A Thesis
Submitted to the Faculty of Graduate Studies and Research
Through the Department of Industrial and Manufacturing System Engineering
in Partial Fulfilment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

July, 1997

0-612-30995-9

I hereby declare that I am the sole author of this thesis. I authorize the University of Windsor to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Xin Yuan

I further authorize the University of Windsor to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Xin Yuan

The University of Windsor requires the signatures of all persons using or photocopying this thesis. Please sign below, and give an address and the date.

Abstract

In the thesis, a prototype object-oriented fuzzy expert system called SAM (Smart Assistant to Machinist) is presented for rotating cutter selection and cutting condition design. Object-Oriented Design and Programming Methods are employed to construct and implement the system. Based on partial and imprecise information, the system is able to select rotating cutter products available in current commercial market, and design cutting conditions for an optimized objective. The system consists of four modules: a database, a rule base, a cutter selection module and a cutting condition design module. The database consists of six kinds of data files: machinability of workpieces, machining plan, fuzzy rule, tool adapter, machine tool, as well as cutter inventory files. The rules in the rule base are developed based on fuzzy set theory. They are used to cross-define unknown information and to determine the relationships between the inputs and outputs information. The cutter selection module is developed based on fuzzy logic, in which the cutter selection is conducted in three steps. First, the input information is "fuzzied". Next, using the fuzzy correlation functions, insert grades and cutters are selected. Then, the selected grades and cutters are searched against the cutter inventory data file to check the availability. The cutting condition design module is developed based on fuzzy non-linear programming where the design constraints could be fuzzy. The model is solved by Hill Climbing Algorithm. Finally, the use of system is demonstrated using two examples.

DEDICATION

To my parents.

ACKNOWLEDGEMENTS

I would first like to thank my supervisor, Dr. R. Du, for his insight and support throughout the course of this thesis. I would like to thank my committee members, Dr. L. Li and Dr. S. Dutta, for their advice over the past years. The encouragement and suggestion form my group members are never forgettable. Last, but certainly not least, I would also like to express my appreciation to Ms. Jacquie Mummery, for her patience and help.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATIONS

FOR CUTTER SELECTION

- X = Input Variable
- Y = Output Variable
- MF(X) = Fuzzy Membership Function of Input X
- MF(Y) = Fuzzy Membership Function of Output Y
- MF(Y/X) = Fuzzy Membership Function of Output Y determined by X
- a, b, c, d = Coefficient Variables for fuzzy region

FOR CUTTING CONDITION DESIGN

Decision Variables:

- V = Cutting Speed (m/min)
- f = Feed (mm/rev)
- SP = Spindle Speed (RPM)
- R_t = Tool Replacement Schedule

Cost Time Factors:

- C_o = Operation Rate ($/min)
- C_t = Tool Cost ($)
- C_p = Unit Production Cost ($/pc)
- T_m = Unit Machining Time (min/pc)
- T_m'= Unit Cutting Time (min/pc) for each insert

- TL = Tool Life (min)
- T_N = Number of Tooth
- T_h = Holding Time (min)
- T_{ch} = Time Required for changing tool (min)
- P_c = Resulting Cutting Power (KW)
- SF = Resulting Surface Finishing (μm)

Decision Dependent Variables:

- F_c = Resulting Cutting Force (N)

Membership Functions:

- U_D = Membership Function of the Decision
- U_C = Membership Function of the Objective
- U_{GH} = Membership Function of the Constraints
- T_i = Maximum Tolerance of the Constraint i
- t_i = Violation Level of the Constraint i

Constant

- L = Length of Cut
- WOC =Width of Cut
- C = Cutting Speed for giving a tool life 1 min
- n = Taylor Exponent depending on the cutter material
- D = Cutter Diameter
- η = machine efficiency,

- MRR = material removal rate,
- Spec. = specific Energy,
- DOC = depth of cut

CHAPTER I

INTRODUCTION

1.1 Background and Overview

In machining operations, process planning plays a very important role in linking the product design and manufacturing together. As shown in Figure 1.1, the process planning includes six tasks: selection of machining operations, design of machining sequences, selection of machine tool, selection of cutters, design of cutting conditions, and tool path generation. This study deals with two tasks: the selection of cutters (including inserts and holders), and the design of cutting condition.

Similar to the other process planning problems, the addressed design problems are subjected to many constraints and have more than one possible or feasible solution. Figure 1.2 shows the representation of inputs, constraints, and outputs of the machining process planning system. The inputs include the material properties, insert (grade), toolholder and cutting conditions. The constraints include technical and economical considerations, such as the part geometry and tolerance, cost factors as well as the machine tool. The outputs are the desirable machine tools, cutters, cutting conditions, estimated profit, and product quality (surface finish and geometric accuracy). The design is usually determined by optimizing certain such as minimization of production cost or maximization of production rate.

This study focuses on rotational machining operations (including milling, boring and drilling), which play a very important role in modern manufacturing engineering. Selecting a proper cutter and designing optimal cutting conditions not only reduces the machining cost by means of extended tool life, increased material removal rate, reduced down time and tool inventory, but also improves the product

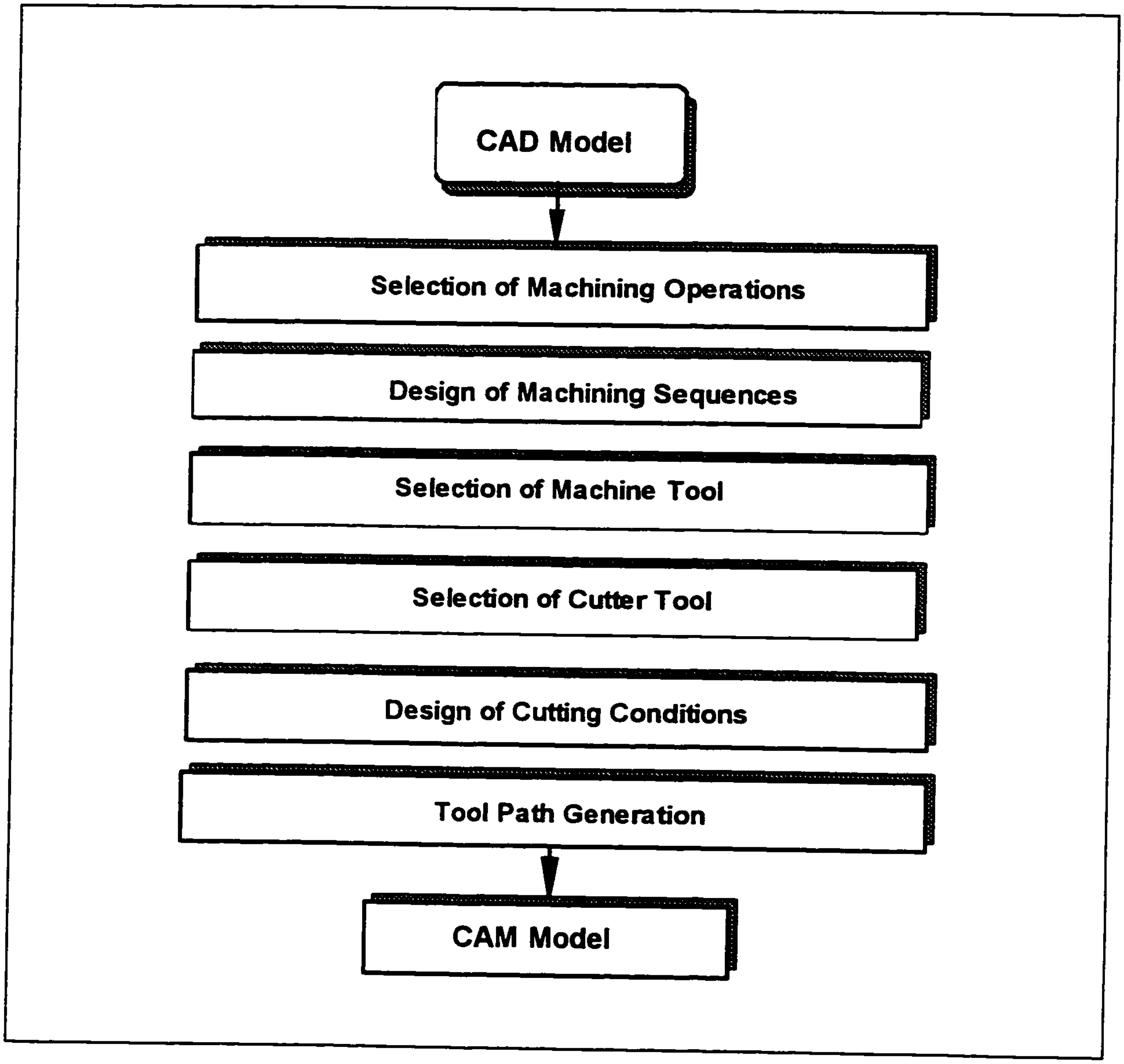


Figure 1.1 The Framework of Process Planning Tasks [Chang, 1990]

quality (better surface finish and geometry accuracy). It is arguably the best way to improve the machining performance with minimum investment.

However, usually the selection of cutter and decision of the cutting condition are also complex tasks. For example, there are thousands of different cutters, and new cutters are being developed

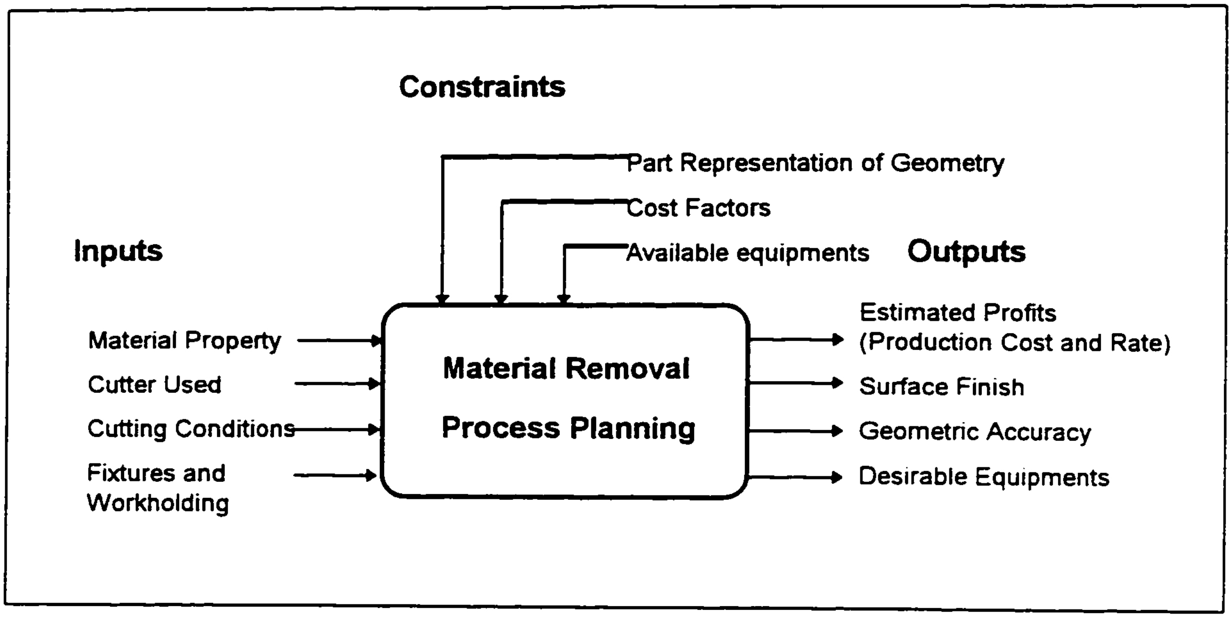


Figure 1.2 Inputs, constraints and outputs of material removal process

and marketed continuously. It also involves a large number of factors and constraints as mentioned above, such as work material, cutter material, cutter geometry, holder, production requirements, cutter availability and cutter cost, etc.

From stand-alone machining centres to transfer machining lines, the design of cutting condition (feed rate and cutting speed) is a key to reduce direct and indirect manufacturing cost and to improve productivity and quality. It is particularly important for high volume production where small savings adds up to significant organizational profit. Cutting condition design again involves a large number of uncertain factors, such as unit product cost, tool life and set-up time, and could be solved as an optimization problem.

Consequently, in practice, cutter selection and cutting conditions design are mostly done manually. It requires highly skilled and knowledge experts who are limited and expensive. In addition, because of the complex and imprecise nature of machining operations, it is very difficult for an human operator to consider all factors and to determine one optimal set-up. As a result, selections are essentially by trial-by-error. Therefore, automation in this area is strongly required to compensate the lack of these high skilled experts.

The objective of this research is to develop a new method for cutter selection and cutting condition design. It focuses on rotational machining operations (including milling, boring and drilling).

1.2 Organization of the Proposal

This thesis is organized into six chapters. Chapter II is the literature review of the scientific publications and commercial software involving cutter (grade) selection and cutting condition design model. Chapter III proposes the methods used in the system. Chapter IV describes the proposed system. Chapter V shows two application examples. Finally, the conclusion and the future work are discussed in Chapter VI.

CHAPTER II

LITERATURE REVIEW

Along with the development of computer technology in the past decades, a large number of research activities have been reported. This chapter will summarize both commercial and research systems for cutter selection and cutting condition design, respectively.

2.1 Reviews on Cutter (Grade) Selections

The most common method for cutter selection is to use databases.

METCAPP/1 was developed by the Institute of Advanced Manufacturing Science Inc. of Ohio U.S.A. It is a database-oriented computer system that consists of a large cutter database. Users must provide all selection criteria in order to perform the database searching. Sequential searching algorithm was used in this system.

COROGUID is a product catalog developed by the largest cutter manufacturer, Sandvik A.B. of Sweden. It is designed to support the application of Sandvik tooling products including cutters, cutter holders and ship-breaker and also provides information such as suitable cutting conditions and tool life. The cutters are searched in a large database which is organized into several levels. Users must make decisions in each level to lead the end selection.

Based on thousands of metal cutting experiments, an equation called ECT (Equivalent Clip Thickness) was developed by Dr. Colding [1960]. Based on the ECT equation coupled with material machinability data (developed by Metcut Research Associate), COMP (stands for Computer

Optimized Manufacturing Planning) and COMP5 was developed to perform the selection of cutter and cutting conditions by Colding International Inc. of Michigan, U.S.A.

EMDB (stands for Electronic Machine Data Book) is developed by National Computer Board of Singapore. It provides recommendations on cutting conditions, cutter geometry and cutting fluid.

Santochi and Dini [1996] verified the applicability of atificial neural networks in cutting tool selection field. The research focused on the design of neural networks for the automated selection of the following technological parameters of a cutting tool: insert grade, normal clearance angle, rake angle, cutting edge inclination angle and so on so forth.

Narang and Fischer [1993] developed a knowledge based system called ESTPAR. Based on predetermined rules, complete and precise input information, the system can select the suitable cutter. This system can also determine the cutting conditions using a mathematical optimization model.

Zhou and Wysk [1993] developed an expert system for cutter selection. Cutters are selected via an expert system which consists of a database, a knowledge base and an inference engine. The inference mechanism tries to match the input information, such as preferred characteristics of cutting tools, process plans, machine tool parameter and workpiece information, with expert rules and hence identify the appropriate tool(s). The output of the cutter selection module also includes the data required for cutting conditions optimization as well as tool replacement schedule.

Dhage and Usher [1993] presented a prototype system that automates selection of tools and tool holders for turning and boring operations. This system was mainly based on database searching. It first determines all applicable toolholders based on those listed in the toolholder database. After querying the user for additional information regarding the nose radius and part material, the system determines suitable inserts. The system can also perform a cost analysis and list all matched sets of toolholders and inserts.

Maropoulous [1992] presented a cutting tool selection method called the Intelligent Tool Selection (ITS). It is a computer based system using Knowledge-Base System (KBS) as well as sequential searching algorithm. It uses the concept of the Minimal Storage Tooling (MST) by linking new and replaced tools in production control and planning. ITS also uses the concept of Tool Resources Structure (TRS) to determine all tooling resources required for producing a part.

Yeo and Rahman [1991] developed a method to model the knowledge from experienced process planners and NC programmers by using expert system techniques. They developed an experimental system for an integrated automated machining system. The system adopted a frame-based approach for the generation of insert and toolholder selections for turned parts. According to the type of operation and surface roughness requirement, the insert nose radius and feed are obtained. The insert material grade is dependent on its selected operating feed range.

Domazet [1990] proposed an automatic turning tool selection process. He used the production rule matrix method (PRMM) which is a knowledge presentation method that formulates the knowledge based production rules in a table form instead of using logical "IF-THEN" rules. Using this method, only the tool class and the toolholder type are selected.

Maropoulous [1990] developed a procedure to select tools for finish turning. Based on input information of different types of finish turning, required surface finish and tool life, the tools are selected by searching the database. The system also calculates the cost of machining with the selected tool based on a nominal length and an average profile diameter.

Chen et al. [1989] developed a method to select tools for rough turning operations. The selection procedure was made from a tool library which was in a form of a tree structure consisting of six levels. A heuristic procedure was used to search the tool library. The procedure started from the top of the tree and proceeded downward. An optimal cutting speed was also given to minimize the production cost.

There are also some other systems, such as: Grade Advisor and Milling Advisor by Kennametal Inc. of Ohio, U.S.A; EXAPT INFOS by Fraunhofer-Gesellschaft, of Aachen, German. In summary, Table 2.1 concludes some of above models with typical methods.

Product and Developer	**Application**	**Process Planning**	**Performance Calculation**	**Method Description**
COMP5 (Colding International Inc.)	turning, boring milling, drilling	assisted	cutting forces, cutting power, MRR, tool life, cost estimation	Based on the ECT (Equivalent Clip Thickness) ECT equation coupled with material machinability data
METCAPP/1 (Institute of Advanced Manufacturing Science Inc.)	turning, milling	suggested	cost estimation	Sequential searching algorithm was used in a database-oriented computer system that consists of a large cutter database
COROGUID (Sandvik Inc.)	turning, milling, drilling	-	cutting power, MRR, tool life	The cutters are searched in a large database which is organized into several levels. Users must make decisions in each level to lead the end selection.
INFOS (Fraunhofer-Gesellschaft Inc.)	turning, boring, milling, drilling	assisted	cutting forces, cutting power, MRR, tool life	Inference engine
GA Advisor	turning,	-	cutting power,	Database searching

(Kennametal Inc.)	boring		MRR	

Table 2.1 Summary of some cutter and grade selection models

2.2 Reviews on Cutting Condition Optimization Model

The design of optimal cutting conditions has long been recognized as an important task in metal cutting. A large number of mathematical models have been developed for the optimization of cutting conditions. These models can be classified into three categories:

(1) Deterministic Models

(2) Probabilistic Models

(3) Fuzzy Models

In general, an optimal cutting condition design model can be represented as follows:

Objective function: 1) Minimization of Unit Production Cost

or 2) Maximization of Unit Production Rate

or 3) Maximization of Profit

Subject to various constraints which include:

1) Maximum and Minimum cutting speed

2) Maximum and Minimum feed

3) Cutting force limitation

4) Power availability limitation

5) Surface finish restriction

6) Tool life requirement

where the objective function and constraints can be expressed or approximated by a combination of the cutting conditions.

For the deterministic models, all the parameters and constraint constants are considered deterministic. On the other hand, in probabilistic models, various probability distributions are assumed to represent the random nature of uncertain parameters such as tool life and cutting force. For example, the tool life could be treated as a random variable whose distribution is parameterized by cutting conditions.

2.2.1 Deterministic Models

According to the literature survey, the problems of minimizing cost or maximizing production rate of single-pass and multi-pass machining operations have been investigated extensively and many optimization techniques have been developed.

Gilbert [1950] used an analytical method to determine the cutting speed that minimized the machining cost for a single pass turning operation where feed and depth of cut were fixed.

Armarego and Brown [1965] presented equations for optimizing the machining cost by determining machining variables with only the depth of cut fixed.

Walvekar and Lambert [1978] considered the optimizing problem as a multi-stage decision process and utilized geometric programming for the simultaneous determination of the optimal levels of speed and feed to minimize the unit production cost. These levels were subjected to certain practical operation restrictions such as maximum cutting force, maximum power and required surface finish.

Petropoulous [1973] developed a method for choosing the optimal cutting speed and feed using geometric programming to optimize the unit production cost under various constraints including a surface roughness model. The model was solved by primal and dual programming.

Hitomi [1977] introduced a new concept of production speed as a decision variable. A model with the production speed was developed to minimize production cost or production time. Optimization was used on a single item in a multistage production system to determine the optimal production speeds for all stages and its corresponding optimal cycle time. For multiple-item production systems, he also considered the minimization of the total flow time as a primary objective, and the minimization of total production cost as a secondary objective.

Rao and Hati [1978] developed a model to determine the optimum cutting conditions for a job requiring multiple machining operations. The major operations considered were turning, drilling and milling process. The Davidon-Fletcher-Powell method, coupled with the cubic interpolation method of one dimensional minimization was used to solve the model.

More recently, Yellowley and Gunn [1989] presented a more advanced analysis for both turning and milling operations. They showed that the optimal width of cut may be evaluated without a prior knowledge of the relevant tool life. The exception to this finding occurs when either a power or torque constraint was used. They demonstrated that in the presence of such constraints, the optimal widths of cut can still be evaluated without extensive computational effort. The optimal cutting conditions were determined using this proposed method.

A multi-criterion decision model was developed by Agapious [1992]. It incorporated a combination of the minimum production cost and production time requirements for optimization. A constant multiplier and weight coefficients were used to normalize the objective function. This multi-criterion decision was solved using the Nelder-Mead simplex method.

Moreover, maximum-profit cutting conditions were also considered by many researchers. Okushina and Hitomi [1964] discovered that in the case of continuous mass production, the profit obtained when machining at the minimum-cost cutting speed is not always maximum. They developed a model without any constraints to determine the optimal cutting speed which maximizes total profit in a given period of time. Wu and Ermer [1966] developed a method similar but their emphasis was devoted to the investigation of a range of optimum cutting speeds, instead of one theoretical optimum speed.

In summary, Table 2.2 presents an overview of the surveyed models in several aspects.

Model's name (Author)	Objective Functions	Decision Variables	Special Method Used	Solution Algorithm	Comment(s)
Gilbert (1950)	1. Min. Production cost	1. Cutting speed	*	Partial Differentiation	One variable is only considered
Armarego & Brown (1965)	1. Min. Production Cost	1. Cutting speed 2. feed	*	Geometric Programming	Depth of cut is fixed
Okushima & Hitomi (1964)	1. Max. Profit	1. Cutting speed	*	Partial Differentiation	*
Wu & Ermer (1966)	1. Max. Profit	1. Cutting speed	Basic Marginal Principles	Partial Differentiation	Range of optimum speed instead of the theoretical optimum
Boothroyd & Rusek (1976)	1. Max. rate of Profit	1. Cutting speed	Worker incentive scheme and batch production were considered	Partial Differentiation	*

Note * Means No

Table 2.2 Summary of the Selected Deterministic Optimization Model

Model's name (Author)	Objective Function	Decision Variables	Special Method Used	Solution Algorithm	Comment(s)
Petropoulous (1973)	1. Min. Production cost	1. Cutting speed 2. Feed rate	Primal and Dual Programme were formulated	Geometric Programming	CLA values of surface finish are introduced
Hitomi (1977)	1. Min. Production time 2. Min. Production cost	1. Cutting speed	-Single and Multiple stages system were considered - production speed was introduced	Partial Differentiation	Job Sequences & Total flow / cycle time were considered in multiple items case
Lambert & Walvekar (1978)	1. Min. Production cost	1. Number of passes 2. cutting speed 3. feed 4. depth of cut	*	Geometric Programming	The number of passes can be determined

Table 2.2 (Continued)

Model's name (Author)	Objective Function	Decision Variables	Special Method Used	Solution Algorithm	Comment(s)
Rao & Hati (1978)	1.Min. Production cost 2. Max. Production rate 3. Max. Profit	1. Cutting speed 2. feed	Multiple Operation	Davidon-Fletcher Powell method coupled the cubic interpolation	Milling, turning and drilling operation were considered
Yellowley & Gunn (1978)	1. Min. Production cost	1. cutting speed 2. feed 3. depth of cut	Multi- pass machining operations	Dual theory of geometric programming	Sensitivity Analysis of the constraints
Agapious (1992)	Combination of Min production cost and Max. production rate	1. Cutting speed 2. feed 3. Depth of cut	Multi-criterion decision making	Nelder - Mead Simplex method	The objective function is normalized thru a constant multiplier

Table 2.2 (Continued)

2.2.2 Probabilistic Models

Fenton and Joseph [1979] showed that the calculation of machining economics based on deterministic tool life yields inaccurate results. There have been some efforts to accommodate the stochastic nature of tool life in economic modelling. In these studies, the tool life is treated as a random variable whose probability distribution is parameterized by the cutting conditions.

Iwata et al. [1977] developed a probabilistic model to simultaneously determine the optimum value of cutting speed, feed, depth of cut and number of passes for a given total depth of cut to be removed, while considering the probabilistic nature of both the objective function and the constraints in the machining processes. They used the concept of dynamic programming and stochastic programming to solve the developed model.

Sheikh et al. [1980] summarized the results of probabilistic nature of the tool life and its effects to the tool replacement strategies and optimum cutting conditions. They considered the preventive, planned, scheduled and failure tool replacement strategies with single and multiple tool systems. The researchers also developed a tool reliability model using the Weibull distribution.

Billatos and Kendall [1990] focused on the replacement model for multi-tool transfer lines. They developed a methodology for the selection of a tool group and the impact of changing this tool replacement interval on tool changing cost. The electro-mechanical equipment failure and tool wear failure were taken into the account using exponential and Weibull probability distribution.

Zhou and Wysk [1992] introduced a concept called tool status measure (tool wear index). In tool status recording, the concern was focused on the fraction of tool life reduction rather than the end of the tool life. The expected tool wear rate was discovered to be approximately constant and inversely proportional to the expected tool life. Based on the tool wear index, the tool cost recovery and probabilistic models were developed to determine the cutting speed and replacement time for minimizing the unit production cost.

Du and Hui [1993] considered the expected unit production cost as the objective function instead of the actual unit production cost. The tool replacement strategies were developed as well as the cutting conditions using a probabilistic model. Moreover, utilization of idle time was considered by Agapious [1992]. He proposed an optimization method which utilizes the idle time to the full extent at all machining stations in the transfer machining system. The intention of this model was to improve tool life, thus achieving cost reduction.

Sakurai and Shimoda [1996] presents an optimal tool replacement procedure in automated multi-stage machining systems. A statistic model for the design of optimal tool replacement and cutting condition is constructed using the minimum production cost criterion.

Reliability of cutting tool concepts were considered by a number of researchers. In a paper by Pandit and Shiekh [1980], a statistical distribution of tool wear was formulated from experimental results, and the distribution of the tool life and the reliability function of cutter were derived. It was shown that the reliability of cutting tools at a certain time can be calculated from machining conditions and tool wear by the use of a reliability function.

In summary, Table 2.3 presents an overview of the surveyed models.

Model's name (Author)	Objective Functions	Decision Variables	Special Method Used	Solution Algorithm	Comment(s)
Sakurai & Shimoda (1996)	1. Min. Production Cost	1. Cutting speed 2. tool replacement	Determination based on K value	Statistic model was developed	Multi-pass Operation were considered
Iwata & Murotsu & Oba (1977)	1. Min. Production Cost	1. Cutting speed 2. Feed 3. depth of cut 4. Number of passes	*	Dynamic Programming & Stochastic Programming	Multi-pass Operation were considered
Sheikh & Kendall & Pandit (1980)	1. Optimal tool replacement strategies	1. Cutting Speed 2. Feed 3. depth of cut	Coefficient of variation	Partial Differentiation	Multitool machining systems were considered
Billatos & Kendall (1990)	1. Min. Production time	1. time replacement interval	Electro-mechanical failure was considered as well	A developed methodology	Reliability Concept was used

Note: * means NO

Table 2.3 Summary of the Selected Probabilistic Models

Model's name (Author)	Objective Functions	Decision Variables	Special Method Used	Solution Algorithm	Comment(s)
Chen & Wysk (1992)	1. Min. unit direct production cost	1. Cutting speed 2. Replacement time	Tool Status recording method was developed	Partial differentiation	*
Agapious (1992)	1. Min. Production cost	1. Cutting speed 2. Feed 3. depth of cut	Zero idling time	A computational algorithm was developed	Multi-stage machining system
Zhou & Wysk (1992)	1. Min. Production cost 2. Tool selection	1. Cutting speed 2. Tool replacement time	Expert system	Two dimensional search algorithm	An integrated system was developed
Du & Hui (1993)	1. Min. Production Cost	1. Cutting speed 2. Feed 3. Depth of cut 4. Tool replacement	Expected unit production cost using probability theory	Frank Wolfe Algorithm	*

Table 2.3 (Continued)

2.2.3 Fuzzy Models

Trappey and Liu [1988] proposed a fuzzy non-linear programming method to resolve the complexity of the manufacturing environment where non-stochastic and vagueness exist in most of the given information. The fuzzy set theory was applied to the mathematical model for the purpose of representing the problem. The fuzzy set theory was adapted to the objective function and constraints. A crisp non-linear optimization model was derived from the fuzzy model for solving the problem numerically. A fuzzy machining economics model was given as an example to find optimal cutting conditions.

Hanna [1996] presents a fuzzy Petri net approach for modelling of a CNC-milling machine center. A technique based on 9 fuzzy rules is developed to determine the cutting speed and feed for required surface roughness. In addition, an artificial neural network is also used combined with fuzzy Petri net.

2.2.4 Solving Algorithms

The optimization models for determining cutting conditions are usually non-convex and non-linear programming models. To solve these models, numerical algorithms are needed. The numerical algorithm must be reliable, accurate, relatively insensitive to the initial value, and relatively inexpensive to run.

According to the literature survey, a number of numerical methods have been developed. However, none has been proven to be superior than that of the others. Moreover, different methods may give different results for the same model. Duffuaa et al. [1993] showed that solving the same model using two different methods result in different solution by as much as 29%. Therefore, attention must be paid to not only the model, but also on the numerical method.

Okushima and Hitomi [1964], Wu and Ermer [1966] have developed optimization models for machining conditions and utilized simple partial differentiation to obtain their results. This method only gives the optimum cutting speed.

Petropoulous [1973] and Walvekar and Lambert [1978] have demonstrated the use of the geometric programming technique as a tool for machining optimization in relatively simple problems.

A combination of linear and geometric programming was employed by Ermer and Kromodihardjo [1981] to minimize cost in multipass turning operation.

Iwata et al. [1972] used the equivalent deterministic problem SUMT (stands for Sequential Unconstrained Minimization Techniques) . Hati and Rao [1978] also utilized SUMT to solve a probabilistic and a deterministic model formulated for a multipass turning operation.

Iwata et al [1977] utilized an algorithm based on dynamic and stochastic programming to determine the optimum cutting conditions.

Kimber et al. [1982] reviewed five optimization techniques and compared only two of them, namely, SUMT and the exterior penalty function method. Duffuaa et. al. [1993] conducted a comprehensive analysis. They evaluated six algorithms, namely, GRG (Generalized Reduced Gradient), SUMT, geometry algorithm, dynamic programming, complex algorithm and constraint Rosenbrock (Hill-Climbing Algorithm). They eventually found that the GRG algorithm and Hill-Climbing algorithm are the best algorithm to solve the machining optimization models in term of their reliability, precision convergence, sensitivity to input vectors and its preparation effort. The Hill-Climbing algorithm will be employed in this thesis to solve the proposed optimization model.

2.3 Motivation of the Research

Although as reviewed in last section, plenty of time and human effort are devoted for the research on cutter selection and cutting condition design, however, it's worth pointing out that most of the methods and systems developed so far have following drawbacks:

- Complete and precise information is needed for both cutter selection and cutting condition design.
- For cutter selection, instead of commercial products used in the shop floor, only some generic cutter features (such as cutter nose radius and thickness) are suggested.
- As a computer-aided software package, their functionality are incomplete or limited.
- The system is not flexible enough. Sometimes it fails to provide a feasible solution, because of internal conflicts.

This thesis focuses on two machining process planning problems:

1. <u>Cutter and Cutter Material Selection</u>

The selections for the most suitable cutter and cutter material are very difficult. Firstly, there are more than two thousand different cutters in the market and many new cutters are being continuously marketed. This makes it almost impossible for an engineer or a machinist to investigate all the available cutters. Secondly, there are many factors involved in the selection such as work material, machine tool, machining requirements and cutter availability. These factors for cutter selection are often correlated, vague and contradictory. For example: one cutter might be the best for machining one particular work material but it does not provide satisfactory machining requirement(s). The traditional cutter selection methods such as database searching may fail to provide a feasible solution. Thirdly, true experts are rare and expensive. Therefore, a more practical system must be developed to accommodate these factors and to resolve these problems.

2. Cutting Conditions Design

Most deterministic or probabilistic optimization models emphasized the quantitative viewpoint for process planners. Only a set of "optimal" cutting conditions was given and optimality was not well justified. Hence, in today's metal cutting industries, most of these models are not being applied.

All of the models in the previous studies required exact values to construct their objective function and constraints. In industries, however, values are not always considered to be exact, but rather to be imprecise or vague. For example: the process planner wants to have a "good" surface finish. But "Good" is a word that is actually imprecise and vague. It's hard and impractical to assign "good" a exact value, because it's hard to judge which one is more feasible between surface roughness of 2μm and 2.1μm. Similarly, it is also hard to differentiate "a very good" and " a good" surface finish. Therefore, the industries are not looking for a quantitative answer of optimum cutting conditions but rather a qualitative solution.

Nowadays, most systems are developed with incomplete or limited functionality. Researching systems just focus on some specific points. They don't design cutting condition or estimate cost, and only suggest generic cutter features but not exact commercial products. Even for commercial software developed by a specific supplier, they are not flexible enough for complex practical environments and sometimes fail to offer operator a satisfactory solution. Moreover, their databases only contain their own products and can not be accessed. Users must buy upgraded versions for new incoming products.

The objective of this thesis is to resolve this practical situation which is full of uncertainties and vagueness. In this thesis, decision-making process for cutter and cutting conditions selection system is designed to be more human-oriented. The rules for decision making process are all represented by the fuzzy set theory. In order to enhance the applicability of this research, a prototype software system with complete functions (including rotating cutter selection, cutting condition design, cost estimation, database management) is designed and developed by applying the method of Object-Oriented Modeling. The proposed system has following distinct features and major functions:

1. By applying fuzzy logic theory, at least one set of feasible solutions is guaranteed for any practical environment.

2. Using Borland C++ windows programming, the proposed system is very user-friendly and human-oriented with graphic interfaces and illustrations.

3. According to the objective of optimization:

    a. Maximization of Unit Production Rate

    b. Minimization of Unit Production Cost

    c. Maximization of Profit

The system not only recommends one or more optimal schedule(s), but also evaluate the user's specific machining parameters, which could be used for skill training by providing a graphic comparison with the optimal one(s).

4. Most commercial cutter products are included. New products could be added into the database whenever necessary.

Table 2.4 shows a comparison between the proposed system and some other typical systems.

System Name or Developers	input information	Cutter and Grade Selection			Cutting Conditions Optimization			
		Methods	**Insert**	**Grade**	**V**	**f**	**N***	**Methods**
Santochi&Dini (1996)	exact	artificial neural networks	yes	yes	No	No	No	N/A
Narrowing (1993)	exact	Knowledge Based searching	yes	yes	Yes	Yes	No	Geometric Programming
COMP (1992)	exact	Database + ECT equation	no	yes	Yes	Yes	No	ECT equation
METCAPP/1 (1992)	exact	Sequential Database searching	no	no	Yes	Yes	Yes	Differentiation
COROGUID (1990)	exact	Database searching in several levels	yes	yes	Yes	No	No	Differentiation
EMDB (1990)	exact	Database searching	yes	yes	Yes	No	No	N/A
Proposed (1997)	*fuzzy*	*Fuzzy Set Theory Object-Oriented Model*	*yes*	*yes*	*Yes*	*Yes*	*Yes*	*Fuzzy non-linear Programming*

* N = Tool replacement time

Table 2.4 Comparison among the developed systems

CHAPTER III

THE PROPOSED METHODS

3.1 Object-Oriented Modeling (OOM) and Implementation

The system is developed using Object-Oriented Modeling (OOM), which includes Object-Oriented Analysis (OOA) and Object-Oriented Design (OOD) and is implemented using Object-Oriented Programming (OOP) language C++.

