

Technical Report: UM-MEAM-91-11

Computer-Aided Process Planning
For Parallel Machining

Debasish Dutta and Jordan B. Levin

Design Laboratory
Department of Mechanical Engineering & Applied Mechanics
The University of Michigan
Ann Arbor, MI 48109

0187

UMR0372

Computer-Aided Process Planning For Parallel Machining

Debasish Dutta and Jordan B. Levin

Design Laboratory

Department of Mechanical Engineering & Applied Mechanics

The University of Michigan

Ann Arbor, MI 48109

August 22, 1991

Abstract

In this paper, we extend the domain of computer-aided process planning to include a new and powerful class of machine tools. A unique feature that characterizes these machines tools is their ability to perform machining in *parallel*. In particular, multiple tools can simultaneously machine a single workpiece, and, multiple workpieces can be machined simultaneously. Such machines can completely finish a part in a single setup resulting in shorter cycle times, and can provide higher machining accuracy by drastically reducing the potential for tolerance buildup. We introduce and formalize a concept for parallel machines which can be extended to include machines more powerful and capable than the ones being used today. Next, we study the effects of parallelism on computer-aided process planning by identifying and examining some new issues that arise in the parallel machining domain. Finally, we propose and discuss a framework for a computer-aided process planning system in which these issues are incorporated.

1. INTRODUCTION

1.1 Process Planning for Sequential Machines

The concept of computer-integrated manufacturing (CIM) is frequently suggested as a means for improving manufacturing productivity. Computer-aided design (CAD) and computer-aided manufacturing (CAM) are certainly important elements in the overall computer-integrated manufacturing scheme. However, human intervention is still required to link CAD and CAM together in order to achieve effective performance. This vital link is commonly referred to as manufacturing process planning. In this paper, we will consider process planning as it applies to the manufacture of machined parts.

Process planning consists of the formulation of the "machining processes and parameters that are to be used to convert [machine] a workpiece from its initial form to a final form predetermined from an engineering drawing."¹ or design. In most cases, this essential step is performed by human process planners who examine the part drawing and:

- determine which operations are necessary to machine the part satisfactorily;
- choose the machines and tools that will perform the operations;
- establish the fixturing methods and setups necessary to machine the part;
- determine a sequence for the machining operations;
- specify the machining parameters for each operation.

In this planning process, the machines being considered are the conventional NC machines. Such machines process one workpiece at a time and, in general, are capable of employing one cutting tool at a time. We refer to these conventional machine tools collectively as *sequential machines*. Therefore, in a sequential machining environment, workpieces will, in general, travel from machine to machine until all required operations have been completed.

With industries emphasizing batch production runs to reduce inventory and achieve just-in-time delivery, the demand for qualified process planners has far outnumbered the supply [31]. Consequently, a great deal of research has been devoted to the development of computerized systems that can perform the task of process planning. The goal of such systems, commonly referred to as computer-aided process planning (CAPP) systems, is to bring fully automated CIM closer to reality.

In principle, computer-aided process planning in a sequential machining environment seeks to provide an automated method for accomplishing the five tasks previously mentioned. In order to perform these tasks, a CAPP system has to automatically carry out the following [16]:

- the extraction and recognition of the machinable features from the 3D representation of the part in the CAD data base;
- the determination of the material volumes to be machined, based on the raw stock size and the knowledge of the machinable features;

¹Chang and Wysk, An Introduction to Automated Process Planning Systems, 1985, p. 15.

- the reduction of these volumes into subvolumes, each of which can be performed in a single machining operation;
- the association of the subvolumes with machining operations and tools (i.e., process selection);
- the sequencing of machining operations;
- the determination of machining parameters for each operation (i.e., tool parameter selection);
- the generation of tool paths and numerical control machine code.

Various CAPP systems for sequential machining are in use today and we elaborate on the existing methodologies in Section 2. In the rest of this paper, the terms "CAPP" and "process planning" are used interchangeably to refer to computer-aided process planning.

