

EXPERT SYSTEM FOR PROCESS SELECTION
AND OPERATION OPTIMIZATION

by
Amit Kar

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1986

Therhardt
LD
3231
.1170m
Kar
A.
Folio

ABSTRACT

Title of Thesis: Expert System for Process Selection
and Operation Optimization

Amit Kar, Master of Science, 1986

Thesis directed by: Dr. Ioannis Pandelidis
Assistant Professor
Department of Mechanical Engineering

The working of an Expert System, PRODUCER, conceived as part of a CIM system for a plant manufacturing discrete-parts, is presented.

PRODUCER starts out by determining if the desired part can be produced. Based on part attributes it then decides on a specific manufacturing process, e.g., Welding, Casting, Forging or Machining. Having selected the process, PRODUCER establishes the particular operation.

With the operation decided, PRODUCER proceeds to find all the feasible combinations, of equipment and tools, that could produce the desired component. The turning operation has been chosen to demonstrate PRODUCER's capabilities.

PRODUCER then sets about the task of identifying the most optimal pair of machine-tool and cutting-tool, which will provide the highest Metal Removal Rate, MRR. This is accomplished at two levels. At the higher level, PRODUCER generates constraints, representing physical limitations of the cutting process, for each machine and tool combination. These constraints are then passed on to an Optimization

program. This is a Fortran program, which operates at a lower level, and returns the optimum values of the process control variables, for each machine-tool and cutting-tool combination. PRODUCER finally yields the highest maximization of the MRR. In doing so it also identifies the particular machine-tool and cutting-tool associated with this global optimum.

PRODUCER, essentially a knowledge-based production system, implemented in the First Order Predicate Logic language of Prolog, also enables intelligent adaptive control.

FOREWORD

The age-old problem in manufacturing industry has been developing skills of the particular craft among its people, improving those skills to maintain competitiveness and at the same time trying to prevent the migration of the better skilled people -- or the experts -- to rival manufacturers. Neither legislation nor lucre has ever been -- or will be -- able to completely eliminate the problem, no matter what the industry or where its location.

Compounded with this problem of cultivation of skills, and preventing their polarization, is another recently emerging difficulty, brought about by automation. Automation has brought with itself high productivity, high precision and even good reliability, in a restricted sense. The problem is a lack of performance and/or skills compatibility of the human interface to automation at various levels.

Human intellect is supreme, but it is safe to say that his performance is not. A human being can never be totally relied on for producing the same response to a given prompt at all times. For example, if at 1:00 am in the morning, the drowsy operator, in the control room of a 5000 MW Power Plant, or a 2000 T Continuous Casting Steel Plant, or a Transfer Line machining Cylinder Blocks of Chrysler's latest

LeBaron engine, sees a flashing red light on the console in front of him, how does he react? He must jerk himself out of his reverie and determine as quickly as possible the cause of the alarm signal, before doing anything so drastic as to cut off the main power. He is unable to think of a better or safer alternative. Yet, the alarm may have been set off by an invalid input from a bad sensor or other malfunctioning equipment. To avoid an improper response, that could result in damage amounting to millions of dollars, all causes must be sorted out and corrected. In fact the operator cannot decide by himself and feels the need to discuss the matter with experts from the relevant disciplines, to find the best course of action in the circumstances.

A strong case, therefore exists in favor of creating some sort of "thinking" device, fashioned after a human being, yet without his limitations! Thus, in relation to the problem discussed in the previous paragraph, this device can immediately search through its own specialized encyclopaedia -- area of specialty depending on if it were the Power Plant, Steel Plant or Automobile Plant -- and come up with the correct answer as quickly as possible.

Thus an automated system, that exhibits some behaviors normally attributed to human intelligence is needed. This is the essence of artificial intelligence. Today "Expert Systems" are being built to not only emulate intelligent

human behavior but also certain animal-like instincts if the particular situation so demands it. A case in point is the Autonomous Land Vehicle, being developed by Martin Marietta. This completely unmanned vehicle will have the capability to travel over the worst possible terrain, negotiate mountains and lakes and even avoid the enemy!

human behavior but also certain animal-like instincts if the particular situation so demands it. A case in point is the Autonomous Land Vehicle, being developed by Martin Marietta. This completely unmanned vehicle will have the capability to travel over the worst possible terrain, negotiate mountains and lakes and even avoid the enemy!

To
MY PARENTS

ACKNOWLEDGMENTS

I am grateful to Dr. William Fourney for his kind consideration and continued support, that enabled me to go through my studies at College Park. I am indeed very grateful to my advisor, Dr. I. Pandelidis, for his encouragement and guidance from the very inception of my M.S. program.

It will be a great omission if I do not acknowledge with thanks the initial words of encouragement from Dr. Colin Marks, during an exploratory visit to this campus, that to a large extent helped me gain the confidence to return to academia after a long break.

I would also like to record my appreciation of the help extended by Dr. S. Azarm and other members of the Department of Mechanical Engineering, during the course of this project.

To my wife Kakoli, and little daughter, Reema, I shall ever be indebted for their infinite patience and understanding, but for which it would not have been possible for me to pursue my learning here with zeal.

TABLE OF CONTENTS

	<u>Page</u>
I. Introduction	1
II. Computerized Process Planning	4
III. Artificial Intelligence	17
IV. Expert Systems	23
V. Process Selection and Optimization:	
The Hierarchical Approach	37
VI. Optimization	51
VII. Intelligent Adaptive Control	66
VIII. Conclusion	73

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
1.	Problem-Driven Search Tree	76
2.	Goal-Driven Search Tree	77
3.	Hierarchical Process Selection Decision Tree	78
4.	Force System Acting on a Continuous Chip in Orthogonal Cutting	79
5.	Three Components of Measurable Forces Acting on a Single-point Turning Tool: Orthogonal Machining	80
6.	Constraint File Created from Prolog	81
7.	Machine Tool N.C. and A.C. Function Using Optimized Process Parameters	82

CHAPTER I

INTRODUCTION

A Computer Integrated Manufacturing system, CIM, is envisioned for a plant undertaking batch production of discrete parts. The plant is assumed to have facilities for metal shaping and joining only.

The CIM is seen to be consisting of several Expert Systems, with each such system playing its distinct role. Conceptually, these different roles would actually be the different department functions contributing to the total manufacturing effort of the enterprise.

The working of one such Expert System, PRODUCER, is presented. PRODUCER starts out by determining if the desired part can be produced. Based on part attributes it then decides on a specific manufacturing process, e.g., Welding, Casting, Forging or Machining. Having selected the process, PRODUCER establishes the particular operation, within the process-technology group, i.e., if a part must be machined then should it be turned or milled, say? In the case of Casting group, this would be choosing between investment-casting, die-casting or sand-casting, say.

With the operation decided, PRODUCER proceeds to find all the feasible combinations, of equipment and tools, that

could produce the desired component, by that particular operation. Only one operation has been chosen for a detailed demonstration of the full extent of PRODUCER's capabilities. The turning operation has been chosen for this, because it is the oldest and still the most popular of all the metal removal operations. Additional frames, mimicking PRODUCER's handling of turning, can easily be included for other operations and processes, to extend the overall range of this Expert System.

PRODUCER then sets about the task of identifying the most optimal pair of machine-tool and cutting-tool, which will provide the highest Metal Removal Rate, MRR. This is accomplished at two levels. At the outer or higher level, PRODUCER generates constraints, representing physical limitations of the cutting process, for each machine and tool combination. PRODUCER finally parses the entire set of local optima, and performs a global optimization to pick out the combination that yields the highest maximization of the objective function, i.e., the MRR. In doing so, it also identifies the particular machine-tool and cutting-tool associated with this global optimum.

PRODUCER, essentially a knowledge-based production system, has been implemented in the first order predicate logic language of Prolog.

The virtue of PRODUCER as an element of a CIM system is pointed out in its effectiveness at enabling intelligent

Adaptive Control, so crucial for maintaining a certain performance index of the machining process. Whenever PRODUCER succeeds in optimizing the feed and cutting speed for turning a given component, a dual purpose is served. The optimized parameters become data to the NC part program function and reference input, as control variables, for the AC system of the machine-tool.

CHAPTER II

COMPUTERIZED PROCESS PLANNING

Process planning has been very aptly defined as the subsystem responsible for the conversion of design data to work instruction (1). Indeed it is a very involved task, requiring much skill and experience on the part of the planner.

It must be clarified beforehand, that the term process planning refers only to the production of discrete parts. Continuous process industries, e.g., steel, cement, etc., do not need process planning, because they work on pretty much established processes. For the discrete-parts industry, however, each differently designed component needs a distinct process plan.

The good process planner must necessarily have a detailed understanding of a large range of manufacturing processes, their pros and cons, scope and limitations, as well as product design (more particularly the aspects of manufacturability) and drafting. Often, a knowledge of the desired product's service requirements can help a great deal in the process planning effort. Further a process planner must have hands-on experience in at least a few related

manufacturing processes, i.e., machining, casting or forging, etc.

The process planner studies the component drawing, selects the process and lays down a sequence of operations for producing the component from a given stock. The stock raw-material and geometry are also determined by the planner. Then for each operation he decides the machine tool and cutting tool, necessary jigs & fixtures, process parameters such as feed and cutting speed as well as machining and non-machining times (e.g., set-ups, tool changing, etc.). Thus, the planner must also have a detailed knowledge of the working environment.

Planning engineers tend to rely heavily on personal experience. As such there is no universally accepted theory about process planning. Process planners from different industrial backgrounds, can create varying process plans for the same component. Yet each process plan would be feasible. Halevi (2) presents an example, wherein a simple circular nut with external threading, M30 x 1.5, and 18 mm bore, was planned differently by four planners working independently. Two of them made plans needing 7 operations, but each with a different sequence. Plans of the third and fourth planners indicated 8 and 10 operations respectively. Clearly, the divergence of process plans, among the different planners, would be lot more pronounced for a more

complex component. So, the need for standardizing process planning methods is appreciable.

Further, process planning needs to account for so many minute details ranging from workpiece material and required production quantities, through availability and capability of each candidate machine tool, that the work itself gets very tedious and time consuming. The human planner, therefore, often tends to lose patience and thereby neglects to make the necessary detailed analysis, in order to save time.

Computer application in this area has already been acknowledged to be very beneficial for industry. The computer's handling of the tedious, repetitive tasks makes for more consistent process planning, while releasing the human process planners for the more creative tasks.

Computer-aided process planning, CAPP, essentially translates the planner's logic into mathematical algorithms. This has been the bane of CAPP until very recently. The difficulty in finding mathematical relationships for all the manufacturing functions proved to be the greatest problem. Thus in many cases approximate analyses had to be used. It was recognized that automation often proved difficult, because either intelligent reasoning or experiential knowledge, or both, were required. These were the limitations of the first generation CAPP systems (3).

With the advancement of Artificial Intelligence as an useful science, new possibilities have begun to open up for circumventing the problems of the earlier CAPP systems. Nowadays, Expert Systems, which are computer systems working with application-specific problem solving knowledge, are being developed for process planning. These work in the manner of a human; and it is expected that these knowledge-based Expert Systems will gradually become more popular in industry than their algorithmic predecessors.

CAPP systems can be classified as follows:

1. Variant.
2. Generative.
3. Semi-generative.

1. Variant Process Planning

This technique is based on GT or Group Technology principles. Application of GT allows one to reorganize the job shop into a flexible arrangement of machines which manufacture groups of similar components or families of parts.

Thus a representative member of a part family is planned in detail and the resulting process plan stored in the computer's memory. When a plan is required for another member of the family, the representative member's plan is retrieved from the database and modified as needed. Modification is very little depending on choice of representative member. Some examples are given below:

(i) CAPP (4) developed by McDonell Douglas Automation and distributed by Computer Aided Manufacturing International (CAM-I).

(ii) AUTOPROS (5) was developed in Norway. It can determine the required operations and their sequencing, selection of machine tools, clamping devices, process routes and time standards.

(iii) MULTICAPP (6).

(iv) MiPlan (7).

2. Generative Process Planning

This approach synthesizes processing information so as to automatically create a process plan for the required component. Generative CAPP systems are more complex than the variant ones because of their higher level of automation.

Development of widely applicable generative CAPP systems has proved difficult, because of the complexity and uncertainty of knowledge as well as problem-solving techniques required to be represented. In fact algorithmic methods have faced the greatest hurdles in this area. The algorithmic CAPP systems are generally designed for simple elementary surfaces that are easily recognizable by some form of algorithm, which then identifies a suitable machining process to generate the surface.

A research group in West Germany has developed a system called AUTAP to carry out process planning for rotational as

UNIVERSITY OF MANCHESTER LIBRARY

well as sheet metal parts (8). The system has been extended to cover the automatic generation of NC tapes as well (9).

CPPP (10) is a system which does process selection and sequencing for rotational parts. To use CPPP, a user provides the name of a part family containing the desired part, and for each surface to be machined, gives CPPP a code describing the surface. The part family name enables retrieval of a process model, which is a previously written simple computer program.

Wysk (11) developed a generative system for planning prismatic parts. The system, called APPAS, is intended for parts that can be made on machining centers and NC drills. To use APPAS, the user enters for each surface a code describing the surface. The system then uses knowledge of the capabilities of various processes to select a process for each surface individually.