3.1.1 Overview of OOM

With an ever increased system complexity, many difficult technical and managerial problems are raised in developing large-scale software systems, which cause software crisis. To overcome the crisis, two key methods are developed: "structuring" and "abstraction". Structuring enables us to decompose systems into components or views, and relate one to another. Abstraction gives us a way of designing away from the details of the system, and this aids comprehension. Object-Oriented Modelling(OOM) is a formal method for structuring and abstraction in software engineering. It provides a new way of thinking about problems using models organized following real-world concepts. It assists the complete cycle of a system from analysis to design and to development.

OOM is a conceptual process independent of programming language and data until the final stage. Its greatest benefits come from helping designers, developers, and customers express abstract concepts clearly and communicate them with each other. The fundamental element of OOM is *object*, which encapsulates both data structure and its associated behavior in a single computing entity existing at a higher level than normal procedure or data structures. An object may be defined as a concept, an abstraction, or a set with crisp boundaries and meaning for the problem at hand [Rumbaugh, 1991].

OOM decomposes a system into a number of objects and establishes the relationships among them. In addition, larger objects can be further decomposed into smaller components. This parent-child hierarchy is used to connect objects to form an OOM system as shown in Figure 3.1. At the higher levels, a system or component may be represented by a single object called a parent. Each parent may have several children represented by objects. The support for object-oriented decomposition is what makes OOM different from traditional structured design: the former uses classes and object abstractions to structure systems logically while the latter uses algorithmic abstractions [Booch 1994].

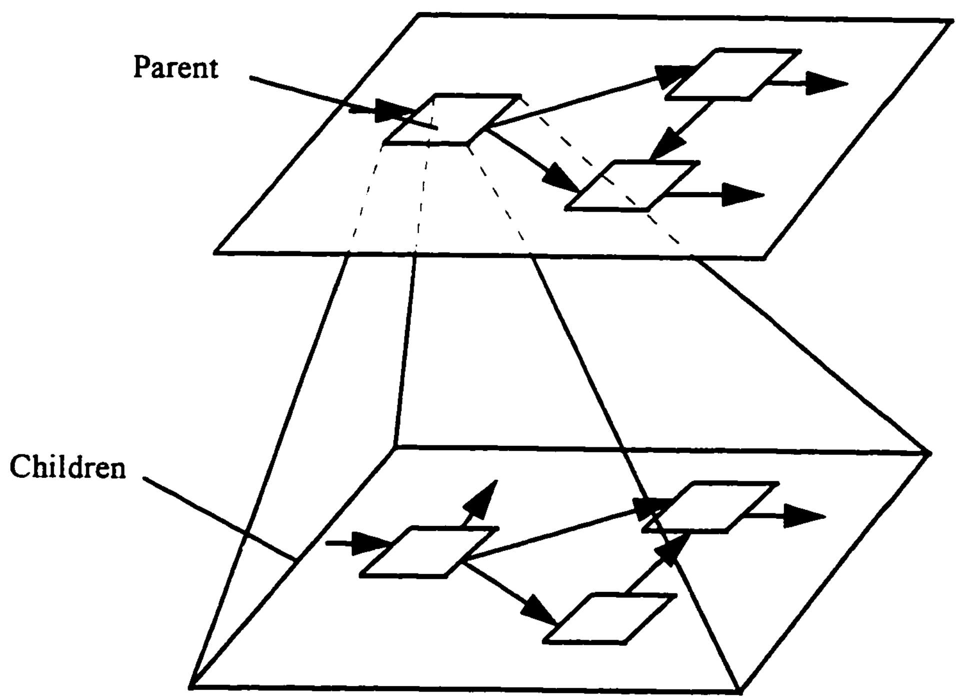


Figure 3.1 Levelling object diagram in OOM [Peterson, 1990]

OOM constructs complex systems with characteristics of *encapsulation*, *polymorphism* and *inheritance*. Encapsulation is the ability of an object to hide information which is internal to it from outside access. It also implies the ability to bind data with operations on the data. Polymorphism allows for generic operations to take on different forms in different classes via dynamic binding, where the method to be executed is identified at runtime. Inheritance is the ability of an object to derive its features from another object or from a number of objects. These concepts can be used in isolation, but together they complement each other synergistically. The use of the OOM exploits the expressive power of object-oriented programming languages, such as C++, Smalltalk. The application of OOM also greatly enhance the reusability and extendibility of the system.

A number of OOM methodologies have been developed. In this research, the OOM approach proposed by Rumbaugh [1991] is used, which has a unified graphical notation for analysis, design, and implementation.

3.1.2 Basic OOM Operations

OOM provides a mechanism for representing information by means of models organized around objects and concepts. An *object class* describes a group of objects with similar properties, common behavior, and common relationships to other objects. Object classes typically contain both attributes and operations. The attributes are data values held by objects. On the other hand, operations are functions or transformations that may be applied to, or by, objects in a class [Rumbaugh, 1991]. Along with objects, object classes, and object operations, this methodology relies on *abstractions* to model real-world constructs. An abstraction consists of focusing on essential, inherent aspects of an entity and ignoring its accidental properties [Rumbaugh, 1991]. There are several other Objected-Oriented modeling principles that can be used to design and enhance the proposed model, including generalization with disjoint or overlapping subclasses, classification, aggregation and association. Additional methods that can be applied to formalize the design of classification models include derived attributes, constraints, ordered lists and restriction. The following sections describe how these OOM organizing principles are applied to proposed model.

1. Generalization with disjoint subclasses

A hierarchical classification divides a large collection of items with mutually exclusive data into mutually exclusive families based on the item types. Only one branch may be selected for each family, and each branch can be further divided. For example, rotating machining operations can be divided as either milling, drilling or boring. Furthermore, a milling operation can be divided either end milling, face milling, and so on.

In OOM, mutually exclusive classification can be modeled by using the *generalization abstraction with disjoint subclasses*. Generalization is the relationship between a class and one or more refined versions of it [Rumbaugh, 1991]. The class being refined is called the superclass and each refined version is called a subclass. Figure 3.2 illustrates the concept. A cutting tool may be either solid cutter or indexable cutter. Thus, a superclass "Cutter" could be generalized from two subclasses "Solid Cutter" and "Indexable Cutter". The notation for generalization with disjoint subclasses to represent mutually exclusive classification is a hollow triangle connecting a superclass to its subclasses as shown in Figure 3.2.

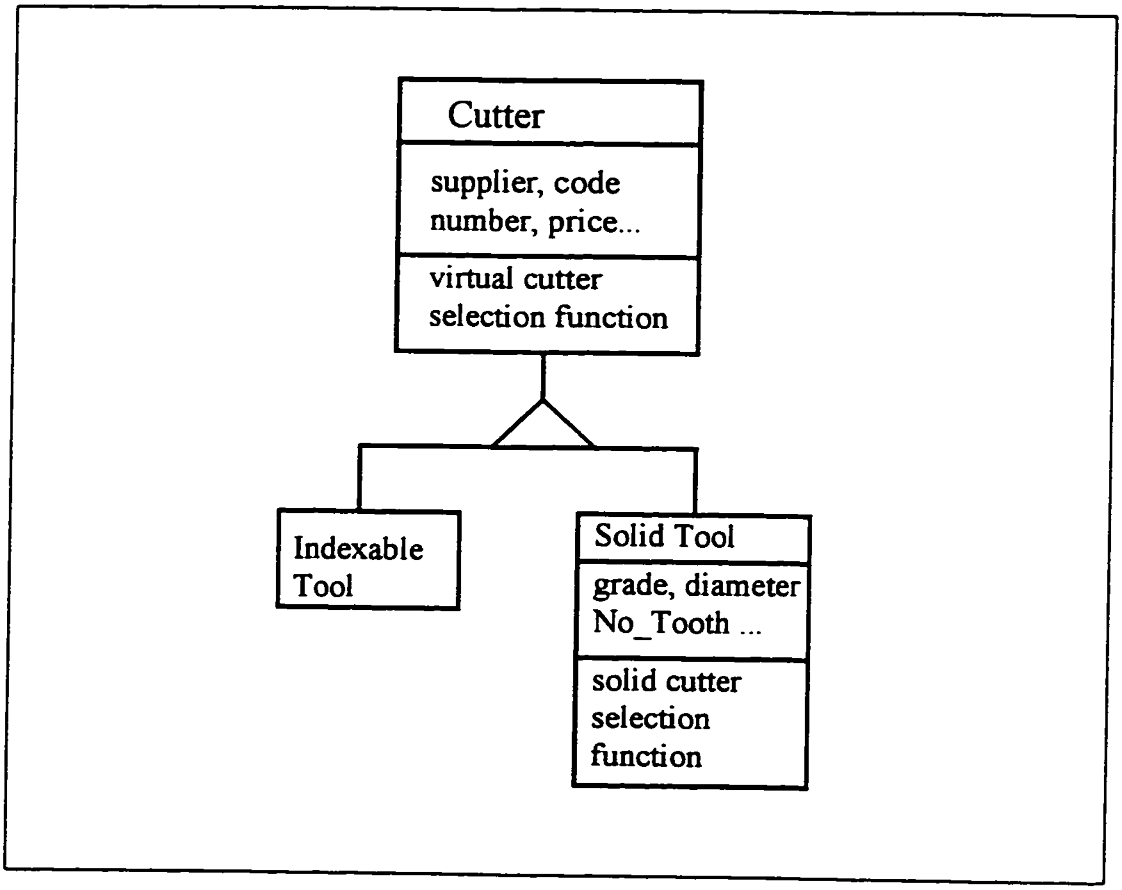


Figure 3.2 Generalization with disjoint subclasses

In class "Cutter", some common features extracted from all kinds of cutters are defined. Their associated behaviors are defined as class member functions. These functions may be virtual abstract functions containing some common or default activities and could be overwritten by their refined versions in subclass "Indexable Tool" or "Solid Tool". For these two subclasses, in addition to inheriting all variables and functions defined in their supperclass, each of them defines its own specific features and their corresponding functions.

The *generalization abstraction with disjoint subclasses* shown as Figure 3.2 is implemented in C++ as follows:

class Cutter{

```
protected:
        float price,diameter;
        int number,lead_angle;
        char *supplier,*code,*solidtype;
public:
        virtual void selectdiameter();
        void selectleadangle();
};

class SolidCutter: public Cutter {
private:
        char *grade,*pitch_type;
        int No_Tooth;
public:
        void selectdiameter();
        void selectpitch();
        void selectgrade();
};
```

When a class is defined, instantiation assigns its values. For example, an instantiation of cutter class could be initialized with following value:

```
solidcutter sample1;
sample1.code="R216.33-04030-AA07N";
sample1.price=55.99;
sample1.number=1;
sample1.solidtype="solid";
sample1.supplier="Sandvik Coroment";
```

```
sample1.pitchtype=coarse;
sample1.diamter=4 mm;
sample1.No_Tooth=3;
sample1.grade=NI45;
```

Please refer to Appendix B for a complete class and function prototype definition.

2. Generalization with overlapping subclasses

Some subclasses may overlap. Allen [1984] describes the overlap as non-exclusive classification, in which multiple paths may be traversed. As shown in Figure 3.3, rotating cutters could be divided in subclasses: "drilling cutter", "face mill cutter", and "end mill cutter". Furthermore, drilling cutters could be divided into "short hole drill" and "deep hole drill", and so on. It is known some of the indexable cutters can be used for both milling and drilling. In this case, the subclasses overlap.

In OOM, the overlap can be best modeled as *generalization with overlapping subclasses*. The overlapping subclasses is depicted by a solid triangle. For above example, the model could be implemented in following C++ code:

```
class Cutter;
class drill: public Cutter {
protected:
        float depth;
        char *shank,*fluid,*drill_type;
public:
        void drillselection1();
        ...
};
```

Combining the generalization with disjointed subclass and with overlapping subclass, one can derive any kind of rotating tool. For example, a solid drilling tool can be derived as follows:

```
class soliddrill: public solidcutter, public drill {
public:
        void soliddrillselection1();
};
```

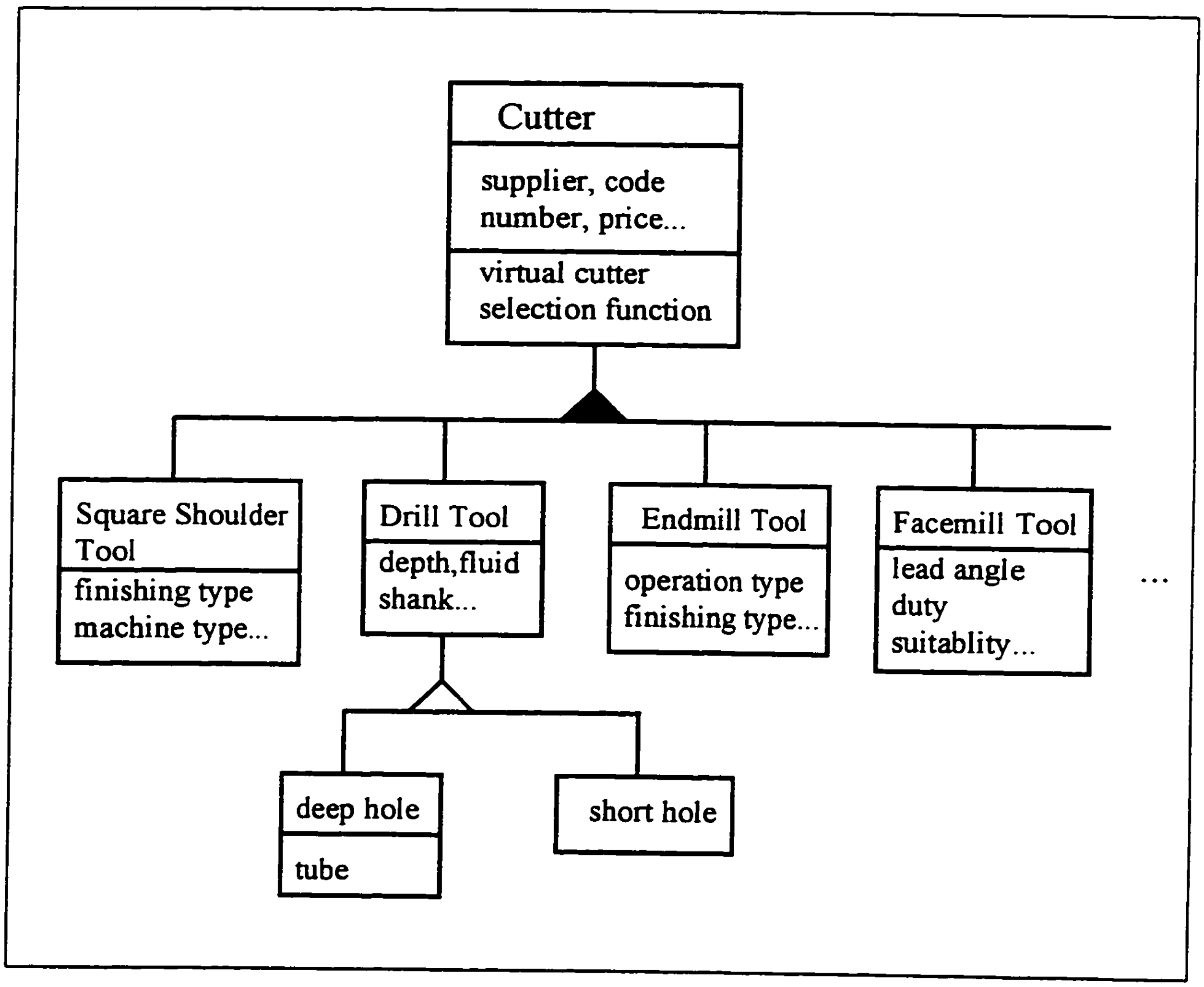


Figure 3.3 Generalization with overlapping subclasses

3. Classification and derived attributes

One of the most capable features of OOM is to use derived attributes for classification. Figure 3.4 illustrates the process for classification of a part. In this example, the available input information includes the Length and Diameter of a clindrical workpiece. Then the expression LD=Length/Diameter is derived.

In OOM, this process can be modeled using the abstraction *Classification* with *derived attributes* [Rumbaugh, 1991]. Classification means that objects that share the same attributes are grouped into a class. In the above example, when designing a cutting condition for a cylindrical workpiece, one should consider not only the size of workpiece(the length and the diameter) but also the shape of the workpiece(described by the ratio of length and diameter). Therefore, the derived attribute, LD, is introduced as the primitive attributes of the class.

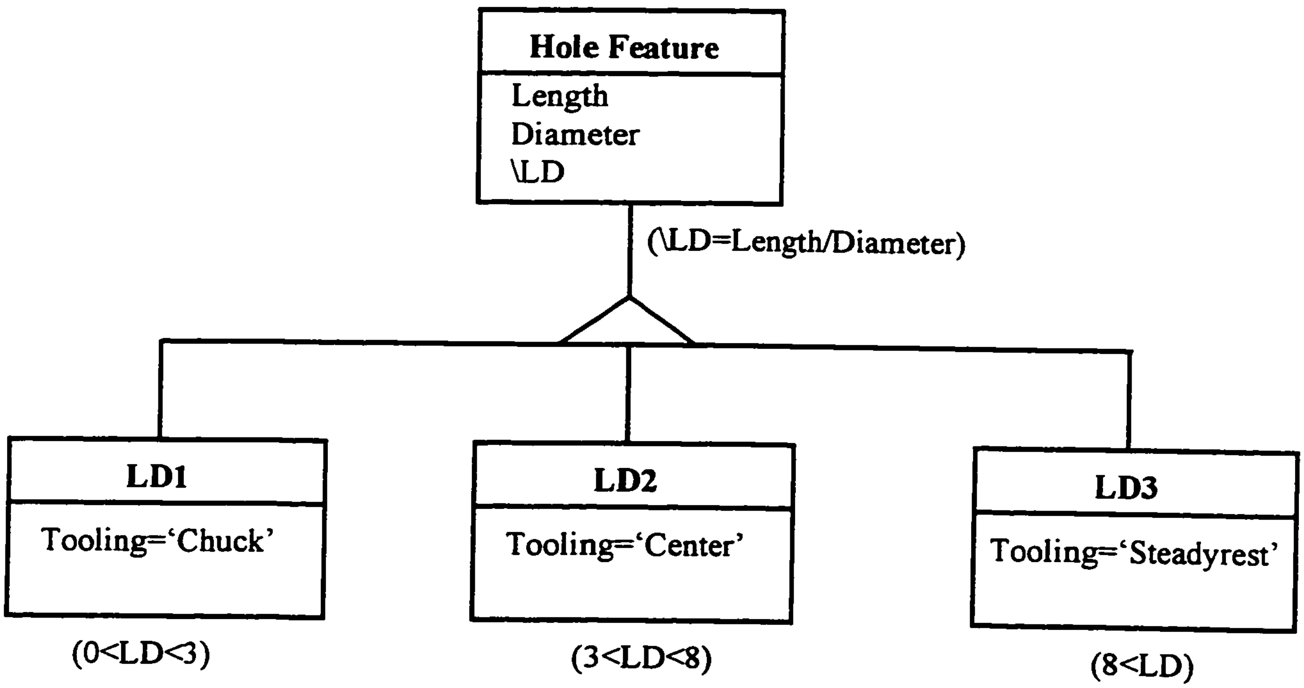


Figure 3.4 Classification, derived attributes, and generalization as restriction

In general, a derived attribute is defined as a function of one or more attributes [Elmasri, 1994]. Mathematical expressions of the derived attributes can be evaluated and stored just like an ordinary attribute. Although it is redundant, derived attributes may be included in the object model to represent important concept. The mathematical expression that determines the derived value is denoted as a *constraints*, which is dictated in a declarative manner within braces below the class box. The notation for a derived attribute is a slash in front of an attribute. In Figure 3.4, the derived attribute LD is a result of the constraint {\LD=Length/Diameter}.

4. Generalization as restriction

Generalization as restriction is the technique to emulate decision rules in OOM, as it places constraints on the attributes of superclasses. In this case, class membership in a subclass is defined by the same rule, and all objects whose values satisfy the rule belong to the class. For example, Figure 3.4 shows how the ratio of length and diameter rule can be modeled. A constraint is placed on each of the subclasses: if an object is to be instantiated in the subclass 'LD 1', its LD attribute in the superclass must be less than 3 units. Subclasses 'LD 2', and 'LD 3' have similar constraints that are placed on the LD attribute in the superclass Size.

5. Aggregation

In OOM, *Aggregation* abstraction allows the designer to have a complete classification model that incorporates mutually exclusive branching, non-exclusive branching, instantiation of data into attributes, evaluation of mathematical expressions, and execution of simple decision rules. Aggregation is a "part-whole" relationship in which objects representing the components of an assembly are associated with an object representing the entire assembly. This is different from generalization concept which represents mutually exclusive or non-exclusive classification in one large collection. Aggregation is denoted by connecting object classes with a straight line. A small diamond indicates the assembly end of the relationship. Figure 3.5 illustrates aggregation. The superclass "Indexable Tool" could be assembled by subclasses "Toolbar" and "Insert". Many of these components in turn represent classes that represent superclasses or aggregates of other classes.

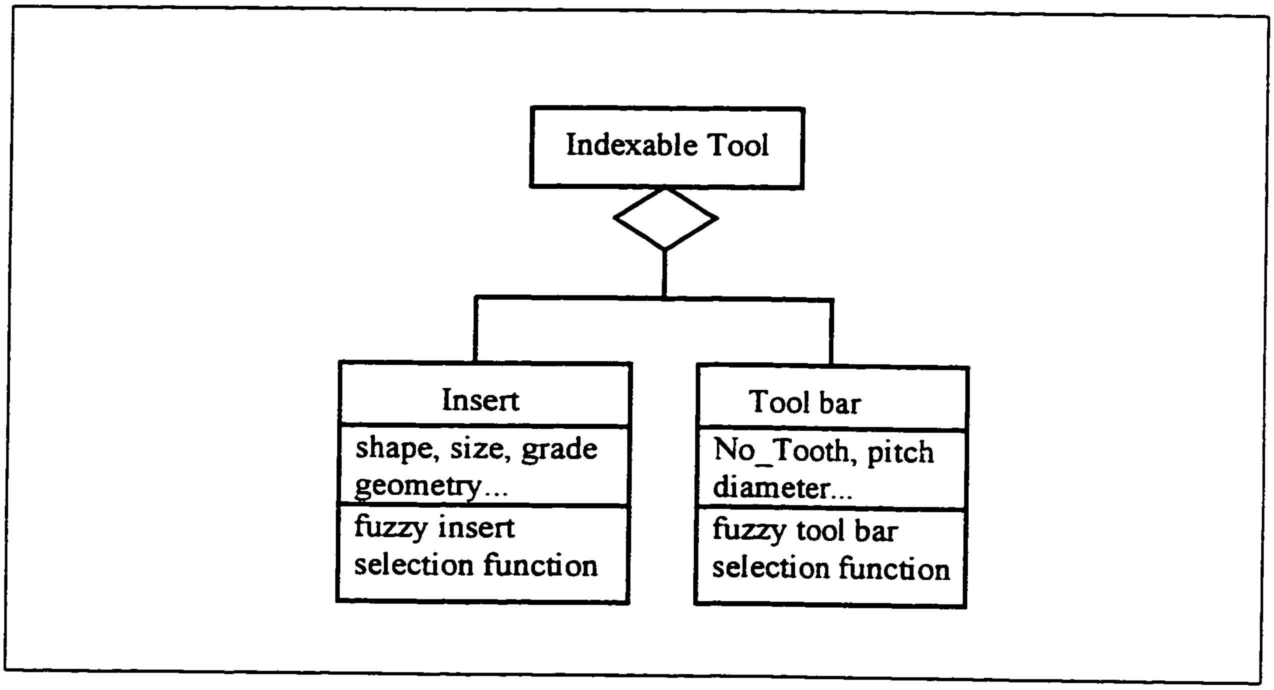


Figure 3.5 Aggregation

Aggregation could be expressed in C++:

```
class Toolbar: public Cutter{
        int No_Tooth;
        char *pitch_type;
        float diameter;
```

```
};
class Insert: public Cutter{
        char *grade,*shape,*geometry;
        float size,thickness,radius;
};
class IndexCutter: public Cutter{
        Toolbar a;
        Insert b;
};
```

6. Enhancing Model Functionality with association, role names, and aggregation

OOM principles can further formalize the design of models by minimizing redundancy and providing a mechanism to enhance the integration of the system with other systems. Additional organizing principles we found to be helpful included *Association* and *Role Names*. In addition, we found Aggregation, as described above, to be helpful in making the model more efficient by eliminating redundant object classes.

An association describes a group of physical or conceptual connection between objects from several independent classes [Elmasri and Navathe, 1994]. Associations can be described as a 'looser' form of aggregation in that aggregation is best expressed as the part-whole relationship whereas association is expressed as a 'linking' of two independent object classes. Associations serve an important role in relating parts of the model with other parts of the model that are not connected through aggregation. It could be further modeled as a class. Consider the model illustrated in Figure 3.6, class Insert represents an object where all possible combinations of insert attributes, such as size, geometry, grade, could be stored. The toolbar information such as pitch type, dimension, number of tooth are kept in the model headed by the aggregate class Toolbar.

The connection that assigns specific insert objects with its toolbar is the association class 'MATCH'. It contains the mounting information between insert and toolbar such as feed direction, lead angle and matching rules. In practice, associations may be binary, trinary, or higher order.

Association also serve to easily connect the model with other external object models, or resulting databases used in market.

Other abstraction techniques, such as *role name*, *qualification* and *ordering list*, could also be used to eliminate redundancy. A role name uniquely identifies one end of an association or an aggregation. It is a helpful tool in reducing redundancy.

A designer can take advantage of aggregation to eliminate repetitive object classes in a model.

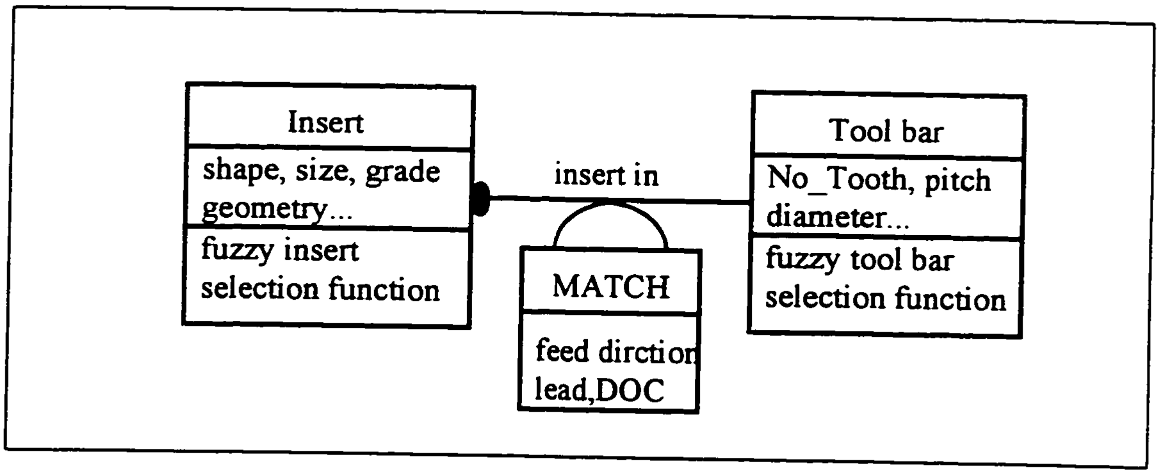


Figure 3.6 Association

3.1.3 OOM Behavioral Principles and Implementation

Although almost all software developed by Object-Oriented language could be implemented by traditional programming language, the former is easy to maintain, extend, reuse and obtain high efficiency. In conventional development method, it's widely recognized that the maintenance stage often accounts for major costs. But for OOM, software is highly structured as a collection of relatively discrete classes in which data and functions are bound together. Any changes in one class will seldom effect others. In OOM, one object may inherit properties from another. This is both beneficial in problem structuring and in achieving efficient implementing code. OOM is usually used for precise information. For uncertain events, OOM classes create additional fuzzy attributes and operations to handle them.

Figure 3.7 illustrates the application of OOM in rotating machining operation domain. All major factors involved are classified in classes. The basic OOM principle introduced in Section 3.1.2 are employed comprehensively to link these classes and establish the model. Please refer to Appendix B for a complete class and function prototype definition.

Figure 3.8 is the critical portion extracted from Figure 3.7 to briefly illustrate how cutter selection works. Based on fuzzrised input information in "Job" class, membership function functions in "Insert" and "Toolbar" class could select the cutter feature one by one. For example, based on machining type, workpiece material, depth of cut, feed rate, select_geometry() function could select proper geometry. Other features of toolbar and insert could be selected in same way. Some features of an insert are related and cross-defined. For example, since the shape of insert is round, the insert nose radius will be automatically same as insert IC without further selection. In the meantime, these functions also select a few alternative toolbar and insert objects, in case some of them might not be available or matchable. These toolbars and inserts may have different dimension and suppliers. The optimal insert may not guarantee to mount on optimal toolbar. Functions in "MATCH" associate class will match available toolbars and inserts with same IC, feed direction, depth of cut, to generate the optimal combination. Because the system is feature-oriented, even products from different suppliers are able to match each other, only if they have identical dimensions.

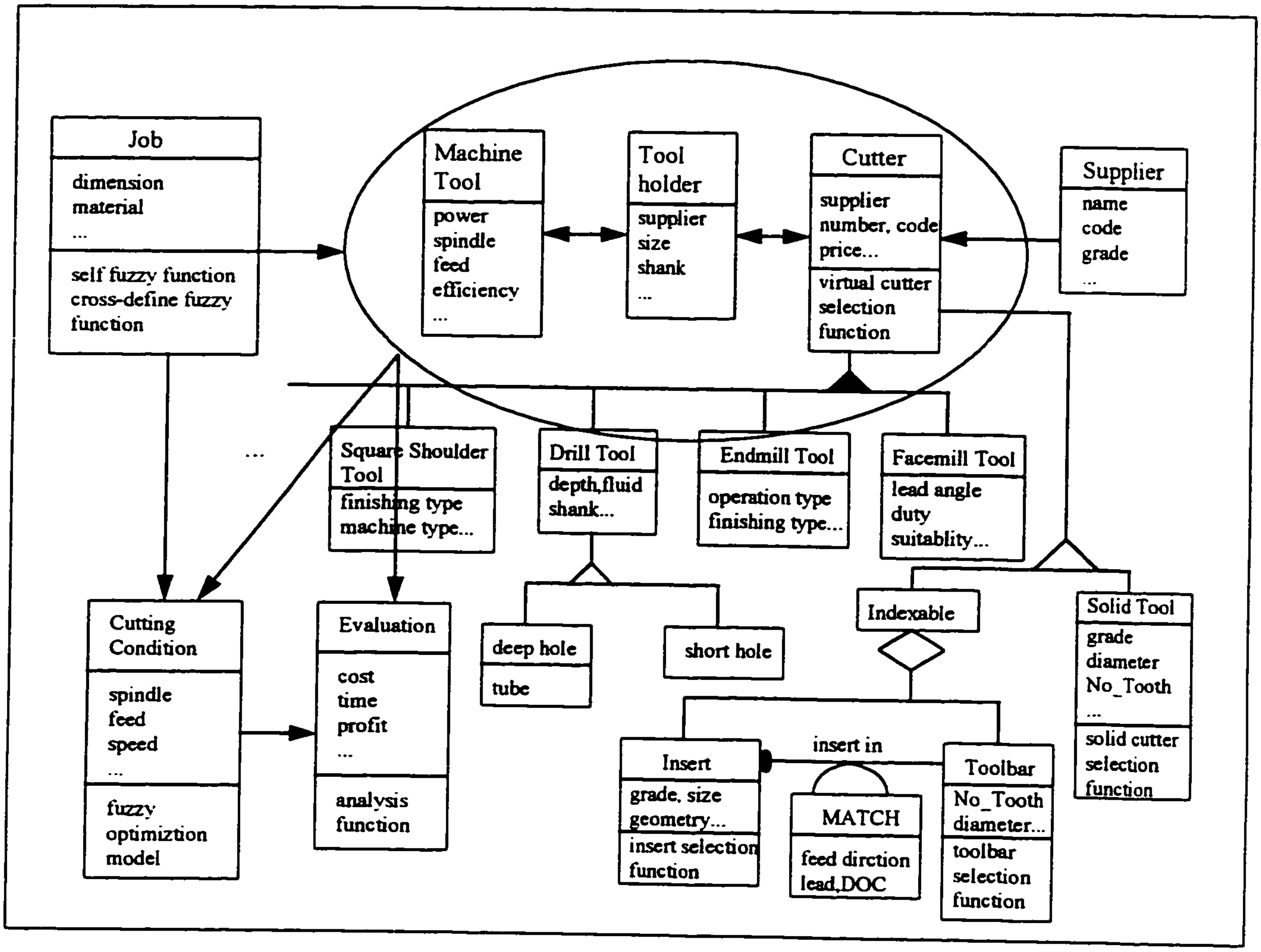


Figure 3.7 Illustration of OOM in rotating operation

The proposed system is not only to develop a commercial system for rotating operation, but also provide a platform for people who are interested in developing such kinds of systems in the future. Complete rotating operation class definitions and their relationship are shown in Figure 3.7. User could reuse these classes, their attributes and member functions by simply instantiating an object or deriving user's own machining object class based on these defined classes in Appendix. Similarly, the object-oriented classes for other machining operations could be easily extended in same manner.

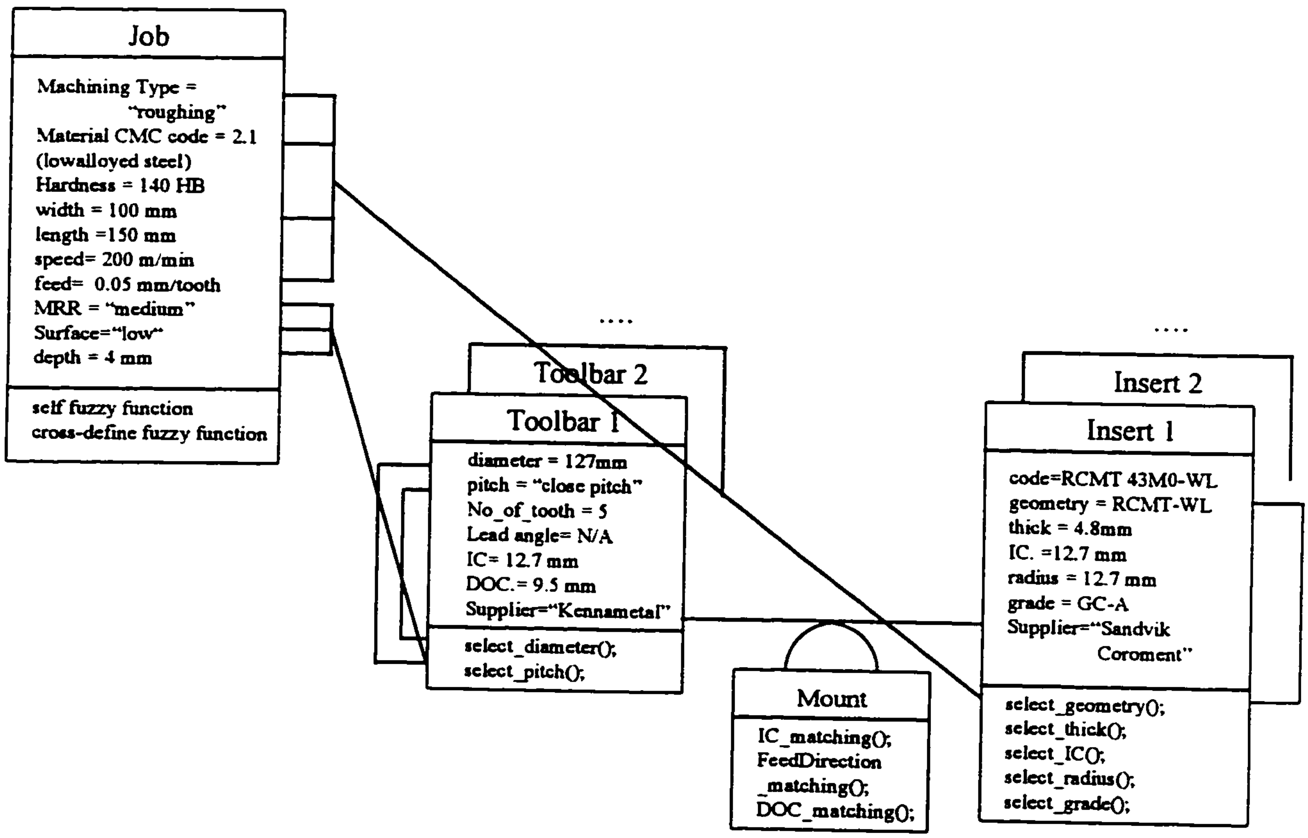


Figure 3.8 Illustration of cutter selection

As pointed out earier, commercial cutter selection systems have a major limitation: they are product-dependent . Hence they cannot deal with more than one product line. For example, the Grade Advisor works with Kennametal product line only and Coroguid works with Sandvik product line only. In addition, they cannot adapt to dynamic changes of the product line. For example, when a new product is added to the product line, the existing version of system must be upgraded in order to adapt the new product. For rotating machining operations involving cutters, toolbars and toolholders, there is an additional problem of searching for matched products (cutter matches toolbar, toolbar matches toolholder).