1.2 Parallel Machines

While computational advances have led to the development of CAPP systems, significant hardware developments have resulted in more sophisticated machine tools. With the realization that the elimination of repeated workpiece setups can lead to a large reduction in part machining time, a new class of machine tools was introduced in the mid-1980's [23], [24]. These machines, similar to conventional 4-axis NC lathes, typically consist of a main turning spindle that holds the workpiece, and two or more turrets each containing several cutting tools. However, unlike the 4-axis NC lathes, the turrets on these new machines can also hold live (powered) tools. Apart from enabling drilling operations, tool motion can be synchronized with the spindle revolutions to allow complex milling operations as well. Therefore, these powerful next generation machines are, in essence, a functional combination of a conventional lathe and a milling machine. In addition, these machines usually possess a second spindle, or subspindle, which allows machining to be done on the rear face of the part. Furthermore, multiple turrets and multiple spindles can be operational simultaneously. The advantages of these machines are immediately obvious:

- (i) At any point in time, a workpiece can be approached and machined by more than one tool (on separate turrets);
- (ii) At any point in time, more than one workpiece can be held and machined by activating the main and subspindles simultaneously, and using the multiple turrets;
- (iii) Elimination of workpiece setups. The workpiece can now be *completely* machined, from raw stock to finished part, on the same machine. This eliminates the time that the part would normally have to wait in a queue to be loaded and set up on a different machine, and may also save factory floor space;
- (iv) An improvement in machining accuracy as the potential for tolerance buildup is significantly reduced with the elimination of setups.

We believe, the most noteworthy characteristic of these powerful machines is their capability to perform machining operations in *parallel* - more than one tool on the same workpiece, or more than one workpiece on the same machine. Formally, we define the following:

Part Machining Location (PML): A part machining location is associated with a machine tool and refers to one valid workholding location on it.

Clearly, the main spindle and subspindle are always valid PMLs for a machine tool. However, by special fixturing, a turret on a machine tool can also be a valid PML.

Machining Unit (MU): A machining unit is associated with a machine tool and refers to one valid tool holding device on it.

Relative motions between the cutting tool on the MU and the workpiece held in the PML accomplish the machining. Conventional machines, such as a 3-axis milling machine will have one PML and one MU.

Parallel Machines: These are machine tools with $N (\geq 1)$ PMLs and $M (> 1)$ MUs, with the capability to activate m cutting tools ($M \geq m \geq 1$) on distinct MUs, in parallel, for the purposes of machining a single workpiece, or activate m tools ($M \geq m > 1$) on distinct MUs for the purposes of machining, in parallel, n workpieces ($N \geq n > 1$) held on distinct PMLs.

For example, a tool on machining unit A can be used to turn one portion of the workpiece while a tool on machining unit B turns another portion of the same workpiece (see Figure 1). In another instance, the activation of the subspindle can permit the machining of the rear portion on the workpiece W_1 by machining unit C, while frontal operations on a new workpiece W_2 are being performed by tools on the machining unit A.

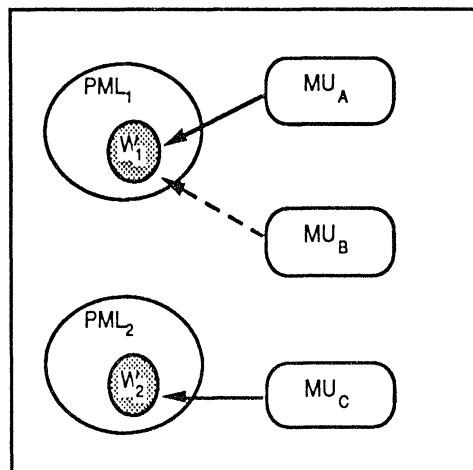


Figure 1: Illustration of Parallel Machining

Clearly, a parallel machine can perform machining in a sequential mode (i.e., when $n = m = 1$). Therefore we use the term *parallel machining* to refer to a parallel machine operating in a non-sequential mode.

The ability to perform machining operations *in parallel* raises some issues with respect to process planning which, to the best of our knowledge, have not been addressed in the past. In this paper, we identify these new issues, discuss their ramifications in process planning and propose a methodology for a CAPP system for parallel machines. The balance of this paper is structured as follows. In the next section, relevant prior work in process planning for sequential machines is described. Section 3 identifies and elaborates on the new issues for process planning in a parallel machining environment. We propose a framework for a process planning system for parallel machines in Section 4. This system includes a generalized algorithm for the formation of a process planning system for any milling, turning, or a combination, parallel machine, and can be adapted for machines with specific configuration. We conclude this paper in Section 5.

2. PRIOR WORK IN PROCESS PLANNING

There are three basic approaches that have been used in the past in the development of CAPP systems in the sequential machining environment. They are variant process planning, process planning using optimization techniques, and generative process planning. We discuss each approach briefly in the following.