Chang and Wysk (12) have proposed an enhancement to the basic APPAS system, by coupling it with a CAD system, through the design database. The part description is retrieved by the system from the database, then detailed instructions for its manufacture are automatically generated.

The proposed integration was actually achieved for a hole-making operation (13). The system making this possible was named TIPPS - Totally Integrated Process Planning System.

In TIPPS, the engineering part drawing is displayed on a CRT screen and surfaces requiring machining are marked interactively. The sequence of operations is determined by reference to a process knowledge database. TIPPS also automatically selects parameters like feed and cutting speed (14).

3. Semi-generative Process Planning

This is like a variant process planning system that based on either codes or specification-analysis, retrieves a general plan which is then automatically modified to fit more correctly the part in question.

Several such systems exist, but all of them are limited in their capabilities, in that they can handle single part types, e.g., prismatic objects. Some examples are given below (3):

GENPLAN developed by Lockheed Corp.

GECAPP developed by General Electric.

ACAPS (Automatic Coding and Parts Selection)
developed by Pennsylvania State University.

A system was developed at Budapest Technical University, Hungary (15), on the justification that purely generative systems tend to be too large, slow and have a limited range of applicability. In fact, creators of the previously mentioned CAM-I have concluded that totally generative process planning is neither practical nor desirable.

A good deal of work has been done in the creation of CAPP systems. For a more extensive survey of previous and current work in this area, see (19). Recently, CAPP systems using A.I. techniques have also been created (16), (17).

Reasonably complex parts consisting of over one hundred features, i.e., groove, slot, keyway, etc., have had their machining plans generated by GARI (17). GARI is a rule-based system, whose control structure combines fact deduction and conflict resolution. It carries out process selection and sequencing, but does not choose machine tools or fixtures. The authors of GARI recently published their subsequent work (18), that attempts to reduce some of its earlier limitations.

Work on integration of CAPP with the NC part programming function, that actually carries out the machining has also been done (19). There is also evidence in the literature, about work involving CAPP, that couples CAD with CAM. However, a good deal of work needs to be done for effective incorporation of the CAD function into the total CIM system.

The work reported in this thesis can be categorized as a generative process planning system. It is a rule-based expert system called PRODUCER, that first determines all feasible combinations of machine tool and cutting tool, that can produce a given part and then proceeds to optimize and pick the combination that best satisfies the optimization

criterion. In this case the criterion has been chosen as the metal removal rate, MRR.

At this point, however, PRODUCER can handle only turned parts. This research indicates PRODUCER's treatment of a single featured part, i.e., one cylindrical surface. This can be extended to accommodate multiple features.

Little or no work has been done on optimization in process planning. There are two kinds of optimization possible in the area of process planning:

1. Sequence optimization
2. Parameter optimization

1. Sequence Optimization

This essentially involves determining the most economical sequence of operations for producing a particular component. It must be clarified, however, that all the operations transforming the given raw-material to its final desired state, cannot be optimized because there are certain operations that have to be carried out before the next can follow. As an illustration, consider creating a keyway on a cylindrical shaft. The proper sequence for this would be shaft turning, series of short-depth drillings (up to keyway depth) along keyway length and finally the actual milling of the keyway slot using an end-mill cutter fed along the rough grooving produced by the drillings. Now sound workshop practice does not permit any other sequence -- it is ridiculous to imagine cutting the keyway before first turning the shaft! Many more such instances can be given.

However, there are also several situations that the manufacturing engineer can visualize, whereby the product design, its service requirement or even manufacturing convenience is not jeopardized by slightly altering the operation sequence. Usually for any component about 20 to 30% of all the operations required, for producing it can tolerate flexibility in sequencing. Thus if "k" operations out of a total of "n" can have variable sequencing, the possibility of determining the "least-cost" sequence definitely warrants attention. Such a problem can be well tackled by A.I. search techniques -- more particularly "least-cost search" where each operation is a node of the search space and the problem is to visit all the nodes at a minimum cost. This is a problem comparable to the "Travelling Salesman Problem", discussed by Nilsson (25). A.I. search techniques are discussed in Chapter 3.

Nau is working on a process planning scheme, SIPP (20) (Semi Intelligent Process Planner) that accounts for this sequencing problem. SIPP uses a least-cost search technique to find the most economical sequencing of the operations required to generate a given surface.

The problem of sequence optimization becomes more complicated when looked at globally from the point of view of all products being manufactured in the factory simultaneously. When all, or even some, of these products are to be processed on the same machines the sequence

determination, of any one product, cannot be done independent of the rest. Considering availability, or otherwise, of necessary machine-tools and fixtures, the problem increases in complexity. Indeed, at this point, the problem goes into the realms of Scheduling, which calls for the application of O.R. methodology and is beyond the scope of this research. Work along these lines has been done at Carnegie-Mellon University (21).

2. Parameter Optimization

This entails finding the "best" cutting condition (for machining), i.e., feed and cutting speed, for a particular operation such as turning. This is modeled as a classical non-linear constrained optimization problem for some agreed objective function, e.g., Metal Removal Rate or Cost of Production, etc. The constraints, which maybe equalities or inequalities, represent the physical limitations of that operation. Typically these may be Tool-tip temperature, Cutting force or power, Surface finish, etc.

The literature does not seem to support evidence of any work in this area. Past researchers in the area of CAM in general and CAPP in particular have not dwelt on the subject of optimum feeds and cutting speeds because for any component the feed and speed are quite reliably assigned by the part program or CAPP function to the NC system of the Machine Control Unit (MCU). For example, the ICAPP system (22) computes the feed and cutting speeds by a formula,

constructed from a least-squares curve-fitting technique applied to recommended feeds and speeds for a wide range of workpiece and tool materials. Perhaps for these reasons the need for optimizing feeds and cutting speeds was never really felt.

The benefits of such optimization are better appreciated when viewed in the greater perspective of an integrated manufacturing system. Adaptive Control is one such critical element of a CIM system. Typically, Adaptive Control seeks to maintain the particular manufacturing process at a desired index of performance. Chapter VII gives a detailed explanation of the Adaptive Control function. In order to maintain performance the AC system must monitor critical process variables, in real time, and continually compare them to reference values of preferably these same parameters or, alternatively, some closely related ones. This research has determined that for metal cutting, feed and cutting speed are functions of the state variables of the process, in terms of the workpiece, cutting tool and machine tool.

In fact the motivation for establishing functional relationships between feed and cutting speed and the different process variables, arose from a necessity to integrate process planning and operation control in a suitable manner. Thus operation control is effected by the machine's NC and AC functions, working in close co-

ordination with each other. The NC function generates the commands for machine operation, based on the part program. The part program in turn is built up from the process plan of the particular component. Feed and cutting speed are common features in all these different functions. In addition, these two variables are also the process "drivers" as it were.

Accordingly, it was considered fit to devise a computerized scheme that would decide on an optimum feed and cutting speed for meeting a certain objective, subject to constraints or limitations of the process itself. Thus, whenever optimization is possible a dual purpose is served. The optimized parameters become data for the NC function and reference input for the AC function. See Figure 7.

At the same time, conventional methods for selection of feeds and cutting speeds, by the particular CAPP system, would also be retained. These methods could easily be embedded in the system architecture of PRODUCER, the expert-system presented in this thesis. Either pattern-matching from a large relational-database or the computation technique could be adopted. This is necessary because optimization will not be possible in all circumstances, e.g., when certain constraints become over-restrictive for a particular workpiece-machine-tool combination. In these situations the default case would apply, i.e., feed and cutting speed decided in the usual way.

CHAPTER III

ARTIFICIAL INTELLIGENCE

Artificial Intelligence, AI, has been an area of research for about 30 years. Some of the early work is as old as the concept of automation itself (23). However, only in the beginning of this decade some commercially useful work has been done in this area.

The main thrust of AI research has been in the computational simulation of human behavior -- or more correctly intelligent human behavior. AI systems are distinguished from conventional computer systems in two ways (24):

1. Processing of symbolic rather than numeric data

Inference is based on well established logic principles. Equipped with suitable semantic notation, it can operate on symbols. The obvious advantage of inferencing is that it does not require an a priori mathematical theory, but instead can be used to manipulate concepts.

2. Heuristic rather than algorithmic problem solving

Most of the real world problems have no algorithmic solution, per se. Also, there are many problems, classified as semi-decideable, i.e., even if their solutions maybe attempted algorithmically, the number of steps for arriving

at the solution is beyond the scope of any computer. A game of chess is a good example.

Heuristic procedures on the other hand, although do not guarantee a formal solution, do provide a good approximation most of the time and work relatively quickly. They may involve trial & error and even guessing.

SEARCH IN AI

Much of the early work in AI was concerned with searching for solutions to problems. One way of representing problem solving in AI is in terms of a hierarchical structure called a tree (cf. Figure 1). Since the solution results from a search among alternative choices, this "tree" depicts the entire search space.

In Figure 1, the root node is the initial state, the "branches" are the operators and the "leaf" nodes are the intermediate states between the initial and goal states. It shows the different ways of getting from point A to point D. Search is a sequence of operators to take a problem from its current to the final or goal state. This can also be seen as problem-driven search or forward reasoning. The inverse case is that of goal-driven search or backward reasoning, as depicted in Figure 2. Here the root node is the goal state.

For a complicated problem this graph becomes very involved to be easily understood. So, only the explicit tree is usually viewed. The explicit tree are the nodes or

UNIVERSITY OF MICHIGAN LIBRARY

states and those branches that lead to the solution. The entire tree becomes explicit only in the case of an exhaustive search.

Some of the better known search techniques are:

1. Blind Search
2. Heuristic Search
3. Uniform Cost Search
4. Depth first Search
5. Ordered Search
6. Best First Search
7. Bi-directional Search

Blind Search is randomly done in that it does not use any knowledge of the problem. For a complex problem, the Blind Search technique can easily be overwhelmed by a combinatorial explosion of possibilities for arriving at the goal state.

Heuristic Search is a thumb-rule method for guiding the search, by using information about the nature and structure of the problem domain. A properly designed heuristic can be efficient, and capable of avoiding combinatorial explosion.

The Uniform Cost Search method applies an evaluation function to each node generated and then pursues those paths that have the least expected cost. Typically, the evaluation function calculates the cost from the root to the particular node being examined, and using heuristics estimates the cost from that node to the goal. Adding the

two produces total estimated cost along the path, and thus serves as a guide as to whether to proceed along that path or continue on another.

Depth-First Search is a search that proceeds from the root node to one of the successor nodes and then to one of that node's successor nodes, etc., until a solution is reached or the search is forced to backtrack. Backtracking is a powerful AI mechanism also used in deductive problem solving. It occurs whenever a leaf node, which really is an intermediate goal state, fails and the search goes back up the branch it had come to the predecessor node in order to try an alternative branch to a next level node.

In Breadth-First Search, the nodes in the search tree are generated and examined level by level, starting with the root node before proceeding deeper. This approach is guaranteed to find an optimal solution if it exists.

There are some problems, like the 8-puzzle (25) (which requires that a jumble of 8 numbered tiles and a blank one be sorted out in a 9-slot square, so that the numbered tiles are arranged in some desired sequence), which can be solved by forward reasoning -- going from the initial state to the desired goal state -- or by backward reasoning, which starts at the goal state, applies inverse moves and work towards the initial state. In the backward reasoning scheme, each inverse move would produce a sub-goal state from which the immediately superior goal state can be reached by one

forward move. Backward reasoning merely reverses the roles of states and goals and uses rules that correspond to inverse moves.

Bi-directional Search is when a solution to a problem is attempted using forward and backward search simultaneously. To do this state and goal descriptions are incorporated into the global data base. The control system must decide at every stage whether to apply an applicable F-rule or an applicable B-rule.

An Ordered search algorithm finds the next best successor to a node depending on a merit function.

Search techniques are now relatively mature. Some good algorithms, e.g., A*, AO*, B*, etc., have been developed by researchers in the field. Search is a very basic AI method and is considered very useful where knowledge of the problem is minimal or incomplete.

PRODUCTION SYSTEMS

The term production system has been used rather loosely in AI, although it usually refers to more specialized types of computational systems. Production systems derive from a computational formalism proposed by Post (26), that was based on rule applications in a certain order. In 1976, Rychener first proposed production systems as a programming method (27).

Production systems are very central to AI methodology, and have variously been called rule-based systems, black-board systems and pattern-directed inference systems. The major elements of an AI production system are:

1. Global Database.
 2. Production Rules.
 3. Control Strategy.
-
1. The global database is the central data structure used by an AI production system. Depending on the application this database may be as simple as a small array of numbers or as complex as a large relational file structure.
 2. The production rules operate on the global database. Applicability of each rule is subject to satisfaction of one or more preconditions to that rule. Rule application changes the database, in that a state change or mode change takes place, as during search.
 3. The control strategy essentially chooses which rule to apply and ceases computation when some termination condition is reached.

UNIVERSITY OF TORONTO LIBRARY
130 St. George Street
Toronto, Ontario
M5S 1A5

CHAPTER IV

EXPERT SYSTEMS

Before the last decade, AI researchers relied on non-knowledge-guided search techniques in trying to find solutions to problems. For elementary problems or well-structured games, e.g., 4-Queens problem discussed by Nilsson (25), these early methods were quite suitable. In fact there are always some problems -- particularly of the Operations Research or Scheduling nature -- whose solutions can only be obtained by search. But for very complex problems, the search space tends to expand exponentially with the number of parameters involved. When traditional search techniques proved inadequate for solving these type of problems, knowledge-guided search approaches began to be conceived. This led to the field of Knowledge Engineering and Expert Systems.