The presented system is an OOD system and can effectively resolve the limitations mentioned above. First, the system is independent on specific product line and can deal with multiple product lines. It is known that cutters, toolbars and toolholders are coded according to either ISO or ANSI standard. For example, a Sandvik milling cutter insert is coded in 10 fields:

T-MAX-145 S E E R 12 04 A Z - WL

where, the first field "T-MAX-145" is the Sandvik product code,

the second field is the shape code and "S" implies the cutter is square in shape,

the third field is clearance angle code and "E" implies the angle is 20°,

the fourth field is the tolerance code and "E" implies small tolerance,

the fifth field is clamp type code and "R" implies screw clamp,

the sixth field is the size code and "12" implies the size is 12 mm,

the seventh field is the thickness code and "04" implies the thickness is 4 mm,

the eighth field is the lead angle code and "A" implies the lead angle is 45°,

the nineth field is the tool nose radius code and "Z" corresponds to 4 mm,

the tenth field is the geometry code and "WL" implies waveline.

In the presented system, cutters are considered as a class and each field is considered as a subclass. The search for a cutter is done by matching the field and then matching the product in database. Thus, it can process multiple product lines from different manufacturers as long as their product database is available. For example, a user needs to use a cutter with a clearance angle of 20°, a size of 12 mm, and a tool nose radius of 4 mm. Then, the cutter above can be used together with many other cutters. Furthermore, if the user prefer to use Sandvik product and screw clamp with a lead angle of 45°, then the cutter above is only one applicable. On the other hand, if the user prefer to use Kennametal product, the equivalent product is: SEC -424. From this example, it is seen that using the OOM principles, the presented system depends only on the standards (ISO and ANSI standard). Since the cutters, toolbars and toolholders are coded based on the standards, the system handle multiple product lines.

More detailed explanation for proposed OOM system could be found in each corresponding sections of Chapter 4.

3.2 Application of Fuzzy Logic

3.2.1 Review of Fuzzy Logic

In machine shops, the process plan is always restricted by a number of constraints imposed by workpiece, the machine tool, cutting tool and the product design specification. The generated process plan must simultaneously satisfy all the objectives and constraints as shown in Figure 3.9. Determination of these constraint specifications, objectives and user's preferences, involves a lot of human decisions. On the shop floor, operators always work with information which is often imprecise and incomplete. [Trappey and Liu, 1988]. Due to the partition of the responsibilities and lack of interaction, the effectiveness and efficiency of the human decision are always affected by these fuzzy information.

To deal with these uncertainties, various methods have been developed to complement and fill in the gap left in the field of knowledge representation. **Fuzzy Logic** is one of the methods. Fuzzy logic was first introduced by Zadeh in 1963. It was developed to improve the oversimplified 0 & 1 logic, thereby developing a more robust and flexible method to solve real- world complex problems.

The fuzzy logic is well known and has been applied in many fields such as operation research, management science, control theory, artificial intelligence, expert systems and human behaviour.

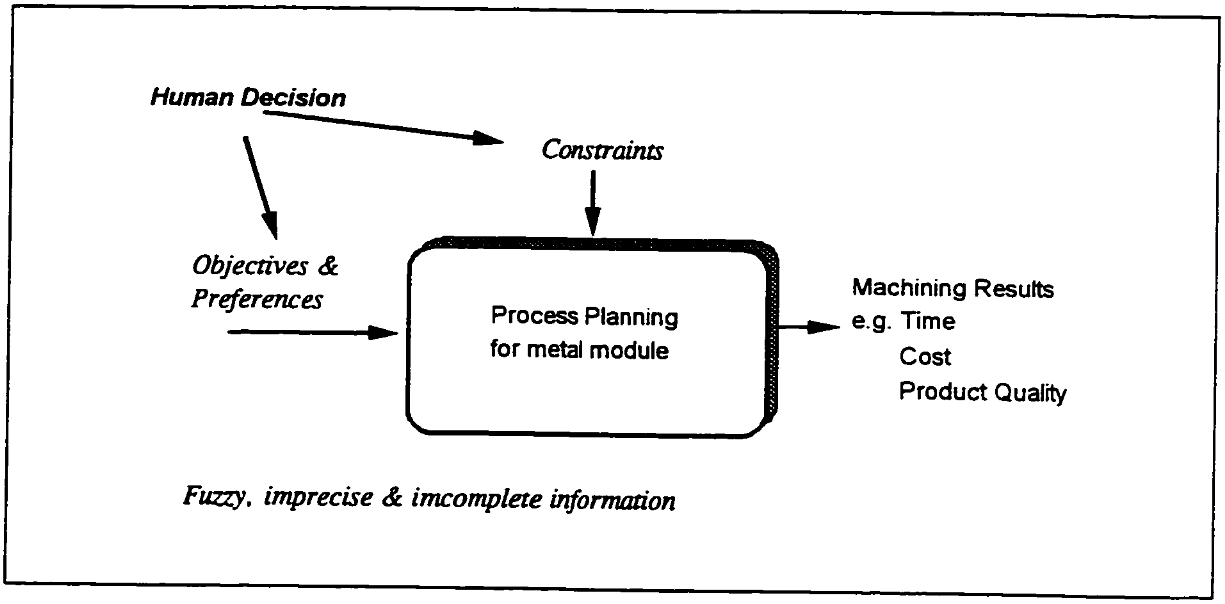


Figure 3.9 A block diagram of objectives and constraints in process planning

Following input information are usually uncertain or incomplete:

1. Workpiece material
2. Machine tool
3. Machining type
4. Machining Removal Rate
4. Machining parameter
5. Cutter preference

The output information, such as applicability of selected cutter and evaluation of machining schedule, could also be expressed fuzzily.

Fuzzy set theory was first introduced by Zadeh in 1963. It revolutionized the simply traditional 0&1 logic and made a huge progress in lots of scientific fields, such as operation research, control theory, artificial intelligence, although it is still controversial in pure mathematics.

In this thesis, the fuzzy logic is used as a tool to deal with the cutter selection and cutting conditions design. In this approach, uncertain events or information are described by means of a fuzzy degree (also called membership function).

Briefly, if A is a fuzzy concept (or uncertain event), and is a subset of the universal set U, then, A can be described by:

$$A = \{ X / U_A(X) \}$$

where the fuzzy degree $U_A(X)$ lies between 0 and 1, that is $0 \le U_A(X) \le 1$, while " 0" means certainly No and "1" implies certainly Yes. Figure 3.10 illustrates the concept of a fuzzy classification compared to the ordinary classification [Du and Hui, 1993]. The h_i, i = 1,2,3 are the fuzzy decision sets which are classified based on their membership functions. More theoretical and practical information about fuzzy set theory can be found in [Kaufmann, 1975].

3.2.2 General Form of Fuzzy Membership Function

In the machining process planning stage, usually only "rough" ideas and desires on the limitations and constraints are available. Therefore, the inputs are often vague, imprecise and sometimes incomplete. For example, a process planner wants to machine the workpiece at a high cutting speed. But, the "High" speed is not clearly defined. Also, the inputs are sometimes contradictory to each other. For example, in roughing, the planner might require a *good* surface finish. This is contradictory because the surface finish is not important in roughing. These kinds of information often make the process planners or existing software systems fail to give a feasible solution. The system transforms each input to be a fuzzy input by assigning it a fuzzy membership function.

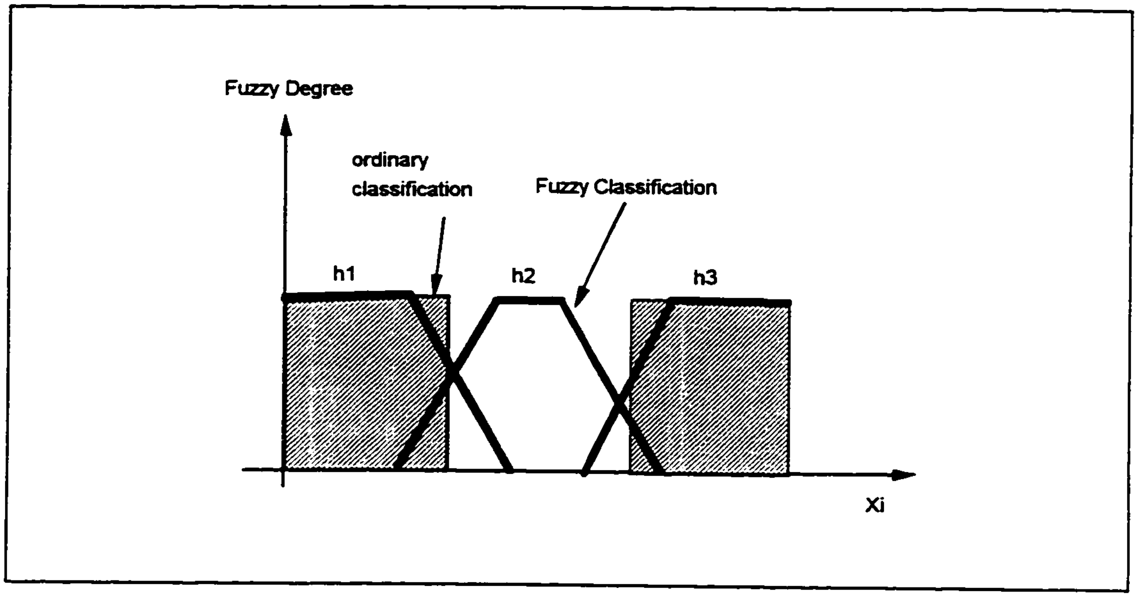


Figure 3.10 Comparison between the ordinary and fuzzy classification

1. <u>In Tabular Form</u>

This technique is designed to accommodate the fuzziness of input variables which are discrete such as machining type. For example, in most of machining handbooks and current industrial catalogues, the machining type is divided into five categories:

MT1 = extreme finishing

MT2 = finishing

MT3 = light roughing

MT4 = roughing

MT5 = heavy roughing

The classification of a given operation into a distinct category is difficult because there is no "exact" differentiation between extreme finishing and finishing. An operation with depth of cut 0.7 mm is classified as extreme finishing. Must an operation with depth of cut 0.71 mm be classified as finishing? Therefore, the machining type classification is actual fuzzy. The system assigns a fuzzy degree to each machining type in its fuzzy region to represent the fuzziness. Their fuzzy membership functions are defined in Table 3.1.

The fuzzy degree of A being classified as B is 0.8. MF(A ∈B) = U(X= A) = 0.8 .

But more often, the input information will be fuzzied by following general fuzzy membership function.

	M(MT1)	M(MT2)	M(MT3)	M(MT4)	M(MT5)
MT1	1.0	0.8	0.4	0	0
MT2	0.8	1.0	0.8	0.4	0
MT3	0.4	0.8	1.0	0.8	0.4
MT4	0	0.4	0.8	1.0	0.8
MT5	0	0	0.4	0.8	1.0

Table 3.1 Fuzzy membership functions for machining type

2. <u>In Analytical Formulas</u>

This technique is designed to accommodate the fuzziness of input variables which are continuous such as cutting speed. The general form of the fuzzy membership function for a quantitative input X is:

$$MF(x,a,b,c,d)= \begin{cases} S(x, a, b) & \text{if } a \le x < b \\ 1 & \text{if } b \le x < c \\ 1 - S(x, c, d) & \text{if } c \le x < d \end{cases}$$

where $S(x,\alpha,\beta)$ is the bell-shape function defined below:

$$S(x,\alpha,\beta) = \begin{cases} 0 & \text{if } x \le a \\ 2(x-\alpha)^2 / (\beta-\alpha)^2 & \text{if } a \le x \le (\alpha+\beta)/2 \\ 1 - 2(x-\beta)^2 / (\beta-\alpha)^2 & \text{if } (\alpha+\beta)/2 \le x \le \beta \\ 1 & \text{if } \beta \le x \end{cases}$$

as shown in Figure 3.11. Theoretically speaking, the higher power used in the function, more accurate solution can we get. But as a compromise between the accuracy and cost of calculation, square is a good choice. Ususally, boundary value b and c are obtained from metal cutting handbook. The value a and d are roughly initialized so that the vlaues (b-a) and (d-c) be 20% of (c-b). And later on these boundaries should be adjusted according to the system performance.

This mathematical model is used throughout the thesis to represent the fuzziness associated with the inputs (MF(X)) and input-output relations (MF(Y/X)).

For example, the fuzzy membership function of the cutting speed(V) is defined below:

$$MF(v) = \begin{cases} S(v,\ 0.5V,\ 0.9V) & \text{if } 0.5V \le v < 0.9V \\ 1 & \text{if } 0.9V \le v < 1.1V \\ 1 - S(v,\ 1.1V,\ 1.5V) & \text{if } 1.1V \le v \le 1.5V \end{cases}$$

as shown in Figure 3.12.

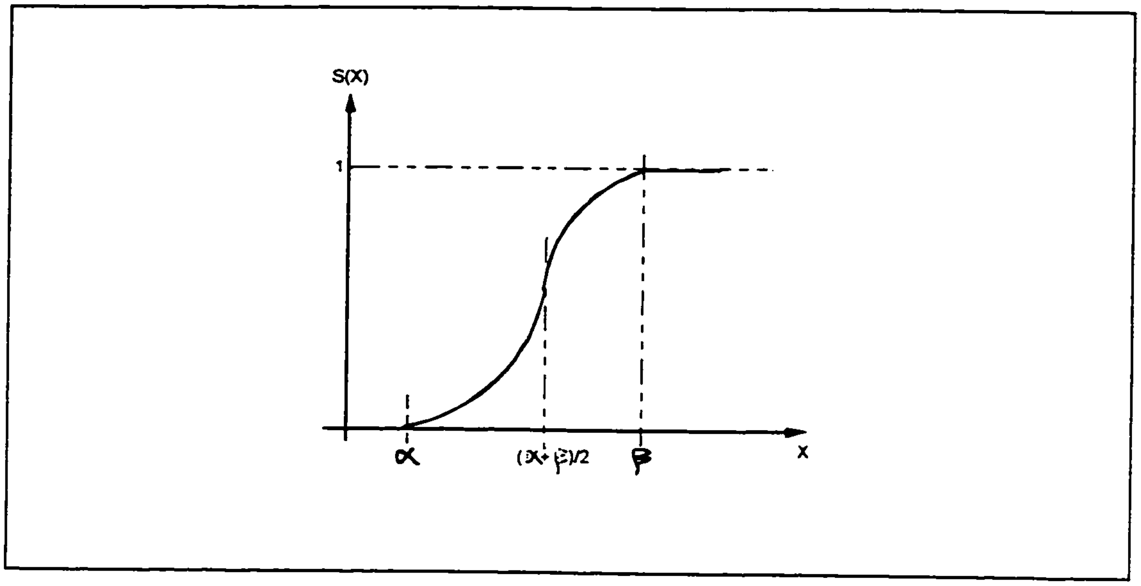


Figure 3.11 Graph of S(x,a,ß) function

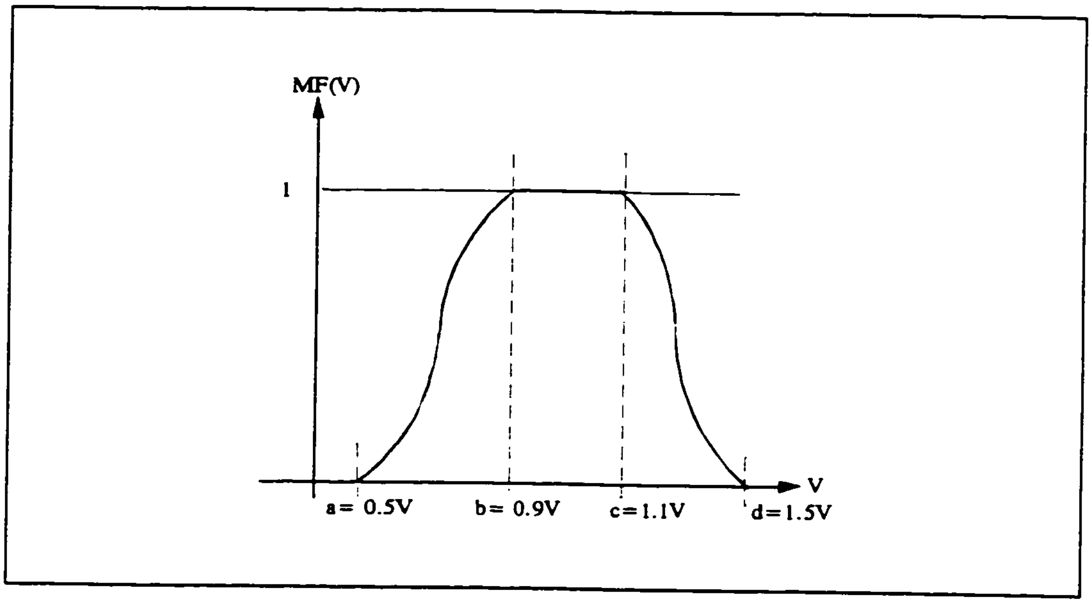


Figure 3.12 The membership function of cutting speed

This general mathematical form could be used not only for fuzzying input information, but also for the determination of cutter selection. For example, the relationship between the insert thickness and depth of cut is defined below and shown in figure 3.13:

$$MF(thickness=4.8/d) = \begin{cases} S(d,1,2) & 1 \le d < 2 \\ 1 & 2 \le d < 3 \\ 1 - S(d,3,4) & 3 \le d \le 4 \end{cases}$$

In the system, over one hundred such functions are defined based on metal cutting principles, cutter manufacturers' manuals, machinists' handbooks, and various technical reports.

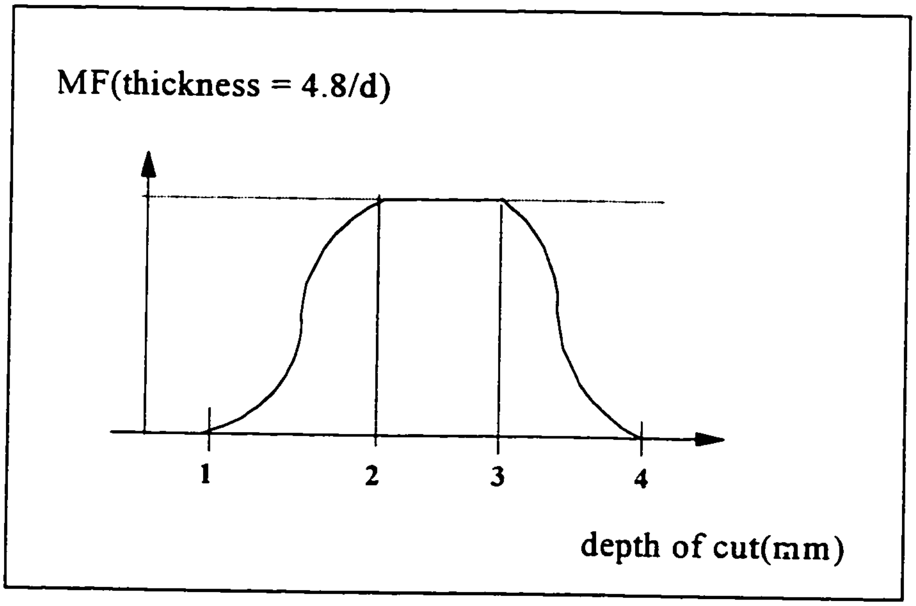


Figure 3.13 Fuzzy membership function for thickness=4.8 using DOC

3.2.3 Fuzzy MAX-MIN and "IF-THEN" Rules

In reality, it is rare and unlikely to have all input information available. Therefore, the input information is often partial. To handle this kind of incomplete or fuzzied environment, related information are cross-defined. Since all the inputs are somehow related, one or more inputs can be used to define the missing information. The correlation between the inputs has been investigated in many publications and machining handbooks. For example; heavy roughing implies that "large" feed and "large" depth of cut are required. Therefore, there must be a relationship between the machining type and feed, and depth of cut. In most of the machining handbooks and cutter product catalogues, this relationship is described quantitatively. [Sandvik, Coromant Product Catalogue, pp.76-77, 1993]

For example: A: Heavy roughing

Fd = 0.5 -- 1.49 mm/rev

Doc = 6 -- 14.8 mm

B: Roughing

Fd = 0.4 -- 1 mm/rev

Doc = 3 -- 10 mm

C: .....

The relationship among the inputs can be used in turn. For example, a process with a "large" feed and "large" depth of cut also implies that the process is a heavy roughing. In other words, the machining type can be determined by the given depth of cut and feed. As can be seen, a process with feed rate of 0.7 mm/rev can be classified in either type of process. This classification provides users with an approximate range for heavy roughing and roughing. The system expresses this kind of fuzzy relationship by the fuzzy set as shown in Figure 3.14.

The Figure 3.14 shows that for heavy roughing, if the feed is between b and c, machining type is classified as heavy roughing because its membership value is 1. If the feed is less than b, it is likely to be roughing than heavy roughing. Similarly, the machining type can also be co-related by the depth of cut.

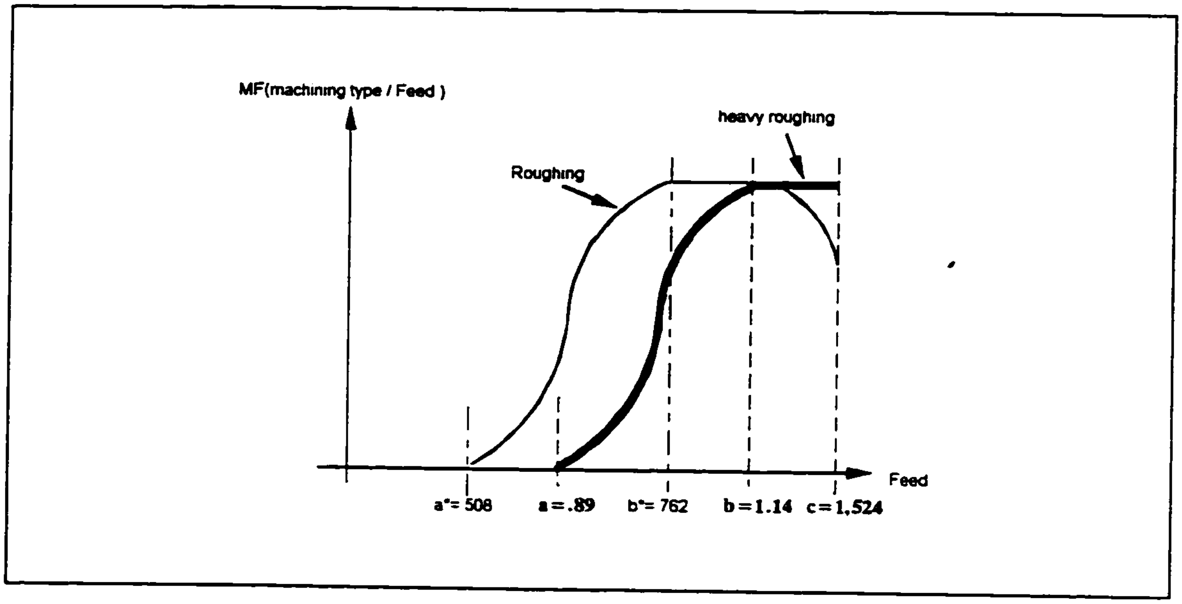


Figure 3.14 Fuzzy relationship between the machining types and feed

The total membership function Y of heavy roughing can be obtained by the MAX-MIN procedure, which was proposed by Zadeh and combines the effects of various inputs to generate the fuzzy membership.

$$M(Y) = \frac{1}{m} \sum_{j=1}^{m} MAX[\sum_{i=1}^{n} M(Y / X_i) \wedge M(X_i)]$$

where $\wedge$ = minimum operator

$\vee$ = maximum operator

Y = missing/unavailable information

X = determining/available information

n = number of qualifiers the variable X are divided

m = number of determining variables X are needed

For above example:

M(heavy roughing) =

$$\begin{aligned}
\tfrac{1}{2}\big(\; & M(\text{heavy roughing}/Fd) \wedge M(Fd \in \text{high}) \\
\vee \; & M(\text{heavy roughing}/Fd) \wedge M(Fd \in \text{medium}) \\
\vee \; & M(\text{heavy roughing}/Fd) \wedge M(Fd \in \text{low}) \\
+ \; & M(\text{heavy roughing}/doc) \wedge M(doc \in \text{high}) \\
\vee \; & M(\text{heavy roughing}/doc) \wedge M(doc \in \text{medium}) \\
\vee \; & M(\text{heavy roughing}/doc) \wedge M(doc \in \text{low}) \big)
\end{aligned}$$

In many cases, some of the inputs are neither clearly defined nor available. The cross defined module provides great flexibility for the system to collect input information.

In the mean time, traditional "IF-THEN" rules are also applied to ensure a proper selection. The combination of both rules simulates two sides of human thinking logic, and make the system more flexible and intelligent.

One example of the rule on nose radius (Ci) selection is as follows:

IF machining type = {Heavy roughing, roughing}

THEN

MF(C1) = MF(C1/Fd)^MF(Fd)

MF(C2) = MF(C2/Fd)^MF(Fd)

......

Else IF machining type {finishing, extreme finishing}

MF(C1) = MF(C1/Surface finish)^MF(Surface finish)

MF(C2) = MF(C2/Surface finish)^MF(Surface finish)

....

END IF

3.2.4 Fuzzy Non-linear Mathematical Model and Hill Climbing Method

1. Membership Functions of the Constraints

It is obviously that the optimal cutting conditions can be determined by solving an optimization problem. In this thesis, a fuzzy non-linear mathematical model proposed by Trappey and Liu [1988] is used. A maximum tolerance is set to each constraint. For example, the surface finish constraint can be expressed as $SF \leq 1.4(mm) + (1-\alpha)0.5(mm)$, where $0 \leq \alpha < 1$ and 0.5(mm) is the maximum tolerance. The fuzzy degree α represents the trade-off between the level of acceptance (satisfaction) and tolerance.

In the following discussion, the primary variables in the proposed model are feed and cutting speed. For the sake of simplicity, they are represented by a decision vector X, which means X=[V,f]. As shown in Trappey and Liu [1988], the model with objective of minimum cost can be generally formulated as follows:

$$\begin{aligned} &\text{MIN} \quad C(X) \\ &\text{S.T.} \quad g_i(X) \leq b^{-}_i + (1-\alpha)T_i \quad i = 1,2,\ldots m \\ &\qquad\quad X \geq 0 \end{aligned}$$

where $C(X)$ = objective function (Cost Function)

$g_i(X)$ = model's constraints equations

T_i = maximum tolerance

b^{-}_i = restriction/specification constants

m = number of constraints

Trappey and Liu considered that the membership function U(i) for the i^{th} fuzzy constraint is:

$$U(i) = \begin{cases} 1 & g_i(X) < b_i \\ 1 - [g_i(X) - b_i] / T_i & b_i \leq g_i(x) \leq b_i + T_i \\ 0 & g_i(X) > b_i + T_i \end{cases}$$

The membership function is graphically presented by Figure 3.15 where T_i is the tolerance of each constraint.

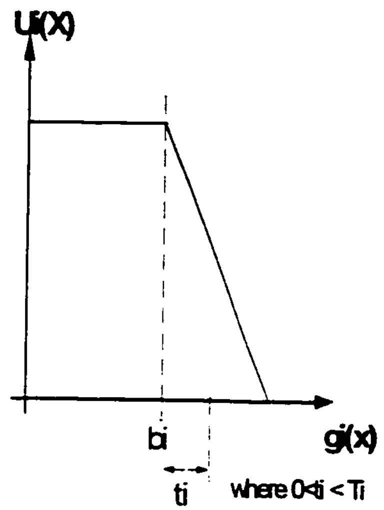


Figure 3.15 Membership function of the constraints

The resource constraints $g_i(x) \leq b_i$, for all i, are satisfied as well as possible. Given that each constraint has a level of violation, t_i, each constraint violates the conditions of the problem to a maximum tolerable degree, T_i, established by the decision maker. For example, if $t_i = 0$, the constraints satisfies the condition with 100% satisfaction. If $t_i = T_i$., then the constraint satisfies the condition with 0% satisfaction, where $0 \leq t_i \leq T_i$. The relationship between the violation rate and the satisfaction rate is described by the Figure 3.15.

2. Fuzzy Decision

In Trappey et al. [1988], they showed that a model with fuzzy constraints can be translated to the crisp constraints. Therefore, the proposed model can be converted to the crisp fuzzy programming for solution.

The objective function (C) and constraints (G) can be connected to each other by an operator "$\wedge$" which corresponds to the intersection of fuzzy sets. A fuzzy decision may be defined as the fuzzy set of alternatives resulting from the intersection of the objective and its constraints. This intersection implies that the decision $D = C \wedge G$ is a fuzzy set resulting from the intersection of C and G with the membership function of $U_D(X) = U_G(X) \wedge U_C(X)$.

The decision at X_m can be represented by both constraints and objective with the highest possible degree of their membership functions as shown in Figure 3.16.

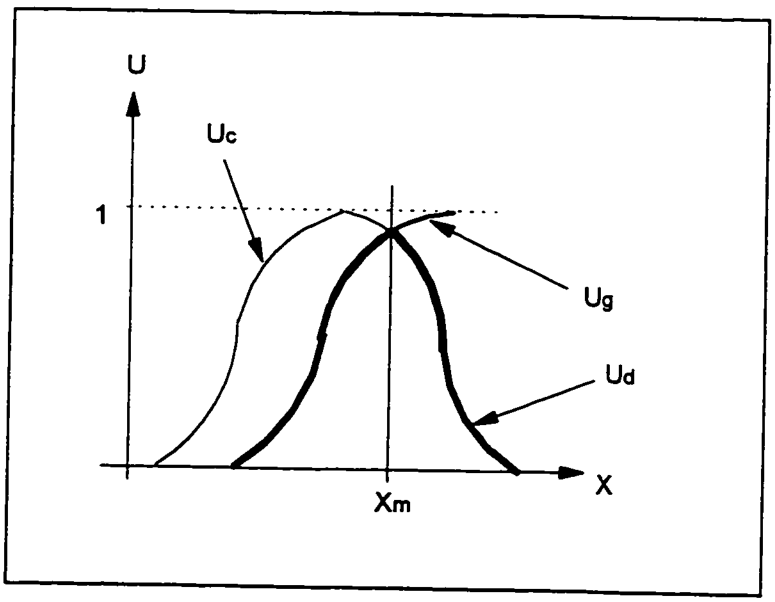


Figure 3.16 The relationship between membership functions of constraints, objective and decision

In the Figure 3.15, with the Max-Min operation, the relationship can be mathematically shown as follows:

$$U_D(X) = U_{G_i}(X) \wedge U_C(X)$$

where i = 1,2.....m. The optimal solution of the proposed fuzzy non-linear optimization model can be obtained by:

$$\text{Max } U_D = \max \{ \min [U_C(X), U_1(X), \ldots\ldots U_m(X)] \}$$

let $\alpha = U_D$, the proposed model is equivalent to

Max α

S.T. $U_C(X) = 1 - (CX - b_0) / T_0 \geq \alpha$

$U_i(X) = 1 - [g_i(X) - b_i] / T_i \geq \alpha \qquad i = 1, \ldots m$

$\alpha \in [0,1]$

After the variables are rearranged, the model becomes as follows:

Max α

S.T. $CX \geq b_0 - (1 - \alpha) T_0$

$g_i(X) \leq b_i + (1 - \alpha) T_i \qquad i = 1, \ldots m$

$X \geq 0$ and $\alpha \in [0,1]$

So finally, a fuzzy non-linear mathematical model is converted into an ordinary non-linear mathematical model like above Equation. This model is actually used to solve the optimization problem.

Let us first consider a linear optimization problem - product-mix selection problem, which could be solved by LINDO, to see how to apply the fuzzy theory in this case. While for non-linear case, which is naturally same as the previous one, Hill Climbing algorithm could be employed to obtain one optimal solution.

Suppose that a company has the option of using one or more of four different types of production processes. The first and second processes yield items of product A, and the third and fourth yield items of product B. The inputs for each process are labour measured in man-weeks, pounds of material y and boxes of material z. The manufacturer's decision on a week's production schedule is limited in the range of possibilities by the available amounts of manpower and both kinds of raw materials. Table 3.2 gives the technology and inputs restrictions. The last row gives tolerance for each restriction. Values larger than sum of the available resource and tolerance are not acceptable.

Items	**Person-week**	**Material y**	**Material z**	**Unit profit**
Process 1	1	7	3	4
Process 2	1	5	5	5
Process 3	1	3	9	9
Process 4	1	2	15	11
Available resource (fuzzy)	***15***	***80***	***100***	***130***
Tolerance	5	40	30	30

Table 3.2 The input data for the product mix selection problem

Suppose that production levels in processes 1, 2, 3 and 4 are x_1, x_2, x_3 and x_4 respectively. the problem can then be formulated as the following model:

Max α

ST $Z = -(4x_1+5x_2+9x_3+11x_4) \leq -130 + 30(1-\alpha)$

$$4x_1+5x_2+9x_3+11x_4 \geq 130 - 30(1-\alpha)$$

$$g_1(x)= x_1 +x_2 +x_3 + x_4 \leq 15+5(1-\alpha)$$

$$g_2(x)=7x_1 +5x_2 +3x_3 + 2x_4 \leq 80+40(1-\alpha)$$

$$g_3(x)=3x_1 +5x_2 +9x_3 + 15x_4 \leq 100+30(1-\alpha)$$

$$x_1 , x_2 , x_3 , x_4 \geq 0 \quad \text{and} \quad \alpha \in [0,1]$$

After setting up the model, the solution can be obtained by LINDO:

$x_1^* = 8.57$, $x_2^* = 0$, $x_3^* = 8.93$, $x_4^* = 0$

$Z^* = 114.55$, $\alpha = 0.65$

While the actually used resources are 17.5, 86.78, 115.01 for man-week, material y and material z respectively.

3. Hill Climbing Algorithm

The mathematical model discussed in Section 5.1 is in a Fuzzy Mathematical Optimization Model. The Model searches the optimum cutting conditions (feed, cutting speed) by using ready-made Hill-Climbing Algorithm. The algorithm is based on the method proposed by Rosenbrock [1960]. This method is a sequential search technique which has proven effective in solving some non-linear programming models with constraints. The algorithm proceeds using unconstrained Rosenbrock procedure until convergence is reached or a boundary zone in the vicinity of the constraints is entered. The boundary zones are defined as follows:

$$G_K \leq X_K \leq (G_K + (H_K - G_K) * 10^{-4})$$

where X_K = decision variables, G_K = Lower bound and H_K = Upper bound and k = 1,2.....M

The search procedure is discussed as follows:

S1. Define by F^o as the current best objective function value for a point where the constraints are satisfied, and F^* the current best objective function value for a point where the constraints are satisfied and in addition the boundary zones are not violated F^o and F^* are initially set equal to the objective function value at the starting point.