2.1 Variant Process Planning

The variant method for process planning uses a coding procedure such as, Group Technology (GT) to form part families and to classify a given part as a member of a certain part family. Each part families has a predetermined process plan associated with it. After a new part has been classified, the part family process plan is retrieved and altered (by the user) to take into account dimensional (but not topological) differences. Some examples of variant process planning systems are AUTOPLAN [29], CAM-I [21], GENPLAN [30], and MIPLAN [28].

The main disadvantage of the variant method is that it can only generate process plans for parts that fall into the predefined part-families. If a part cannot be categorized, a human process planner must develop the process plan manually.

2.2 Process Planning using Optimization Methods

This is a procedure in which optimization methods are used to solve for the machining operation scheduling problem. The operation scheduling problem is typically formulated as an integer program. Most often, this method has been applied to multiple machine scheduling problems using minimum cost, maximum tool life, or minimum cycle time as the main criterion; see for example [8, 18, 19, 32]. Feature extraction and feature recognition are not performed explicitly, since in this approach there is no interaction with the CAD description of the part. In work to date, geometric constraints and other kinds of feature precedences have not been taken into account. Such information about the part must be known *a priori*, and be incorporated into the integer programming formulation of the problem. After the model is completely defined, optimization techniques are applied to produce a sequence of operations along with the machining parameters.

However, in most cases, the complexity of the integer programming problem mandates the use of heuristics, producing suboptimal results. Furthermore, the complexity associated with formulating the model and in its solution procedure prevents this approach from being fully automated; each model must be formulated “by hand” and solved individually.

2.3 Generative Process Planning

Generative process planning seeks to create an individual process plan for *each* part, based on the following [3]:

- part geometry, which includes surface characteristics and tolerances;
- manufacturing databases, which provide the necessary information for each step of the process planning routine;

- decision making algorithms, which use the database information and the part description to form the final process plan. These can exist in the form of decision trees, tabular decision tables, or the more recent rule-based expert systems.

Generative process planning is considered to be more promising than the variant process planning and optimization-based methods, as a means towards achieving totally automated CIM. Examples of generative process planning systems are AUTAP [9], CIMS/PRO [15], GARI [7], PROPLAN [26], SIPS [25], etc. However, sub-problems remain unsolved, or partially solved, in the generative process planning methodology (e.g., recognizing complex and nested features, deriving machinable sub-volumes, etc.). In addition, the decision making algorithms and logic have not yet been uniquely established. As a result, expert systems need large numbers of rules to cover all eventualities.

Some of the more recent generative planners have been developed as integrated CAD/CAPP/CAM systems--not as isolated systems which attempt to allow input from any geometric modeler [3]. The main enhancement of these newer systems is in the way that the part is designed and the feature recognition is performed. A CAD system is included as a front end module, designed to function with the planning system. When parts are designed using this "feature-based" CAD system, design features are tagged with the specific purpose of making feature recognition easier for the planning module. An example of an integrated generative process planning systems is QTC [3].

3. PROCESS PLANNING ISSUES FOR PARALLEL MACHINING

It is well known that currently (in sequential machining environments), the actual machining consumes approximately ~25% of the total time a part spends in the machine shop. It spends the rest of the time in transit from one machine to another, waiting in queues, or being realigned for machining in a new setup. Thus, it was considered unwise to expend a great deal of effort on the optimization of the actual machining processes, since only minimal time savings would result. In a parallel machining environment, however, the situation is quite different. Because parallel machines are capable of machining parts completely in one setup, machining time is now approximately 90% or more of the total time. Clearly, the reduction of machining cycle time is now an attractive goal to pursue.

There are several items which require attention when pursuing process plans in a parallel machining environment. These are concerned with: the calculation of the machining parameters for each operation, the "machine mode," the avoidance of collisions, and planning for batch machining of parts. Future efforts may lead to the development of methods for achieving optimal process plans.²

3.1 Relationship between Operation Sequence and Machining Parameters

One of the major differences between process planning for parallel machines and process planning for sequential machines involves the determination of machining parameters. The order in which the machinable features are removed has a much greater effect on the total machining time for parts made in parallel machines than for parts made in sequential machines.

²In this paper, "optimal" refers to minimum time.

This is so, because operations performed on parallel machines may *share* machining parameters, whereas operations performed on sequential machines will not.