The ability of the knowledge-based approach to minimize, if not eliminate, the combinatorial complexity encountered by classical search techniques when applied to real world problems, springs from its use of rule-based heuristics as opposed to numerical heuristics. Numerical heuristics, involve merit or evaluation functions typically employed by classical search techniques, discussed in the previous chapter. Thus, the rule-based system is able to

reason about its own search strategy as well as reasoning about the specific problem domain.

Expert knowledge is generally thought of as knowledge accumulated from many years of experience. However, knowledge is not necessarily from experience alone. In fact, no matter how specialized or restricted the domain, it is naive to think that a single individual (or group) can at any time claim complete knowledge of it. For one thing, the domain's knowledge frontier is always expanding with time, while the respective individual's memory span is not. Thus, the super-successful lawyer or the brilliant cardiac surgeon can never afford to cease the learning effort. They find it necessary to continue acquiring their domain knowledge for retaining the label of "expert".

Thus, the expert needs to have two kinds of knowledge:

1. Knowledge of the domain (or a specific sub-set of it).
2. Knowledge of where to find more knowledge about that domain.

The expert can analyse a problem, assemble parts, make inferences, give advice along with an explanation of his reasoning. If giving advice on a specific problem, is the expert's goal he can arrive at that goal either on the strength of his experiential knowledge alone or on his ability to acquire the knowledge, required to give that advice, or both. Further, the expert must determine which

body of knowledge is applicable to the problem and/or its sub-parts rather than merely proceeding step by step, in algorithmic fashion.

An expert system is constructed to emulate the human expert as closely as possible. It is a computer program, which deals in a specialized field requiring some expertise to provide solutions to problems and/or give advice. Since expert systems deal with knowledge, they must have the ability to handle symbolic representations of data.

The real justification for an expert system is that it has the potential to rise above the limitations of the human expert. Several human experts collaborate to help develop the expert system, which then becomes a repository of the accumulated knowledge of those specialists. Accordingly, it is more competent than any single specialist and also has much greater objectivity. However, to quote an U.S. Department of Commerce report of May '82 (28), prepared for NASA, "There are not yet many examples of expert systems whose performance consistently surpasses that of an expert".

Feigenbaum, a pioneer in expert systems (29) states:

An expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such a level, plus the inference procedures used, can be thought of as a model of the expertise of the best practitioners of the field . . . The performance level of an expert system is primarily a function of the size and quality of the knowledge base that it possesses.

Conventional computer programs differ markedly from those functioning as experts. Their main drawbacks in respect of expert problem-solving are as follows:

1. They are usually complex and difficult for anyone but their designers to understand.
2. For purpose of computational efficiency, the knowledge with which these programs work, are inseparably intertwined with its control parts. This makes alteration very difficult.

On the other hand, in an expert system the program is only a general reasoning mechanism (30), and the system can ideally be changed by adding or taking away rules from the database.

3. They have no means of suggesting to the user, what, if any, facts are required to solve the given problem.
4. They cannot justify their solutions.

The three basic elements of an expert system are as follows:

1. A Knowledge Base:

This consists of domain-specific facts and heuristics associated with the problem.

2. An Inference Engine:

This manipulates the knowledge base as necessary for problem-solving.

3. A Working Memory, or "global database":

This maybe thought of as a kind of scratch-pad that continually updates solution status of the problem, beginning with input data. As such this is transparent to the user, unless the system is commanded otherwise.

Most expert systems are custom-built, i.e., constructed for use in a specific domain. However, there are domain-independent expert systems called "shells". These have empty databases and knowledge bases, and so may be used for a variety of applications. In practice, however, shells are not suitable for some applications.

Comprehensive reviews of existing expert systems and system shells can be found in Hayes-Roth et al. (31), Buchanan and Shortliffe (32), Allwood et al. (33) and Rychener (34). Some recent commercial expert system shells in use are KEE (35) and ART (36). Expert Systems Limited of U.K. have developed a shell written in Prolog, that has been used to determine building regulations (37).

Since knowledge is the fundamental feature here, the manner in which it is represented in and/or used by the expert system is of utmost importance. Indeed, the realization that bore the concept of Knowledge Engineering, as it is known today, is that the knowledge of the problem

domain is more important than the procedure for arriving at its solution.

KNOWLEDGE ENGINEERING

Knowledge Engineering focuses on how to bring expert knowledge to bear in problem solving. It is concerned with the acquisition, representation and manipulation of human knowledge in symbolic form. It means more than just a capacity to acquire facts -- as a human might from a dictionary or handbook. It may be thought of as a process that maps the available domain-specific knowledge, onto the problem space of the AI system.

The importance of the representation, which is abstracted from the main problem cannot be overemphasized. The scheme proposed here simulates the thinking of the process-planning and manufacturing engineers.

The thinking process of the expert and his almost intuitive reasoning, implicitly employs such logic tools as tautologies, modus ponens (38), Robinson's resolution theorem, etc. Mathematical logic likewise works in the same pattern -- beginning with certain axioms, making some unifications, negations, then deductions leading to the final answer. This is the very basis of the predicate calculus (39), which is perhaps one of the most effective AI tools -- and definitely appropriate for designing expert systems. Prolog, a first order predicate logic language

developed from predicate calculus, is therefore ideal. The knowledge used by the system can be classified as follows:

1. Declarative Knowledge.
2. Procedural Knowledge.
3. Control Knowledge.

These are explained below.

Declarative Knowledge: Object Representation

This is essentially a database of facts or axioms. For the problem of this research it has been referred to as a Variable Data Bank (VDB). The VDB has been named thus because the information it holds is subject to change, as and when necessary.

The VDB stores every bit of information, pertinent to the domain, in the form of facts. Typically these are complete specifications of machine tools, cutting tools, jigs and fixtures; process information such as possible cutting forces for a given geometry and material of a tool operating at a certain speed; maximum allowable tool-tip temperature for different materials, values of mechanical equivalent of heat, hardness to ultimate tensile strength relations, etc., are also stored. These constitute the declared or factual knowledge collected from experts in the area, handbooks and available data bases.

The relevant objects in a manufacturing environment are the workpieces, machine tools, cutting tools, etc. Each

object may be described in terms of a frame structure such that all relevant information about the particular object is stored in one place. Such representation is modular in the sense that objects can be added to or omitted from the databank easily. As an example, a specific lathe identified by the unique qualifier "lathe_1" is represented in the following manner:

```

/* machine tool specs */
lathe_1 is_a machine_tool.
type(lathe_1,turret).
max_power(lathe_1,30.0).
max_torque(lathe_1,20.5).
.
.
```

Use of the `is_a` predicate as an infix operator is made here purely for clarity purposes. The representation of a cutting tool and a workpiece is done in a similar manner as above:

```

/* Cutting Tool Specs */
ct_1 is_a cutting_tool.
material(ct_1, carbide).
back_rake_angle(ct_1,20).
side_cutting_edge_angle(ct_1,12).
front_cutting_edge_angle(ct_1,5).
.
.
```

```

/*Workpiece Specs */
wp1 is_a workpiece
shape(wp1, cylindrical).
length(wp1,4.0).
diameter(wp1,4.0).
.
.
.

```

It is not necessary however that all information about the "world" objects reside in the VDB. Shown below is a method by which problem-solving will proceed even if the required data is not residing in the VDB. This is an useful tool, particularly in view of a very complex problem when the user may quite easily omit putting in a certain value. Also, for an open-ended system, such as PRODUCER, where world information can periodically be updated, this provides an option to obtain the most current data interactively from the user.

The general-purpose predicate "find" was created to retrieve information irrespective of the source. The predicate "find(S)" will instantiate the structure S either from the VDB, or interactively from the user, or from default values. The structure "S" is of the form attribute (object_name, Value), where Value is the unstantiated

variable. As an example, find (back_rake_angle(ct_2,X)) will first attempt to get a a value for X from the VDB, failing which it will request the user for a value. In case the user is unable to find a value, a default value will be used.

Procedural Knowledge

Knowledge about the problem, represented in the rules is often called procedural knowledge. For intelligent information retrieval, the procedural knowledge would include rules that permit manipulation of the facts in the declarative knowledge base.

An illustration follows:

```
Shear_angle(CT,Phi): - back_rake_angle(CT,R),
                    friction_angle(CT,Theta),
                    Phi is 45 + R/2 - Theta*(0.75 + 0.005*R).
```

Thus, the back rake angle of the specific cutting tool is picked up from the database and the friction angle is determined from another production rule. Then the shear angle is easily computed.

The final goal of the A.I. system is to find out how a particular workpiece, WP, may be produced. It is stated below.

```
produce(WP, Process, Process_Params):-
    not(material_type(WP, metal)),!, fail.
produce(WP, Process, Process_Params):
```

```
choose_oper(WP, Process)
selected_optm_equip(WP, Process_Params).
```

Thus, if the workpiece is non-metallic it cannot be produced, and Prolog does not try to resatisfy the "produce" goal alternatively. The left-hand clause of the rules, is the head goal, which succeeds by satisfaction of the sub-goals on the right. Thus, the argument variables of the "produce" predicate get instantiated by the satisfaction of the conjunction of the "choose_oper" (i.e., choose operation) and selected_optm_equip (selected optimum equipment) subgoals.

These two sub-goals decompose into many lower level sub-goals. Thus, in order to decide the process, i.e., casting, welding, machining, etc., part design attributes are retrieved from the database and compared to processing limitations. Once, the process is decided, operation possibilities are checked out, i.e., if the workpiece is established as a machine part, first the particular machining operation (turning/milling/boring, etc.) and thereafter the equipment and tools are decided. The different levels of selection are clear from the decision tree shown in Figure 3. A feasible combination is generated by a complete path along those nodes actually visited by the tree search. As such a feasible combination is an ordered list of the form (Workpiece, Process, Operation, Equipment, Tool).

Control Knowledge

The control knowledge basically guides the entire A.I. problem solving scheme. When a particular inferencing rule is to be applied it is directed by the control knowledge. Even housekeeping and administrative functions, of the system, as it were, are carried out by virtue of this control knowledge. Thus opening and closing of files, calls made to other programs, sending data to other programs, etc., are all within the purview of control knowledge.

Generation of each feasible combination of machine-tool and cutting-tool is done by a depth first search using the built-in inference engine of Prolog which includes automatic backtracking. The search is done in a top-down backward chaining fashion. Pruning of unnecessary branches in the search tree is done at each level of selection. Thus, a particular lathe will be examined for feasibility only after the turning operation has been determined to be appropriate. Note that a particular lathe may still not be available or suitable for the job.

The set of feasible combinations is then parsed, and for every pair of machine-tool and cutting tool, the constraints are automatically generated and furnished as a subroutine to the Fortran optimization program. The optimization program returns the optimum feed and cutting speed to the A.I. module, which is the main driver of the

software. The A.I. function then calculates the resulting MRR and generates a parameter set of the form (MRR, Fopt, Sopt, WP, MT, CT), where, Fopt and Sopt are optimum feed and cutting speed for producing the workpiece, WP, on machine tool, MT, using cutting tool, CT. In this way, with every MRR the associated equipment also get identified.

Thus, for every machine tool and cutting tool pair, for a given workpiece, one such parameter set is obtained. Finally, the global optimization is carried out by picking the parameter set with the highest MRR.

The indicated approach prevents a combinatorial explosion and accesses the relevant information only as needed.

THE A.I. TOOL FOR SYSTEM EXECUTION

The choice of Prolog (2) as the language of implementation was motivated by the following considerations:

- a. Since Prolog is essentially a database coupled with an inference mechanism, it is particularly useful for this sort of application.
- b. The rule-based production system, used by Prolog, seems more appropriate, for this application, rather than decision tables because of the former's simplicity and generality (37).

- c. Once the A.I. scheme has been developed by the domain expert, it becomes user friendly even for relative novices.
- d. Prolog can be regarded as either a declarative or a procedural language. This facilitates choice of coding perspective according to convenience.
- e. Prolog data bases of arbitrary complexity may be easily constructed without affecting modularity. A direct benefit from this is the possibility of getting queries answered at any level of the goal tree.
- f. The inference engine of Prolog is implicit in its automatic depth-first search. This frees the system designer to concentrate more on the knowledge representation of the system.
- g. Prolog uses a tentative control strategy, rather than an irrevocable one, in that after application of a selected rule, provision is made to later return to this point and apply some other rule if necessary (38). This is the very important and useful backtracking mechanism, inherent in Prolog.

UNIVERSITY OF TORONTO LIBRARY
130 St. George Street, 4th Floor
Toronto, Ontario M5S 1A5
416-978-2082

CHAPTER V

PROCESS SELECTION AND OPTIMIZATION: THE HIERARCHICAL APPROACH

Before proceeding to explain the system architecture of PRODUCER and how it works in a hierarchical fashion, it is necessary to point out the significance and context in which the Process Selection problem has been posed in this research.

Process Selection in the Job Shop

Process Selection is the work of choosing the appropriate basic manufacturing technology for producing a certain commodity or material.