S2. If the current point objective function evaluation, F, is worse than F^o or if the constraints are violated, the trial is a failure and the unconstrained procedure is continued.

S3. If the current point lies within a boundary zone, the objective function is modified as:

$$F(New) = F(Old) - (F(Old) - F^*)(3d - 4d^2 + 2d^3)$$

where d = (distance into boundary zone) / (width of boundary zone)

$= ((G_K + (H_K - G_K) * 10^{-4} - X_K) / (H_K - G_K) * 10^{-4})$ if Lower zone

$= ((X_K + (H_K - G_K) * 10^{-4}) / (H_K - G_K) * 10^{-4})$ if upper zone

At the inner edge of boundary zone, d = 0, i.e. the function is unaltered (F(new)= F(old)). At the constraints, d = 1, and thus F(new) = F^*. Thus the function value is replaced by the best current function value in the feasible region and not in a boundary zone. For a function which improves as the constraints is approached, the modified function has an optimum in the boundary zone.

S4. If an improvement in the objective function has been obtained without violating the boundary zone or constraints, F^* is set equal to F^o and the procedure is continued.

S5. The search procedure is terminated when the convergence criteria is satisfied

The programming flow chart is constructed as shown in Figure 3.17.

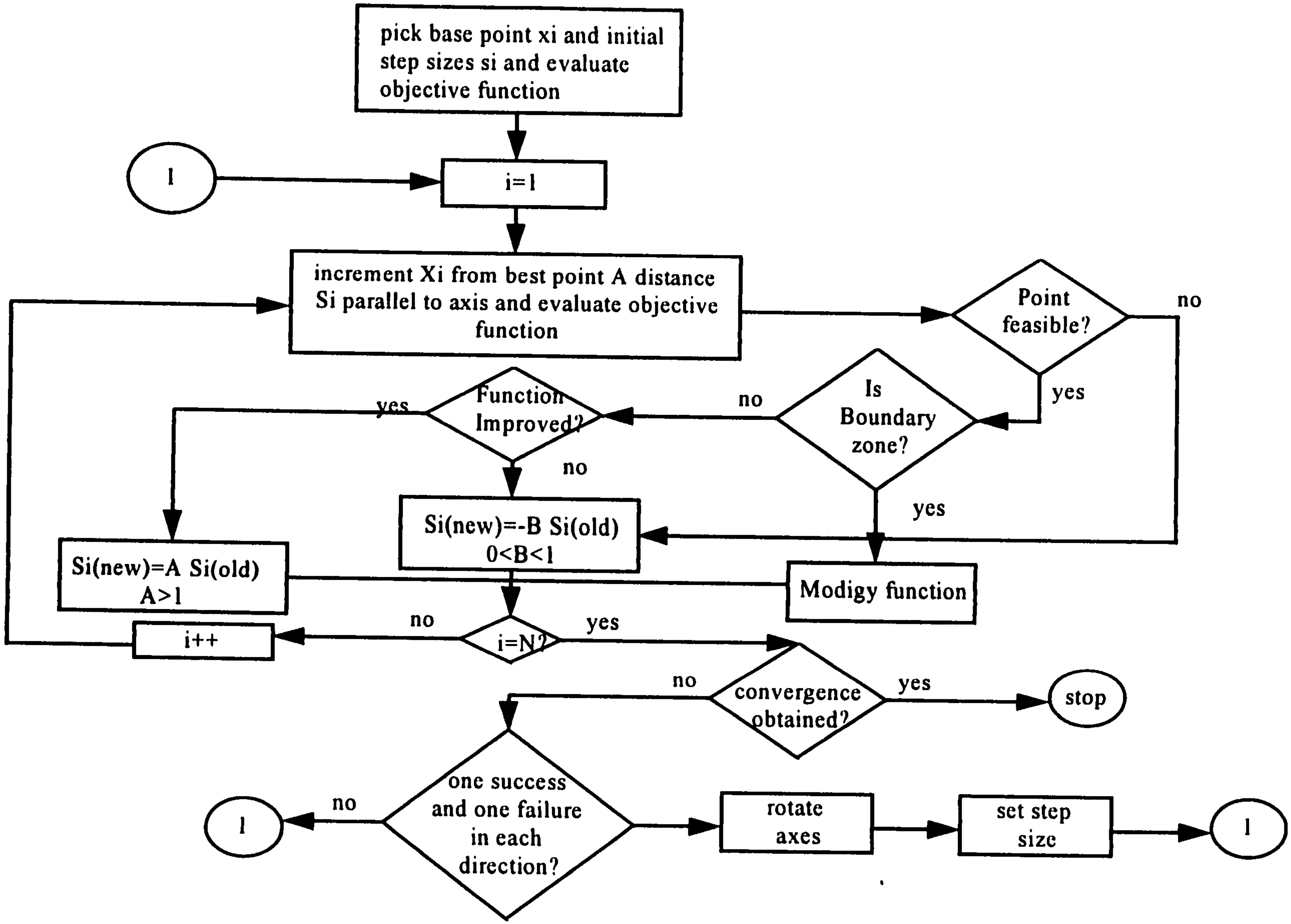


Figure 3.17 Programming flow chart of the Hill Algorithm

CHAPTER IV

DESCRIPTION OF THE PROPOSED SYSTEM

A prototype application software system called SAM (Smart Assistant for Machinist) is developed for the cutter selection and cutting condition design for rotating processes, including milling, boring and drilling. The system contains input/output interfaces, pre-compiled fuzzy decision rule base, databases, a mathematical programming model and a learning mechanism. The system is designed with the method of OOM and written by Borland C++ under Microsoft Window™ environment. With all the sophisticated features of Windows programming, the proposed system is very user-friendly and human-oriented. Users who possess minimum knowledge of computer software can use it easily. The system is designed to deal with practical working environments which are full of uncertainties or fuzziness. Therefore, the system is very flexible with respect to collecting input data and delivering outputs. At least one solution is guaranteed. In other words, no such errors like" Infeasible solution" will be given by the system. Since the system emphasizes on providing qualitative results, a final decision should be made by users. The system is able to not only recommend user the optimal solution, but also estimate and compare it with user's desire.

4.1 System Structure Design

As shown in Figure 4.1, the system is composed of a number of modules: an input/output interface, fuzzy decision rule base, a database, a mathematical programming module and a learning module. The system is designed following the principle of Objective Oriented Modelling (OOM) and is written using Borland C++ under Microsoft Window environment. With all the sophisticated features of Windows programming, the proposed system is user-friendly and human-oriented. The system is designed to deal with shop floor applications which are full of uncertainties or incompleteness. Given input information, no matter how limited, at least one feasible solution would be found. It often provides multiple feasible solutions, and asks users to make a final decision. The system is also able to recommend an optimal solution and compare it with the users' design.

Under the OOM, rotating machining operations are classified into drilling, milling (face milling, end milling,...), and boring; cutters are classified into solid cutters and indexable toolbars with inserts. This makes system easy to develop, maintain, expand. Fuzzy logic is used to generate proper cutter selections and optimise cutting conditions. Based on partial and imprecise information, the system is able to select rotating cutter products which are available in current commercial market, and design cutting conditions for an optimised objective. At least one satisfactory solution is guaranteed.

The Object-Oriented Modeling (OOM) approach is employed to design the system. OOM encapsulates both data structure and its associated behaviour in a single computing entity existing at a higher level than normal procedure or data structure. With the application of OOM, rotating machining operations are classified into drilling, milling (face milling, end milling), and boring classes. Cutters are classified into solid cutters and indexable toolbars with inserts as shown in Figure 3.2. This makes the system much easier to develop, maintain, expand. Fuzzy logic is used to generate proper cutter selections and optimise cutting conditions. Based on partial and imprecise information, the system is able to select rotating cutter products which are available in current commercial market, and design cutting conditions for an optimised objective. At least one satisfactory solution is guaranteed.

Figure 4.1 is Object-Oriented Model and information flow in the system. It's based on the OOM method described in section 3.1. Each frame in diagram is a class. The arrows represent the linkage between two classes and information flow. The system consists of four major classes: a database, a rule base, a fuzzy operation, and a learning class. The database could be further divided into six subclasses: machinability of workpieces, machining plan, fuzzy rules, tool adapter, machine tool, as well as insert and cutter inventory classes. Currently, database contains over three thousand kinds of cutters and inserts from several big companies. Each cutter item is an instance of the Cutter class. The rules in the rule base are developed based on fuzzy set theory. They are used to cross-define unknown information and to determine the relationships between the inputs and outputs information. Fuzzy Operation class is an abstract class and could be derived into three association classes: input fuzzization, cutter selection, and cutting condition design subclasses. These trinary associations are inference engine of the expert system. They express the complicated relationship between the classes they link. Three basic functions of the system (cutter selection, cutting condition design and learning) are implemented by these association classes respectively. With input fuzzization class, unknown and imprecise input information are cross-defined and fuzzied. The cutter selection module is developed based on fuzzy logic. The cutting condition design module is developed based on fuzzy non-linear programming where the design constraints could be fuzzy. The model has three optimization objectives (Min. cost, Min. machining time and Max. profit) and is solved by Hill Climbing Algorithm. Combined with selected cutters, these cutting conditions and performance are saved as process plan. The performance of the system could be further improved by an internal self-learning mechanism by fine-tuning fuzzy membership function coefficients. The modified fuzzy rules update the fuzzy rule base. The function and information flow of the system is described as follows:

The first function module is implemented by the cutter selection association class, in which the cutter selection is conducted in three steps. First, the user's basic and preferred input information, which may be imprecise or incomplete, are fuzzirised or cross-defined by applying fuzzy member function rules and generate the job object. Next, according to the fuzzy decision rules and fuzzirised input information, cutter selection mechanism generates the features of rotating cutters, including geometry, size, grade, ..., and its corresponding toolbars and/or tool holders. Finally, after the fuzzy matching with cutter inventory database, a list of available cutters is outputted as the result.

The second step is cutting condition design. Depending on the basic and preferred input information (like workpiece, machining type, optimization objective and so on), selected cutters, and the technical support information retrieved from machinability and machine tool database, the association class Fuzzy Optimization sets up the fuzzy non-linear mathematical model, which includes the construction of an objective function (Min. unit cost, Max. production rate, Max. profit) and constraint functions. The model could be solved by Hill-Climbing algorithm, and cutting data, such as feed rate, cutting speed, are its outputs. In the meantime, the performance of the whole cutting process (such as machining time, unit cost and tool replacement) is estimated by Performance class. Combined with the selected cutter information, a complete process plan is generated and saved in database.

The third major function of the expert system is self learning. Because the fuzzy membership functions, which are very critical to the performance of the system, are usually empirical and their initial values are far from precise, the generated process plans probably are not satisfactory. Through this learning module, experts and skilled workers are capable of teaching the system to improve the performance or accommodate their own preference by modifying the fuzzy decision rule base, especially by modifying the coefficients of these functions.

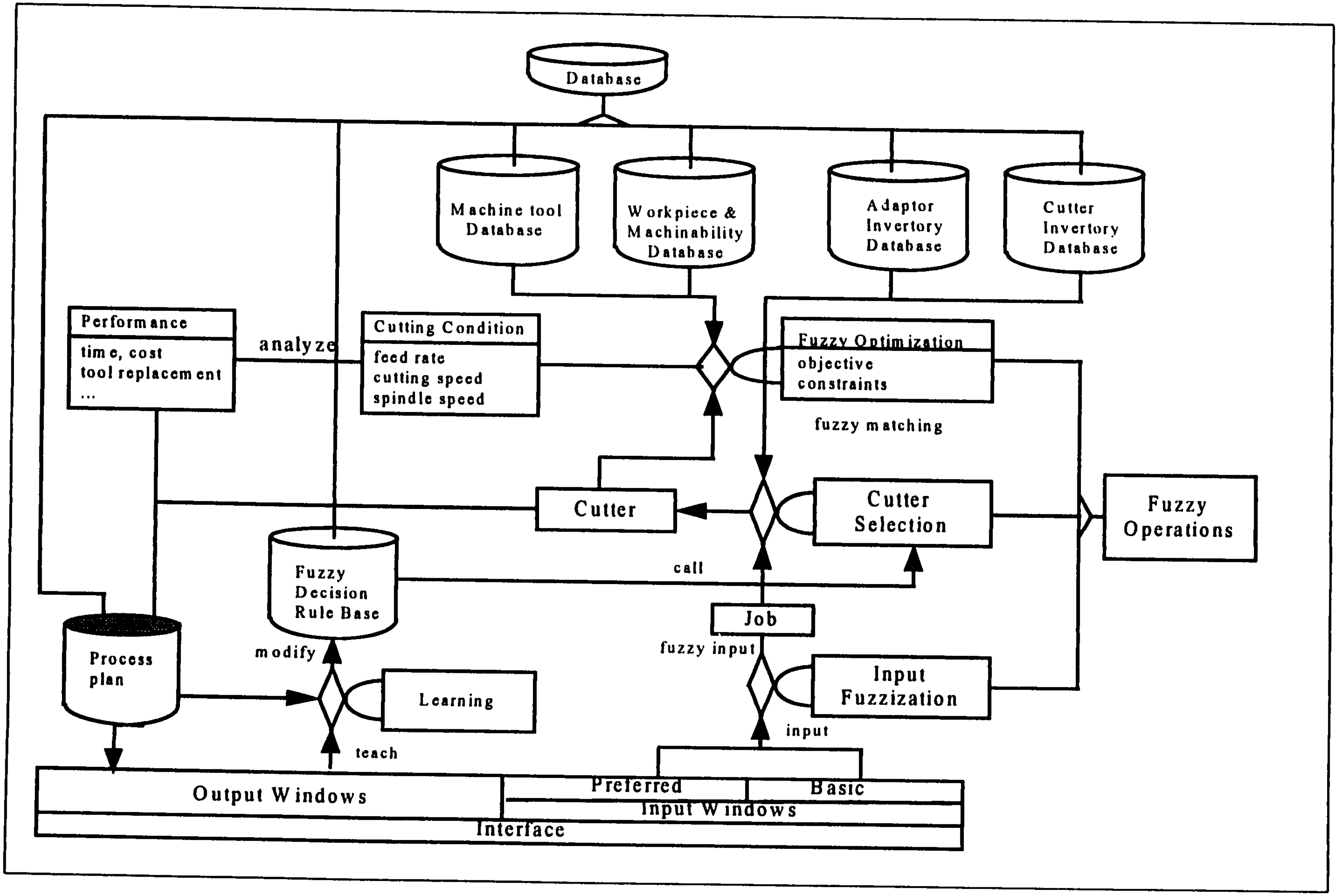


Figure 4.1 Object-Oriented Model and Information Flow

4.2 Fuzzy Cutter Selection Subsystem

The inputs of the system include work material and machining preference, cutter preference, job specification and machine tool, as listed in Table 4.1. Many inputs can be defined by either numerical values or qualitative terms or not defined at all.

The outputs of the system include:

1) The recommendations of commercial products including grades and product catalogue number from major tool manufacturers.

2) The recommendations of cutting conditions, and

3) The estimation of machining cost and machining time, power consumption, surface fininshing, etc.

The selection procedure consists of three steps:

1. Obtain input information from either users or Cross-Defined Relationship Modules. Hence, using the fuzzy set theory, a membership function is assigned to each input information to represent its fuzzy nature.
2. Select the most suitable cutter and cutter grade based on information extracted from the tooling, machinability database and the rule base by the fuzzy decision mechanism, which consists of "MAX-MIN" procedure and "IF-THEN" rules decision procedure.
3. Check for the availability of the selected cutter and grade from the built-in cutter inventory database.

Figure 4.2(a) shows the structure of the cutter selection system while Figure 4.2(b) shows the information flow.

Work material (1.1) material code: in ISO standard with CMC code, or in ANSI standard (1.2) material type: steel alloy, stainless steel, ... (1.3) hardness: in Rockwell C scale, Rockwell B scale, or Brinell scale (1.4) heat treatment: annoyed, oil punching, ...
Machine tool (2.1) power and maximal power: KW, HP (2.2) torque and maximal torque: Nm, lb, -ft (2.3) maximum positioning speed: m/min, ft/min (2.4) maximum cutting speed: (m/min, ft/min) (2.5) accuracy, (2.6) efficiency,
Machining plan (3.1) machining type (3.1.1) milling: face milling, end milling, contour milling, ... (3.1.2) drilling: general drilling, counterbored hole, ... (3.1.3) boring: rough, fine, ... (3.2) machining type: heavy roughing, roughing, light roughing, finishing, ... (3.3) material removal rate: large, medium, small or cm^3/min, $inch^3$/min (3.4) surface finish: rough, medium, fine or μm, μinch (3.5) dimension tolerance: mm, inch (3.6) cutting speed: fast, medium, slow or m/min, ft/min (3.7) feed: fast, medium, slow or mm/tooth, inch/tooth (3.8) depth of cut: mm, inch (3.9) width of cut: mm, inch (3.10) length of cut: mm, inch (3.11) machining cost: $/min (3.12) time factor: $
Cutter and holder preference (4.1) cost: cents (4.2) supplier: Sandvik, Kennametal (4.3) cutter geometry: nose radius, thickness.... (4.4) cutter holder (4.3.1) geometry: lead angle, rake angle, side rake angle, relief angle, ... (4.3.2) size: (4.3.3) pitch type:coarse, close, extra close (4.5) tool life: long, medium, short or min

Table 4.1 Descriptions of the input factors of the system

4.2.1 Fuzziation of Input Information

Shown as Fig 4.2(a), every input information item, whether it's from users or not, should be fuzzied as the first step. The fuzzy membership functions needed are discussed in section 3.3, 3.4.

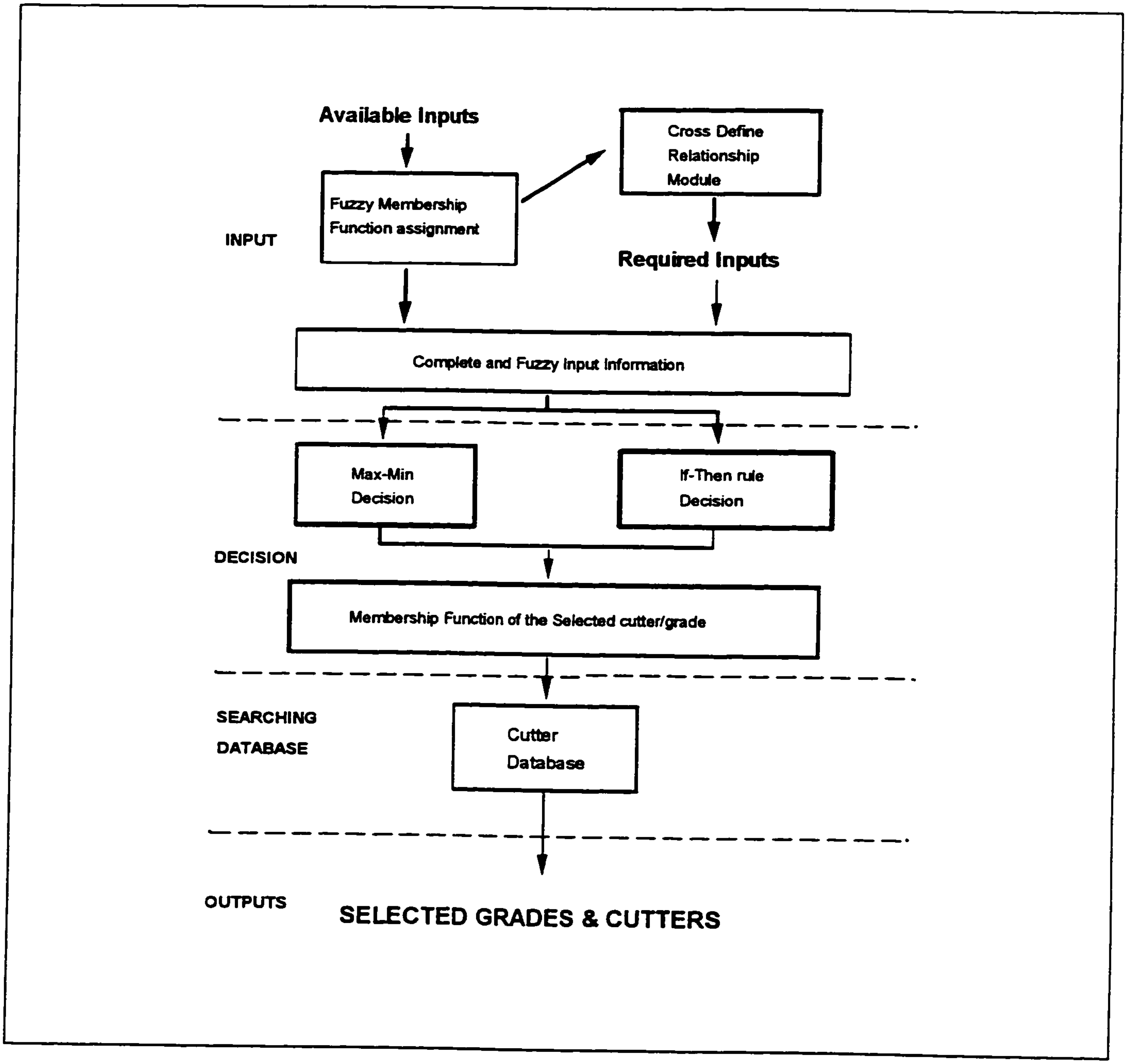


Fig 4.2 (a) The structure of the cutter selection system

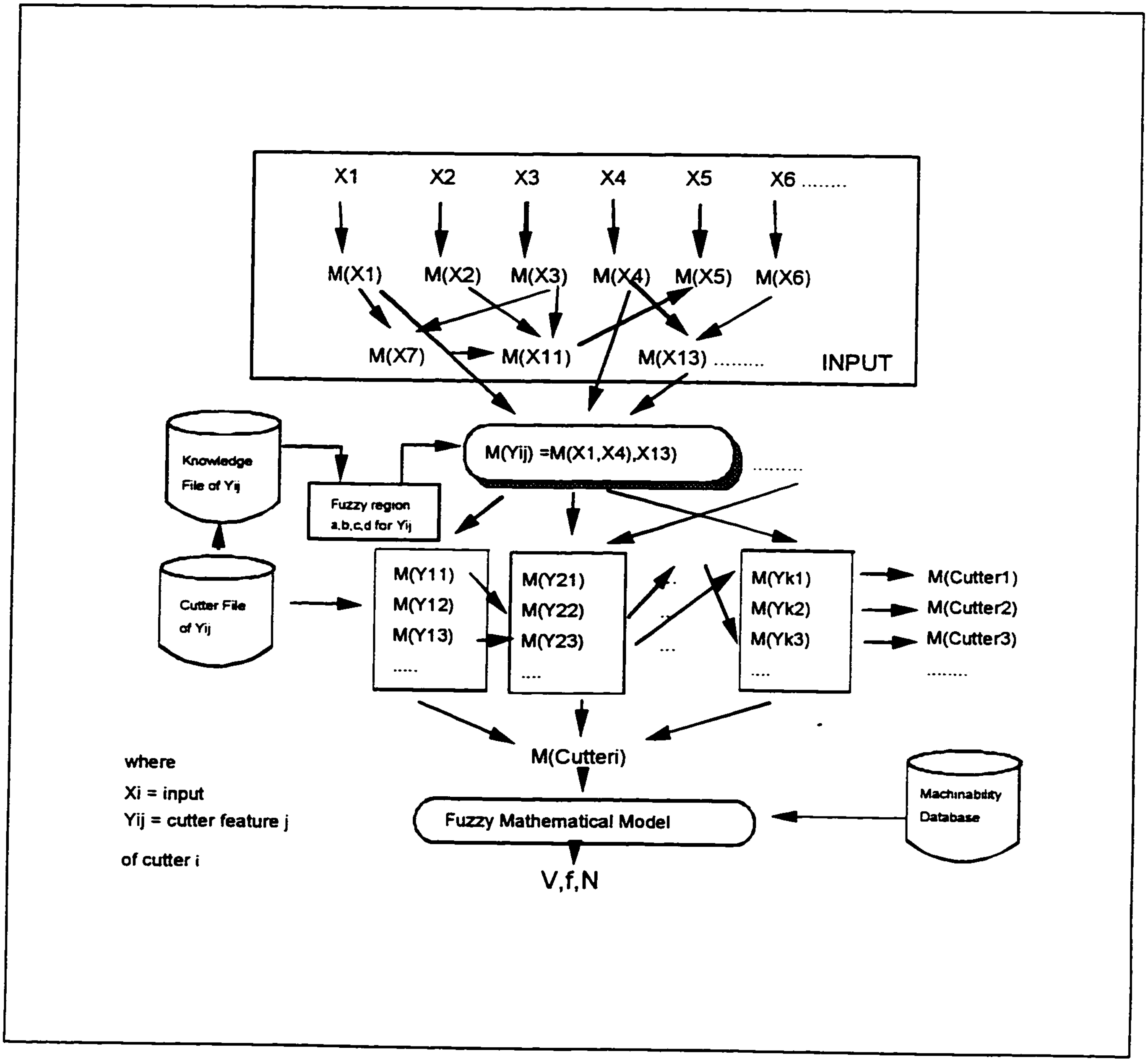


Figure 4.2 (b) The information flow of the system

1. Inputs for Cutter Selection from User

All the input information in the system is represented by a qualifier and a variable with a fuzzy degree. The qualifiers represent the qualitative information such as high, medium or low, while the variable represents a numerical value. All the qualifiers and choices are listed in Table 4.2. For each qualifier, there's a fuzzy membership function assoicated. The variables, their units and coefficients of the fuzzy degree are listed in Table 4.3.

Qualifiers	Variables	Choices
1. Supplier	SUPP	1. Sandvik 2. Kennamental ...
2. Tool Life	TL	1. Long 2. Medium 3. Short
3. Material Removal Rate	MRR	1. High 2. Medium 3. Low
4. Surface Finish	SF	1. Rough 2. Medium 3. Fine
5. Cutting Speed	CS	1. High 2. Medium 3. Low
6. Feed	Fd	1. High 2. Medium 3. Low
7. Depth of Cut	Doc	1. High 2. Medium 3. Low
8. Material Code	Ml_Code	1. 1.1 2. 1.2 3. 1.3 4. 2.1 5. 2.21 6. 2.22 ...
9. Material Type	Ml_Type	1. Unalloyed steel 2. Low alloyed steel 3. Chilled cast iron 4. High alloyed steel 5. Heat resistant alloyed .....

Table 4.2 List of qualifiers and choices in the system

Variable Description		Unit	Variable Name	Fuzzy Descriptions a	b	c	d
Surface Finish:	1. Rough	mm	SF	7	12	20	-
	2. Medium			5	8.75	12.25	15.5
	3. Fine			-	0	5.5	8.5
Cutting Speed	1. High	m/min	V	275	425	500	-
	2. Medium			125	237.5	312.5	425
	3. Low			-	50	125	275
Feed	1. High	mm/tooth	Fd	.375	.625	.75	-
	2. Medium			.125	.3125	.4375	.625
	3. Low			-	0	.125	.375
Depth of Cut	1. High	mm	Doc	7.5	12.5	15	-
	2. Medium			2.5	6.25	8.75	12.5
	3. Low			-	0	2.5	7.5
Material Removal Rate	1. High	cm^3/min	MRR	20	30	60	-
	2. Medium			15	20	25	45
	3. Low			-	5	25	30

Table 4.3(a) List of Variables in the system for Machining Plan

2. Inputs for Cutter Selection by Cross-Defined Relationship Function

Table 4.4 shows all the cross-defined relationships between the different input information.

Variable Description	Unit	Variable Name	a	b	c	d
Cutter Cost	$	C_Cost	0.4X	0.8X	1.3X	1.6X
Tool Nose Radius	mm	NRa	0.5X	0.95X	1.05X	1.5X
Thickness	mm	Tks	0.5X	0.9X	1.1X	1.5X
Cutter Diameter	mm	Cdia	1.2X	1.27X	1.35X	1.45X
Insert Size	mm	ISz	0.5X	0.9X	1.1X	1.5X
Lead Angle	deg.	LdA	0.2X	0.7X	1.3X	1.8X
Tool Life 1. Long	min	TL	35	50	90	-
2. Average			25	30	45	55
3. Low			-	5	25	30

Table 4.3(b) Cutter Preference from users

where X represents its corresponding variable.

	Y	X	a	b	c	d
Machining Type	Heavy Roughing	Feed	.89	1.143	1.524	-
		doc	9	12	15	-
	Roughing	Feed	.508	.762	1.016	1.143
		doc	6	9	10.5	12
	Light Roughing	Feed	.254	.508	.762	1.016
		doc	4.5	6	7.5	9
	Finishing	Feed	0	.254	.508	.762
		doc	1.5	2	5	6.5
	Extreme Finishing	Feed	-	.01	.254	.508
		doc	-	0	2	4.5

Table 4.4(a) Cross-Relationship between the machining type and Feed and Doc

	Y	X	a	b	c	d
Material	High	CS	275	425	500	-
Removal	MRR	Feed	0.375	0.625	0.75	-
Rate	Med.	CS	125	237	312	425
(MRR)	MRR	Feed	0.125	0.315	0.437	0.625
	Low	CS	-	50	125	275
	MRR	Feed	-	0	0.125	0.375

Table 4.4(b) Cross relationship between the MRR and cutting speed and feed
where CS = Cutting Speed doc = Depth of Cut

4.2.2 Rotating Cutter Selection

After the input information is "fuzzied", the next step is to determine the fuzzy membership for the cutter and insert selections.

1. Toolbar Selection

(1) Toolbar type selection

There are hundreds of toolbar series available in the market. The selection of toolbar type is dependant on three factors:

1) workpiece material

2) machining type and surface requirement

3) duty of machining operation

Figure 4.3 summarizes fuzzy relationship between tool bar and workpiece material and duty which could be estimated by MRR (Material Removal Rate) and specific energy of workpiece. More relationship between workpiece material and their corresponding suiltable tool bar are summarized in Appendix A.

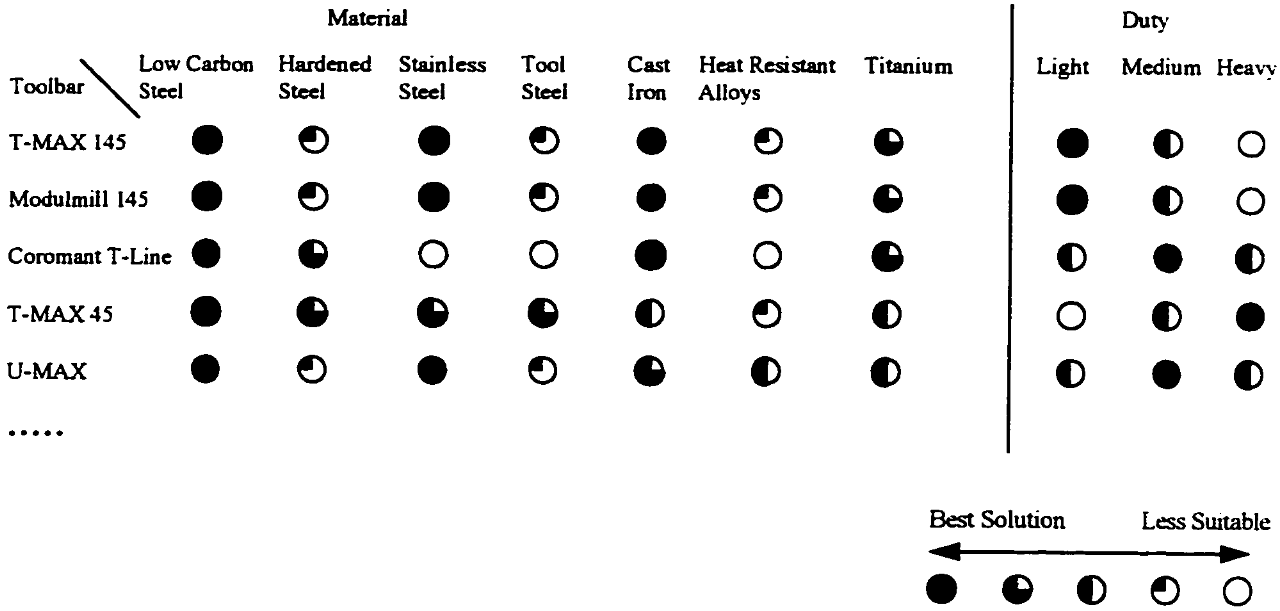


Figure 4.3 Relationship between toolbar and material, duty [Sandvik Tool Handbook, 1995]

The lead angle is important to machining type and surface requirement. For a finishing operation which implies a better surface requirement, usually the smaller lead angle is suggested. Table 4.5 lists the coefficients of associated fuzzy membership functions.

Lead angle	**Extreme finishing**	**Finishing**	**Light roughing**	**Roughing**	**Heavy roughing**	**Surface requirement (μm)**			
						a	**b**	**c**	**d**
0°	1	1	0.8	0.6	0.1	0	0	3	6
15°	0.6	0.9	1	0.8	0.5	0	1	5	8
30°	0.5	0.7	1	1	0.8	1	3	7	11
45°	0.4	0.6	1	1	1	3	6	10	13
60°	0.3	0.4	0.8	1	1	6	9	12	15
Round	0.1	0.3	0.7	1	1	9	12	15	-

Table 4.5 Cross relationship between lead angle and machining type and surface requirement

(2) Dimension, pitch type and number of teeth selection

Usually for a toolbar series, there are tens of toolbar with different dimension, pitch type, number of tooth, rotating hand and so on so forth. To choose a specific toolbar from its families, three major factors, dimension, pitch and number of teeth play a important role.

Diameter of cutter is dependent on dimensions of the workpiece. To satisfy the machining part dimension requirement, the selection of cutter diameter will affect surface quality, machining efficiency and power consumption. With a large diameter cutter, the machine will run out of power, which leads to an uneconomical feed per tooth and deph of cut being used. A large cutter also gives an unsatisfactory ratio between the spindle's front bearing and the cutter diameter, which may cause vibration. When long tool overhang is involved always use smallest possible cutter diameter. If there is mismatching of the surface, a larger cutter may be required for a finishing cut. For Facemilling, diameter of cutter is around 4/3 times of the width of workpiece [Sandvik, 1995]. The membership function of diameter D could be defined as follows:

$$\text{MF(Diameter)}=\begin{cases} S(D,1.2X,1.27X) & \text{if } 1.2X \le D \le 1.27X \\ 1 & \text{if } 1.27X \le D \le 1.35X \\ 1-S(D,1.35X,1.45X) & \text{if } 1.35X \le D \le 1.45X \end{cases}$$

The pitch type of a toolbar is classified as "coarse", "close", "extra close". For a toolbar with same dimension, coarse pitch type has minimum tooth number, and extra close pitch type has maximum number of tooth. Apparently, extra pitch type is more suitable for heavy duty, and coarse one is only suitable for light duty, otherwise the inserts will wear out much faster. So for a given toolbar, its pitch type could be chosen by the ratio duty:diameter . Table 4.6 roughly describes how to select pitch type based on the ratio. The boundary values also depend on the characteristic of different inserts.

Pitch Type	Duty/Diameter (W/inch)			
	a	b	c	d
Extra Close	1000	1500	2300	3000
Close	800	1200	1700	2200
Coarse	0	400	1000	1500

Table 4.6 Fuzzy relationship between pitch type and duty/diameter

The number of teeth is important in rotating operations such as Facemilling. It is related to not only cutter dimension, but also cutter pitch type. When cutter diameter and pitch type are chosen, the number of teeth is also determined.

2. Insert Type Description:

The system uses both ISO and ANSI standard to code the cutter inserts. The inserts are described into nine fields, as listed in Table 4.7:

Field No.	Description
1	Shape:R - round, S - square, T - triangle, ...
2	Relief angle:N - 0°, A - 3°, B - 5°, ...
3	Tolerances:A - ±0.0002, B - ±0.0002, ...
4	Type:A - with hole, B - with hole and one countersink, ...
5	Size:4 = 1/2"I.C., 5 - 5/8"I.C., ...
6	Thickness:number of 1/32nds on inserts less than 1/4"I.C., ...
7	Insert nose radius: 1-1/64", 2-1/32",..., A-square 45° chamfer,...
8	Cutting edge condition:F, E, T, ...
9	Feed direction:R -right, L -left, ...