Consider a workpiece being turned on a parallel machine. When two turning operations are being performed on the workpiece simultaneously, as might often be the case, both operations must be carried out using the same spindle speed. Similarly, when two milling cutters of a parallel machine are simultaneously machining a part held on an X-Y table, both operations must be performed using the same feed rate which is supplied by the table. Therefore, there is an interdependence between the machining operations that are performed in parallel. The machining parameters chosen will require some adjustments for at least one of the operations.

In sequential machines, on the other hand, it is desirable to carry out each operation using locally optimized parameters. This stems from the fact that there is no relationship between adjacent operations within the machining sequence. A process planning system for parallel machines must therefore be able to calculate the machining parameters taking into account the current machine situation and restrictions resulting from dependent machining operations.

3.2 Relationship between Operation Sequence and Machine Mode

Parallel machines are capable of performing various machining operations such as, turning, facing, milling, drilling, boring, etc., by a local simulation of the corresponding sequential machine environment. That is, the relative motion between the cutting tool and the workpiece for any machining operation is brought about by starting or stopping the spindle and/or the (live) tool. We refer to each such instance of a parallel machine, in operation, as a "machine mode". The typical machine modes that might be encountered in parallel machining are:

- the part is rotating, such as in turning;
- the part is stationary, such as in drilling or milling;
- both the part and the tool are in motion, such as in contour milling.

The machine mode has severe implications during the operation sequencing in parallel machines. While the presence of live tooling increases the flexibility and capability of the parallel machines, there are situations which prevent this flexibility from being completely exercised. This is because in each machine mode of a parallel machine only certain operations can be performed in parallel. For example, a radial drilling operation, in which the drill bit enters the part perpendicular to the axis of rotation of an axisymmetric part (-R direction, see Figure 2), cannot be performed while the part is rotating. Consequently, it is not possible to perform a radial drilling operation and a turning operation in parallel, since for a turning operation the rotation of the part is requisite. None of the prior work in operation sequencing has had to address this unique problem. For example, in the optimization approaches of [8], [32] all machining operations were considered to be available for scheduling at all times. Thus, for parallel machines with turning and milling capabilities, the machine mode restricts the set of selectable operations at any time.

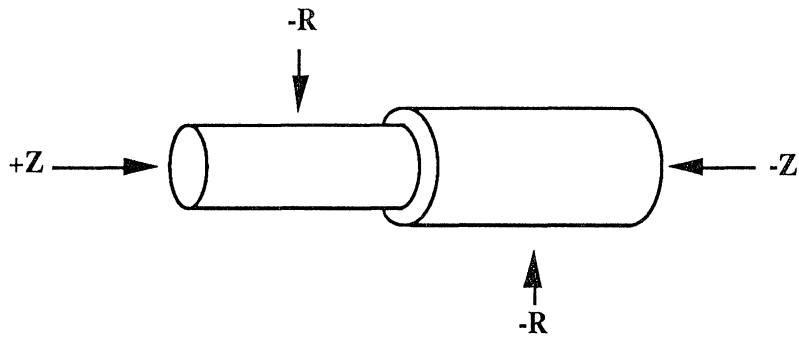


Figure 2 : Example axisymmetric part with primary approach directions

3.3 Collision Avoidance

In sequential machines, a complete process plan typically needs to insure avoidance of collisions between: (i) the tool-holder and workpiece and, (ii) between the tool and fixtures. Existing methods include on-screen graphical verification and swept volume intersections. The process plan for a parallel machine, on the other hand, must in addition to the above seek to prevent collisions between the (multiple) tools that are active simultaneously. The following example illustrates the situation.