Generally speaking, Process Selection has no relevance in continuous process industries, e.g., oil refineries, steel, cement, sulfur etc., since the basic technology of manufacture of these commodities is essentially set.

So, Process Selection can be said to apply to discrete-part manufacturing alone. However, the vast majority of products both visible and invisible to the consumer are discrete parts, whose methods of manufacture are also well established. Thus, the processing technique and operation sequencing in the production of say, semi-conductor chips,

electric motors, lawn-mowers, transformers, telephone-jacks, etc. are not changed from item to item. Only, a major design change, duly tested on a proto-type, and approved, can cause processing change during volume manufacturing.

Consequently, the application-domain of Process Selection gets restricted to the job-shop. This research has precisely attempted to address the problems of a typical medium-scale mechanical job-shop. Nonetheless, there should be no problem in suitably applying the current work to a support department (of a large manufacturer) undertaking maintenance work or batch production.

The typical mechanical job shop definitely has a machine shop, with lathes, drills, shapers, mills and other metal-cutting machines. Some shops would also have attached foundry, welding, press shop and perhaps even a forge. Needless to mention, the actual existence of these extra facilities would vary widely, depending on the job-shop's location, clientele, financial stability and other factors.

The job shop is expected to be able to produce a wide range of items, albeit in small quantities. The fundamental requirement that enables the job-shop to live up to such an expectation, is a very good command of manufacturing and engineering knowledge. The job-shop must, therefore, have on its payroll skilled process planners and operators, who on receipt of an order can conclusively decide which process or processes can best produce a given part that will meet

the desired geometric, functional and engineering specifications. Indeed, to be able to do this at a minimum cost to the job-shop owners, is highly desirable. The requirement of these skills, increase almost exponentially with the variety of products handled by the job shop.

The existence of good job shops is very crucial for the subsistence of industry as a whole. Yet, the pace of automation introduction in job shops is dismally low, and much behind their larger counterparts. Also, big industry seems to monopolize the attention of researchers, while the job shop is left to struggle along, innovate and fend for itself or simply die out. Perhaps, academic institutions could take the lead in extending help to the job shop -- by standardizing and computerizing its procedures from receipt of an enquiry to delivery of final product. In this way job shops can maintain their rightful place in the industrial infra-structure.

The job shop is in perpetual difficulty to retain skilled personnel. Perhaps its need for "Expert Systems" is the greatest. These would be knowledge-based Expert Systems, that will contain the accumulated experience of the skilled workmen. PRODUCER is one such Expert System, intended for use in a job-shop doing metal shaping and joining work.

Process Selection must be distinguished from another very similar term, Processing Planning. The two are often

used interchangeably. Process Planning involves laying down the sequence of operations that transform the raw material into the end-product, whereas Process Selection determines the processing technique. Indeed, Process Selection maybe considered to be a subset of Process Planning.

PRODUCER does not do any process planning as such. It selects a process, and a specific operation, and optimizes the variables thereof. This is explained in detail in the next section.

The use of the terms "operation" and "process" must also be clarified in the context of this thesis. "Process" indicates the major manufacturing technology, while "operation" is a member of that group. Thus, if Welding is the process then arc-welding, resistance welding, fusion welding, etc. are the operations, or if Machining is the process then drilling, turning, boring, milling, broaching, etc. are the operations.

Purpose of Producer

The purpose of this research was to devise an automated scheme that will give the "best" way of making an item.

In other words, the intent was to come up with a computerized system, which can "understand" a given item, in terms of its physical features and functional requirements, and prescribe the appropriate process and optimum operating conditions for its manufacture.

For a computerized system to "understand" and "infer", cognitive capabilities are called for. Having such capabilities, imply picking up part attributes from a database and trying to recognize them in the light of some in-built knowledge.

As a very simple example: if the system "knows" that any 2-legged creature that can fly and has a feathered coat is a bird; and if this system is asked if an object called "tweety", that can fly and has feathers, is "bird", "airplane" or "man", it can reply "bird".

Since the conventional algorithmic computer program cannot deal with such type of problems, it was decided to adopt A.I. methodology. This led to the development of the Expert System, "PRODUCER".

The Hierarchical Approach

In order for PRODUCER to determine the "best" way of making a certain item, it must deal with the problem in a hierarchical fashion, i.e., take the decisions level by level. This maybe thought of as a top-down approach to the problem solving.

The different steps in the decision tree are given below, the first step being the root node:

STEP 1. Can it be made at all?

If yes go to Step 2

Else stop.

- STEP 2. (a) Find major process-family.
(b) Is it possible to process the item under given shop limitations?

If yes go to Step 3

Else stop.

- STEP 3. Choose specific operation.

- STEP 4. (a) Find Equipment and Tools, by which operation chosen in Step 3 is feasible.

This requires:

Availability of m/c and Tools

Satisfying certain preconditions for using the respective m/c and tools.

- (b) Compile a set of Equipment + Tool combinations.

- STEP 5. (a) For each member-combination of set obtained in Step 4(b) determine the optimum operating condition.

- (b) Make a set of local optima and the respective associated equipment and tool.

- STEP 6. Find Global maximum and identify the associated equipment and tool.

The hierarchical decision making proceeds stepwise from 1 through 6, going from the problem-state to the goal-state. The particular decision at each step, however, is effected by backward reasoning. This is the underlying concept of

combined top-down backward-chaining methodology used in this Expert System.

STEP 1.

PRODUCER is designed to work with machining, casting, welding and forging processes. Since it is an A.I. system, other processes like drawing or forming can be incorporated later without affecting its current features.

So, PRODUCER must find out if the part is metallic.

See Page 32 for an illustration.

STEP 2:

From the attributes of the metallic part, it must determine the major process technology group, i.e., casting, or welding or machining. Thus, if the part is bulky, has complex contours and is not required to be finished on all its surfaces, it is a candidate for casting. On the other hand, if the part must be well finished on all surfaces, have regular geometric features that have close dimensional tolerancing, it must be machined.

Once the major process group is established, PRODUCER determines if the existing shop facilities permit manufacture by that process. Thus parts above or below a certain size, weight, or other feature, cannot be cast, or welded, or machined as the case maybe.

The following lines of code serves to illustrate this:

```

choose_oper(WP, Process):- casting(WP), !,
    Process = casting.
choose_oper(WP, Process):- machined_pt(WP), !,
    find_machining_opern(WP,Process).
casting(WP):- casting_design(WP), cast_possibl(WP).
cast_design(WP):- (complex_shape(WP);
    all_surfaces_not_finished(WP)).
cast_possibl(WP):- vol(WP,V),
    0.01 < V, 50 > V.
vol(WP,V):- leng(WP,L),
    breadth(WP,B),
    height(WP,H), V is (L*B*H).
.
.
.

```

STEP 3:

If shop facilities are no bar, the specific operation, within the major process family must be pinpointed. Thus, if found that a part can be welded, then should it be arc-welded or resistance welded, or fusion-welded. Or, if the part must be cast then should it be sand-cast, gravity die-cast, or investment-cast, etc.

For these decisions, PRODUCER must not only be able to pick out the respective detailed specifications of the part, e.g., surface finish, minimum tolerance, number of pieces required, shape, material, etc., but also have the necessary domain knowledge for carrying out inferences.

At present, the machining process has been chosen to demonstrate the capability of "PRODUCER", because of its wide popularity. In this regard, part of PRODUCER maybe seen as an Expert System shell, wherein inferencing knowledge for other process, can be filled in as desired.

Accordingly, PRODUCER chooses turning for a cylindrical component such as a piston-rod.

Prolog rules effecting this are shown below:

```

      .
      .
      .
machine_pt(WP):- (finish_criterion (WP);
                  closely_toleranced_dimensions(WP)).
finish_criterion(WP):- surface_finish(WP,S),
                       S < 0.000300.
find_machining_opern(WP, Process):- cylindrical(WP),
                                     turning_possibl(WP), !,
                                     Process = turning.
find_machining_opern(WP, Process):- cylindrical(WP),
                                     boring_possibl(WP),
                                     Process = boring.
turning_possibl(WP):- weight(WP,W)
                       0.05 < W,  50 > W.
      .
      .
      .

```


STEP 4:

To find a feasible combination of equipment and tool, whereby the chosen operation can be carried out to produce the component, PRODUCER must find which machines and tools can do the job and then if they are available.

To determine if a machine is feasible, a match must be obtained between the part's features, and the relevant features of the machine. For turning this would imply finding out if the diameter of the bar-stock can be accommodated in the chuck or collet of the lathe, and if the particular size of workholder is available. Further, this machine can only use those cutting-tools whose sizes are within the tool-holder capacity of the lathe. Finally, of course, both machine and cutting tool must be available for use.

Accordingly, a set of feasible machine and cutting tool combinations can be obtained.

STEP 5:

This implies finding the local optimum for operation with each combination of m/c and tool.

The objective function was chosen as productivity, which is sought to be maximized subject to the physical limitations of the particular operation, modeled as constraints. For machining, this implies maximizing the metal removal rate, MRR.

For metal cutting operations in general, and turning in particular, Feed and Cutting Speed are the most critical control variables. Thus, it was decided to model the constraints with Feed and Cutting Speed as the variables. Also, $MRR = F.S.D.$, but modeling has been done leaving "D" out, since depth of cut remains constant for a single pass in machining (as such NC m/cs have $2\frac{1}{2}$ -D capability for this reason).

The parameters and constants in each constraint change from combination to combination, as these are typically the tool back rake angle, m/c's h.p. rating, m/c's spindle torque rating, etc.

The optimization problem is purely mathematical in nature, and so more conveniently tackled by the algorithmic-type programs such as Fortran.

The constraints, however, are generated in the A.I. module, for each m/c + tool combination, and passed on to the algorithmic module, which is then made to execute from the A.I. module. The optimized F & S values are returned to the A.I. module, which then performs a simple calculation to find the consequent MRR. This gives the system a hybrid character.

For every combination a local optimum is generated. Then, a set of the local optima and the associated equipment is compiled. Eventually all these sets are assimilated in one single set, to give a set of sets.

The rule given below shows how for a certain combination of workpiece (WP), machine-tool (MT) and cutting-tool (CT), constraints are generated, optimized values of feed and cutting speed are obtained from the Fortran program, metal removal rate (MRR) is calculated and the entire information put in a set of the form:

```
[MRR, Fopt, Sopt, WP, MT, CT]
optimize([WP,MT,CT], List):- find_constrnts(WP,MT,CT),
                               get_optm_varbls(Fopt,Sopt),
                               MRR is 12*Fopt*Sopt,
                               concatenate ([MRR,Fopt,Sopt], [WP,MT,CT], List).
```

Also see Page 65 of this thesis.

STEP 6:

Finally, the A.I. module performs a global optimization, to pick the set enabling the highest maximization of the objective function. Thereby, the associated equipment and tool are also identified.

Since the Expert System essentially deals with metal cutting, its theory is briefly discussed.

Metal Cutting

Metal cutting also called machining, is the removal of unwanted metal from a workpiece in the form of chips, so as to obtain a finished product of desired size, shape and finish.

Over the years a great amount of research and experimentation on the process has led to improved productivity; but the complexity of it has resisted progress in obtaining a complete theory of chip formation. This is because most theories have ignored the plastic deformation properties of the work material and/or have not been able to accurately characterize the interactions at the sliding contact surfaces between the tool and the chip.

Metal cutting is a very large - strain plastic deformation process operating at exceptionally high strain rates, which makes it quite unique. The problem is further complicated by tool geometry variations, wide variety of tool materials used in the process, temperature or heat problems, and the great variation in operating conditions of the machines performing this process. Basically, the chip is formed by a localized shear process which takes place over very narrow regions. The shear process itself is discontinuous in which a series of shear fronts or narrow bands produce what is called a lamellar structure.

For all metal cutting process it is necessary to distinguish between speed, S , feed, F , and depth of cut, d . Speed, S , is the primary cutting motion, which relates the velocity of the tool relative to the workpiece, given in surface feet/min. Feed, F , is the amount of material removed per revolution, or per pass of the tool over the workpiece, usually given in inches/revolution. For a single

pass in turning, the depth of cut remains constant and metal removal rate, MRR, is given by: (see Figure 1 and 2).

$$\text{MRR} = 12 F S \text{ sq inches/min}$$

This is the objective function, which is sought to be maximized. F & S cannot be made indefinitely high, since that would violate lot of physical laws in machining.

CHAPTER VI

OPTIMIZATION

For a specific machining operation on a workpiece using a particular machine tool and cutting tool a realistic goal would be maximization of the metal removal rate. Assuming a constant depth of cut for a single pass in turning, the metal removal rate is given by,

$$\text{MRR} = F * S$$

Maximization of the metal removal rate, MRR is a realistic goal in metal cutting. A high MRR translates to a low cost of production and is therefore a very desirable goal

Some alternative process objectives could be:

- Minimizing cost
- Maximizing profit
- Maximizing tool life

The decision on the choice of most appropriate objective function depends on the particular processing job on hand, and other considerations such as batch size, dimensional accuracy, work piece geometry and material, etc.