Table 4.7 The information of the selected insert

In each yield $y_i(i=1....8)$, there are only a limited amount of elements available in the market. Based on metal cutting principles and cutting tests, the relationships between the input(s) and each element of the fields can be correlated. The selection rules and procedure of each insert field are discussed in detail in the following sections:

(1) Insert Shape

The insert shape determines the strength of the insert and surface finish [SME ToolHandbook, 1985]. Six commonly used standard insert shapes are listed in Table 4.8. Each shape offers different benefits and limitations. They are listed in order of decreasing insert strength.

Shape	**Code**
1. Round	RCMT
2. Square	SNMG/SCMT
3. 80° Diamond	CCMT
4. Triangle	TNMM/TCMT
5. 55° Diamond	DCMG
6. 35° Diamond	VNMG/KNMX

Table 4.8 List of insert shape available in the system

The rule for selecting a particular insert shape is strongly associated with its own insert's geometry. In order to reduce system internal conflicts, the system only has rules to select the most suitable insert geometry. The insert shape will be automatically selected corresponding to the chosen insert geometry.

(2) Tool bar and Insert Geometry

Choosing insert geometry is a complicated process. The insert geometry is actually a combination of the factors of clearance angle (field 2 in ISO standard), tolerance(field 3) and the insert type (field 4) [SME ToolHandbook, 1985]. Different tool manufacturers offer a spectrum of products to customers for different machining operations. For rotating operations, the selection of insert geometry is strongly related to that of toolholder (including pitch type). For the example of facemilling operation, tens of basic holder types are included. For each holder, several geometry elements are available. Table 4.9 lists part of the available choices.

Machining Type	Toolholder Type	Choice	Machining Type	Toolholder Type	Choice
End Milling (Sandvik)	U-MAX-C Endmill 216.2	SPMT -WL -WH -AAH -AAM	Rough Boring (Sandvik)	T-MAX U 391.68	CCMT-UF -UR -UM CCGT-UM CCGX-AL
Face Milling (Kennametal)	High Shear MFCN	SEAN -AFN -AFTN LECN -AW -AWT	Face Milling (Sandvik)	T-MAX AL 265.2	SEC SPCN-A0 -A1 -X1 -SX1 SFC SPCX

Table 4.9 List of available choices and elements of insert geometry

The selection of tool bar depends on the following factors [Sandvik Tool Handbook, 1995]:

1. Workpiece material

2. Machining type: Heavy roughing, roughing .....

3. Duty

4. Depth of cut

The selection of a proper insert geometry depends on the following factors [SME Tool Handbook, 1983].

1. Workpiece material

2. Machining type: Heavy roughing, roughing .....

3. Depth of cut

4. Feed

The selection procedure for insert geometry is divided into the following steps:

Step 1. Take the first database record of insert geometry and set X = TMAX-20AL -SFA. (For facemilling)

Step 2. The relationship between X and workpiece is retrieved in the database base. The relationship is expressed by a fuzzy membership as shown in Figure 4.4.

Step 3. The relationship between the X and machining type can be expressed by a fuzzy membership function as shown in Figure 4.5. If the machining type is not given by the user, it can be determined by the cross-defined module.

The membership function of X with respect to different machining type can be expressed as follows:

$$MF(SFA/Machining\ Type) = MAX[\ MF(SFA/Machining\ Type_i) \wedge MF(Process \in machining\ type_i)]\ \text{for all } i$$

where i = heavy roughing, roughing....

Step 4. The average of the two membership functions generates the membership

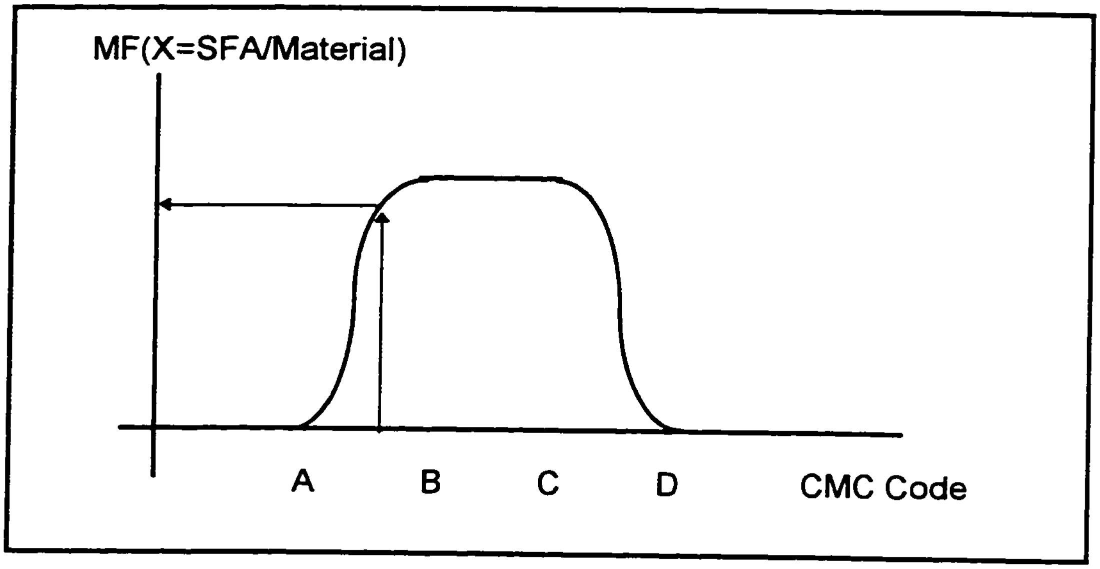


Figure 4.4 Membership function relationship between the insert geometry (SFA) and workpiece material

where A = long chipping material
B = stainless steel
C = alloyed steel
D = heat resistance steel
E = unalloyed steel
F = cast iron/casting steel

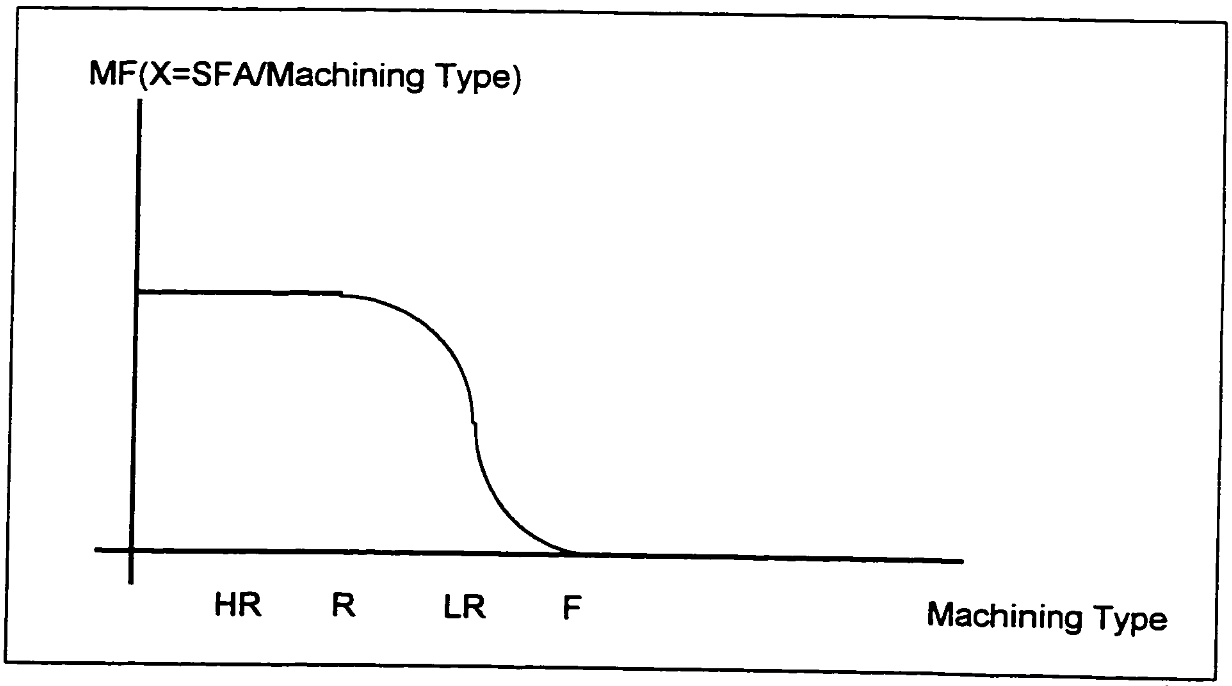


Figure 4.5 Membership function between insert geometry (SFA) and machining type function of Y.

$$MF(Y = SFA) = \frac{1}{n}\sum_{i=1}^{n} MF(SFA / Factor_i)$$

where n = number of input factors

$factor_1$ = material type

$factor_2$ = machining type

Step 5. If Y is not equal to "EOR" (End Of Database Record), then record = record + 1 and return to Step 1.

The above procedure represents the MAX-MIN procedure. It is used throughout the system while specifying the input variables and output results. For example, the membership value of insert geometry SFA can be obtained by calling the following procedure.

MAX-MIN (insert geometry = SFA, material type, machining type)

For some rotating cutters, especically short hole drills and end mills, solid cutters are used instead of inserts, because their diameters are too small to be inserted. Selection procedure for this kind of solid cutters is similar and is also included in the system.

(3) Insert Size

For most standard insert geometries, the insert size is specified by the diameter of the largest cycle that can be inscribed within the perimeter of the insert. This is generally referred to as the IC (inscribed circle) of the insert. When selecting an insert size, the IC should be large enough to meet the demands of the operation to be performed. For economical purposes, the smallest insert size that produces the desired depth of cut should be selected [SME Tool Handbook, p.8-35, 1985]. The insert shape, nose radius and insert size are correlated to the length of cutting edge. Figure 4.6 shows the graphical presentation of their relationships.

The variables are combined and expressed as the following equation:

$$L = \frac{d}{\cos(Ld)}$$

A rule of thumb is not to let depth of cut exceed 2/3 of the cutting edge length [Sandvik Tool Handbook, 1995].

$$I = 3/2 * L = \frac{3*d}{2*\cos(Ld)}$$

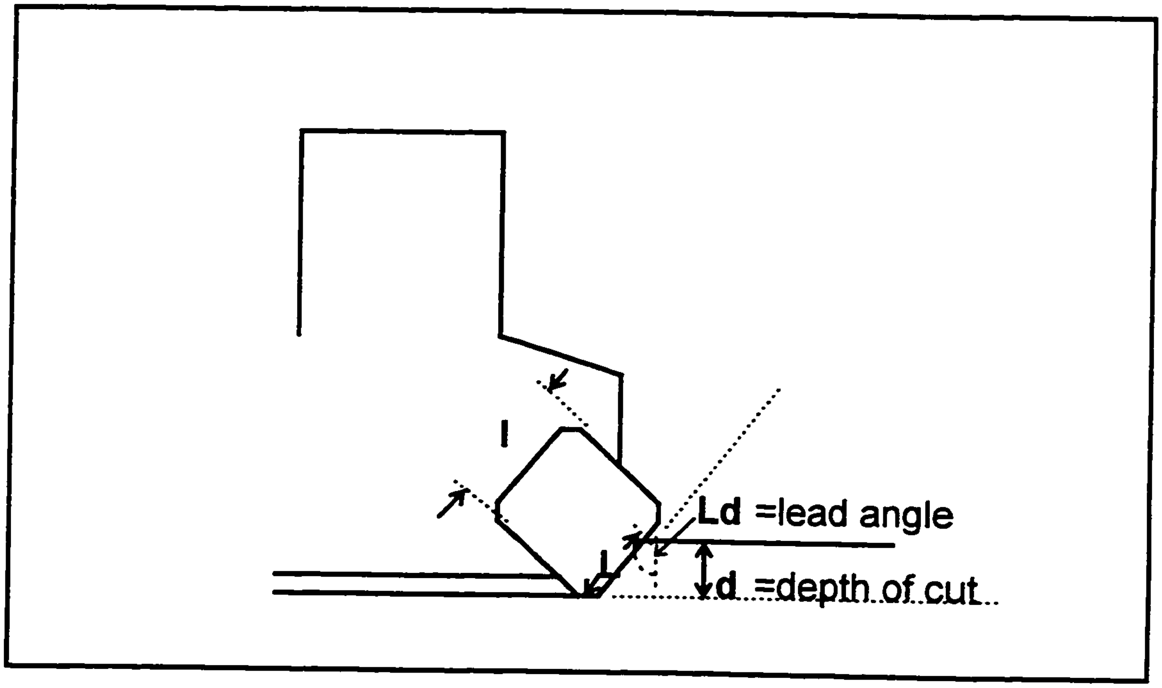


Figure 4.6 The Schematic presentation of insert size and others input variables where I = cutting edge length (insert size), Ld = lead angle, d = depth of cut and L=effective cutting edge.

The function can be mathematically expressed as followed:

$$\text{MF(insert size)} = \begin{cases} S(I,\ 0.8I,\ 0.9I) & \text{if } 0.8I \le I < 0.9I \\ 1 & \text{if } 0.9I \le I < 1.1I \\ 1 - S(I,\ 1.1I,\ 1.2I) & \text{if } 1.1I \le I < 1.2I \end{cases}$$

Figure 4.7 shows the membership function graphically.

The selection procedure is divided into the following steps:

Step 1: Determine the value of I using above Equation.

Step 2: Test the first database record of insert size i.e. TNMG-QM named by Y.

Step 3: Determine the fuzzy degree for Y using the predetermined membership function.

Step 4: Proceeds if EOR is reached. If not, go back to Step 1.

Step 5: Select the best insert size with highest fuzzy degree.

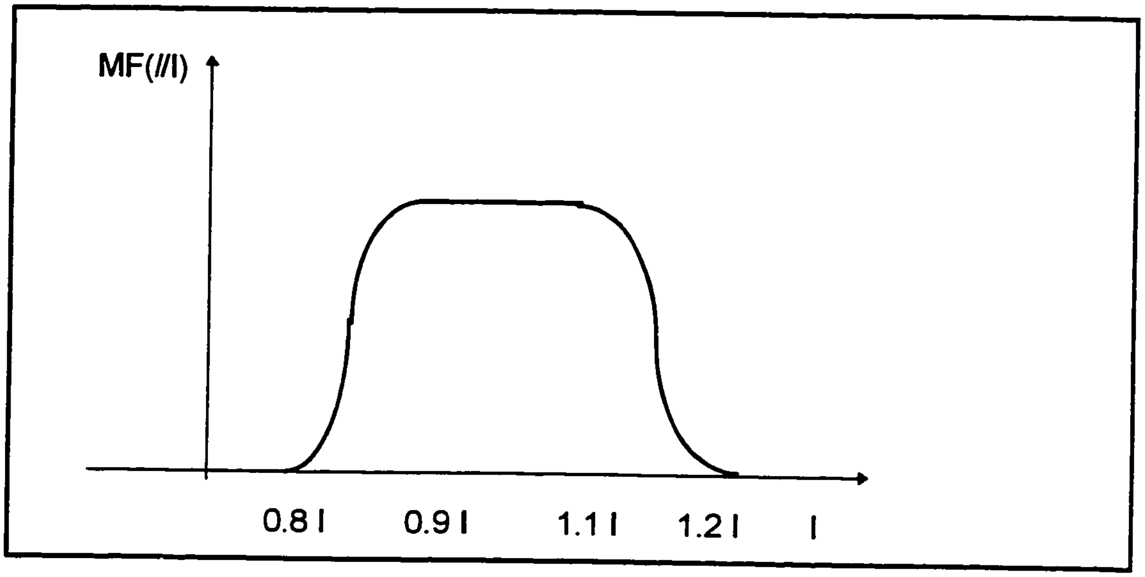


Figure 4.7 Membership function of insert size

(4) Insert Thickness

The thickness of the selected insert depends on the depth of cut and feed. [Devries, 1991]. When these two factors are known, the insert thickness can be selected from Table 4.10 [SME ToolHandbook, 1985].

The selection procedure is divided into the following steps:

Insert Thickness	Depth of Cut (mm)				Feed (mm/tooth)			
mm	a	b	c	d	a	b	c	d
2.4	2.2	2.8	3.6	4.2	.08	.11	.15	.2
3.2					.15	.22	.27	.32
4.8					.29	.37	.42	.49
2.4	3.8	4.4	5.2	5.8	.05	.09	.13	.17
3.2					.11	.17	.23	.29
4.8					.24	.32	.4	.46
6.4					.28	.38	.43	.48
4.8	5.6	6.2	6.6	7.2	.2	.29	.36	.42
6.4					.26	.35	.41	.44
10.5					0.29	0.38	0.46	0.5
4.8	7.1	7.5	8.4	8.8	.17	.26	.32	.38
6.4					.23	.31	.39	.42
10.5					.26	.34	.4	.45

Table 4.10 Thickness of insert for various depth of cut and feed

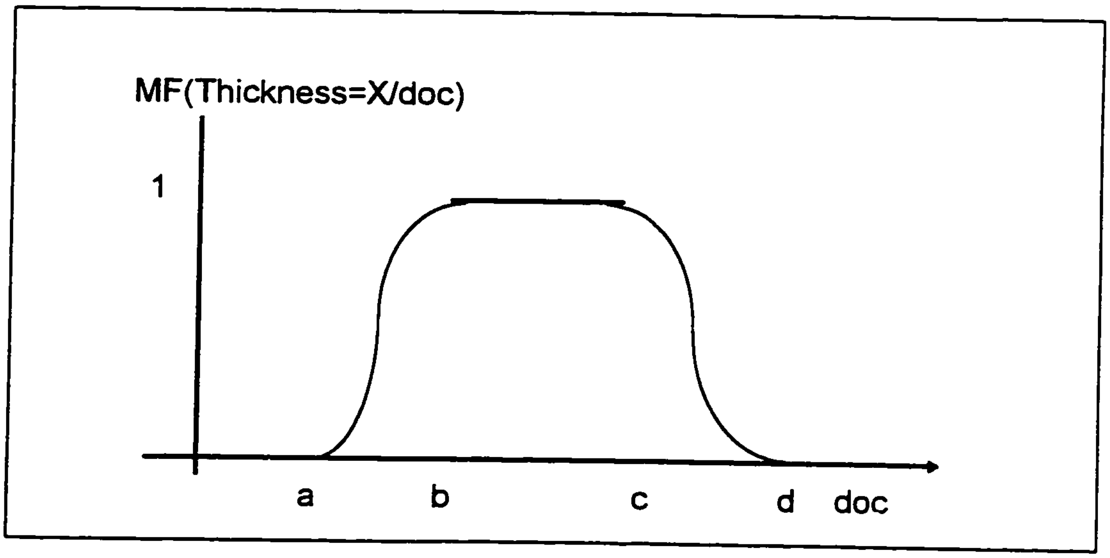


Figure 4.8 Membership function between X and the depth of cut

Step 1: Find the membership function of thickness X corresponding to its depth of cut. Figure 4.8 shows the membership function.

Step 2: Find the membership function of thickness X corresponding to its feed.

Step 3: Call the Max-Min procedure by MAX-MIN (thickness, doc, Fd) to obtain MF(Thickness = X).

If the value of depth of cut and feed are not given by the user, the cross-defined module is used to determine the membership function of the depth of cut and feed based on the selected machining type. Their cross relationships are given in the Table 3.5(a). Therefore,

$$MF(\text{insert thickness} = X) = 1/2\ \{MF(Doc \in \text{machining type}) \wedge MF(Doc/X) + MF(Feed \in \text{machining type}) \wedge MF(Feed/X)\}$$

(5) Insert Nose Radius

The configuration of the workpiece(necessary radii) and required surface finish dictate the nose radius used on an insert [SME ToolHandbook, 1985]. The applicability of the nose radius varies if different machining types are determined to be used. In addition to the MAX-MIN procedure, the system also possess "IF-THEN" rules to ensure a proper selection. For example, the selection of the cutter nose radius $\{c_1, c_2 \ldots\}$ is dependent on feed in machining type of heavy roughing, but in roughing, it is dependent on the required surface finish . Hence, the "IF-THEN" rules can be applied in this situation as follows:

IF machining type = {heavy roughing, roughing} THEN

$$MF(C_1) = MF(C_1/ Fd \in I) \wedge MF(Fd \in I)$$

$$MF(C_2) = MF(C_2/ Fd \in I) \wedge MF(Fd \in I)$$

....

where I = the selected machining type

ELSE machining type = {finishing, extreme finishing} THEN

$MF(C_1) = MF(C_1/\text{Surface finish}) \wedge MF(\text{Surface finish}) \wedge MF(\text{Feed} \in \text{low}) \wedge MF(\text{feed})$

$MF(C_2) = MF(C_2/\text{Surface finish}) \wedge MF(\text{Surface finish}) \wedge MF(\text{Feed} \in \text{low}) \wedge MF(\text{Feed})$

....

END IF

[Reference: Analysis of Material Removal Processes, Springer Verlag, p.104]

For *roughing* processes, Table 4.11 shows the relationship between the nose radii and the feed. [Sandvik Coromant product catalog, p.94]

Nose Radius (mm)	**Feed(mm/tooth)**			
	a	b	c	d
0.4	0.04	0.09	0.14	0.18
0.8	0.07	0.12	0.17	0.21
1.2	0.1	0.16	0.20	0.25
1.6	0.13	0.19	0.23	0.28
2.4	0.2	0.26	0.3	0.35
3.2	0.28	0.34	0.38	0.43
4.8	0.32	0.38	0.42	0.47

Table 4.11 Relationship between the nose radius and feed

For *finishing* processes, surface requirement becomes important. It is expressed as:

$$Ra = \frac{f^2}{32R}$$

where Ra = surface finish and R = insert nose radius. With this restriction, the desired surface finish can be obtained by choosing an appropriate insert nose radius. Table 4.12 shows the relationship between the nose radius and the surface finish [SME Tool Handbook, p.3-4].

Nose radius	**Surface finish required (μm)**			
(mm)	a	b	c	d
4.8	0	0.1	0.80	1.1
3.2	0	0.52	1.2	1.9
2.4	0.6	1.24	2.1	3.1
1.6	1.32	2.1	3.1	4.42
1.2	1.94	2.84	4.1	6.81
0.8	2.89	3.5	5.5	8
0.4	3.2	4	8	11.5

Table 4.12 The relationship between the nose radius and surface finish

4.2.3 Insert Grade Selection

The indexable inserts available in the system are made from high speed steel, uncoated and coated carbides, ceramics and cement with carbides being predominant. For example: Table 4.13 lists four common types of insert material offered by Sandvik and Kennametal Inc..

The selection of a grade for insert material depends on many factors. The factors affecting the selection of an insert material for a specific application include: [SME ToolHandbook, p.4-2, 1985]

1. hardness and conditions of the workpiece material
2. machining type to be performed
3. amount stock to be removed

4. accuracy and finish required

5. type, capability and conditions of the machine tool to be used

6. production requirement - speed and feed used

7. operating conditions - cutting force and temperature

8. tool cost per part machined

Selection of a proper cutting tool material for a specific machining application can provide productivity, quality and reduce costs. No single insert material is available to meet the needs of all these machining requirements. This is because of the wide range of conditions and requirements encountered. Each insert material has its own combination of properties making it best for a specific operation.

Insert material	**Choices**
Uncoated Carbide	H10A H13A S6 K68 ...
Coated Carbide	GC215 GC235 GC320 GC415 GC3015 GC4025 KC250 KT150 ...
Cermet	CT525 CT1525 CT5015 ...
Ceramic	CC620 CC650 CC670 CC690 K090

Diamond	CD100 KD100 ...

Table 4.13 List of common insert material available in the market

Important requirements for selecting the insert material are good wear resistance and toughness [SME ToolHandbook, p.3-1]. Wear resistance to the various tool wear is essential for the tool to retain cutting efficiency, as well as to provide part quality. Toughness is critical for tools to withstand high mechanical shock loads without chipping or fracturing, and heat deformation. Toughness is generally considered to be the ability of the material to absorb energy and withstand plastic deformation without fracturing under compressive load [SME ToolHandbook, p.3-3]. Unfortunately, for cutting tool application, any increase in wear resistance is generally accompanied by reduction in toughness.

Based on many studies and experiences, the wear resistance and toughness can be relatively measured by the following factors [SME Tool Handbook, p.8-4, 1985]:

1. Workpiece's Machinability Rating (**MR**) which will be discussed in next chapter
2. Tool life requirement
3. Cutting speed selected
4. Feed selected

There are two general rules for selecting the most suitable insert material. They are linguistically shown as follows:

Rule 1: IF MR is *HIGH* and the Tool Life required is *LONG* Then

The insert material must be *MORE* wear resistance

& Cutting speed must be *LOW*

END IF

Rule 2: IF MR is *HIGH* and the selected Speed and Feed are *HIGH* Then

The insert material must be *TOUGHER*

& The tool life must be *SHORT*

END IF

They can also be expressed as follows:

Rule 1: MF(wear resistance $\in$high) = MF(MR $\in$high) $\wedge$MF(Tool life $\in$long) also implies that

MF(cutting speed $\in$low) and MF(Feed $\in$low) and MF(toughness $\in$low)

Rule 2: MF(toughness $\in$high) = MF(MR $\in$high) $\wedge$MF(cutting speed $\in$high) ^ MF(feed $\in$high)

also implies that MF (tool life $\in$short) and MF (wear resistance $\in$low)

Before discussing the selection procedure, the fuzzy membership function of (machinability rating $\in$ high) must be defined as shown in Figure 4.9.

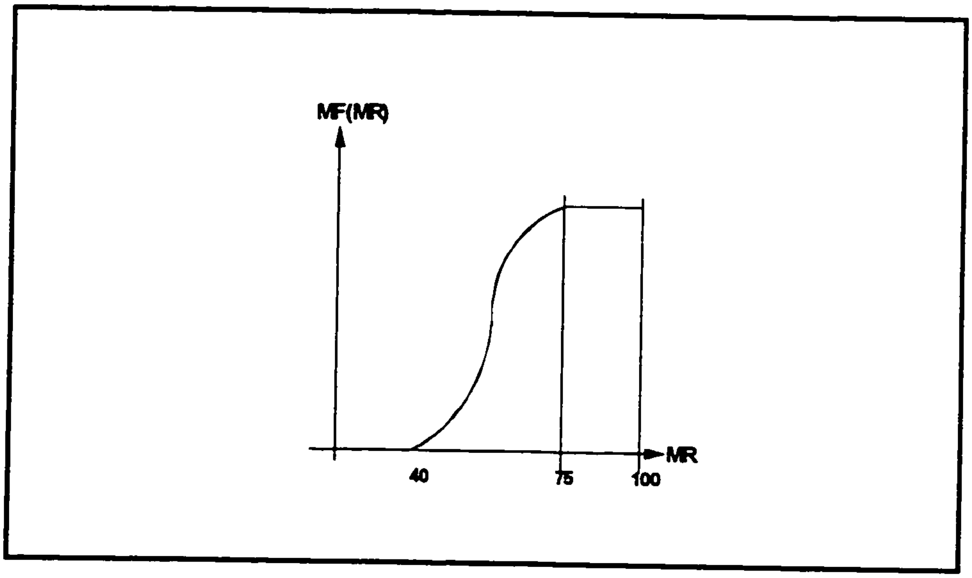


Fig 4.9 Fuzzy membership function of high machinability rating

The selection procedure for insert material is divided into the following steps:

Step 1: Look up the specific insert grade table which designates to the workpiece materials in each product supplier's handbook.

Step 2: Find MF(wear resistance) and MF(toughness) by the rule 1 and rule 2.

Step 3: In each geometry column, a fuzzy degree is assigned to each grade element according to its MF(wear resistance) and MF(toughness) by the rule 1 and rule 2.

Step 4: Combining the MF(wear resistance) and MF(toughness) gives a fuzzy degree to each grade element in the table 4.13.

Step 5: Select the best grade with the highest fuzzy degree.

4.2.4 Cutter Database Searching

When the fuzzy memberships of all the cutter fields are determined, they are combined to generate the fuzzy membership of the cutter and grade:

$$M(\text{cutter}_i) = \frac{1}{m} \sum_{j=1}^{m} M(\text{field}_j)$$

where j = 1,...m denotes the related fields. The combined fuzzy membership function $M(\text{cutter}_i)$ determines the applicability of the cutter. (M = 1 means the best, M = 0 means totally unsuitable)

The final step is to search against the built-in cutter databases to determine the availability of the selected cutters. The search is conducted by the sequential searching method as shown in Figure 4.10. The available cutter(s) and grade(s) with their fuzzy membership(s) are the outputs of the system.

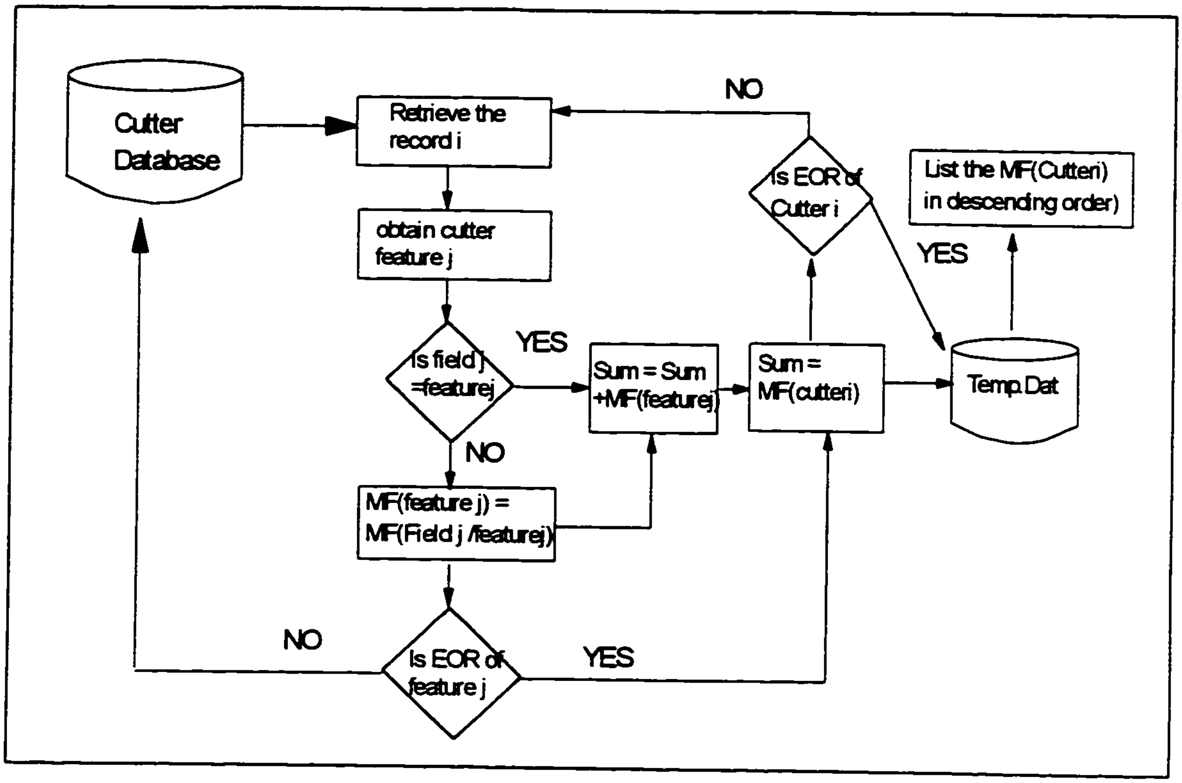


Figure 4.10 The flowchart used for the database searchinig

4.3 Database Management

In the expert system, there are three types of databases.

1. Machinability Database

The term "machinability" can be generally defined as the degree of "ease of machining" [Schey, 1988]. For a given set of machining conditions, the ease of machining varies with the workpiece material variables. Common workpiece material variables affecting ease of cutting are: [SME Tool Handbook, p.1-41]

1. Hardness
2. Tensile properties

3. Chemical composition

4. Microstructure

5. Shape and dimensions of workpiece

6. Rigidity of workpiece

Common criteria for judging ease of cutting are classified in terms of three factors:

1. Tool life between resharpening

2. Magnitude of the tool forces, machining work as energy or power consumption

3. Quality of surface finish produced on the workpiece

These criteria are very important for the cutter selection as well as the cutting conditions design. Measurement of machinability is a very difficult task because it is not only the function of metal's own metallurgical properties such as microstructure, but also a function of machining type. [Amargeo and Brown, 1965]. Therefore, a general measurement called machinability rating was developed by METCUT Research Association in Cincinnati, U.S.A. to express the metal's machinability in terms of relative values. It is based on a tool life of 60 minutes. The standard is AISI 1112 steel, which is given a rating 100. Thus for a tool life of 60 minutes, this steel should be machined at a cutting speed of 100ft/min (30m/min). Higher speeds will reduce tool life, and lower speeds will increase it. For example, a metal with a rating of 50 should be machined at *roughly* half the speed and feed used for a material with a rating of 100 to obtain same tool life of 60 minutes.

A data file called MACH.DEX is designed to store all the necessary information. The data contents of the machinability databases are described in Table 4.14 [SME Tool Handbook, p.2-15, 1985].

Material Type	Conditions	Typical Hardness	Specific Energy ($hp*min/in^3$)	Machinability Rating	HSS		Carbide		Cement		Ceramic	
					n	C	n	C	n	C	n	C
Unalloyed Steel	Annealed & cold drawn	90-110	0.8-1	120	0.16	250	0.29	1000	0.33	2000	0.42	3000
Low alloyed steel	Annealed & cold drawn	115-120	1-1.2	100	0.11	236	0.24	946	0.34	1893	0.4	2840
High Alloyed Steel	Hot rolled & cold drawn	140-155	1.1-1.3	80	0.1	216	0.25	866	0.3	1733	0.43	2600
Chilled Cast iron	Cold drawn	250-350	0.6-2	55	0.1	137	0.26	547	0.31	1093	0.43	1640
Heat resistance steel	Annealed & cold drawn	151-161	1.2-1.8	75	0.1	170	0.25	680	0.36	1360	0.41	2040
Aluminium alloy	-	120-140	0.15-0.4	185	0.11	260	0.28	1040	0.32	2080	0.42	3120
Stainless steel	-	180-220	1.1-1.9	87	0.14	183	0.21	733	0.3	1466	0.42	2200
Steel casting	Annealed & cold drawn	170-230	0.8-1	65	0.11	130	0.19	520	0.27	1040	0.38	1560
....	....	......	.....	......	....	...	....	...	....	...	....	...

Table 4.14 Data content of Machinability File [Schey, Introduction to manufacturing process, McGraw Hill, 1987 p.436]

* *machinability rating listed are based on cold drawn low alloyed steel AISI 1112 for 60 mins tool life. This value involves machining at a cutting speed of 30 m/min for feed up to 0.20mm/rev.*

where C = constant in the Taylor's equation whose values depends on each combination of workpiece material andtool material. The value of C is numerically equal to the cutting speed that gives a tool life of 1 minute.

n = exponent in the Taylor equation whose value varies to some extent with the tool material and workpiece material variables.

2. Cutter Inventory Database

The system also possess a cutter inventory database (including adaptor tool bar and insert information) to check availability of the selected cutters. Currently the inventory database contains more than three thousand different dimensional tool bars from more one hundred series, hundreds of inserts with different grade from couples of big tooling companies. The record is stored by method of Random Access(RA). The database can be modified by record addition, deletion and edition. The data content is summarized in Table 4.15.

Cutter Information	Variable	Field Space	Field Type
Supplier	Supp	10	Text
Cutter name	C_Name	10	Text
Diameter	Dia	3	Numeric
Thickness	Tks	2	Numeric
Pitch Type	PTy	4	Text
Geometry	Geod	5	Text
Nose Radius	NRa	3	Numeric
Lead Angle	Ld	3	Numeric
Size	Sz	3	Numeric

Table 4.15 Data content of cutter inventory database

3. Machine Tool Database

A machine tool database is developed and named as MACHINE.DAT. It contains more than 10 machine tools. User must specify the machine tool from the database before the system performs the selection activity. The machine tool database has the ability to be modified. The data content of the MACHINE.DAT is presented in Table 4.16.