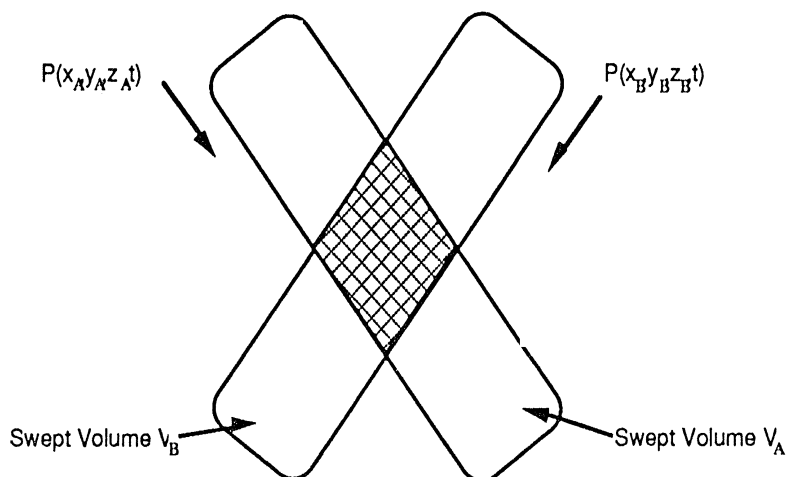


Figure 3: Intersection of tool sweep volumes

Consider a workpiece being machined simultaneously by two turning tools A and B, on two different machining units. Let their single-pass trajectory define swept volumes V_A and V_B along tool paths $P(x_a, y_a, z_a, t)$ and $P(x_b, y_b, z_b, t)$ as shown in Figure 3. If V_A and V_B do not intersect, clearly there is no collision between the tools. However, if the tool swept volumes V_A and V_B do intersect, it does not necessarily imply collision between the tools. This is because the tools A and B could occupy the same position (shaded portion in Figure 3) but at different instances in time. Therefore, process planning systems for parallel machines must be capable of performing such dynamic collision tests, so that acceptable tool path combinations will not be rejected incorrectly.

3.4 Fixture/Setup Planning and Scheduling

Fixture and setup planning is an important step in every machining operation. This topic has received attention primarily with a view toward automation; e.g., [5, 6, 22]. Integration of such modules with process planning systems have also been researched. For example, Hayes

and Wright [10, 11] have developed a system for three axis milling machines which takes into account "feature interactions"³ to establish an order for the different setups and also to determine which features must be machined during each setup. Chang and Kanumury [3], [17] use a branch and bound procedure which "clusters" the features into different groups to achieve the minimum number of setups.

In the context of parallel machining, fixturing and setup planning is an important step. The problem is rather complicated for parallel machines because of the occurrence of multiple spindles or fixturing locations. In such cases, it is necessary to come up with a schedule that specifies the following:

- (i) the different locations L_i that a part will occupy during the machining process
- (ii) the order in which the part will assume the locations L_i
- (iii) the set of machining operations M_{ij} that have to be performed at each location L_i

3.5 Process Planning for Batch Machining

In addition to allowing the machining of multiple features on a single part at the same time, parallel machines also have the ability to machine multiple parts simultaneously. This is the case when the machine possesses more than one PML. The desire to utilize this capability whenever possible requires a different overall approach to the process planning for parallel machining. We remark on this further.

Machining capabilities at each PML may change when more than one part is being machined simultaneously. Such a multi-part scenario arises frequently when the rear face of one part is being machined on the subspindle while the machining of a new part has begun on the main spindle. Consider the parallel machining environment depicted in Figure 4. Suppose MU-2 can access parts held in PML-1 and PML-2, while MU-1 and MU-3 can access PML-2 and PML-1, respectively. When only one part is on the machine, say at PML-1 (PML-2), then both MU-2 and MU-3 (MU-1) can be activated, if necessary, to machine the part. However, when both PMLs are occupied (i.e., two parts on the machine), MU-2 can access only one part at any time. Therefore, a part machined *in isolation* in either location can be allocated two machining units exclusively, whereas when two parts are in the machine, MU-2 is a shared resource.

This example illustrates that there will be differences between the process plan for a part made in a batch mode and the process plan for a part made in isolation. It must also be noted that for certain machine configurations, the first part could be machined in isolation for a greater share of its total cycle time than would be the case for subsequent parts. If this condition were to be allowed to occur during the formulation of the process plan, the generation of as many different process plans as parts in the run could result, since the machining cycle time would gradually progress from that of the first part towards a steady state level approached by the last part in the run. The occurrence of as many process plans as parts would form an extremely long NC program.

³Hayes and Wright. "Automating Process Planning: Using Feature Interactions to Guide Search," Journal of Manufacturing Systems, 1990, p. 1.