For this research problem the choice of objective function was made on the productivity criterion, i.e., maximization of the MRR.

$$\begin{aligned} \text{MRR} &= (\text{Volume of metal removed})/\text{unit time} \\ &= (F.S.D.)/\text{unit time} \end{aligned}$$

In a single machining pass the depth of cut, D , remains constant. So,

$$\text{MRR} = 12 F S \text{ sq.in/min}$$

where,

F = Feed in inch/rev

S = Cutting Speed in ft/min

F and S are the design variables of the optimization problem. The aim of the local optimization problem will be to determine optimum values of F and S in order that their product is a maximum, subject to the limitations of the process itself, for a particular combination of machine-tool and cutting tool.

The problem would indeed be trivial if F and S could be made very high so that their product is high. This is not possible owing to physical limitations of the machine tool and the cutting tool as well as quality restraints on the workpiece itself.

Some of these limitations are studied, and an attempt is made to represent them as suitable predictive models in terms of F and S so that a well-posed problem is identified.

With the objective function chosen as the MRR, the local optimization problem may be stated as (41)

$$\text{Maximize MRR} = J(F,S) = F*S$$

subject to a set of constraints:

$$g_i(F,S) \leq 0 \quad i = 1,2,\dots,n$$

Each constraint represents a specific physical limitation of the cutting process. An example is given below, with more constraints discussed in the next section.

Lolaldze found (42) that the tensile stress on the tool rake face was found to increase with increase in feed rate.

$$\sigma = C * F \quad \text{or} \quad F = \sigma / C$$

where C is determined experimentally or by numerical analysis (43).

The constraint to avoid tool fracture may thus be modeled as,

$$C * F - \sigma_a < 0$$

where $\sigma_a = 100 \times 10^{-3}$ psi is the transverse rupture strength of the ceramic tool.

Essentially, the optimization problem in this paper has been formulated to construct reasonably close predictive models of the different process factors affecting the key variables of feed and cutting speed. Indeed, due to interplay between all these influential factors, and even others; e.g., secondary shear because of interface friction

at the base of the chip, which have been accounted for, it becomes quite difficult to establish feed and speed charts that could apply in all cases. Nevertheless several empirical and semi-empirical relationships and tables have been developed over the course of several years which may be used in conjunction with each other to define a feasible region of operation. It would be best, however, to run individual machining tests with state-of-the-art on-line sensors (44) for monitoring tool condition and part measuring on a part that is to be mass produced, together with a sensitivity analysis of the optimization problem. Optimum production, that is, higher metal removal rates with reasonable tool wear can be obtained only by tempering the theoretical approach to the limitation of the existing shop conditions.

Keeping in mind the above considerations, the global objective function J_{\max} takes the following form:

$$J_{\max} = \max_{\text{Oper,MT,CT}} \{ \max_{F,S} J(F,S;\text{Oper,MT,CT}) \}$$

subject to the appropriate constraints. The above formula represents the global optimization of local optima.

A sequential quadratic algorithm is employed for the local optimization of the process variables (45). The program is written in Fortran and has the objective function

and constraints in different subroutines. While the main program and the objective function subroutines are compiled only once, the constraints subroutine changes for each different combination of workpiece, machine tool and cutting tool.

At first a dry run of the optimization program was made with an Augmented Lagrangian algorithm. However, the final choice was made in favor of the SQP algorithm (45), due to its greater reliability and robustness. The fact that the SQP algorithm can handle up to 30 constraints and 30 variables, also contributed to this decision, thereby permitting ample freedom in including more constraints and/or variables, as may be found necessary in course of its use; when PRODUCER is applied to processes such as casting, forging, etc.

Every set of constraints generated by Prolog is passed on to the Fortran program as a subroutine, compiled and linked with the object codes of the existing main program and objective function subroutines to produce an executable. The output is then redirected to a file, which is thereafter consulted by Prolog to read off the optimized values of the process control variables, feed and cutting speed.

The intercommunication between Prolog and Fortran is quite convenient, in the C-Prolog version 1.5 (46). A built-in predicate "system" is provided that calls the

operating system and thereby gains access to the UNIX command mode, from within Prolog.

THE CONSTRAINTS

Torque Constraint

$$g_1 : T_c - T_{\max} \leq 0$$

The actual cutting torque must be less than the maximum torque, T_{\max} , the machine is capable of. T_c is given as:
(47)

$$T_c = h_1 F_m S^{-m} D^{1+m} d^n$$

where,

$$m = 0.7$$

D = work piece diameter, in ft.

$$n = 1.0$$

d = depth of cut, in inches

$$h_1 = 115045922$$

Power Constraint

$$g_2 : P_c - P_{\max} \leq 0$$

The power required in cutting cannot exceed, the power available from the machine. P_c is given by (47)

$$P_c = h_2 F_m S^{1-m} D^m d^n$$

where,

$$h_2 = 6904.28$$

and m , D and d as above.

Power Constraint (by cutting force)

$$g_3 : P_c - P_{max} \leq 0$$

f_c , the primary cutting force, acting in the direction of cutting speed, S , is generally the largest force and accounts for 99% of the power required by the process.

so,

$$P_c = f_c S/33000$$

where, value of f_c is picked up from the database for the particular tool-workpiece combination.

P_c & P_{max} are in h.p.

Power Constraint (by compounded parametric effect)

$$g_4 : P_c - P_{max} \leq 0$$

In this case the cutting power is calculated as:

$$\frac{\tau_s dw \cos(\theta - \alpha) S}{\sin \phi \cos(\phi + \theta - \alpha)}$$

where, α is the cutting tool back rake angle and θ is the friction angle given by,

$$\theta = \tan^{-1} \mu$$

$$\mu = \frac{f_T + f_C \tan \alpha}{f_C - f_T \tan \alpha}$$

Values of cutting and thrust forces, f_C and f_T , are given by conservative empirical estimates for given operating ranges, and placed in the VDB of the AI system.

The force system in orthogonal metal cutting is shown in Figure 4. Figure 5 indicates relevant variables and parameters involved in turning.

As per Kronenberg (48) the Shear angle ϕ , is given by:

$$\phi = 45 + \alpha/2 - \theta(0.75 + 0.005\alpha)$$

The Shear stress along shear plane (Figure 2) is given by:

$$\tau_s = \frac{f_C \sin \phi \cos \phi - f_T \sin^2 \phi}{dw}$$

With the above parameters determined, the power constraint as a function of S is of the form:

$$KS - P_{\max} \leq 0$$

where, $P_c = KS$

and

$$K = \frac{\tau_s dw \cos(\theta - \alpha)}{\sin\phi \cos(\phi + \theta - \alpha)}$$

Tool Hardness Constraint

$$g_5 : S - S_{\max} < 0$$

where, $S_{\max} = 500 \text{ ft/min}$

For a sintered carbide cutting tool, when the tool/chip interface temperature rises to 1800° F the hardness falls below $R_c 75 (49)$. This approximately corresponds to a cutting speed of about 900 ft/min . However, at a temperature of 1800° F the low-carbon steel workpiece is prone to damage, so the cutting speed is actually kept limited to 500 ft/min .

Temperature Constraint

$$g_6 : T_c - T_{\text{allowable}} \leq 0$$

Nearly all (99%) of the energy available at the cutting edge of a vibration-free metal cutting process is converted into heat (50). The generated heat is directly proportional to the product of feed and cutting speed. It was experimentally demonstrated by Brackenburg and Meyer (51) as early as 1911 that a major portion of this heat (about 70%) is carried away by the chip. It was later found that this proportion increased with higher metal removal rates. The temperature rise in the body of the chip is given by:

$$T_c = \frac{0.88 Q_s}{p_c S F d}$$

An average of 88% of Q_s , the heat generated in the shear plane, is considered reasonable for the higher MRR values attainable by ceramic and carbide tools.

Now,

$$Q = Q_s - Q_f$$

where,

Q_f = heat due to chip sliding over the tool

As per Boothroyd's method (52) of calculating approximate temperature rise in chip from measurement of tool forces,

$$Q_s = S/J (f_c - f_T \tan \phi)$$

thus,

$$T_c = \frac{0.88 (f_c - f_T \tan \phi)}{J p c F d}$$

Values of the different parameters are retrieved from the VDB.

$$T_{\text{allowable}} = 1200^\circ \text{ F}$$

1200°F is the suggested maximum temperature, at the cutting point, for a carbide tool cutting mild steel. If the temperature is permitted to exceed this value, both workpiece and tool would be damaged. Since this would in turn upset the cutting force balance, it could well lead to spindle motor overloading and possible burnout.

Lower Bounds on Design Variables

$$g_7 : S > 300 \text{ ft/min}$$

For cutting relatively soft metals, speed should not be allowed to drop below a certain minimum in order to avoid "built up edges" on the tool rake face. Also when the cutting speed is low -- the tool is more likely to chatter.

Another temperature constraint was modeled, on the lines of g_6 , on the basis of shear specific energy with reference to other cutting temperature investigators, Trigger and Chao (53) and Jaegar (54), where the shear plane temperature is given by

$$T_s = \frac{u_s}{J p c \left[1 + \frac{1.33}{12S} \left(\frac{K v_s}{d \sin \phi} \right)^{1/2} \right]} + T_0$$

where,

T_0 = Ambient work piece temperature
 Shear specific energy,

$$u_s = \frac{T_c \cos \alpha}{\sin \phi \cos (\phi - \alpha)}$$

K = Thermal diffusivity, $k/p c$

k = Thermal conductivity

$$v_s = \text{Shear velocity} = \frac{12 S \cos \alpha}{\cos (\phi - \alpha)}$$

However this constraint proved to be over-restrictive, for the class of workpiece and cutting tool materials taken for this study, and was thus dropped from consideration.

Work by other temperature investigators like Loewen and Shaw (55), Holzer and Wright (56) and many others were studied, but not included for the modeling.

A surface finish constraint that imposes a restriction of the desired surface finish (or roughness height) of the workpiece, in terms of micro-inches or microns, can easily be modeled. Turning operations can typically provide a surface finish ranging from about 16 micro-inches to 250 micro-inches (49).

When a bar is turned by a single point tool, feed marks form on the surface, spiralling around as the tool moves axially at a given feed rate. Thus, a predictive model of the constraint can be constructed as a function of feed rate and tool geometry. The actual form of the model would, however, depend on nose-radius (or absence thereof) of the tool.

Thus, $h_v - h_{vs} \leq 0$ where h_{vs} is say, 120 micro-inches. For sharp-point tool, $h_v = F/(\cot y_f + \tan y_s)$, and y_f = front cutting edge angle, y_s = side cutting edge angle.

However, for this optimization problem, which seeks to maximize the MRR, the surface constraint is considered to be overrestrictive. During a dry run of the optimization program, independent of the AI module, inclusion of the

surface finish constraint, resulted in an empty feasible space. In fact, a high grade surface finish requires a very low feed, whereas the temperature constraint compliance demands a feed on the high side. Generally speaking, the temperature constraint is considered more important. As such, if a good surface finish is desired, then a final grinding operation is more advisable.

Constraints pertaining to workpiece and/or cutting tool size and shape, machine tool workholding, etc., are accounted for in the pre-optimization stage when processing feasibility is decided. Rules enabling this are built into the procedural knowledge base of the A.I. module -- and, as such, is identifiable at the higher levels of the decision tree.

Automatic Constraint Generation

The constraints are automatically generated by the A.I. production system. The first part of the program module generates the feasible combination of machine tool and cutting tool for a certain workpiece. The second module retrieves the necessary data, pertaining to the respective machine tool and cutting tool from the database, then computes and puts together a set of constraints for each machine, tool and workpiece combination. These constraints are then automatically written into a file, in the syntax understood by the Fortran optimization program.

A brief excerpt from the A.I. program shows the generation of the power constraint based on Bedini and Pinotti's work (47).

```
find_constraints(WP,MT,CT):-tell (constraints_file),...,
    printstring("G(1) = 115045922*(X(1)**0.7)*
    (X(2)**(-0.7))*",tab(1),
    product_term(WP,A), write(A), tab(1),
    max_torque_lbft(MT, T_max), write(-T_max),..., told.
```

In the above procedure, the file "constraint_file" is opened and then closed by the predicate "told", after all constraints have been generated for the instantiated workpiece, machine tool and cutting tool combination. The product_term A was calculated using the predicate

```
product_term(WP,A):- depth_of_cut(WP,Dcut)
bar_dia(WP,D), A is (Dcut/12.0)*D 0.7.
```

The contents of the generated file as it was created for a cylindrical workpiece of diameter = 1 ft, depth of cut = 0.25" and $P_{max} = 17$ h.p., $T_{max} = 203$ lbft using a carbide tool are given in Figure 6. The optimum process conditions were returned as $F = 0.82E-3$ in/rev, $S = 500$ ft/min.

CHAPTER VII

INTELLIGENT ADAPTIVE CONTROL

The work presented in this thesis was actually prompted by a need to alleviate some of the problems found in Adaptive Control systems in the metal cutting industry. Even in attempting to do this, hopefully a significant step will have been taken towards the greater goal of at least a quasi-ideal Computer Integrated Manufacturing system for the production of machined components.

It is therefore considered fit and proper to include in this report, some of the impressions and analyses of published research work in the area of Adaptive Control in machining. These studies led to the idea of and need for developing an Expert System, which would perform optimization in a global manner, based on a certain criterion or objective function. Thereby, constraint activity is known before-hand and the A.C. function "knows" which parameters are critical. This permits it to be more efficient in steady-state conditions. Another advantage from this is that AC-capability extension to dynamic conditions, to account for tool-wear, for example, becomes less problematic.