Field Name	Unit	Variable	Field Space	Field Type
Machine Name	--	M_Name	10	Text
Max. Spindle Speed	m/min	CS_{max}	3	Numeric
Max. Feed	mm/rev	Fd_{max}	3	Numeric
Min. Spindle Speed	m/min	CS_{min}	3	Numeric
Min. Feed	mm/rev	Fd_{min}	3	Numeric
Max. Cutting Dimension	mm	Dim_{max}	3	Numeric
Max. Cutting Power	KW	CP_{max}	2	Numeric

Table 4.16 The data content of Machine Tool Database

4.4 Fuzzy Rule Base Schema

The fuzzy set theory is used for handling uncertainty and imprecision inherent in the knowledge representation. The performance of the following modules is based on the rules presented in the knowledge base.

1. Cross-Defined Modules for the missing input information
2. "IF-THEN" rules for cutter selection
3. Input-output relationships for cutter and grade selection

The rules in the rule base are not based on predicate logic (Either Yes or No). The rules presented using the fuzzy set theory can better simulate the human logical thinking and decision process [Ishibuchi and Fujioka, 1993]

The system is composed of multiple inputs and multiple outputs which can be presented as a block diagram shown in Figure 4.11.

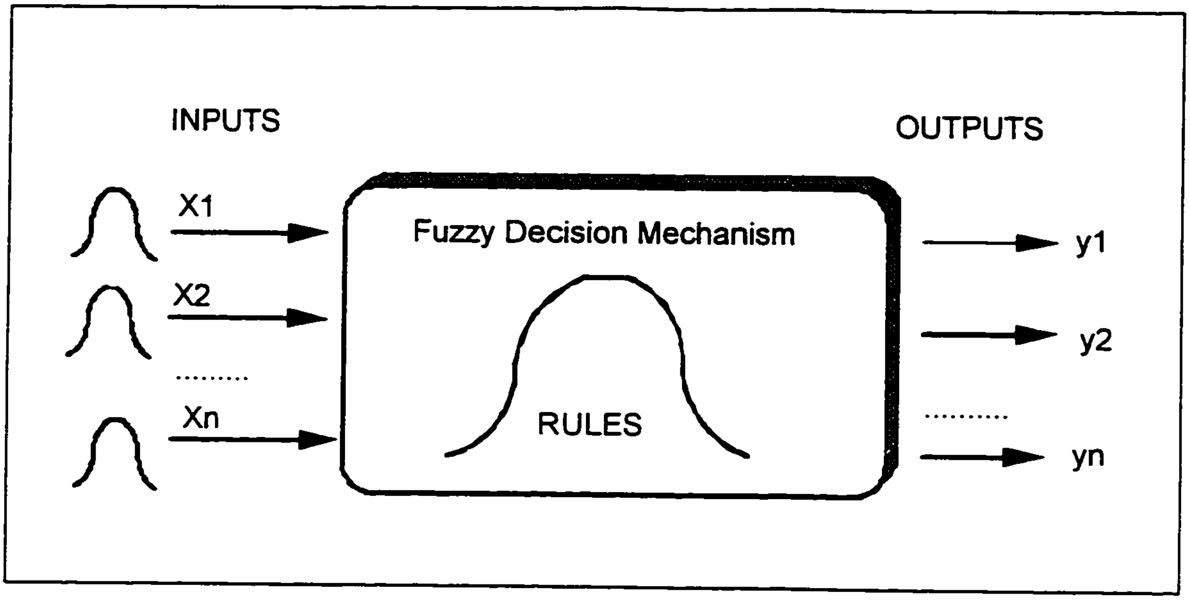


Fig 4.11 A block diagram for a multi-input and multi-output system

In the system, each rule presented in the knowledge base deals with only one output. The output can either be the selection of insert's features such as insert thickness or the determination of missing information such as machining type. The rules can be generally shown as follows:

R^1: IF $X_1 \in A_1^1$ AND $X_2 \in A_2^1$ AND ..... $X_n \in A_n^1$ THEN Y_1 is B^1

R^2: IF $X_1 \in A_1^2$ AND $X_2 \in A_2^2$ AND ..... $X_n \in A_n^2$ THEN Y_1 is B^2

....

....

R^r: IF $X_1 \in A_1^r$ AND $X_2 \in A_2^r$ AND ..... $X_n \in A_n^r$ THEN Y_1 is B^r

where $A_i^K \in MF(X_i)$; $U_{Ai}^K(X_i) : X_i \longrightarrow [0,1]$

$B_j^K \in MF(Y_y)$; $U_{Bj}^K(Y_j) : Y_j \longrightarrow [0,1]$

Two examples are given:

Example 1: The membership function of insert thickness of 3.2mm can be determined by the following rule.

R5: IF Doc $\in A_1$ and Feed $\in A_2$ Then

MF(insert thickness = 3.2) = B_1

where A_1, A_2 are the shape functions parameterized by a,b,c,d which can be found in the Table 4.4.

Example 2: In the Cross-Defined Modules, the machining type can be determined by the feed and depth of cut. Hence,

R14: IF Doc $\in A_1$ and Feed $\in A_2$ Then

MF(Machining type $\in$heavy roughing) = B_1

where A_1 and A_2 can be found in the Table 4.8. The value of fuzzy degree B_1 varies if the process is classified as a different machining type. Currently, the system consists of 89 rules to construct the rule base. The rules and their duties are organized in Table 4.17.

Rule number	Description
Rule 1 to 11	Cross definition between the machining type and feed and doc
Rule 12 to 15	Cross definition between tool life and cutting speed and feed
Rule 16 to 22	Cross definition between MRR and cutting speed, feed and doc
Rule 23 to 29	Rules for insert geometry selection
Rule 30 to 35	Rules for insert size selection
Rule 36 to 46	Rules for insert thickness selection
Rule 47 to 50	Rules for insert nose radius selection
Rule 51 to 62	Rules for insert material selection
Rule 63 to 89	"If - then " rule

Table 4.17 The rules and their duties in the system

4.5 Optimal Cutting Condition Design and Cost Analysis Using Fuzzy Non-linear Programming Model

After selecting the appropriate cutters and tool grades, the next step is to design optimum cutting conditions. The proposed model is based on the assumptions below:

- The tool life can be reasonably predicted by the Taylor equation.
- The objective function and constraints can be represented by fuzzy sets.
- The tolerance T_i for each constraint is known.
- Lubricant is used when applicable.

Defining a machining optimization model, which includes the construction of an objective function and constraints, is a complicated process. Depending on the tool material, workpiece material, and type of turning process, constraining resources have to be decided before the entire non-linear programming model can be constructed. Determination of the machining limitations and specifications may involve many human decisions or factors. For example, when surface finish of 1.4 mm is specified, it may be imprecise or fuzzy. Can the surface finish be 1.41mm instead of 1.4mm ? It may be argued that the surface finish of 1.41mm is still "acceptable". The proposed method is to use fuzzy set theory to deal with the vagueness found in the machining optimization model.

1. Objective Function of the Proposed Model

Prior to the development of the model, all of the cost elements in the objective function are defined in this section. In metal cutting industries, there are two categories of cost. These categories are direct cost and indirect cost.

(1) Direct Cost includes:

Operator Cost which can be expressed as the product of operator rate C_0 and unit machining time T_m.

$$\text{operator cost} = C_o * T_m$$

Tool Depreciation Cost which can be expressed by real cutting time T_m' of each insert as follows:

$$Tooling\text{-}Cost = \frac{T_m'}{T_L} T_N C_t$$

where

$$T_m = \frac{\pi DL}{V * F}$$

$$T_m' = T_m * \frac{\arcsin(\frac{WOC}{D})}{\pi}$$

$$T_L = (\frac{C}{V})^{\frac{1}{n}}$$

note that C and n can be obtained in the machinability database.

(2) Indirect Cost includes

Tool Changing Cost can be expressed as:

$$\text{Tool Changing Cost} = \frac{T_m'}{TL} T_{ch} C_o$$

Holding Cost can be expressed as the product of operator rate and unit holding time.

$$\text{Holding Cost} = C_o * T_h$$

Repair/Maintenance Cost can be determined by the time spent and the frequency for the tool changing. In fact, the unit of production cost and product quality are significantly influenced by the frequency required by the tool changing.

Capital Cost is the consumption of captial investment, for example: machine tool, workpiece material cost. They are either constants or affect total cost slightly, and are not considered.

In summary, the unit production cost can be mathematically expressed as follows:

$$C_p = C_o T_m + \frac{T_m'}{T_L} C_t + \frac{T_m'}{T_L} T_{ch} C_o + C_o T_h$$

The objective (Max. production rate, profit) functions could be obtained Similarly:

$$\text{Production rate} = \frac{R_t}{R_t * T_m + T_{ch} + R_t * T_h}$$

Profit=Production rate*(Price-C_p)

2. Constraints of the Proposed Model

(1) Machine Tool Specification

a. The maximum and minimum cutting speed:

$$\tilde{V}_{min} < V < \tilde{V}_{max}$$

b. The maximum and minimum feed range:

$$\tilde{f}_{min} < f < \tilde{f}_{max}$$

where the variables with a " ~ " are fuzzy.

(2) Machine Tool Dynamic

a. Power Availability limitation

$$Pc < \tilde{Pc}_{max}$$

$$P_c = MRR * Spec./\eta$$

$$MRR = f*SP*DOC*WOC$$

$$SP = V/(DL)$$

b. Cutting force limitation [Schy et al, 1988]:

$$Fc < \tilde{Fc}_{max}$$

$$F_c = \frac{P_c * \eta}{V}$$

(3) Part Design Specification

Surface Finish requirement:

$$SF < \tilde{SF}_{max}$$

$$SF = 2.2 * 10^4 V^{-1.52} f$$

The equivalent crisp non-linear programming model [Trappey et al, 1988] is shown as follows:

MAX α

S.T. $$C_o T_m + \frac{T_m'}{T_L} T_N C_t + \frac{T_m'}{T_L} T_{ch} C_o + C_o T_h \le b_0 + (1-\alpha) T_0$$

$$-V \le -V_{\min} + (1-\alpha) T_{V_{\min}}$$

$$V \le V_{\max} + (1-\alpha) T_{V_{\max}}$$

$$-f \le -f_{\min} + (1-\alpha) T_{f_{\min}}$$

$$f \le f_{\max} + (1-\alpha) T_{f_{\max}}$$

$$\frac{\eta P_c}{V} \le F_{max} + (1-\alpha) T_{F_{max}}$$

$$\frac{MRR * Spec.}{\eta} \le Pc_{max} + (1-\alpha) T_{Pc_{max}}$$

$$MRR = f * V * DOC * WOC / (D * L)$$

$$2.2x\, 10^4 V^{-1.52} f \le SF_{\max} + (1-\alpha) T_{SF_{\max}}$$

$$1 \geq \alpha \geq 0$$

3. Time Replacement Schedule

After the cutting conditions (cutting speed, feed) are determined, the system can automatically schedule the tool replacement time in order to achieve the minimum unit production cost since the prolongation of tool usage raises the reduction of part quality. Base on the Taylor Equation, the predicted tool life with respect to the optimum cutting speed is:

$$T_{L\,opt} = \left(\frac{C}{V_{opt}}\right)^{\frac{1}{n}}$$

The insert cutting time is:

$$Tm' = \frac{\arcsin\left(\frac{WOC}{D}\right) DL}{V_{opt} f_{opt}}$$

The optimal tool replacement schedule is:

$$R_t = \frac{T_{L\,opt}}{Tm'}$$

therefore, after R_t number of parts are produced, the cutter should be replaced.

4. The Interpretation of this Model

The total unit cost for each workpiece, Cp must be at least less than b_o, an aspiration level selected by the job shop manager. The aspiration level can be violated t_o, up to a certain maximum tolerable T_o.

The constraints are satisfied as well as possible, Given that each constraint has a level of violation (t_i, i=1,,,,n). each constraint violates the conditions of the problem to a maximum tolerable degree, T_i. Each value of T_i is established by the job shop manager and depends on his best judgement of the job shop environment. If the violation are increased, the manager's satisfaction decreases. Therefore, the fuzzy optimization machining model becomes more flexible and human oriented.

4.6 Learning Mechanism

In the system, a large number of fuzzy membership functions are used. These functions are critical to the performance of the system. Based on these functions, the optimal cutter and cutting conditions can be obtained. But these functions themselves are usually empirical functions and may not be precise, because the perdetermined initial points of functions are very roughly selected. Hence, a learning module could be developed so that users can fine-tune these functions (by changing the coefficients of the functions) to accommodate their own preference.

Here, because of limitation of time, effort and experiment equipment, only the algorithm of learning mechanism is proposed, but it is not fulfilled in present system.

As pointed out in the chapter III, an element y_k in a given field is related to a mumber of inputs, x_j, $j = 1, 2, ..., n$, with the fuzzy membership function $M (y_k/x_i) = S(x_i, a_{ki}, b_{ki}, c_{ki}, d_{ki})$. An inappropriate selection (e.g., inadequate thickness) may be caused by any one of the $M(y_k/x_j)$, $j=1, 2, ..., n$. The learning mechanism is controlled by the user; it starts by identifying the input which has the largest influence on the selection. Suppose this input is x_i , then the learning procedure modifies the fuzzy membership function by:

$$a_{ki}= a_{ki}+\alpha(a_{k\pm 1,i} - a_{ki})$$

$$b_{ki}= b_{ki}+\alpha(b_{k\pm 1,i} - b_{ki})$$

$$c_{ki}= c_{ki}+\alpha(c_{k\pm 1,i} - c_{ki})$$

$$d_{ki}= d_{ki}+\alpha(d_{k\pm 1,i} - d_{ki})$$

where, $0 \leq \alpha \leq 1$ is the learning coefficient, and the subscript $k\pm1$ represents the change toward the neightboring elements in the field. when it is felt that the selection is too large, the negative sign is taken and when the selection is too small, the positive sign is taken. Furthermore, if the current element is the largest (smallest) in the field, then $a_{k+1},i = b_{k+1},i = 0$ ($c_{k-1},i = 0$). Next, to ensure consistency, the fuzzy membership functions of all elements in the field (except the first one and the last one) are also modified using the same formula with the learning coefficient β, where $0 \le \beta < \alpha$. This process is illustrated in Figure 4.12. Suppose it is felt that the selection is to be too small and $M(y_k/x)$ is identified to have the largest influence; then it will be moved towards $M(y_{k+1}/x)$. This is done by resetting the parameters a_k and b_k. Assuming $\alpha=1/2$, it follows:

$$a_k=a_k+(a_{k+1}-a_k)/2=(a_k+a_{k+1})/2$$

$$b_k=b_k+(b_{k+1}-b_k)/2=(b_k+b_{k+1})/2$$

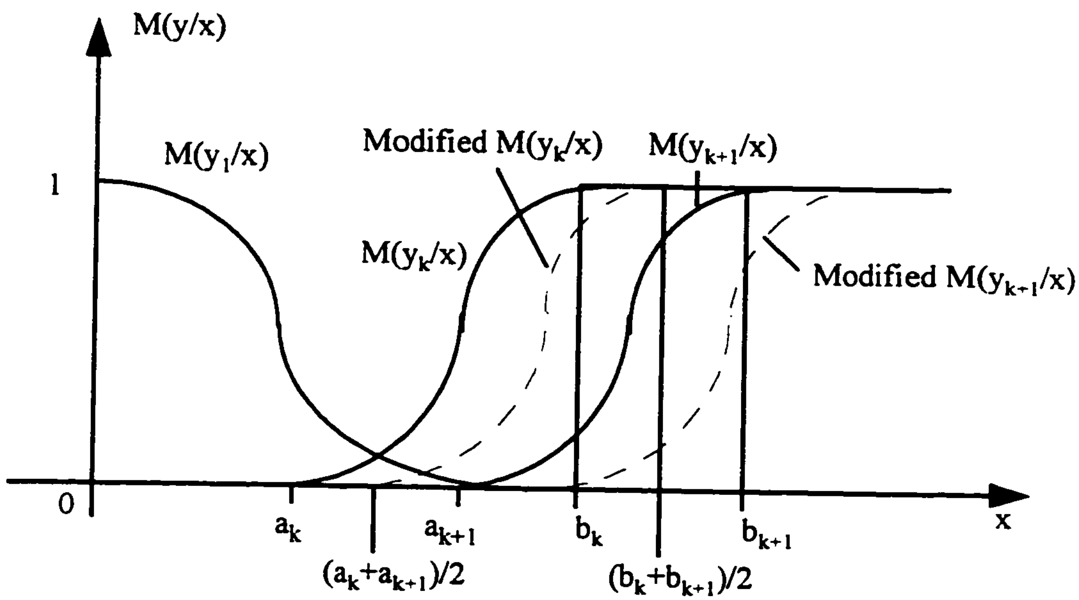


Figure 4.12 Illustration of the modificationof a fuzzy membership function ($\alpha=1/2$, $\beta=1/4$)

Next, $M(y_{k+1}/x)$ is also modified. With $\beta=1/4$, it follows:

$$a_{k+1}=(3a_{k+1}+a_{k+2})/4$$

$$b_{k+1}=(3b_{k+1}+b_{k+2})/4$$

The modified fuzzy membership functions are then used to obtain a new selection. This process is repeated until user felts the selecion is satisfactory. The learning procedure is given in Table 4.18.

For the example of cutter thickness selection, if a user feels that the selected thickness T_1 is too small, then the learning results in a number of changes to the fuzzy membership functions. In particular, the fuzzy membership function between T_1 and depth of cut becomes:

$$M(T_1/d)=\begin{cases} S(d,\ 0.035,\ 0.055), & \text{if } 0.035 \le d \le 0.055 \\ 1, & \text{if } 0.055 \le d \le 0.075 \\ 1-S(0.075,\ 0.095), & \text{if } 0.075 \le d \le 0.095 \end{cases}$$

The learning procedure is also adopted to justify the fuzzy membership functions used in the cutting condition design.

So once equipped with this learning model, only if the rules reflect the cross related between inputs and output, and are qualitatively correct, the satification of the selection can be reached.

```
Procedure learning(field, related_input):
//identify the fuzzy membership functions to be modified
for k=1 to m do y[k]=field_element[k];
for j=1 to n do x[j]=related_input[j];
for k=1 to m for j=1 to n retrieve M(y[k],x[j],p[k,j]);
//note that p[k,j]=[a[k,j],b[k,j],c[k,j],d[k,j]] is a vector
repeat
//identify the input xi that has largest influence
x[i]= max{M(y[k],x[j],p[k,j])};
//modify M(yk/xi) by α
        if y[k] is too large
                        if l>1 then p[k,i]=p[k,i]+ α(p[k-1,i]-p[k,i]);
                        else p[k,i]=p[k,i]- αp[k];
                        end;
        if y[k] is too small
                if k<m then p[k,i]=p[k,i]+ α(p[k+1,i]-p[k,i]);
                else p[k,i]= p[k,i]+ αp[k];
                end;
//modify all the M(yk/xi) k=1,...,m by β
        for kk=2 to m-1 and kk !=k do
                if y[k] is too large p[kk,i]=p[kk,i]+ β(p[kk-1,i]-p[kk,i]);
                if y[k] is too small p[kk,i]=p[kk,i]+ β(p[kk+1,i]-p[kk,i]);
        end;
//determine the new selection
calculate the new M(y[k],x[j],p[k,j]);
        calculate the new M(y[k]);
        calculate the new selection y[k];
until y[k] is right;
end;
```

Table 4.18 The learning procedure

CHAPTER V

CASE STUDIES

A case study illustrates this fuzzy expert system described above. Two general rotating operations are chosen as examples for cutter selection and cutting conditions design.

5.1 Example 1

1. <u>Inputs to the system</u>

This example deals with a face milling operation with the objective of minimizing the product cost. The specifications of the machine, the workpiece, the cost factors, and the job are listed in Figures 5.1, 5.2, 5.3, and 5.4 respectively. As previously mentioned, the system is capable of understanding imprecise and/or incomplete information. As a result, it is not necessary to fill all the slots in the specifications of the machine, the workpiece, the cost facts and the job. The users can also pre-select a preferred cutter and specify a set of preferred setup data including cutting speed, feed, spindle speed and tool life as the initial setup as shown in Figure 5.5 and Figure 5.6. The system will estimate and compare the performance of user preference and system recommandation.

Since the input information is incomplete, the cross-define relationship module is used to determine the missing information. According to the fuzzy degrees defined in the Table 4.4(a), the machining type can be cross-defined by the inputted feed and depth of cut. Table 5.1 shows the membership functions for different machining types.

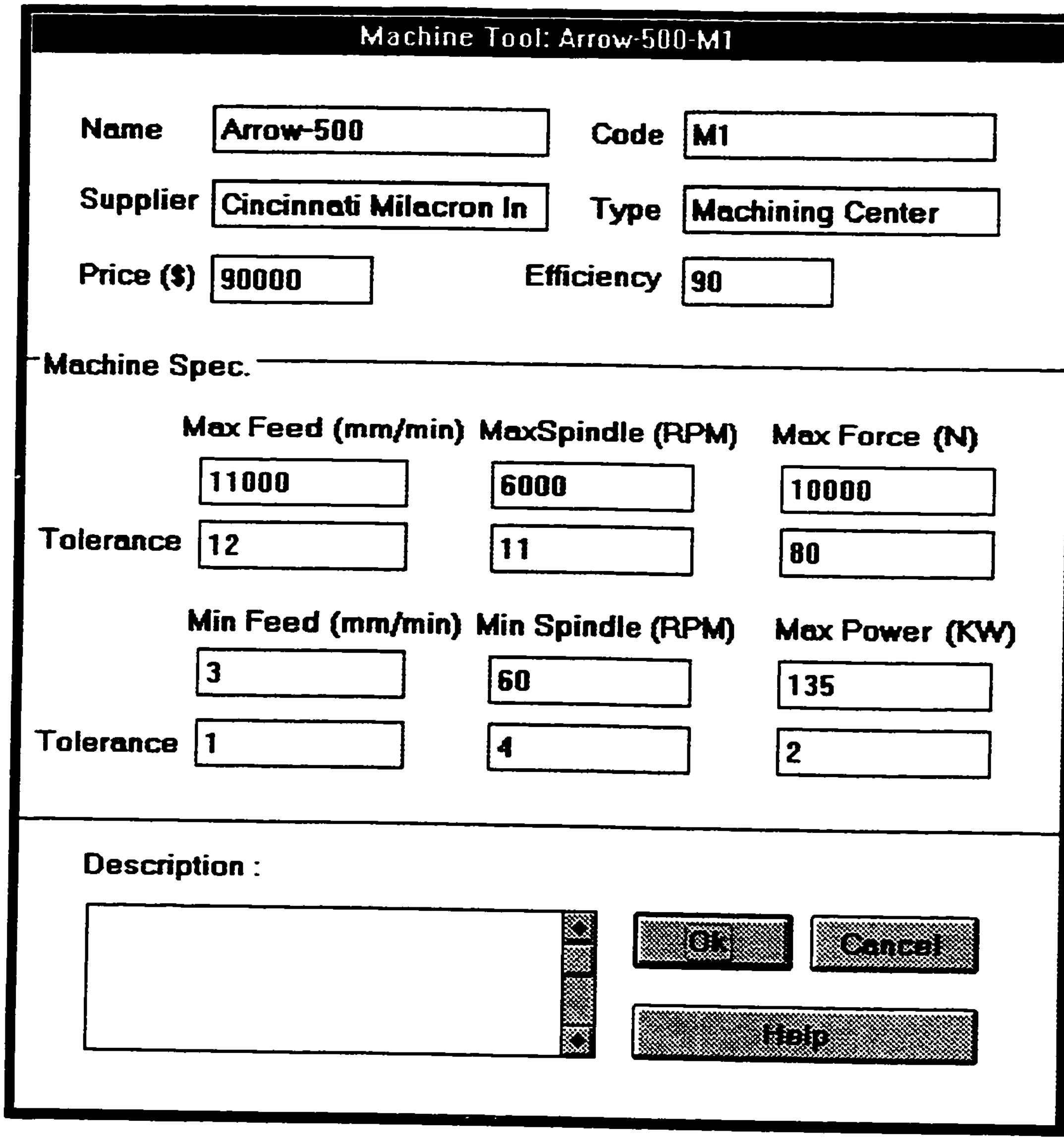


Figure 5.1. Machine Tool Specifications

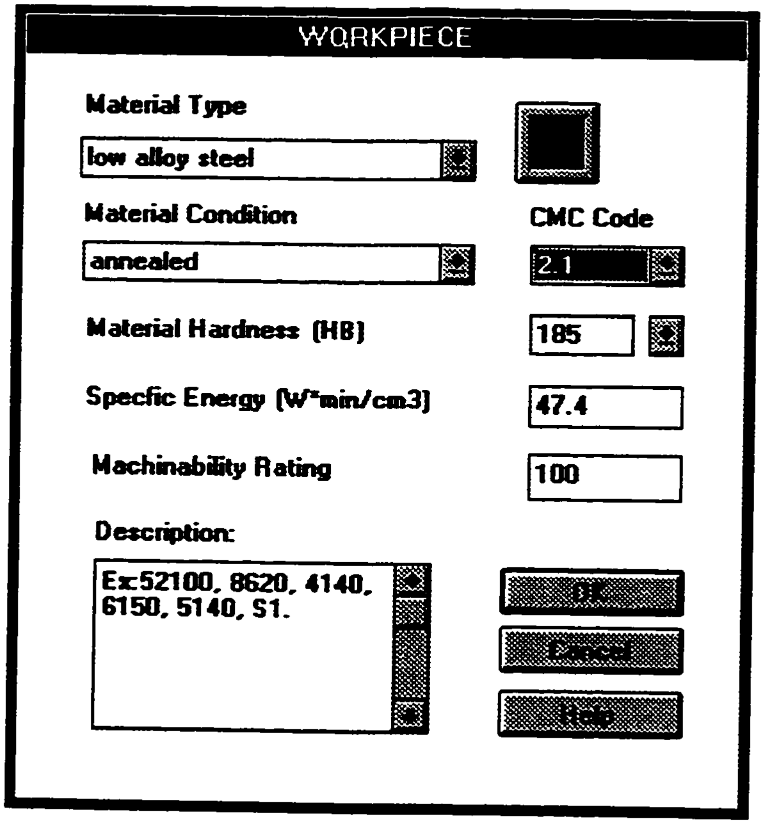


Figure 5.2. Workpiece information for example 1

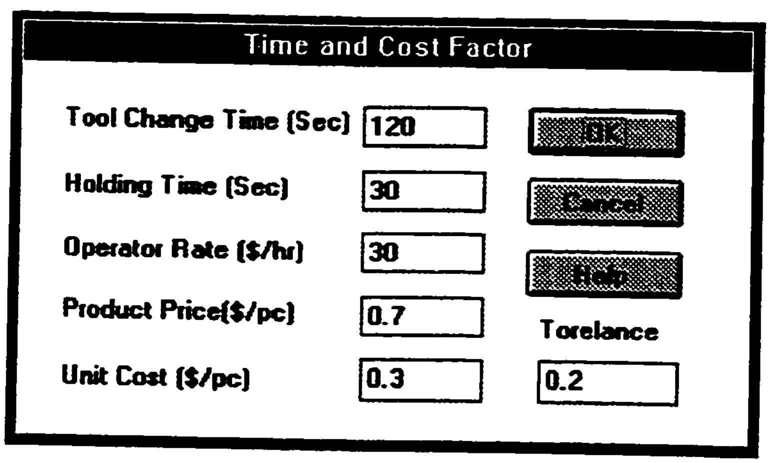


Figure 5.3. Time and Cost information

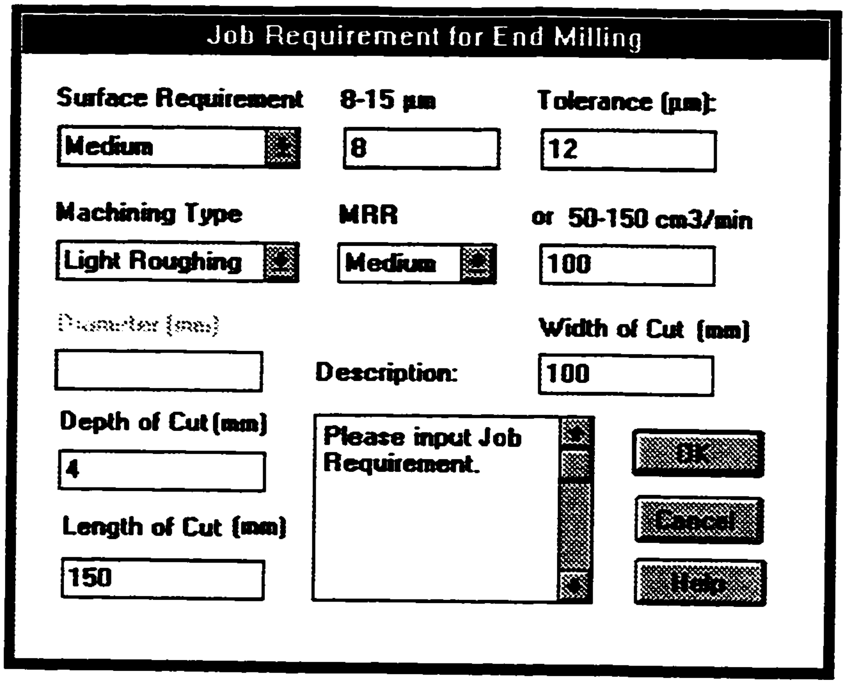


Figure 5.4. Job Requirement for example 1

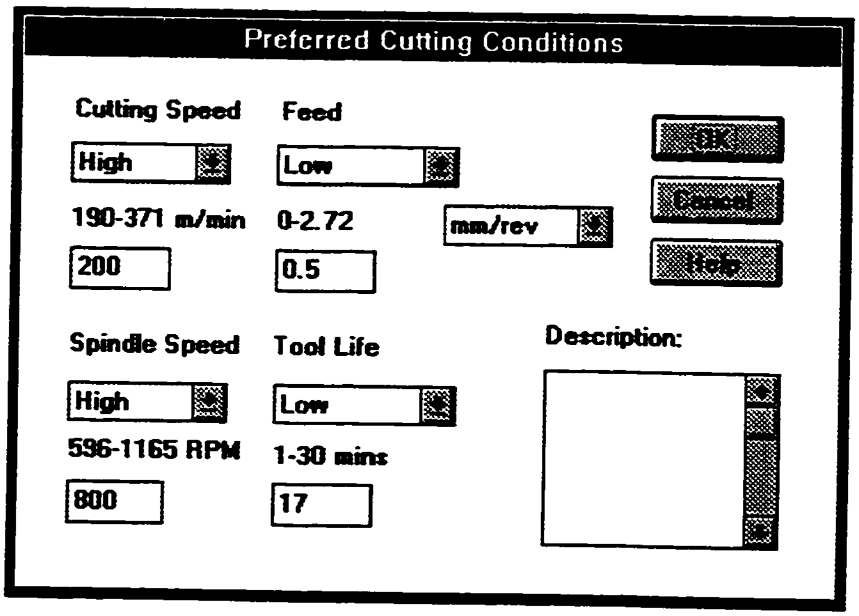


Figure 5.5. User Preferred Cutting Data

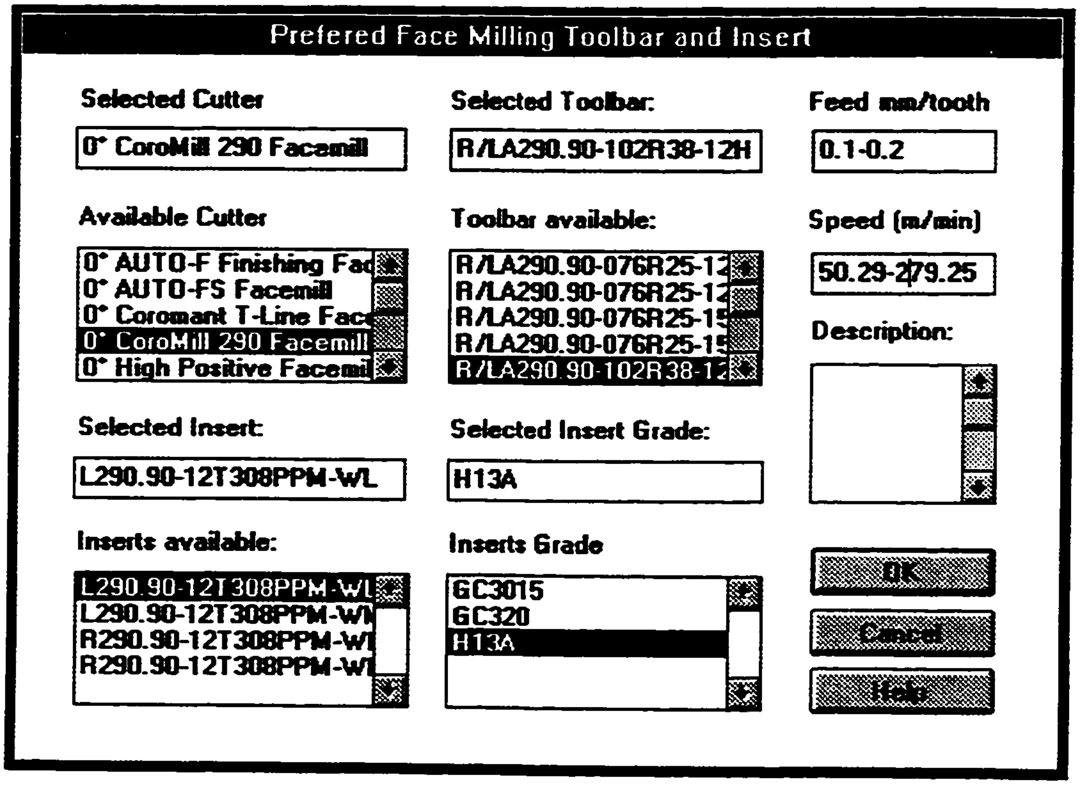


Figure 5.6 User preferred cutter

Machining types	**Fuzzy membership degree**
Heavy roughing	0
Roughing	0.4
Light roughing	**1**
Finishing	0.74
Extreme finishing	0.11

Table 5.1 Membership function classification of machining type

* The surface finish is being classified as:

-M(surface= *medium*)=M(8,5,8.5,12,15)= 0.74

-M(surface= *rough*)=M(8,7,12,20,20)= 0.53

-M(surface=*fine*)=M(8,0,0,5,8.5)= 0.34

* The MRR is being classified in a similar manner:

- *medium* with a fuzzy degree of 1

- *high* with a fuzzy degree of 0.65

- *low* with a fuzzy degree of 0.41

Table 5.2 summarizes the fuzzy degree obtained after the cross-defined module is performed.

Inputs	**Qualifier**	**Fuzzy Degree Associated**
Machining type	Light roughing	1
MRR	Medium	1
Surface finish	Medium	0.74
Cutting speed	High	0.86
Feed	Medium	0.5

Table 5.2 Complete list of input information with a fuzzy degree associated

2. Cutter Selection

(1) Toolbar selection

According to the Figure 4.3 discussed in the Section 4.2.2, Table 5.3 shows the fuzzy degree of the tested toolbar types for low alloyed steel.

Toolbar Type	**Fuzzy Degree Associated**
T-MAX 145	**1**
T-MAX 45	**1**
Modular Style	0.8
Modulmill	0.8
....	....

Table 5.3 Toolbar Type selection

Since machining type of the project is light roughing, and surface requirement is medium (6μm), according to Table 4.5 of section 4.2.2, the fuzzy degree of lead angle of toolbar is shown in Table 5.4.

Lead Angle	**Fuzzy Degree Associated**
0°	0.4
15°	0.98
30°	**1**
45°	**1**
60°	0.4
Round	0.35

Table 5.4 Lead Angle selection

Dimension of tool bar should be around 4/3 of cutting width. Table 5.5 shows the fuzzy degree of proper tool bar dimension in selected toolbar families.

Tool Bar Type	**Selected Tool Bar**	**Diameter (mm)**	**Fuzzy Degree**
T-MAX 145	RA260.22-125R38-12L	**127**	**1**
	RA260.22-100R38-12L	102	0.45
T-MAX 45	RA260.7-125-40	**127**	**1**
	RA260.7-160-40	152	0.57
Modular Style	KMCR5S	**127**	**1**
	KMCR4S	102	0.45
	KMCR6S	152	0.57

Table 5.5 Toolbar Diameter selection

The ratio duty:diameter is:

duty/diameter=MRR*Spec./Diameter

=Doc*Woc*Feed*Spec/Diameter

=(4*100*0.35*800/16387)*777/127

=1063 W

According to Table 4.6 in section 4.2.2, fuzzy degree of pitch type is shown in Table 5.6.

Pitch Type	**Fuzzy Degree Associated**
Croase	**0.98**
Close	0.85
Extra Close	0

Table 5.6 Pitch Type selection

(2) Insert Geometry

According to the procedure discussed in section 4.2.2, Table 5.7 shows the fuzzy degree of the tested elements including the optimum selection.

Tool bar type	**Insert Geometry**	**Fuzzy Degree**
T-MAX 145	SEMR- WL **- WM** - ...	0.44 **0.61** ....
Modulmill	SMKR- WH - 1 -...	0.32 0.25 ....
Modular Style	SEHW- EFR - EFTR - ...	0.29 0.33 ...

Table 5.7 Tool bar and Insert Geometry selection

(3) Insert Size

The effective cutting length is approximately equal to

$$L = 3*4/2*\cos(0) = 6 \text{ mm}$$

According to the L, the fuzzy degree of some of the tested elements are shown in Table 5.8.

Geometry	Insert Size (mm)	Fuzzy degree
SEMR- WL - WM - ...	**12.7**	**0.87**
SMKR- WH - 1 - ...	9.78	0.61
SEHW- EFR - EFTR SEEW -EFR1 -EFTR1	12.7	0.61

Table 5.8 Insert Size selection

(4) Insert Thickness

According to the Table 4.10, the fuzzy degree of the thickness are shown in Table 5.9.

Thickness (mm)	Fuzzy degree from Max-Min Module	Fuzzy degree from User's preference	Average
3.2	0.41	0.675	0.58
4.8	**0.64**	**1**	**0.82**
6.4	0.2	0.61	0.615
...	...	...	...

Table 5.9 Insert Thickness selection

(5) Insert Nose Radius

The "IF-THEN" rules determines that the selection of the nose radius should be based on the inputted feed only. Table 5.10 gives the fuzzy degrees of various radii.

Nose Radii (mm)	Fuzzy Degree from Max-Min	Fuzzy degree from user preference	Average
0.4	0.64	*-	0.64
0.8	**1**	-	**1**
1.12	0.58	-	0.58
1.6	0	-	0
2.4	0	-	0
...	...	...	...

Table 5.10 Insert Nose Radius selection

* Note that user has no preference for the nose radius

(6) Insert Material (Grade)

Before the insert material is selected, the machinability database MACH.DEX is retrieved to provide the workpiece material information. The machinability rating of the workpiece material is 75. Since the MF(machining type $\in$ light roughing) = 1, the process can be classified as a light roughing. -WL geometry has the highest fuzzy degree for light roughing process.

Hence, MF(wear resistance $\in$high) = 0.75

MF(toughness $\in$high) = 0.44

For the selected -WL geometry, GC3015 provides the best combination for requirement of wear resistance and toughness. Therefore, the GC3015 is the most suitable insert material to finish the required job.

Summary of the Selection

In summary, the most suitable cutter for this job is shown in Table 5.11.

Description	Selection	Fuzzy Degree
Toolbar Type	T-MAX 145	1
Lead Angle	45°	1
Pitch Type	Croase	0.98
Dimension (mm)	127	1
Shape	SEMR	-
Relief Angle	-WM characterizes all the required feature information	0.61
Tolerance		
Type		
Size (mm)	12.7	0.87
Thickness (mm)	4.8	0.82
Nose Radius (mm)	0.8	1
Insert material (Grade)	GC3015	0.86

Table 5.11 Summary of the Insert selection for example 1

In cases, the best suitable cutter may not be available in the shops or even exist in the market. The cutter inventory database searches the most suitable and, most importantly, available cutter to finish the required job. For above example, four cutters are selected from the Sandvik and Kennametal products. They are listed in Table 5.12. The selected toolbar and insert in schedule 1 are shown in Figure 5.7, 5.8 and 5.9.

#	1	2	3	4
Tool Bar	T-MAX-145	T-MAX 45	Modulmill	Modular Style

	(RA260.22-125-15)	(RA260.7-125-40)	(RA285.2-125-20)	(KMCR5S)
Supplier	Sandvik	Sandvik	Sandvik	Kennametal
Lead Angle	45°	45°	15°	15°
Pitch Type	Croase	Croase	Croase	Close
Diameter(mm)	127	127	127	127
Tooth Number	6	6	6	8
Insert Code	SEMR1204AZ-WM	LNCX 18 06 AZR-32	SMKR 43E2R-WH	SEHW 43E4R/T
Tool Cost ($/pc)	9.55	8.99	9.5	11.45
Size (mm)	12.7	10	12.7	12.7
Thickness (mm)	4.6	6.4	4.8	4.8
Nose radius (mm)	0.8	1.2	0.8	0.8
Grade	GC3015	GC235	GC-A	KC992M
MF	0.87	0.8	0.76	0.71

Table 5.12 Selected cutters for the Example 1

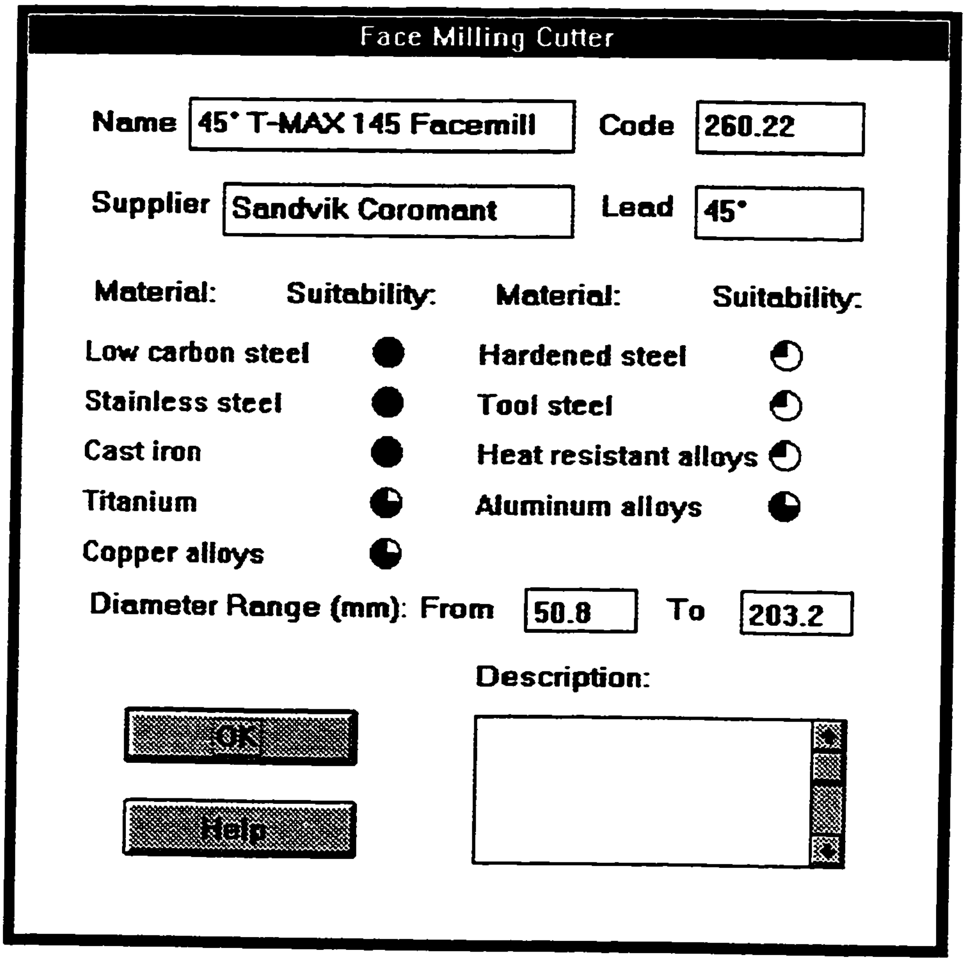


Figure 5.7 Selected Toolbar Type for example 1

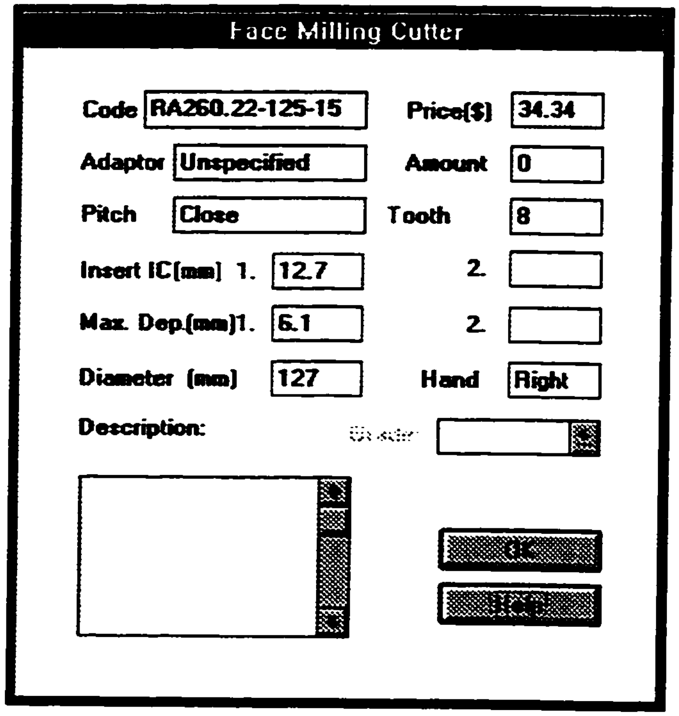


Figure 5.8 Selected Toolbar for example 1

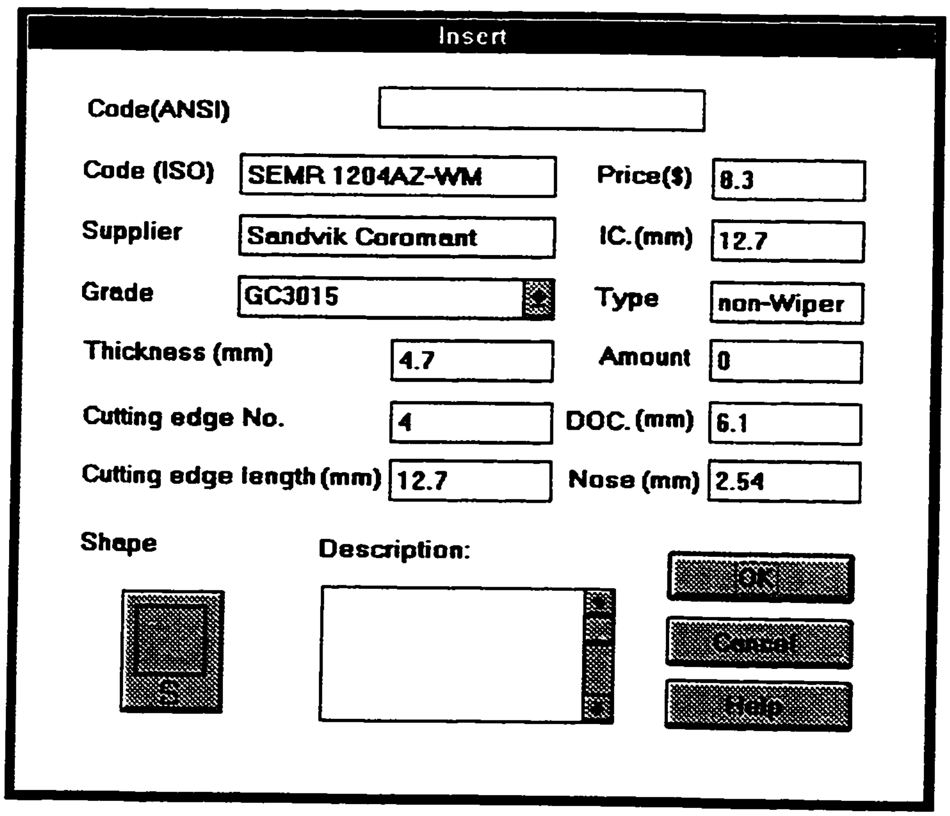


Figure 5.9 Selected Insert and Grade for example 1

3. <u>Design of Cutting Conditions</u>

Before the cutting conditions are designed, the data file MACH.DEX is retrieved to obtain the necessary data of the selected insert and workpiece material for constructing the Taylor equation. In the Table 4.14, the C value for the workpiece material is 946 m/min and the n exponent value for a carbide tool (GC3015) is 0.24. These data are important to construct the mathematical model to design the cutting conditions. With the machine tool specification data retrieved from the MACHINE.DAT, the formulation of the fuzzy optimization model for first selected cutter is shown as follows:

MAX α

ST $$0.5Tm + \frac{Tm}{TL}(6)(9.55) + \frac{Tm}{TL}(2)(0.5) + (0.5)(0.5) \leq 1.3 + (1-\alpha)(0.1)$$

$$-V \leq -31 + (1-\alpha)(2)$$

$$V \leq 2194 + (1-\alpha)(6)$$

$$-f \leq -0.06 + (1-\alpha)(0.025)$$

$$f \leq 1.8 + (1-\alpha)(0.004)$$

$$Fc = \frac{(0.9)Pc}{V} \leq 9300 + (1-\alpha)(50)$$

$$Pc = f(800)(4)(100)(777/25.4^3)/0.9 \leq 138 + (1-\alpha)(12)$$

$$220000\, V^{-1.52} f \leq 6 + (1-\alpha)(2)$$

$$TL = \left(\frac{710}{V}\right)^{\frac{1}{0.25}}$$

$$Tm = \frac{\pi(127)(150)}{(Vf)}$$

$$0 \leq \alpha \leq 1$$

$$V \geq 0$$

$$f \geq 0$$

The model is solved by a Hill-climbing searching algorithm which depends on thousands of iterations, the expert system running on a Pentium takes at least 15-25 seconds to obtain the optimum solution. One of the complete machining plan recommended is summarized in Figure 5.10. The unit cost function curves for both user preference and system recommandation are plotted in Figure 5.11 against the cutting speed.

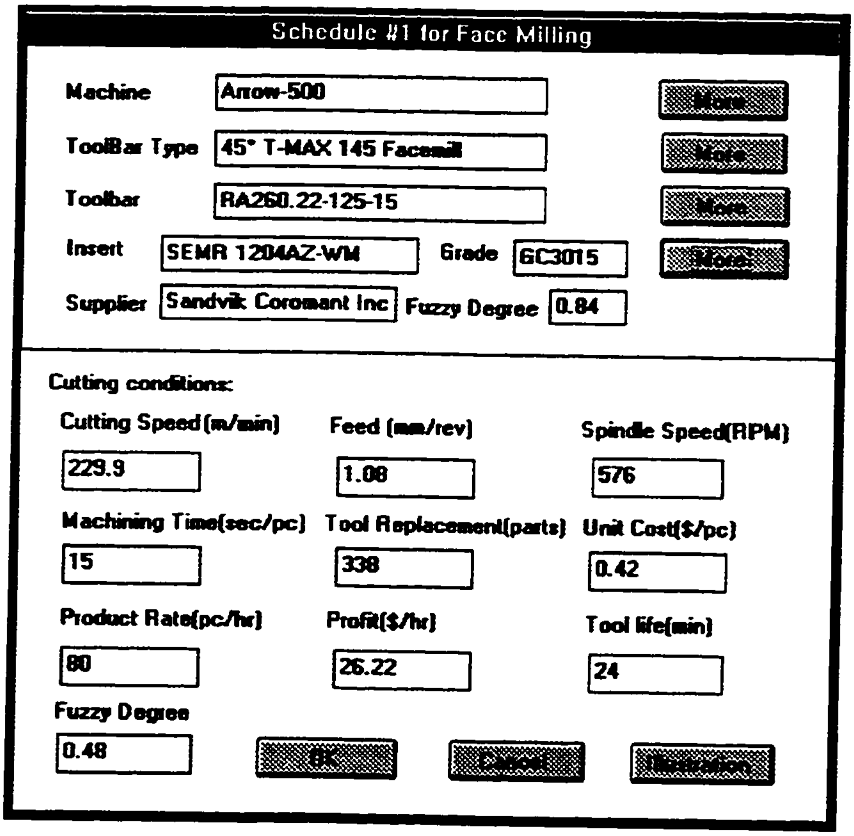


Figure 5.10 Optimal Cutting Condition and all the associated results for example 1

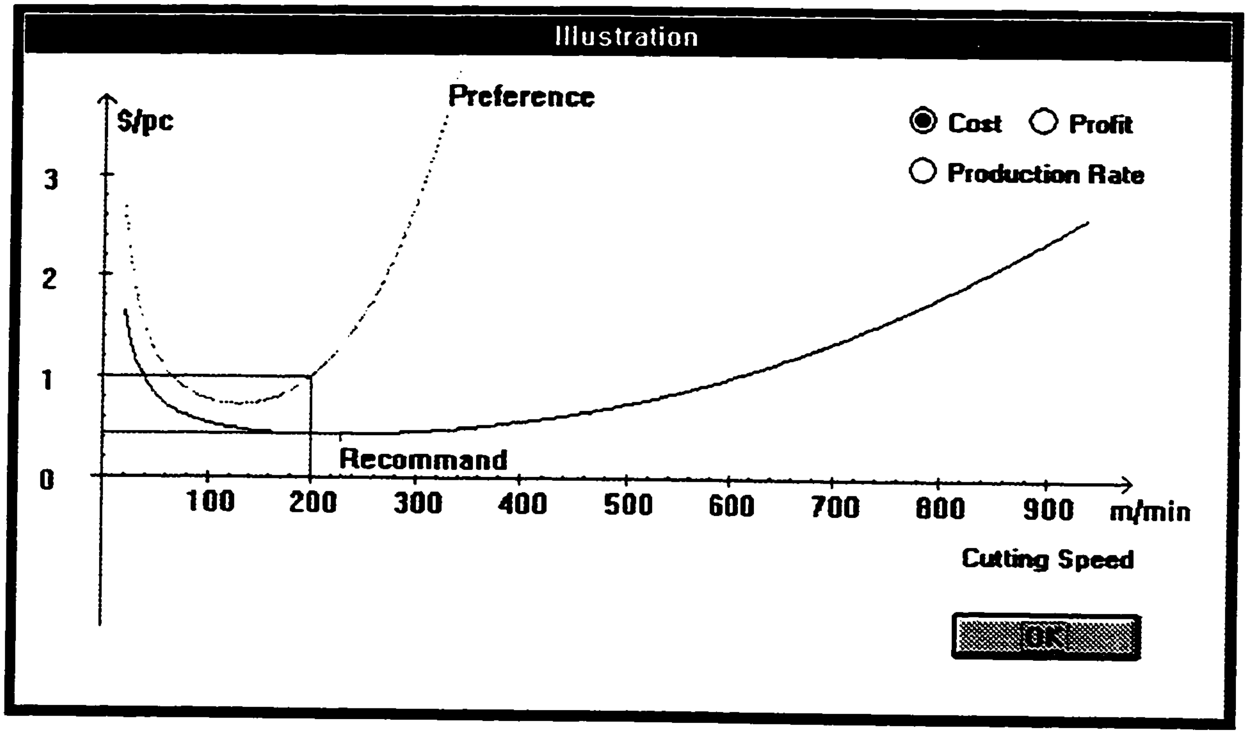


Figure 5.11 System Performance for current machining plan and user preference (optimization under minimum unit cost)

5.2 Example 2

1. Inputs to the system

Example 2 is an end milling operation with the objective of maximizing production profit. The workpiece, cost factor, job requirement and user initial cutting data are listed in Figure 5.12, 5.13, 5.14 respectively. Figure 5.15 and 5.16 shows initial user cutting data and user preferred cutter. Table 5.13 summarizes all fuzzy input information on machining plan.

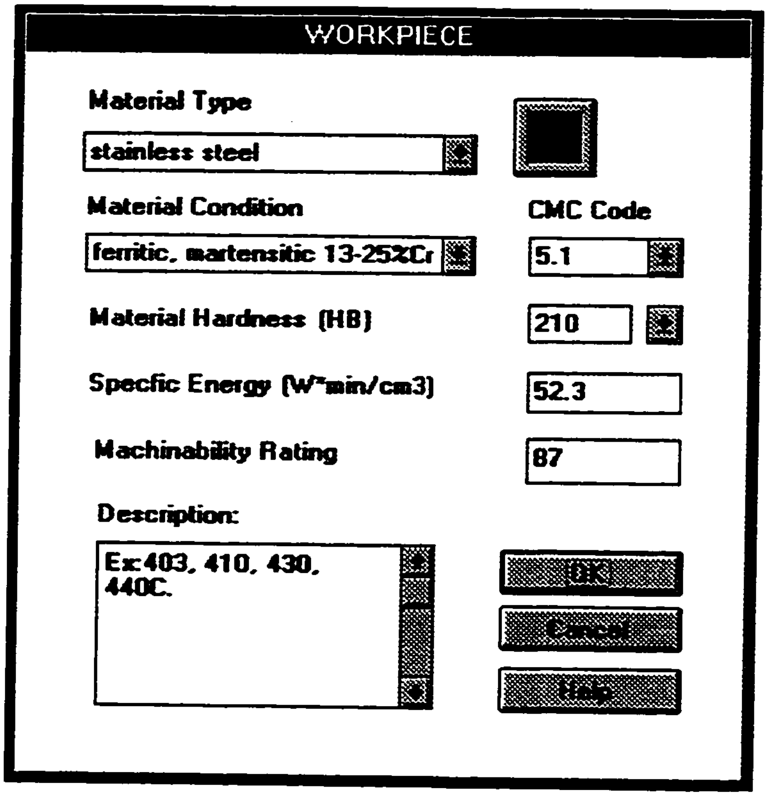


Figure 5.12 Workpiece information for example 2

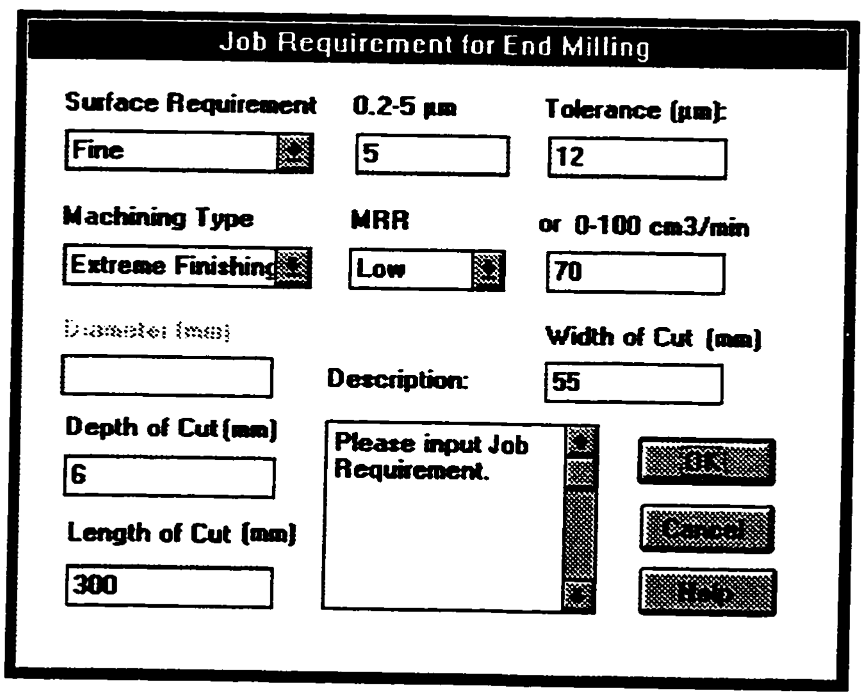


Figure 5.13 Job Requirement for example 2

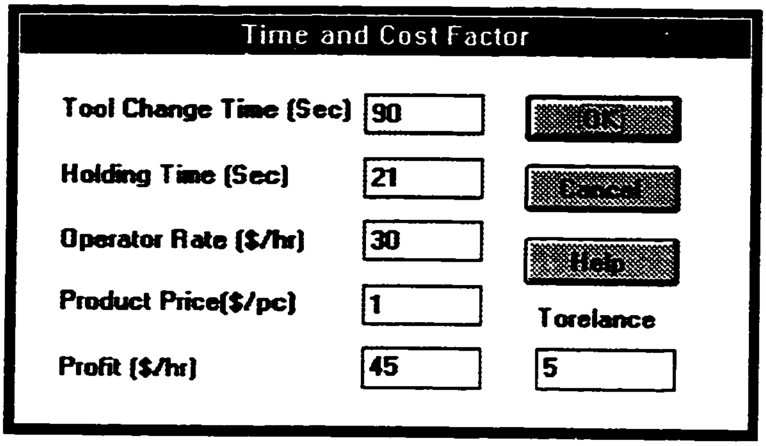


Figure 5.14 Time and Cost information

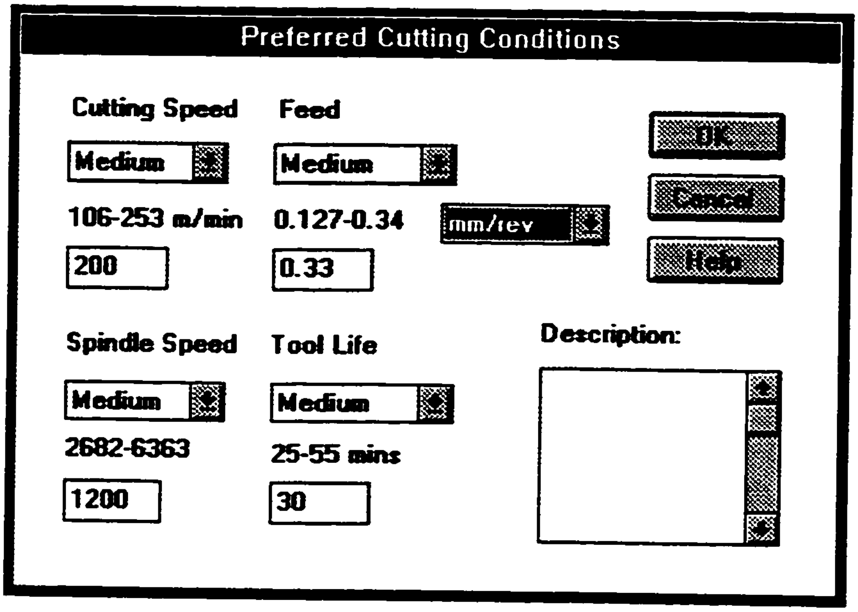


Figure 5.15 User Preferred Cutting Data

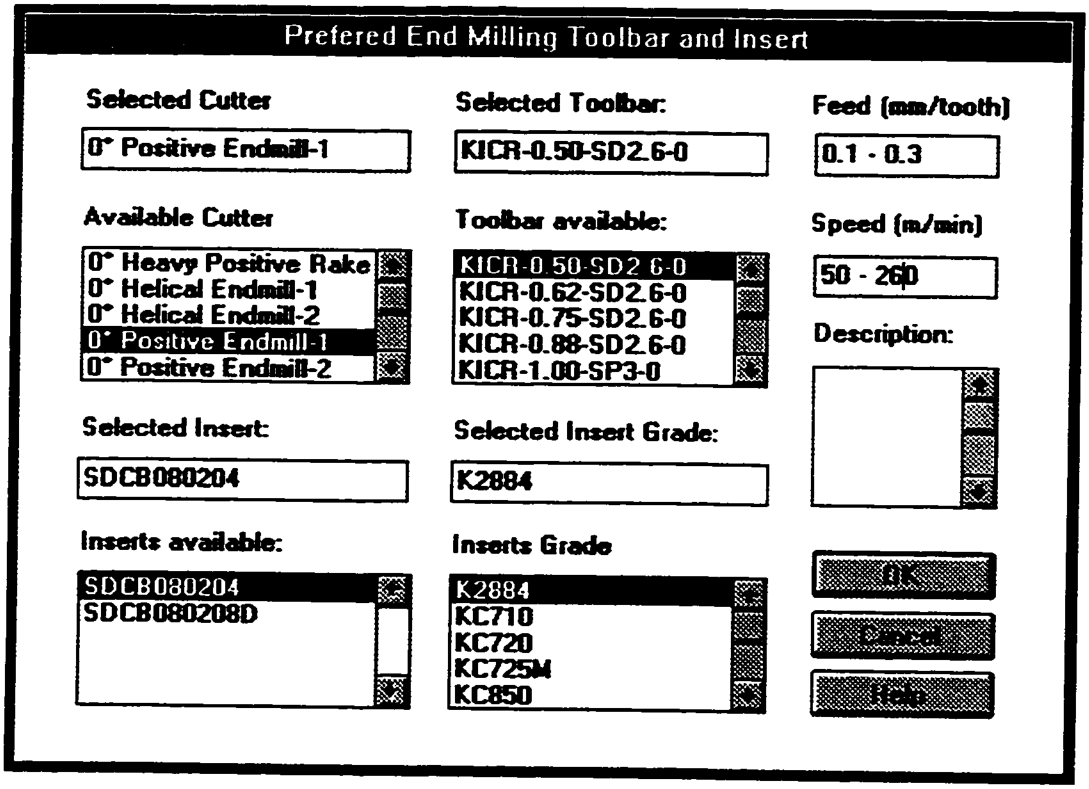


Figure 5.16 User preferred cutter

Inputs	Qualifier	Fuzzy Degree Associated
Machining type	finishing	0.86
MRR	low	.764
Surface finish	fine	0.72
Cutting speed	medium	0.89
Feed	medium	0.75

Table 5.13 A complete input information list for example 2

2. Cutter Selection

The most suitable cutter to finish this operation must carry the following features as summarized in Table 5.14.

Description	Selection	Fuzzy Degree
Toolbar Type	U-MAX Square	1
Dimension (mm)	50	0.8
Lead Angle	0°	1
Shape	86° Rombic	-
Relief Angle	-WL characterizes all the required feature information	0.92
Tolerance		
Type		
Size (mm)	12.7	1
Thickness (mm)	4.8	0.91
Nose Radius (mm)	1.6	0.87
Insert material (Grade)	GC-A	1

Table 5.14 Summary of the Insert selection for example 2

Then, the system searches cutter inventory database to find the most proper available cutters listed as Table 5.15. The selected toolbar and insert in schedule 1 are shown in Figure 5.17, 5.18 and 5.19.

#	1	2	3	4
Tool Bar	U-MAX Square (RA252.44-051R19-15L)	Coronite (Solid) (R215.34-18030 - AA38N)	Positive Rake (KIPR-150 AN1623-0-5)	Positive Endmill (KIPR-0.75-SD2.6-30)
Supplier	Sandvik	Sandvik	Kennametal	Kennametal
Diameter(mm)	50	18	38.1	19
Lead Angle	0°	0°	0°	30°
Tooth Number	12	4	6	2
Insert Code	RA252.44-15T316M-WL	-	ANGT-1.62.3P3R	SDEB-2.61.52
Tool Cost ($/pc)	7.78	50	10.5	12.19
Size (mm)	12.7	-	6.8	8.31
Thickness (mm)	4.8	-	3.6	2.4
Nose radius	1.6	0.8	0.5	0.8

Grade	GC-A	NI45	KC992M	KC725M
MF	0.95	0.87	0.79	0.72

Table 5.15 Selected cutters for the Example 2

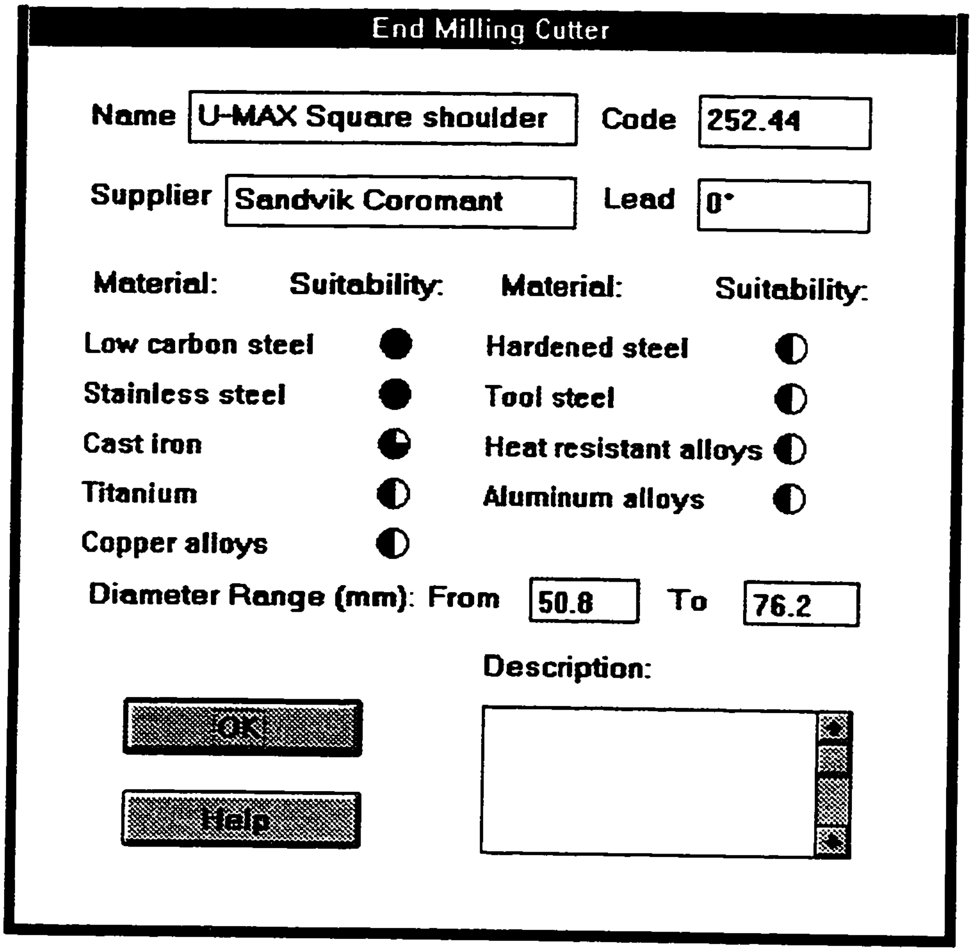


Figure 5.17 Selected Toolbar Type for example 2

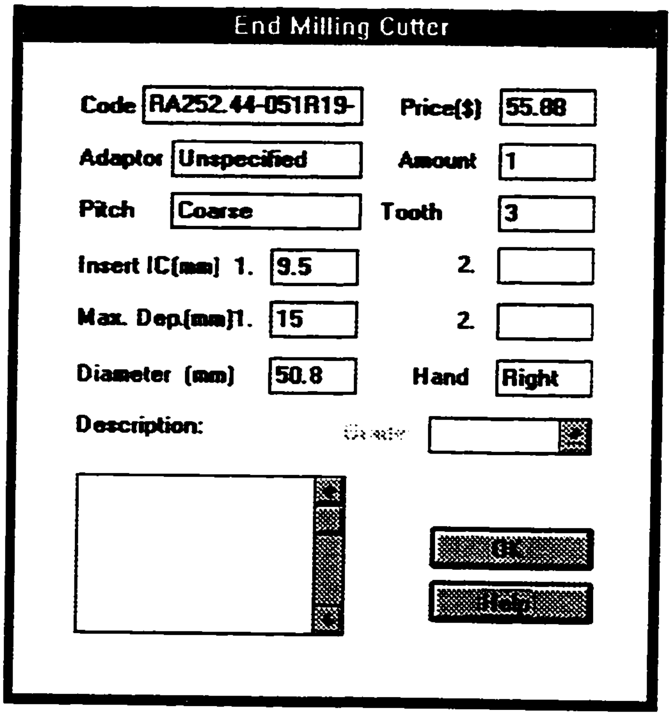


Figure 5.18 Selected Toolbar for example 2

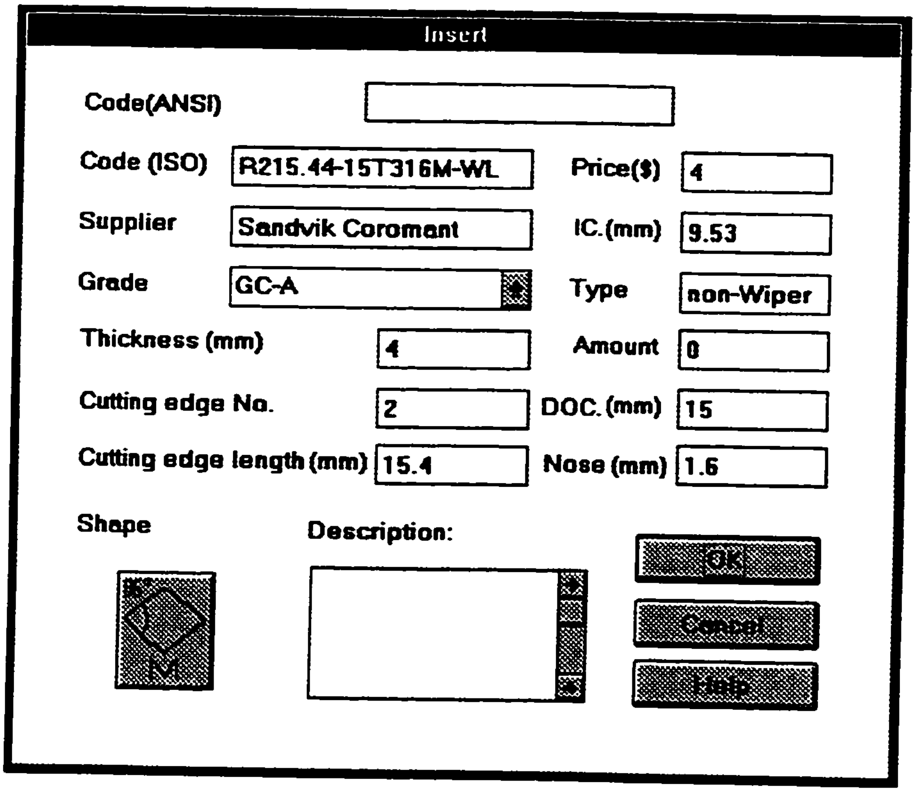


Figure 5.19 Selected Insert and Grade for example 2

3. <u>Design of Cutting Conditions</u>

The optimum cutting conditions and cost factors are summarized in Figure 5.20. The performance (profit) for both user preference and system recommandation are plotted in Figure 5.21 against the cutting speed.