Adaptive Control is essentially a computerized system that enables a machine tool -- or any plant equipment for

that matter -- to perform its functions with no (at least ideally speaking) human intervention. To quote Landau (57): "An adaptive system measures a certain index of performance (IP) using the inputs, the states, and the outputs of the adjustable system. From a comparison of the measured I.P. and a set of given ones, the adaptation mechanism modifies the parameter of the adjustable system or generates an auxiliary input in order to maintain the IP close to the set of given ones." Clearly, therefore, the adaptive system must have two distinct features:

1. Closed Loop Control
2. Optimization in some form

Accordingly, this presupposes a full theoretical understanding of the process to the extent needed to develop the mathematical models, that can be programmed into the computer of the AC system. The basic elements effecting the adaptive control are:

1. Measurement(s) from process output
2. Decisions, i.e., optimization within the computer
3. Modification, i.e., "adjusting" signal to the process controller to alter inputs.

This level of automation has been readily achieved in the realm of continuous processes, e.g., oil refineries, where theories of fluid mechanics and heat transfer are better understood and parameters easily measured. The metal cutting process being less well understood and its different

parameters (many in number) being less conveniently measurable, fully successful implementation of AC has not yet been possible (68, 70) in the metal removal or even the metal forming industry.

AC of machine tools can be classed into two groups:

1. Adaptive Control of Optimization (ACO)

This uses some I.P. of machining, e.g., minimum cost, as a control criterion subject to constraint modeled from Taylor's tool life equation (or its modified version) for a particular tool-workpiece combination, to optimize feed and cutting speed (57-61).

2. Adaptive Control of Constraints (ACC)

This attempts to maintain the operation within a feasible region, bounded by constraints representing the physical limitation of the process (62-67).

Investigations (68, 69) indicate problems with either approach. In the case of ACO, Taylor's tool life equation does not suitably represent all the physical realities of the cutting process. Besides, a huge database is also called for. For the ACC approach, on the other hand, there is always a conflict as to which parameters are more critical. This is due to an incomplete knowledge of the interplay between the various parameters, e.g., cutting forces, tool geometry, tool tip temperature, shear angle, etc., and their individual or combined influence on the cutting process. Thus, work in this area hithertofore has

dealt with either machine tool constraints (47) or with material constraints, relating to physical realities of cutting for a specific tool-workpiece combination (71).

Thus, a global approach to the Adaptive Control problem in metal cutting, is an yet unresolved issue.

The work in this thesis has attempted to take this global approach. Accordingly, an Expert System was developed to tackle the problem of optimization. several constraints have been modeled to represent different aspects of the cutting process. In a few cases, the same constraint, e.g., Power consumption, has been modeled from more than one approach, in order to be conservative. The optimized cutting speed and feed, from this off-line optimization scheme, are simultaneously furnished as data and reference input respectively to the NC and AC functions of the machine tool. The feed and cutting speed are included as data for the part-program of the NC function, thereby ensuring optimum operation. The AC function, using appropriate displacement and speed transducers for feed and cutting speed respectively, can compare them with the reference feed and cutting speeds to maintain operation at or near the optimum point. A schematic showing this arrangement can be seen in Figure 7.

Since the optimization analysis points out the active constraints the AC function is not handicapped, as in previous ACC approaches. In fact it becomes intelligent,

because now it can concentrate only on those process parameters that would be most likely to destabilize, i.e., those of the active constraint models.

This implies information passing from the expert system "PRODUCER" to the dedicated microcomputer performing the A.C. function. The information passed will consist of the identity of the active constraints and the limiting value of the particular constraint parameters, i.e., Force and Power, or Temperature and Power, etc. The on-line A.C. function can then do an Adaptive Control of Optimization -- with the same objective function, maximizing the MRR. However, this A.C. function will be more efficient, since it needs to contend with only two constraints -- the active ones!

Thus, the optimized feed and cutting speed values from the off-line optimization, performed by PRODUCER, become starting values for controlling the turning operation of the particular workpiece, on a given machine tool using a given cutting tool. Also, from identification of the active constraints and knowledge of limiting values of the critical parameters, intelligent real-time adaptive control of optimization (ACO) becomes possible.

A major advantage of the off-line Expert System is its open-endedness. Since this is a subject of continuing research, additional constraints will doubtless be modeled to supplement, or even replace the current ones. There is also an implicit learning property in this system. Adding

or eliminating constraints would pose no problem at all. The system architecture as well as the robustness and capacity of the optimization algorithm ensures this. Further, the need for a large data base felt by past researchers in the ACO approach, is also satisfied by this Expert System.

This present research also presumes that the burden of the optimization being with an off-line system, the AC function becomes more effective in real-time control of the machining process -- its *raison d'etre*. This presumption should be borne out by subsequent research and experimental verification.

Scrutiny of Figure 7 raises an important question. What about the effectiveness of the AC for non-steady-state situations? The single most critical factor responsible for deterioration in the cutting process is tool wear. There are some very good commercially available tool wear sensors (44). Thus when tool wear is detected, the N.C. function should be signaled to effect a tool change. This is more advisable than generating an auxiliary input, that reduces cutting speed and/or feed to prevent further abuse of the same cutting tool, at the cost of productivity.

For better overall control of the cutting process it is, however, prudent to monitor as many parameters/variables as possible. Such a data acquisition system would require state-of-the-art sensors, with very good signal-to-noise

ratio. Finally, another Expert System would have to be designed for evaluating sensed signals and directing appropriate correctional commands to the machine control unit. There is substantial scope for future research in this area.

CHAPTER VIII

CONCLUSION

The basic framework of an Expert System, PRODUCER, that selects an appropriate process, for producing a discrete part, and optimizes that process thereafter, has been constructed. Its capability with one process -- machining -- has been demonstrated from selection of the process-family, through identification of the specific operation -- turning -- and finally its optimization.

The optimized process parameters for turning the given component, also serve as a useful input for the Adaptive Control System, employed by the machine tool actually performing the operation.

The main features of PRODUCER are given below:

1. It is a rule-based production system, that uses backward chaining, and is implemented in Prolog.
2. The system architecture affords easy inclusion of additional production rules or other data.
3. Knowledge representation in this depth-first search system, is such that fruitless search is avoided by applying restrictions at each level.
4. It has a hybrid character, in that it combines A.I. and algorithmic techniques, in providing the final solution.

Thus process selection and constraint generation, is automatically done by A.I. The actual optimization is carried out by a Fortran program, which is "run" by A.I.

5. Has a robust optimization algorithm that can handle up to 30 variables and 30 constraints.
6. The modular construction allows great convenience in deletion or replacement of any "knowledge" or "data" that is used by the system, including the constraints. This implies an in-built "learning" capability. Accordingly, whenever it is determined that a certain constraint can be better modeled, or a more important one should be included, these changes can easily be effected.
7. Permits information retrieval at any level of the hierarchical structure.
8. Enables intelligent adaptive control.

The thesis also refers to previous and current research in the area of computerized process planning, and points out differences between them and the present work. Short theoretical backgrounds of A.I. and Expert Systems, in general, are also given.

Subsequent research beyond this thesis may be conducted towards extending PRODUCER's capabilities for a wider range of machined parts as well as cast, forged or welded items.

At present part attributes are described to PRODUCER, by the user as "facts" in the VDB. It would be very interesting to examine the possibilities of substituting this with a suitable CAD interface. Constructive Solid Geometry techniques -- whereby any component maybe thought of as a combination of primitives could serve as a good start. Thus, a linear extrusion of a 2-D figure such as a circle or square, along a direction orthogonal to its plane, produces a cylinder and cubic prism respectively. This approach, would, however, be unable to account for anything but the major part features. On the other hand, the macroscopic features may help in deciding at least the basic operations, e.g., turning or boring vs. shaping, planning or broaching.

It should be attempted to integrate PRODUCER with an existing CAM system, to determine its effectiveness and identify areas of improvement.

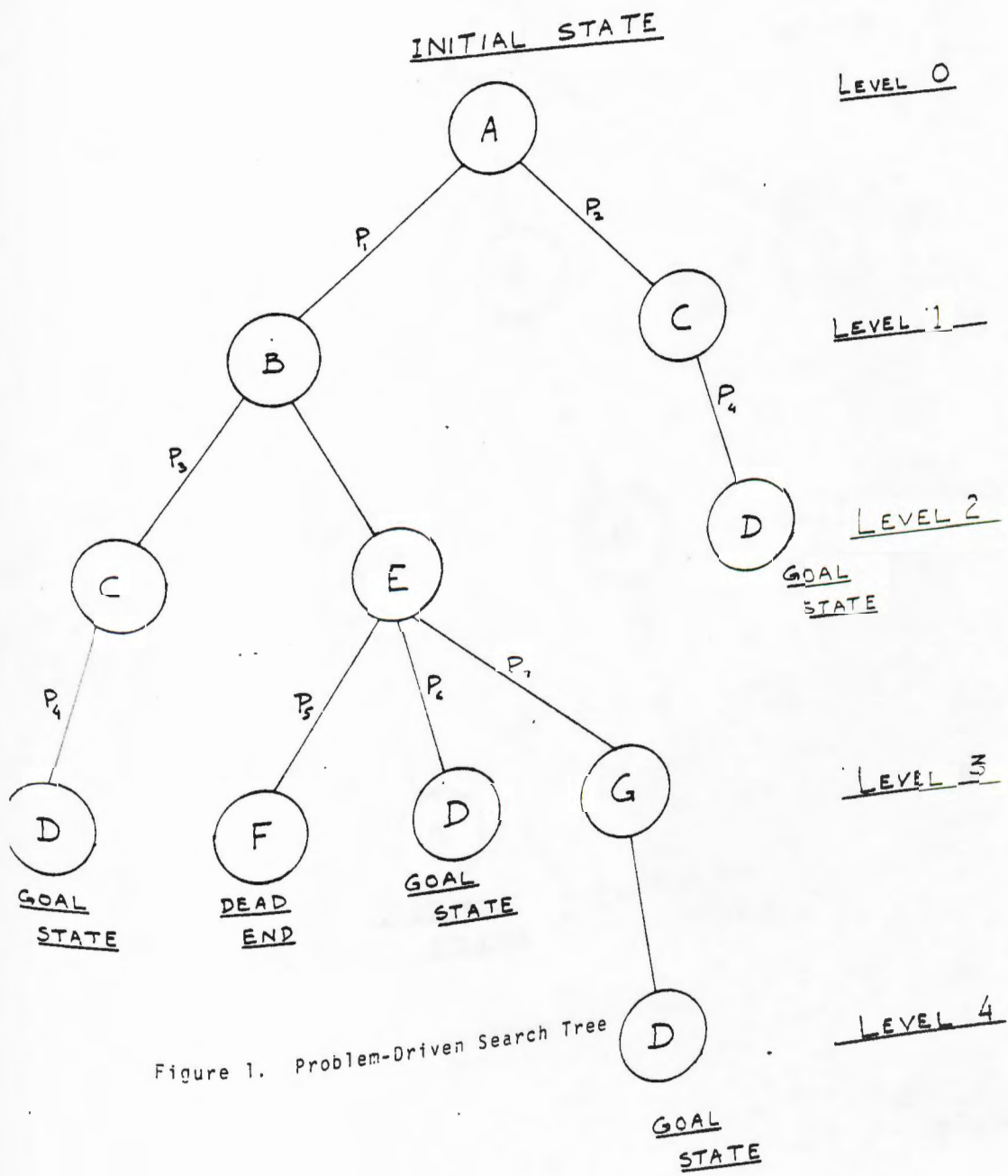


Figure 1. Problem-Driven Search Tree

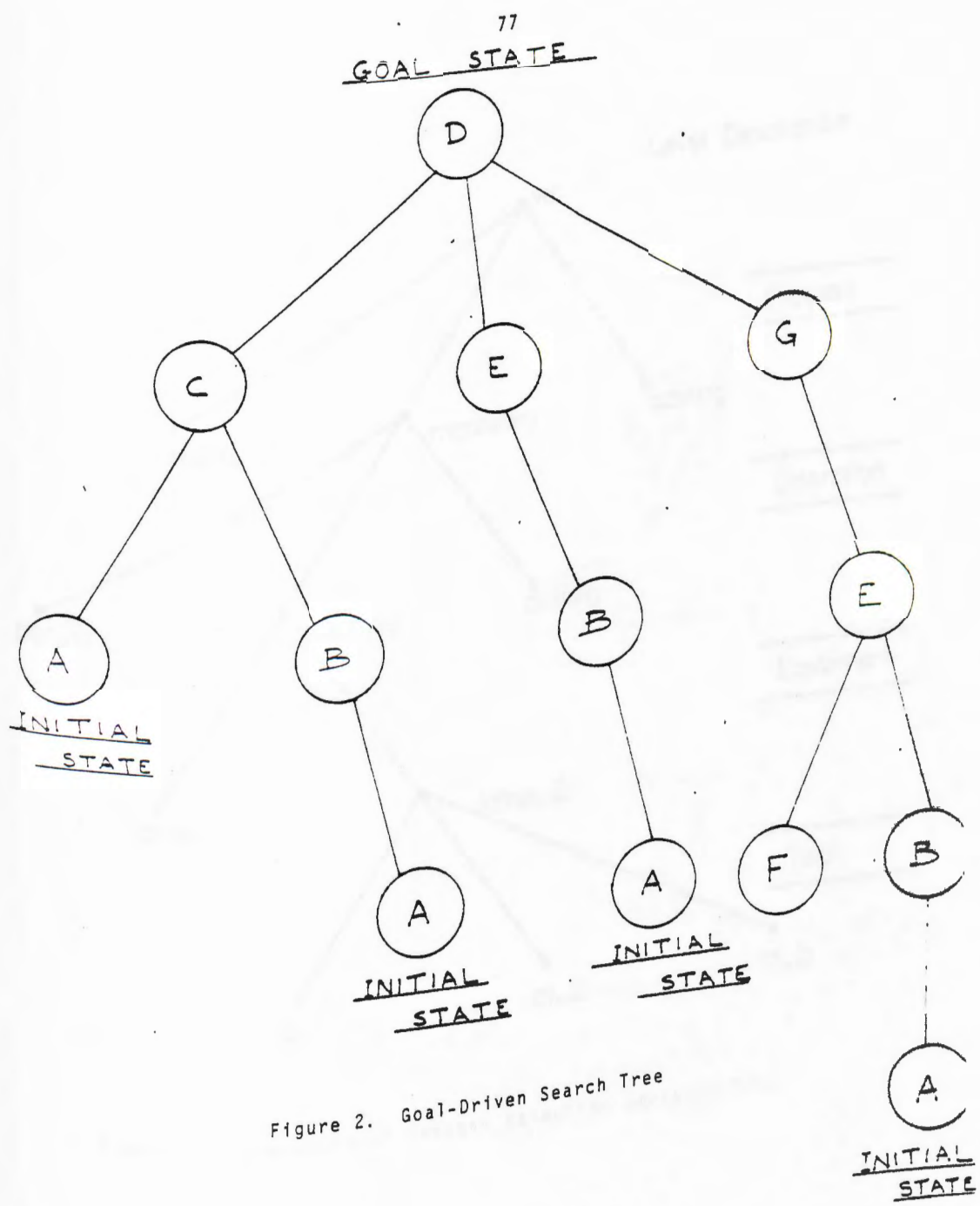


Figure 2. Goal-Driven Search Tree

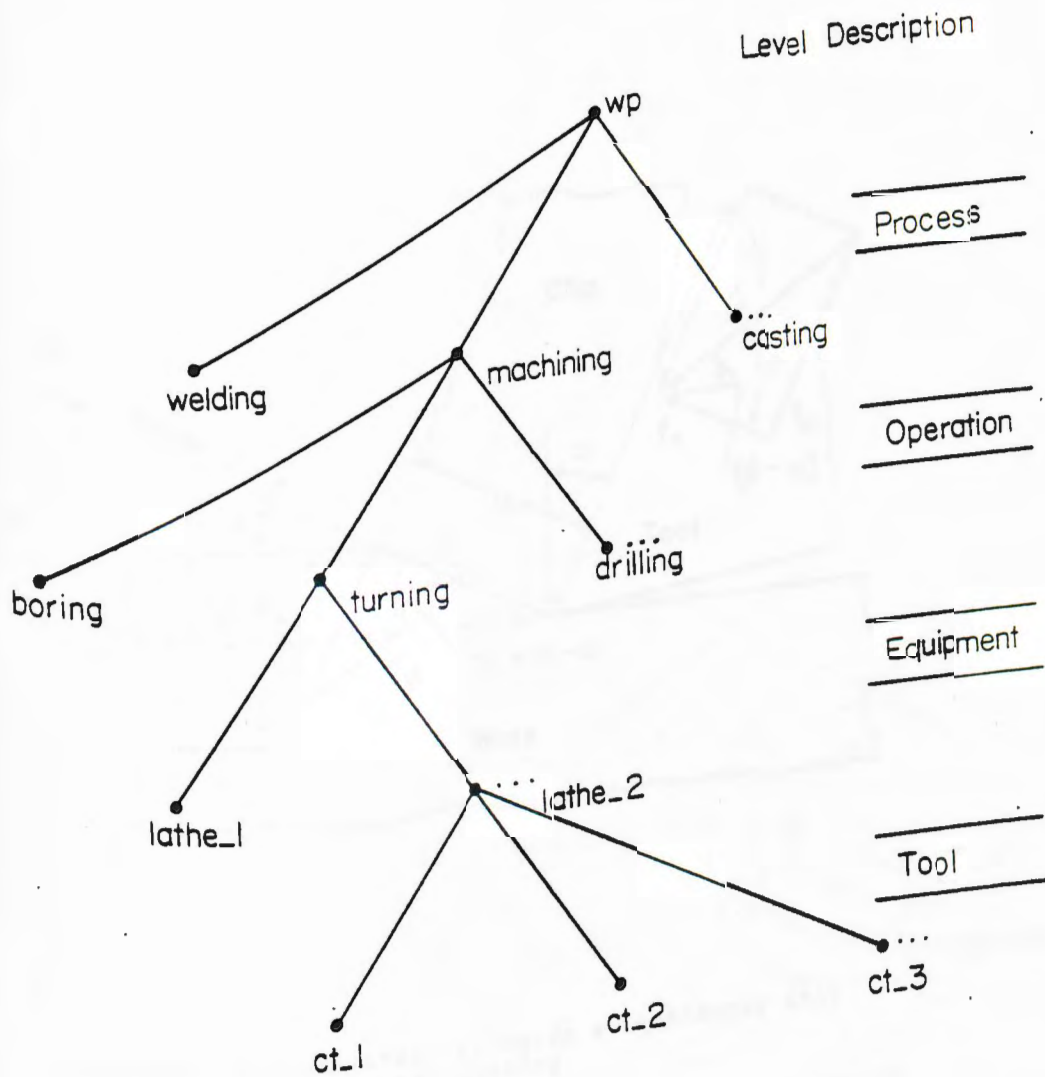


Figure 3. Hierarchical Process Selection Decision Tree

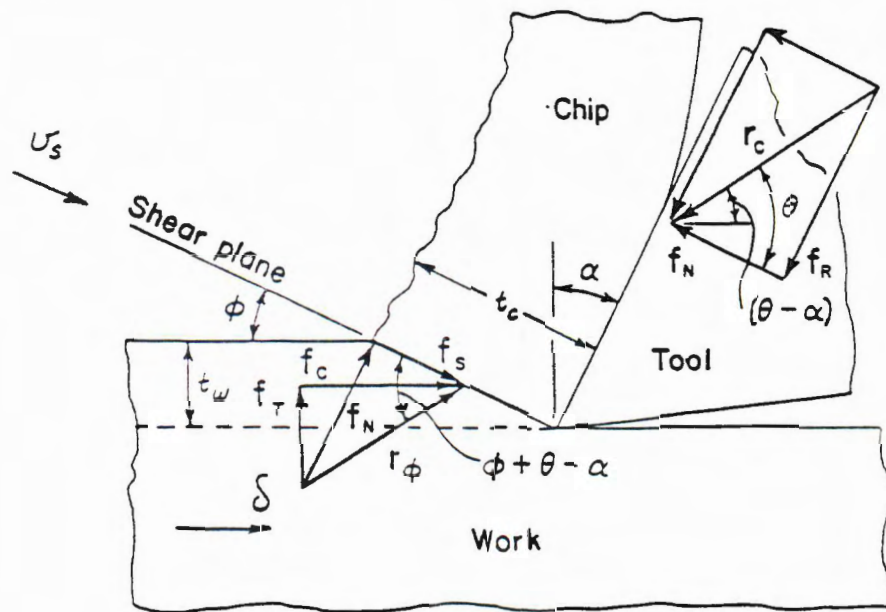


Figure 4. Force System Acting on a Continuous Chip in Orthogonal Cutting

80

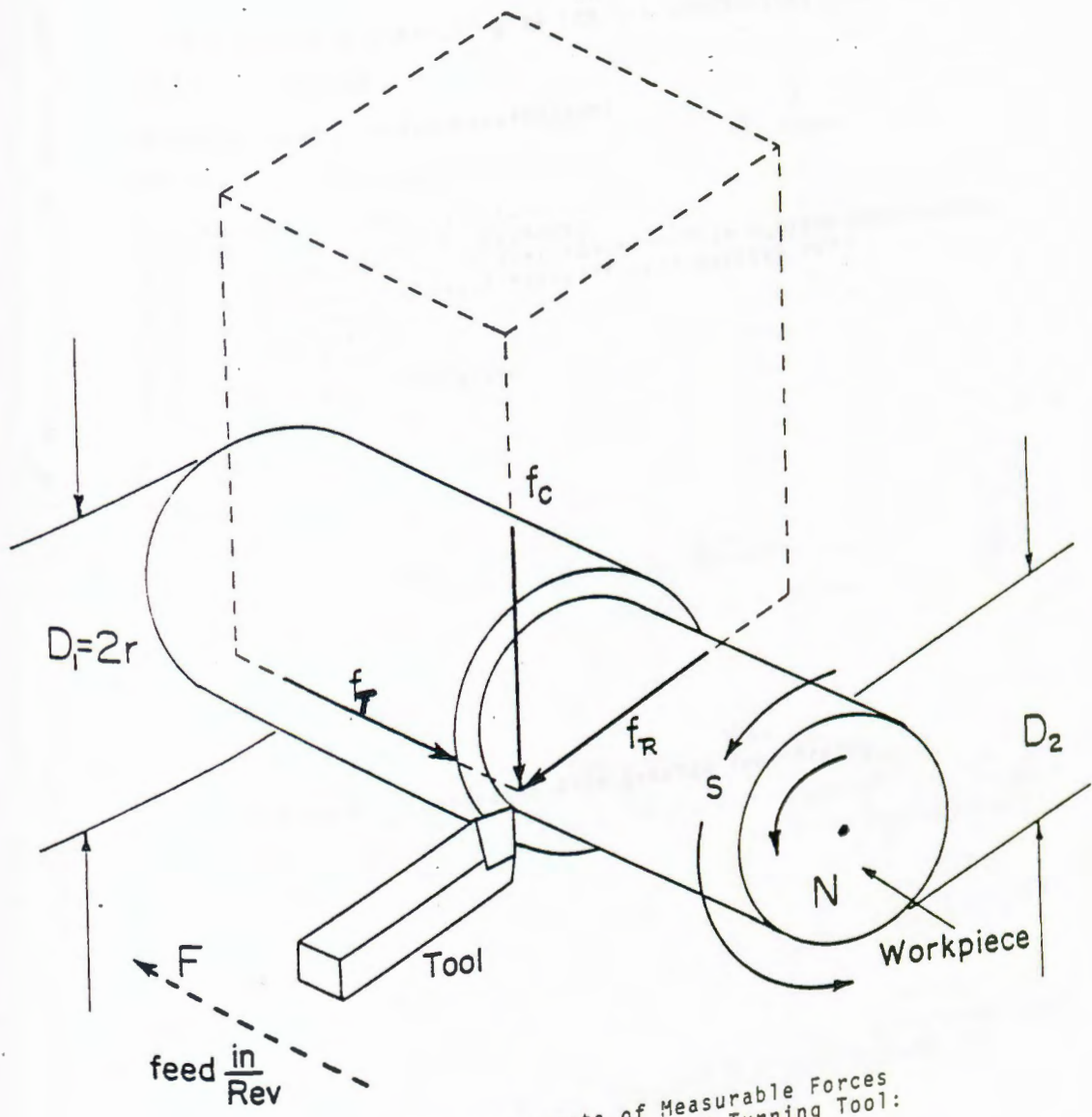


Figure 5. Three Components of Measurable Forces Acting on a Single-point Turning Tool: Orthogonal Machining