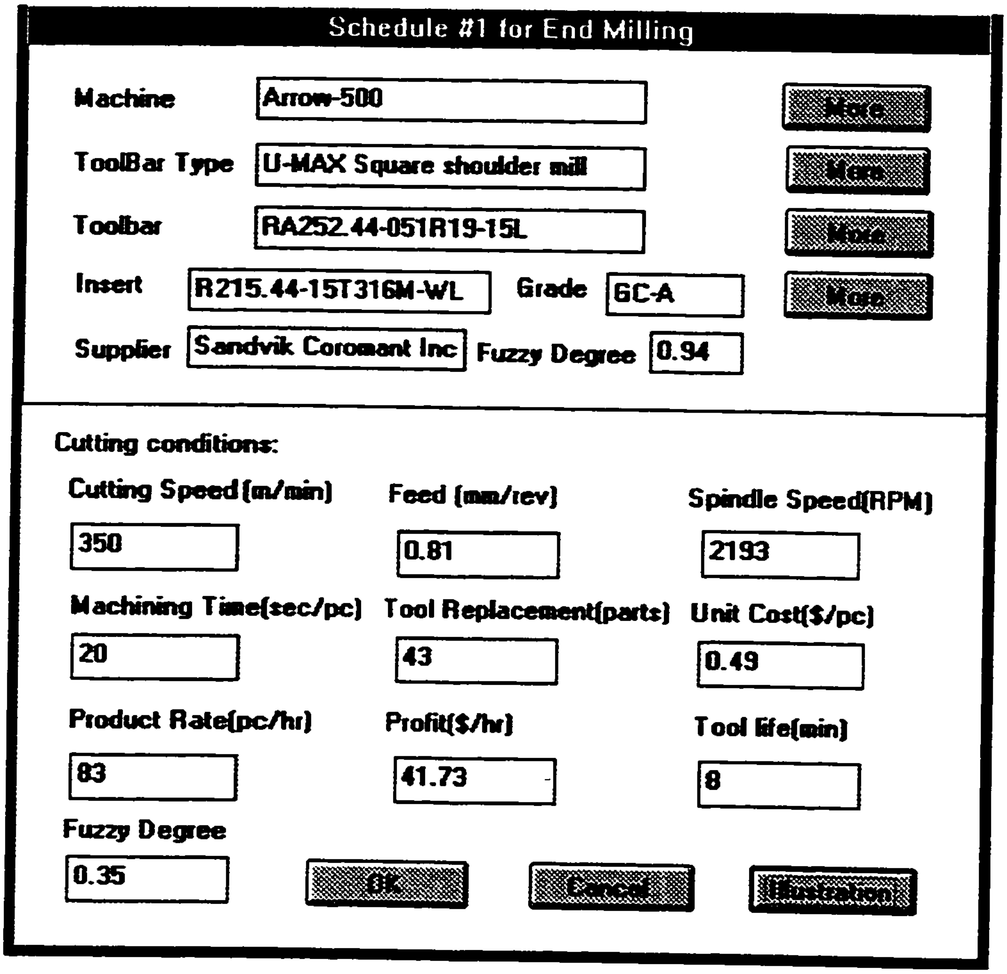


Figure 5.20 Optimal cutting condition and all the associated results for example 2

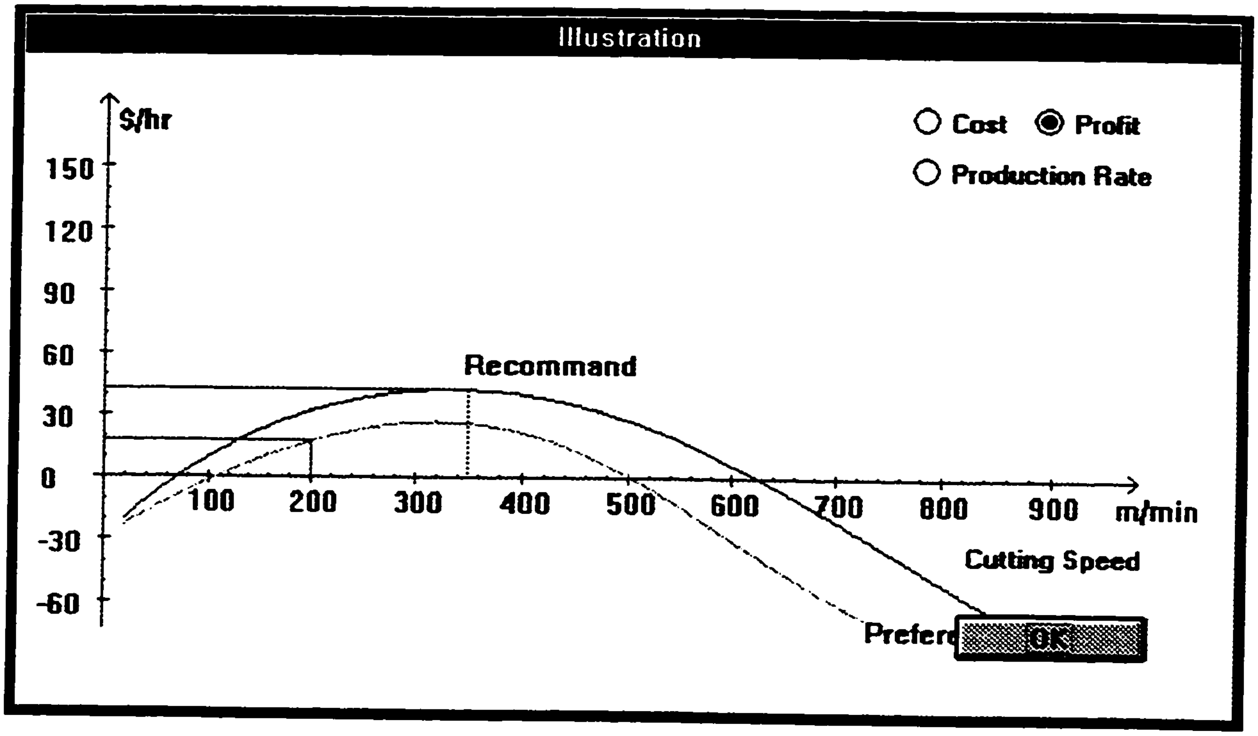


Figure 5.21 System Performance for current machining plan and user preference (optimization under maximum profit)

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

1. Conclusion

In this thesis, a prototype fuzzy expert system for the design of rotating operations is presented. The system is designed based on the object oriented modelling (OOM) approach. It has two major functions: 1)cutter and grade system and 2)cutting condition design. Based on the discussion in previous chapters, following conclusions can be drawn.

In the cutter selection, the fuzzy set theory is used as a tool to represent the fuzziness associated with all input information and input-output relationships. Based on incomplete and/or imprecise input information, the Fuzzy Decision Mechanism is able to automatically select the cutters (including toolbar and insert) and insert grades according to the pre-complied fuzzy decision rules. The cutter and machinability databases are also developed to store the necessary information.

For the cutting condition design, a fuzzy non-linear programming approach is employed. The objective of the model is to provide user an optimal cutting speed and feed rate while considering imprecise or fuzzy constraints. The system also estimates unit production cost, tooling cost and production time which are very important for workshop to (a) set up unit price (b) estimate profit and (c) predict system productivity.

The application of the OOM approach simplifies both the design and the implementation of the system. It encapsulates both the structural elements of the data and the behavioral aspects of the software. As a result, great cost savings are achieved in several significant ways, including model design and redesign time, implementation and maintenance effort, breakthrough of database limitation from each individual supplier. Furthermore, it also establishes a platform for future metal-cutting application development.

A windows application software package is developed to implement the proposed methods for selecting cutters and designing cutting conditions. The system is user-friendly and graphical. Compared to other similar systems, it has two distinct features: (1) it does not require precise and complete design information, and (2) it support industrial products used in shop floor. These make the system practical and effective as demonstrated by the two examples.

2. Future Research

The system is only a prototype at this developmental stage. Further works and studies are aimed at:

1. Only single pass processes are considered for most rotating operations in the current system. Multi-pass processes should be attempted to be solved using the fuzzy set theory in the future research.

2. There are other factors that are still fuzzy which the present system has not considered. For example, the operator rate is actually fuzzy. The current model can be reconstructed to accommodate these factors

3. In the proposed system, a large number of fuzzy membership function are defined. These functions are so critical to the cutter selection, yet, they are empirical functions which may not be precise. Although a learning algorithm has been proposed to fine-tune these functions to accommodate their own preferences and to improve the selection capability, it is not fulfilled yet. Because it is hard to figure out which rule is bottleneck of the system performance, and how much the rule and its corresponding rules should be adjusted.

4. All system outputs are based on theoretical calculation, experiments are needed to verify the correctness of system, especially when implementing the learning algorithm.

5. Currently, the system do not offer transparent integration with external databases. This is due to the lack of unified implementation in modern relational and object-oriented database management systems. In the future, the expert system could be redeveloped, for example, by Java in network environment, on-line survey, selection, book will become reality.

REFERENCES

Agapious, J.S., 1992, "The Optimization of Machining Operations Based on a Combine Criterion, Part 1: the use of Combine Objectives in Single Pass Operation", *Transactions of ASME, Journal of Engineering for Industry*. Vol.114, pp.500-506

Agapious, J.S., 1992, "Optimization of Multistage machining Systems Part 1: Mathematical solution", *Transactions of ASME, Journal of Engineering for Industry*, Vol.114, pp.524-531

Alting, L. and Zhang, H. 1989, "Computer Aided Process Planning: the state-of-art survey." *International Journal of Production Research*. Vol.27, No.4, pp.355-383

Amargeo, E.T. and Brown, R., 1965, "*The Machining of Metal*", Prentice Hall, New Jersey

Billatos, S.B. and Kendall, L.A., 1990, "A Replacement Model for Multi-Tool Transfer Lines", *Transactions of ASME, Journal of Engineering for Industry*, Vol.112, pp.253-259

Billo, R.E. and Bidanda, B., 1995, "Representing group technology classification and coding techniques with object oriented modeling principles", *IIE Transaction*, Vol. 27, pp. 542-554

Bley, H. and Stadelmeyer, V. 1993, *"25th CIRP International Seminar on Manufacturing Systems"*, pp. 335-345

Booch,G., 1994. *"Object-oriented analysis and design with applications"*, The Benjamin/Cummings Publishing Company, California

Boothroyd, G. and Rusek, P., 1976, "Maximum Rate of Profit Criteria in Machining", *Transactions of ASME, Journal of Engineering for Industry*, Feb, pp.217-220

Chen, S.J., Hinduja, S. and Barrow, S., 1989, " Automatic Tool Selection for Rough Turning Operation ", *International Journal of Machine Tools and Manufacture*. Vol. 29, No.4, pp.535-553

Chen, Y.D., Hui, A. and Du, R., 1995, "A Fuzzy Logic System for Cutter Selection and Cutting Condition Design," *International Journal of Machine Tools & Manufacture*. vol. 35, No.12 pp.1605-1621

Chang, T.C., 1990, "*Expert Process Planning for Manufacturing*", Addison-Wesley Inc, New York

Colding, B., 1991, "*Computer Optimization Manufacturing Planning Menu*", Colding International Corp, Ann Arbor

Devries, W., 1991, "*Analysis of material removal processes*", Springer Verlag, New York, pp.110-123

Dhage, S. and Usher, J., 1993, "Computer Aided Tool Selection for turning and boring", *Computers and Industrial Engineering*. Vol.25, No.1-4, pp.207-210

Domazet, D., 1990, "The automatic tool selection with the production rules matrix method", *Annals of the CIRP*, Vol.39, No.1, pp.497-500

Drozda, T.J. and Wick, C., 1983, " Tool and Manufacturing Engineers Handbook," 4th ed., Vol.1, *SME*

Du, R.X. and Hui, A., 1993, "Optimal Design of Cutting conditions and Tool Replacement time for transfer machining systems", *Proceeding of IIE in California*, pp.345-367.

Duffuaa, S.O. Shuaib, A.M. and Alam, M., 1993, "Evaluation of Optimization method for machining Economics model". *Computers and Industrial Engineering.*, Vol. 20, No.2, pp.227-257

Elsayed, E.A. and Turley, R.E., 1980, "Reliability Analysis of Production system with buffer storage", *International Journal of Production Research*, Vol.18, pp.737-645

Ermer, D.S. and Kromodihardjo, S., 1981, "Optimization of Multipass turning with constraints", *Transactions of ASME, Journal of Engineering for Industry*, Vol.103, pp.462-468

Fenton, R. and Joseph, N., 1979, " The Effect of Statistic nature of Tool Life on the Economic of Machining", *International Journal of Machine Tool Design*, Vol.19, pp.43-50

Gilbert, W. 1950, " *Economic of machining*", Machining theory and Practice, ASME pp.465-485.

Hanna, N. 1996, "Modelling products quality from CNC machining center using fuzzy Petri nets with neural networks", *IEEE Emerging Technologies & Factory Automation*, vol.. 2, pp. 440-445.

Hati, S.K. and Rao, S.S., 1976, "Determination of optimum machining Conditions Deterministic and Probabilistic Approach", *Transactions of ASME, Journal of Engineering for Industry*, Feb, pp.354-359

Hitomi, K., Nakanura, N. and Inous, S., 1979, "Reliability Analysis of Cutting Tools", *Transactions of ASME, Journal of Engineering for Industry*, Vol.101, pp.185-189

Hitomi, K., 1977, "Optimization of multistage production system with variable production time and cost", *International Journal of Production Research*, Vol.15, No.6, pp.587-597

Hitomi, K. and Okushima, S., 1964, "A study of Economic Machining: an analysis of the maximum - profit cutting speed", *International Journal of Production Research*, Vol. 3, No.1, pp.73-79

Ishibuchi, H. and Fujioka, R., 1993, "Neural Networks that learn from Fuzzy If-Then rules", *IEEE Transactions on Fuzzy Systems*, Vol.1, No.2, pp.85-92

Iwata, K., Murotsu, Y. and Oba, Y., 1977, "Optimization of Cutting Conditions for Multipass Operations Considering Probabilistic nature in Machining Process", *Transactions of ASME, Journal of Engineering for Industry*, Ser B. pp.210-217

Iwata, K., Murotsu, Y., Iwatsubo, T., and Fuji, S., 1972, "A Probabilistic Approach to the Determination of the Optimum cutting conditions", *Transactions of ASME, Journal of Engineering for Industry*, Vol.94, pp.1099-1107

Kaufmann, A., 1975, *Introduction to the Theory of Fuzzy Sets*, Academic Press, New York.

Kalpakjian, S., 1995, *Manufacturing Engineering and Technology*, Addison-Wesley Publishing Inc, New York

Kimbler, D., Wysk, R.A. and Davis, R., 1982, "Alterative Approach to the machining parameter optimization problem", *Computers and Industrial Engineering* Vol.21, pp.195-262

Lambert, B. and Walvekar, A., 1978, "Optimization of Multi-pass machining operation", *International Journal of Production Research*, Vol.16, pp.259-268

Lee, B.H., Lim, B.S. and Nee, Y.C., 1991 "IKOOPS: an intelligent knowledge based object-oriented process planning system for manufacturing of progressove die," *Expert Systems*. Vol.8, No.1, pp.19-32

Lai, Y.J. and Hwang, C.L., 1993 ," *Fuzzy Mathematical Programming*", Springer Verlag, New York, pp. 102-104

Maropoulous, P., 1990, "Automatic tool selection for finish turning", *Proceeding of the Institute of Mechanical Engineering*, Vol.204, pp.43-51

Maropoulous, P., 1992, "Cutting tool selection: an intelligent methodology and its interface with technical and planning function", *Proceeding of Institute of Mechanical Engineering* Vol.206, pp.49-60

McDermidm, J., and Rook, P., 1993 "Software Development Process Models", *Software Engineer's Reference Book*, CRC Press

Narang, R.V., and Fischer, G.W., 1993, "Development of a framework to automate process planning functions and to determine machining parameter", *International Journal of Production Research*, Vol.31, No.8, pp.1921-1942

Pandit, S.M. and Shiekh, A.K., 1980, "Reliability and Optimal Replacement via Coefficient of Variation", *Journal of Mechanical Design*, Vol.102, pp.761-768

Peterson, G.E., 1990, " *Tutorial: Object-Oriented Computing*", IEEE series, Computer Society Press

Petropoulous, P., 1973, "Optimal Selection of machining rate variables by geometric programming", *International Journal of Production Research*, Vol.11, pp.305-314

Rao, S.S and Hati, S., 1978, "Computerized Selection of Optimal Machining Conditions for a Job Requiring Multiple Operations". *Transactions of ASME, Journal of Engineering for Industry*, Vol.100, pp.356-362

Rumbaugh, J., 1991, "*Object oriented modeling and design*", Prentice-Hall Inc., New Jersey

Sandvik Coromant Product Catalogues, 1995, "*Rotating Tools and Inserts*" , Sweden

Sandvik Coromant Product Catalogues, 1994, "*Turning Tools and Inserts*" , Sweden

Santochi, M. and Dini, G., 1996, "Use of neural networks in automated selection of technological parameters of cutting tools", *Computer Intergrated Manufacturing systems*, vol. 9, pp. 137-148.

Schey, J.S., 1988, " *Introduction to Manufacturing Process 2nd Edition*", McGraw series, New York.

Sekulic, S., 1982, "Cost of Cutting tools and total machining cost as a function of the cutting tool reliability in automatic flow lines", *International Journal of Production Research,* Vol.20, No.2, pp.187- 200

Sheikh, A.K., Kendall, L.A. and Pandit, S., 1980, "Probabilistic Optimization of Multitool Machining Operation", *Transactions of ASME, Journal of Engineering for Industry,* Vol.102, pp.239-246

Trappey, J. and Liu, L., 1988, "Fuzzy non-linear programming: theory and application" *International Journal of Production Research,* Vol.26, No.5, pp.975-985

Walvekar, A.G. and Lambert B.K., 1978, " The Optimization of Multi-pass Machining Operations", *International Journal of Production Research,* Vol.16, No.5, pp.259-268

Wu, S.M. and Ermer, D.S., 1966, "Maximum Profit as the Criterion in the Determination of the Optimum Cutting Conditions". *Transactions of ASME, Journal of Engineering for Industry,* Vol.93, pp.35-44

Yellowley, I. and Gunn E., 1989, "The Optimal Subdivision of Cut in Multi-pass Machining Operation", *International Journal of Production Research,* Vol.27, No.9, pp.1979-1986

Yeo, S., Wong, Y.S., and Rahman, M. 1991, "Integrated knowledge based machining system for rotational parts", *International Journal of Production Research,* Vol.29, No.7, pp.1325-1337

Zhou, C. and Wysk, R.A. 1992, "Tool Status Recording And its use in Probabilistic Optimization", *Transactions of ASME, Journal of Engineering for Industry,* Vol.114, No.4, pp.494-499

Zhou, C. and Wysk, R.A, 1992, "An Integrated System for selecting optimum cutting speed and tool replacement time", *International Journal of Machine Tools and Manufacture,* Vol.32, No.5, pp.695-707

Appendix A The relationship between tool bar and workpiece material

Toolbar \ Material	Low Carbon Steel	Hardened Steel	Stainless Steel	Tool Steel	Cast Iron	Heat Resistant Alloys	Titanium	Aluminum Alloys	Copper Alloys
T-MAX 145	●	◔	●	◔	●	◔	◕	◕	◕
Modulmill 145	●	◔	●	◔	●	◔	◕	◕	◕
Coromant T-Line	●	◕	○	○	●	○	◕	○	○
T-MAX 45	●	◕	◕	◕	◐	◔	◐	○	○
0° U-MAX	●	◔	●	◔	◕	◐	◐	◐	◐
CoroMill 290	●	◔	●	◔	●	◔	◔	◐	◐
Round U-MAX	●	●	●	●	●	◕	◐	◐	◔
0° Modulmill	●	◔	◐	◕	◕	◔	○	◔	◐
15° Modulmill	◔	●	○	◐	◕	○	○	○	○
T-MAX AL	○	○	○	○	○	○	○	●	◔
AUTO	◔	○	○	○	●	○	○	◔	○
15° Positive	◐	○	◐	○	◐	○	○	◔	○

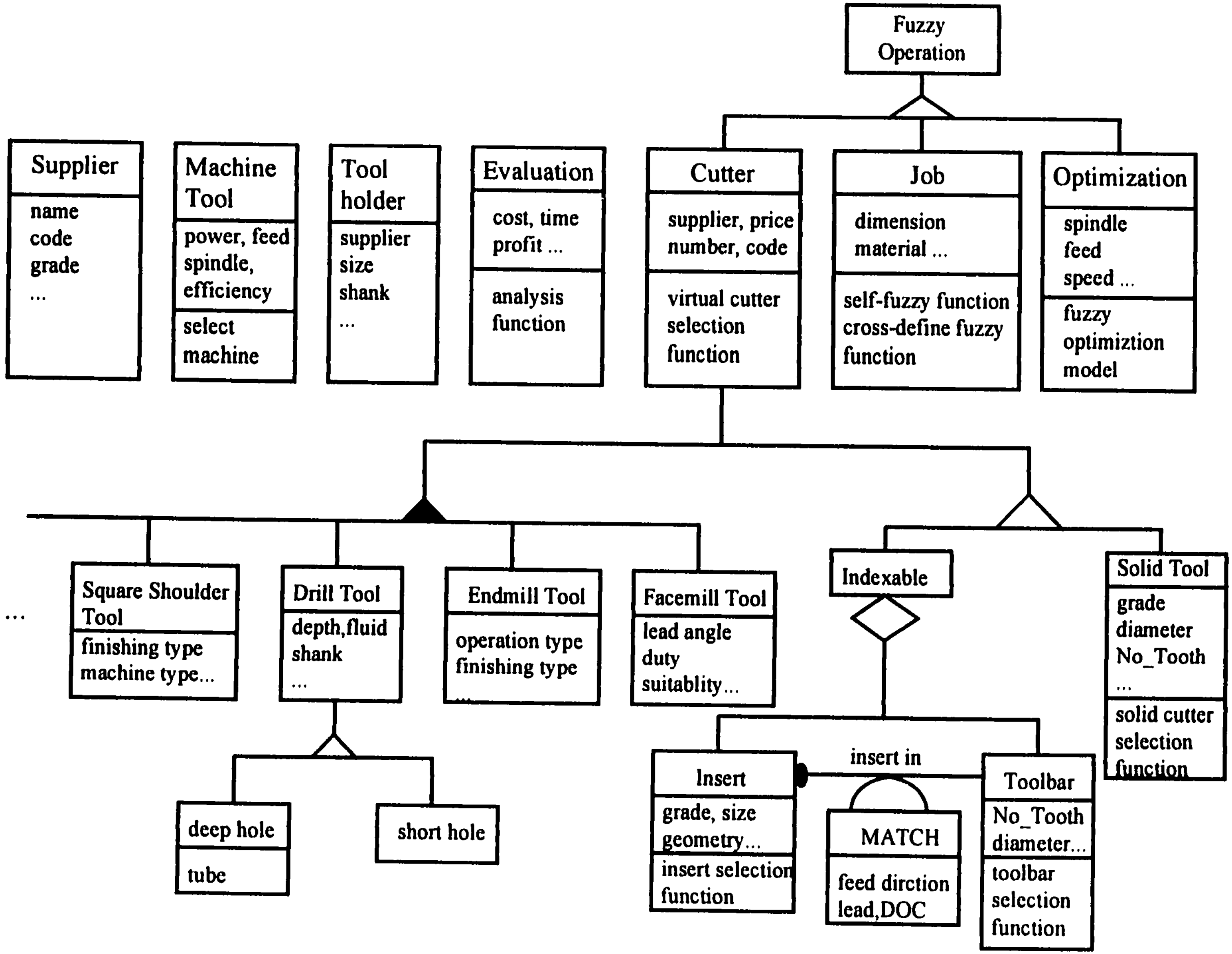
Toolbar \ Material	Low Carbon Steel	Hardened Steel	Stainless Steel	Tool Steel	Cast Iron	Heat Resistant Alloys	Titanium	Aluminum Alloys	Copper Alloys
0° High Shear	◔	○	●	◕	◐	○	●	●	●
45° High Shear	●	●	●	●	●	◐	●	◐	●
0° Shear	●	◐	●	◐	◕	◐	◐	○	◐
20° Shear	●	●	●	●	●	◔	◐	○	◐
0° Positive	◐	○	◐	○	◐	○	○	◔	○
0° Double Negative	◐	◐	◐	◔	◐	◕	○	○	○
0° High Positive	○	○	◔	◔	○	○	○	●	◐
0° Double Positive	◐	○	◐	○	◐	◐	◐	◐	●
45° Positive	◐	○	◐	○	◐	○	○	◔	○

Best Solution ← → Less Suitable

● ◕ ◐ ◔ ○

[From Sandvik Rotating Tool Handbook (1995) and Kennametal Milling Tool Handbook (1995)]

Appendix B SAM class hierarchy and function prototype



```
class FUZZY
{
        FUZZY();
        ~FUZZY();
        float S(float a, float b, float x);
protected:
        float fuzzymembership(float a, float b, float c, float d, float x);
};

class CUTTER: public FUZZY
{
        CUTTER();
        ~CUTTER();
        char comment[70], supplier[25], purpose[10],name[35], code[10], solid;
        float suit[9],dialow,diahigh,dialowmetr,diahighmetr,price,diameter,radius,depth;
public:
        friend class MachiningPlan;
        friend class JOB;
        virtual void LoadData();
        virtual float LeadAngleSelection(int lead,char machiningtype[]);
        float cuttertypeselection(WorkPiece piece,char code[]);
        virtual float DiameterSelection(int width, float diameter);
        virtual float PitchTypeSelection(float diameter,float feed,int width);
        virtual float ToothNumberSelection();
        float select_noseradius();
        virtual void writeplan();
};

class IndexTool
{
        IndexTool();
        ~IndexTool();
        class INSERT;
        class ToolBar ;
public:
};

class SolidTool: public CUTTER
{
        SolidTool();
        ~SolidTool();
        char grade[15];
public:
        void writeplan();
        float select_DOC(float depth);
```

```
        float select_grade();
};

class FaceMill: public CUTTER
{
        FaceMill();
        ~FaceMill();
        int lead;
        char pitch[15];
public:
        float DiameterSelection(int width, float diameter);
        float LeadAngleSelection(int lead,char machiningtype[]);
        float PitchTypeSelection(float diameter,float feed,int width);
        float ToothNumberSelection(float diameter,char pitch[]);
};


class SquareMill: public CUTTER
{
        SquareMill();
        ~SquareMill();
        char pitch[15];
public:
        float DiameterSelection(int width, float diameter);
        float PitchTypeSelection(float diameter,float feed,int width);
        float ToothNumberSelection(float diameter,char pitch[]);
};

class EndMill: public CUTTER
{
        EndMill();
        ~EndMill();
public:
        float DiameterSelection(int width, float diameter);
        float ToothNumberSelection(float diameter);
};

class ToolBar: public CUTTER
{
        ToolBar();
        ~ToolBar();
        friend class MATCH;
        int lock,toothnum,num;
public:
        Ic *ichead,*icp,*icq;
```

```
        Grade *bar2head,*bar2p,*bar2q;
        void LoadData();
        float writeplan();
        float select_DOC(float depth);
};

class INSERT: public CUTTER
{
        INSERT();
        ~INSERT();
        friend class Data;
        char codeansi[30],standard,codeiso[30];
        float length,thick,thickmm,radius,size;
        int wiper,shape,num;
public:
        BARCODE *head1,*p1,*q1;
        Grade *head2,*p2,*q2;
        INSERT* next;
        void LoadData();
        float select_geometry(char machiningtype[],WorkPiece piece);
        float select_wiper(char surface[]);
        float select_size(float surface,float feed,char machiningtype[]);
        float select_thickness(float feed,float depth);
        float select_DOC(float depth);
        float writeplan();
};


class Grade
{
        friend class MATCH;
        char ID,grade[13];
        int numavail;
        float toughness();
        float wear_resistance();
public:
        Grade();
        ~Grade();
        void LoadData();
        float select_grade(GRADEindex grade,int toollife,WorkPiece piece);
        Grade* next;
};
```

```
class JOB: public FUZZY
{
        JOB();
        ~JOB();
        friend class Optimization;
        friend class Evaluation;
        char surface[8],Mrr[8],mach[20],fdunit,piecetype,condition[48];
        int mrr,hard,premachining,energy,MR,life,C;
        float workpiececode,lengthmm,widthmm,depthmm,feed,speed,
                surfacemm,surfaceto;
        int rate,rateto,change,hold,operate;
        float cost,costto,price;
public:
        float machiningtype_fuzzy(char machiningtype[],char surface[]);
        float surface_fuzzy(char surface[],char machiningtype[]);
        float speed_fuzzy(float speed):
        float feed_fuzzy(float feed);
        float MRR_fuzzy(char surface[],char machiningtype[]);
        void LoadData();
};

class Adapt
{
        Adapt();
        ~Adapt();
        friend class MATCH;
        char purpose[9],name[25],company[25],comment[40],shank;
        int vari;
public:
        void LoadDate();
        Adapt *next;
};

class Company
{
        Company();
        ~Company();
        friend class MATCH;
        char company[30],code[15],stand;
public:
        void LoadData();
        Grade *gradehead,gp,gq;
        Company *next;
};
```

```
class WorkPiece
{
        WorkPiece();
        ~WorkPiece();
        char Material[34],Condition[48],description[45],type;
        int hardness[3],desnum,energy,MR,C;
        float Code;
        friend class CUTTER;
public:
        void LoadData();
};

class Machine
{
        Machine();
        ~Machine();
        char type,name[20],supplier[50],code[5],purpose[10];
        int eff; //num in head
        unsigned int maxfor,maxspind,minspind,maxpow,maxfd,minfd;
        int maxforto,maxspindto,minspindto,maxpowto, maxfdto,minfdto;
        long price;
public:
        void LoadData();
        Machine *next;
} ;

class Ic
{
        float ic,maxdep;
public:
        Ic();
        ~Ic();
        Ic *next;
};


class BARCODE
{
        char barcode[10];
        int num;
public:
        BARCODE();
        ~BARCODE();
        BARCODE* next;
};
```

```
class GRADEindex
{
        class TOUGH
        {
                char machining[8];
                int p1,p2,m1,m2,k1,k2,PC,MC,KC;
        public:
                TOUGH();
                ~TOUGH();
                TOUGH* next;
        };
        friend class MATCH;
        friend class Grade;
        char ID,grade[13],type,company,comment[25];
public:
        GRADEindex();
        ~GRADEindex();
        TOUGH *thead,*tp,*tq;
        GRADEindex* next;
};


class CUTTING
{
        char ID,grade[12];
        float feedlow,feedhigh;
        int  speedhigh,
         speedlow;
public:
        CUTTINGDATA();
        ~CUTTINGDATA();
        CUTTINGDATA  *next;
};

class MILLINGDATA
{
        class CMC
        {
                char cmccode[6];
                float num;
        public:
                CMC();
                ~CMC();
                CMC* next;
        };
```

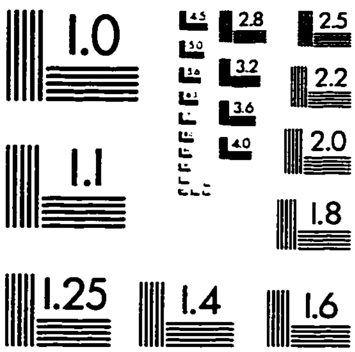
```
        friend class MATCH;
        char solid,geo[8],fdunit,comment[30];
public:
        MILLINGDATA();
        ~MILLINGDATA();
        BARCODE *barhead,*barp,*barq;
        CMC *cmchead,*cmcp,*cmcq;
        CUTTINGDATA *cuthead,*cutp,*cutq;
        MILLINGDATA *next;
};


class MACHCOD
{
        char machinecode[5];
public:
        MACHCOD();
        ~MACHCOD();
        MACHCOD *next;
};

class MachiningPlan
{
        MachiningPlan();
        ~MachiningPlan();
        friend class Optimization;
        friend class Evzluation;
        char bar1name[35],//bar1code[10],
                bar2code[25],codeansi[30],codeiso[30],supplier[25],solid,unit;
        char grade[15],machine[20],gradetype;
        int spind,object,C,toothnum,turn,life;
        float time,replace,rate;
        float radmm, N, toolcost, profit, bar2diain, speedin, feedin, fuzzy, cuttingfuzzy;
        double  cost;
public:
        void LoadResult();
        void saveResult();
};


class MATCH
{
        MTCH();
        ~MATCH();
        float match_IC();
```

```
        float match_DOC();
        float match_feed();
        float match_direction();
public:
        void Match_toolbar_hold(Adapt,ToolBar);
        void Match_toolbar_insert(ToolBar,INSERT);
};

class Optimization: public FUZZY
{
        Optimization();
        ~Optimization();
        friend class Evaluation;
        int object;
public:
        float cost_optimization(JOB job);
        float rate_optimization(JOB job);
        float profit_optimization(JOB job);
        void writeplan();
};

class Evaluation
{
        Evaluation();
        ~Evaluation();
public:
        float evaluate_cost(JOB job, MachiningPlan&plan);
        float evaluate_profit(JOB job, MachiningPlan&plan);
        float evaluate_rate(JOB job, MachiningPlan&plan);
        float evaluate_replacement(JOB job, MachiningPlan&plan);
        void writeplan();
};
```

VITA AUCTORIS

Xin Yuan was born in ShangHai, China on 30th of May, 1971. He attended Dalian University of Technology from 1989 to 1993, graduating with a B.Eng in Computer Engineering. Since September 1994, he has been studying toward the M.A.S.c in Industrial&Manufacturing Systems Engineering at University of Windsor.

IMAGE EVALUATION
TEST TARGET (QA-3)



150mm

6"

APPLIED IMAGE . Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989