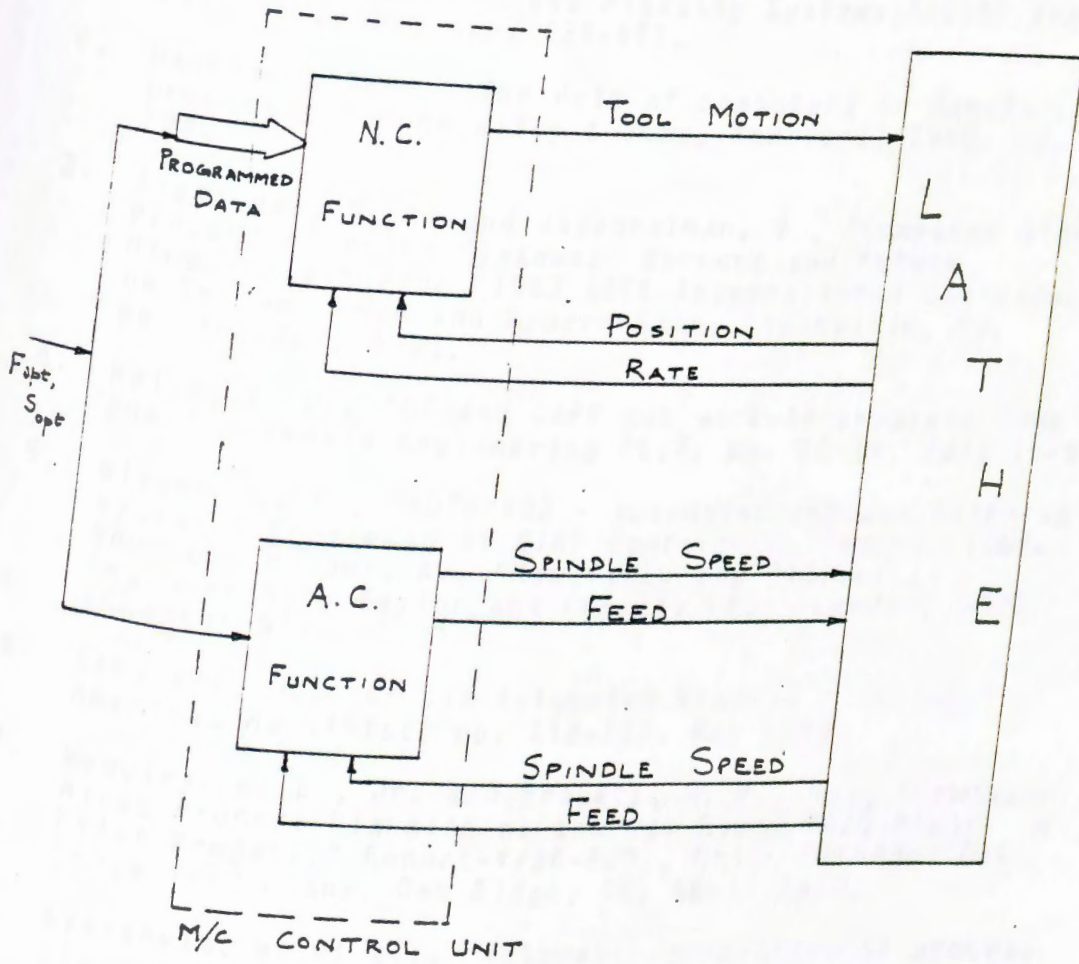


Figure 7. Machine Tool N.C. and A.C. Function Using Optimized Process Parameters

BIBLIOGRAPHY

1. Weill, R., Spur, G. and Eversheim, W., "Survey of Computer Aided Process Planning Systems," CIRP Annals, Vol. 31/2/1982, pp. 539-551.
2. Halevi, Gideon, "The role of computers in manufacturing processes," John Wiley & Sons, New York, 1980, pp. 289-294.
3. Alexander, S. M. and Jagahnathan, V., "Computer Aided Process Planning Systems: Current and Future Directions," Proc. 1983 IEEE International Conference on Systems, Man and Cybernetics, Louisville, Ky, pp. 462-465, 1983.
4. Holtz, R. D., "GT and CAPP cut work-in-progress Time 80%," Assembly Engineering 21,7, pp. 16-19, July 1978.
5. Nissen, C. K., "AUTOPROS - automated process planning system," Presented at CIRP Conference, Geneva, 1969. Source: Hitomi, K., "Manufacturing Systems Engineering", Taylor and Francis Ltd., London, 1979 (Chapter 5).
6. Schaffer, G., "GT via Automated Process Planning," American Machinist, pp. 119-122, May 1980.
7. Hewgley, R. E., Jr. and Prewett, H. P., Jr., "Computer Aided Process Planning at the Oak Ridge Y-12 Plant: A Pilot Project," Report-Y/DA-8297, Union Carbide, Oak Ridge Y-12 Plant, Oak Ridge, TN, April 1979.
8. Eversheim, W. et al., "Automatic generation of process plans and NC tapes for sheet metal parts," The Production Engineer, March 1981.
9. Eversheim, W. and Holz, B., "Computer-aided programming of NC-machine tools by using the system AUTAP-NC," CIRP Annals, Vol. 31/1/1982.
10. Mann, W. S., Jr. et al., "Computerized Production Process Planning," Report R77-942625-14, United Technologies Research Center, Nov. 1977.
11. Wysk, R. A., "Automatic Process Planning and selection-APPAS," Ph.D. Thesis, Purdue University, 1977.

12. Chang, Tien-Chien and Wysk, Richard, A., "An integrated CAD/Automated Process Planning System," AIIE Transactions, vol. 13, no. 3, September 1981, pp. 223-233.
13. Chang, Tien-Chien, Wysk, R. A. and Davies, R. P., "Interfacing CAD and CAM -- A study in hole design," Computers and Industrial Engineering, vol. 6, no. 2, 1982, pp. 91-102.
14. Chang, Tien-Chien, "TIPPS -- A Totally Integrated Process Planning System," Ph.D. Thesis, Virginia Polytechnic Institute and State University, 1982.
15. Horvath, M., "Semi-generative process planning for part manufacturing," SME Technical Paper, SER. MS79-158.
16. Nau, D. S. and Chang, T. C., "Prospects for Process Selection Using Artificial Intelligence," Computers in Industry 4, pp. 253-263, 1983.
17. Descotte, Y. and Latombe, J. C., "GARI: A Problem Solver that Plans How to Machine Mechanical Parts," Proc. Seventh International Joint Conf. Artif. Intel., August 1981.
18. Descotte, Y. and Latombe, J. C., "Making Compromises among Antagonist Constraints in a Planner," Artificial Intelligence 27 (1985), pp. 183-217.
19. Ssemakula, M. E., "Further development of a computer-aided process planning system and its integration with NC part programming," Ph.D. Thesis, UMIST, Manchester, U.K., 1984.
20. Nau, D. S., SIPP Reference Manual, Tech. Report 1515, Computer Science Department, University of Maryland, 1985.
21. Fox, M. S., Allen, B. P. et al., "ISIS: A constraint-directed Reasoning Approach to Job Shop Scheduling," Tech. Report CMU-RI-TR-83-8, Carnegie-Mellon University, Pittsburgh, PA, June 1983.
22. Eskicioglu, Hakki, "An interactive computer aided process planning system for non-rotational parts," Ph.D. Thesis, UMIST, Manchester, U.K., 1981.
23. Arbib, M. A., "Brains, Machines and Mathematics," McGraw-Hill, New York, 1964.

24. Barr, A. & Feigenbaum, E., "Handbook of Artificial Intelligence," Three volumes, W. Kaufman Inc., 1981-82.
25. Nilsson, Nils J., "Principles of Artificial Intelligence," Tioga Publishing Co., California, 1980.
26. Post, E., "Formal reductions of the general combinatorial problem," American Journal of Mathematics, vol. 65, pp. 197-268.
27. Rychener, M. D., "Production Systems as a Programming Language for A.I. applications," Ph.D. dissertation, Department of Computer Science, CMU, 1976.
28. U.S. Department of Commerce, National Bureau of Standards, "An Overview of Expert Systems," NBSIR 82-2505, May 1982.
29. Feigenbaum, E. A., "Knowledge Engineering for the 1980's," Computer Science Department, Stanford University, 1982.
30. Duda, R. O., "Knowledge-Based Expert Systems Come of Age," Byte, Vol. 6, No. 9, September 1981, pp. 238-281.
31. Hayes-Roth, F., Waterman, D. A et al., "Building expert systems," Addison-Wesley, Reading, Massachusetts, 1983.
32. Buchanan, B. G and Shortliffe, E. H., "Rule-based expert systems," Addison-Wesley, Reading, Massachusetts, 1984.
33. Allwood, R. J., Stewart, D. J. et al., "Interim report on expert system shells evaluation for construction industry applications," Loughborough University, U.K., 1984.
34. Rychener, M. D., "Expert Systems for engineering design," Expert Systems, Vol. 2, No. 1, pp. 30-44, 1985.
35. Intellicorp, "Representation, reasoning and explanation," 707 Laurel Street, Menlo Park, CA 94025-3445, U.S.A.
36. Inference Corporation, "ART programming primer," 5300 W. Century Boulevard, 3rd Floor, Los Angeles, CA 90045, U.S.A., 1984.

37. Roseman, M. A. and Gero, J. S., "Design Codes as Expert Systems," Computer Aided Design, Vol. 17, No. 9, pp. 399-409, November 1985.
38. Charniak, E., and McDermott, D., "Introduction to Artificial Intelligence," Addison-Wesley Publishing Co., Reading, Massachusetts, 1984.
39. Chang, C. L. and Lee, R. C. T., "Symbolic Logic and Mechanical Theorem Proving," Academic Press, 1973.
40. Clocksin, W. F. and Mellish, C. S., "Programming in Prolog," Springer-Verlag, Berlin, Heidelberg, New York, 1981.
41. Reklaitis, G. V. et al., "Engineering Optimization Methods and Applications," John Wiley & Sons, Inc., New York, 1983.
42. Lolaldze, T. N., "Nature of Brittle Failure of Cutting Tools," Annals of the C.I.R.P., Vol. 24, No. 1, pp. 13-16, 1975.
43. Tlusty, J. Y. and Masoud, Z., "Chipping and Breakage of Carbide Tools," ASME Journal of Engineering for Industry, Vol. 100, No. 4, pp. 403-412, 1978.
44. Livingston, D., "Sensors guide the way for machine tools," High Technology, p. 49, June 1985.
45. Powell, M. J. D., "A fast algorithm for non-linearly constrained optimization calculations," Proceedings of Dundee Conference on Numerical Analysis, 1977.
46. C-Prolog Users Manual, Version 1.5, SRI International, Menlo park, California, November 1985.
47. Bedini, R. and Pinotti, P. C., "Experiments on Adaptive Constrained Control of a CNC Lathe," Journal of Engineering for Industry, Vol. 104, pp. 139-149, May 1982.
48. Kronenburg, M., "A new approach to some relationships in the theory of metal cutting," Tech. paper American Society of Tool Engineers, paper no. 86, Volume 58, pp. 1-8, May 1958.
49. Degarmo, E. Paul et al., "Materials and Processes in Manufacturing," Macmillan Publishing Co., 1984.

50. Trent, E. M., "Metal Cutting," Chapter 4 & 5, Butterworths and Co., London, 1984.
51. Brackenburg and Meyer, "The heat generated in the process of cutting metal," Engineering, 13th January, 1911.
52. Boothroyd, G., "Fundamentals of Metal Machining and Machine Tools," McGraw-Hill, New York, 1975.
53. Trigger, K. J. and Chao, B. T., "An analytical evaluation of metal cutting temperatures," ASME Transactions, Vol. 1, pp. 57-68, January 1951.
54. Jaegar, "Moving sources of heat and temperature at sliding contacts," Proceedings of Royal Society of New South Wales, 76, pp. 203, 1946.
55. Loewen, F. G. and Shaw, M. C., ASME Transactions, Vol. 76, pp. 217, 1954.
56. Holzer, A. J. and Wright, P. K., "Material Science Engineering," ASME Proceedings, Vol. 51, pp. 81, 1981.
57. Landau, Y. D., "Adaptive Control: The Model Reference Approach," Marcel Dekker, New York, 1979.
58. Centner, R., "Final Report on Development of Adaptive Control Techniques for Numerically Controlled Milling Machines," Bendix Research Laboratory, Technical Documentary Report No. ML-TDR-64-279.
59. Davis, R. P., Wysk, R. A., et al., "Optimizing Machining Parameters in a Frame-work for Adaptive Computer Control," Computers and Industrial Engineering, Vol. 4, pp. 143-151, 1980.
60. Opitz, H., "Adaptive Control - Fundamental Principles for Numerical Optimization of Cutting Conditions," ASME Proc. of the International Conference on Manufacturing Technology, pp. 25-28, September 1967.
61. Yonetsu, S., Inasaki, I., and Kijima, T., "Optimization of Turning Operations," 5th North American Manufacturing Research Confce. Proc., pp. 17-23, 1978.
62. Jaeschke, J. R., Zimmerly, R. D., and Wu, S. M., "Automatic Cutting Tool Temperature Control," International Journal of Machine Tool Design and Research, Vol. 7, No. 4, pp. 465-475, Dec. 1967.

63. Koren, Y., and Masory, O., "Adaptive Control with Process Estimations," *Annals of the C.I.R.P.*, Vol. 30, No. 1, pp. 373-380, 1981.
64. Tlusty, J., and Elbestawi, M., "Constraints in Adaptive Control with Flexible End Mills," *Annals of CIRP*, Vol. 28, pp. 253-255, 1979.
65. DeFilippi, A., and Ippolito, R., "The influence of Constraints on Cutting Condition Optimizations," *Annals of the CIRP*, Vol. 24, pp. 417-421, 1975.
66. Cook, N. H., "Adaptive Machining Parameter Selection," 5th North American Metal Working Research Conference Proceedings, pp. 344-356, 1977.
67. Mathias, R. A., "Adaptive Control for the Eighties," Society of Manufacturing Engineers, Technical Report, M580-242, 1980.
68. Kaminskaya, V. V., et al., "Trends in the Development of Adaptive Control," *Machines and Tooling*, Vol. 45, No. 8., pp. 66-70, 1974.
69. Groover, M. P., "A Definition and Survey of Adaptive Control Machining," SME, Manufacturing/Numerical Control Systems, Technical Paper MS70-561, 1970.
70. Wick, C., "Automatic Adaptive Control of Machine Tools," *Manufacturing Engineering*, Sept., 1977, pp. 38-45.
71. Yen, D. W., and Wright, P. K., "Adaptive Control in Machining -- A New Approach Based on the Physical Constraints of Tool Wear Mechanisms," *Journal of Engineering for Industry*, Vol. 105, pp. 31-37, February, 1983.