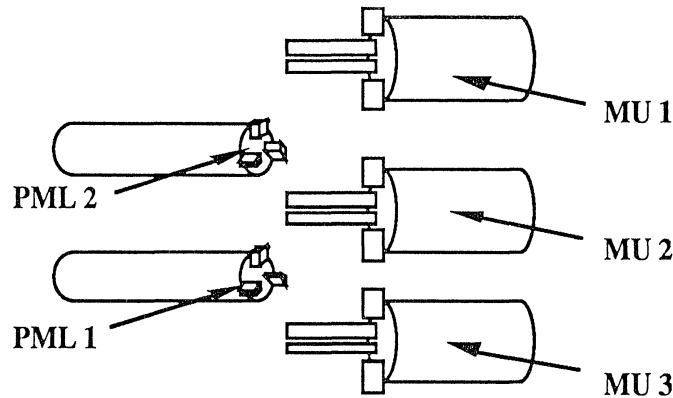


Figure 4: Configuration showing 3 MUs (turrets) and 2 PML (spindles)

In summary, this section has identified and elaborated on five major differences between process planning for parallel and sequential machining. In the next section, a process planning methodology which takes into account these new issues will be presented.

4. A PROCESS PLANNING ALGORITHM FOR PARALLEL MACHINING

The algorithm revealed in this section incorporate the issues highlighted in Section 3 into a process planning system for parallel machines. We discuss the methodology in details and present the algorithm. Procedures are suggested for the individual modules within the overall system. In this paper, we exclude discussions about feature recognition/extraction, and process selection. A functional description of the input to the algorithm, the preprocessor, and the planning modules is explained in the following sections.

4.1 Input to Algorithm

The features are presented as input to the system in the form of the boundary representations of the volumes to be machined which relate to each feature. These feature BREPs also possess several attributes which describe the type of feature, the location and dimensions of the feature, tolerances, surface finish information, approach directions, and relationships to other features (some of these attributes may be *derived* from the BREP). This information is the output of a feature extraction and recognition system which has access to the design model of the part. The “type of feature” attribute points to a manufacturing template which specifies the operations that must be performed in order to machine the feature. Each manufacturing template contains the set of operations that correspond to a manufacturing process. If there is more than one kind of manufacturing process that may be used to produce a specific feature, heuristics (rules of thumb) may be used to select the most effective process. An example template is shown in Figure 5 for a threaded hole; actual selection of the optional operations depends on the tolerances, surface finish, and the tool used to machine the feature.

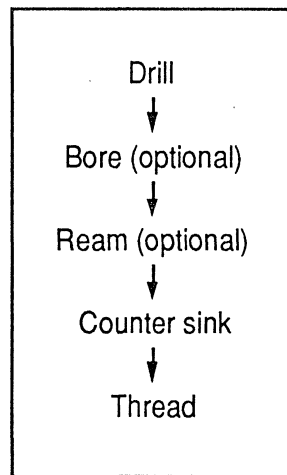


Figure 5: Manufacturing Template for Threaded Hole

4.2 Preprocessor Module

The preprocessor module is used to:

- establish the setups and the positions on the machine that the workpiece will assume as the machining proceeds;
- form precedences between features, due to geometry, manufacturing practices, and tolerances;
- assign approach directions and initial machine modes to the different features.

The first section in the preprocessor is a high-level planner which schedules the setups and PMLs. This part of the preprocessor determines the order of the setups and divides the operations into groups that will be performed in each setup. Next, the preprocessor tests for accessibility, and uses the approach directions associated with each feature to establish geometric precedences within the setups. Subsequently, each feature is labeled with an initial machine mode, which will be in effect for the first operation to be done on the feature. The final step in the preprocessor is to incorporate additional precedences resulting from tolerances and machining practices relevant to the part. When the preprocessing is finished, it is possible for each feature to have a different number of prerequisites. The steps in the preprocessor module are shown in Figure 6 and outlined in the succeeding sections.

4.2.1 Setup/Fixturing Planner

The type of reasoning used to accomplish this step might be considered as an extension of the work done by Hayes and Wright [10, 11]. In their *Machinist* system, the number of setups is determined, the setups are ordered, and it is decided in which setup each feature will be machined. The incorporation of a scheduling algorithm will also establish the PML for each setup on the machine tool, *the conditions on the machine at "time zero"* (in the batch planning scenario, there may be several workpieces on the machine at any point in time, each in a different stage of machining), and the orientation of the part in each setup. Time zero occurs when the machining of a new part begins during batch machining. The input essential for this step includes the feature BREPS, tolerances, surface finishes, reference faces, descriptions of the available fixtures, and information regarding PMLs and MUs on the particular parallel machine. Upon completion of this step, the features are separated into groups corresponding to the different setups, and each setup has an assigned PML.

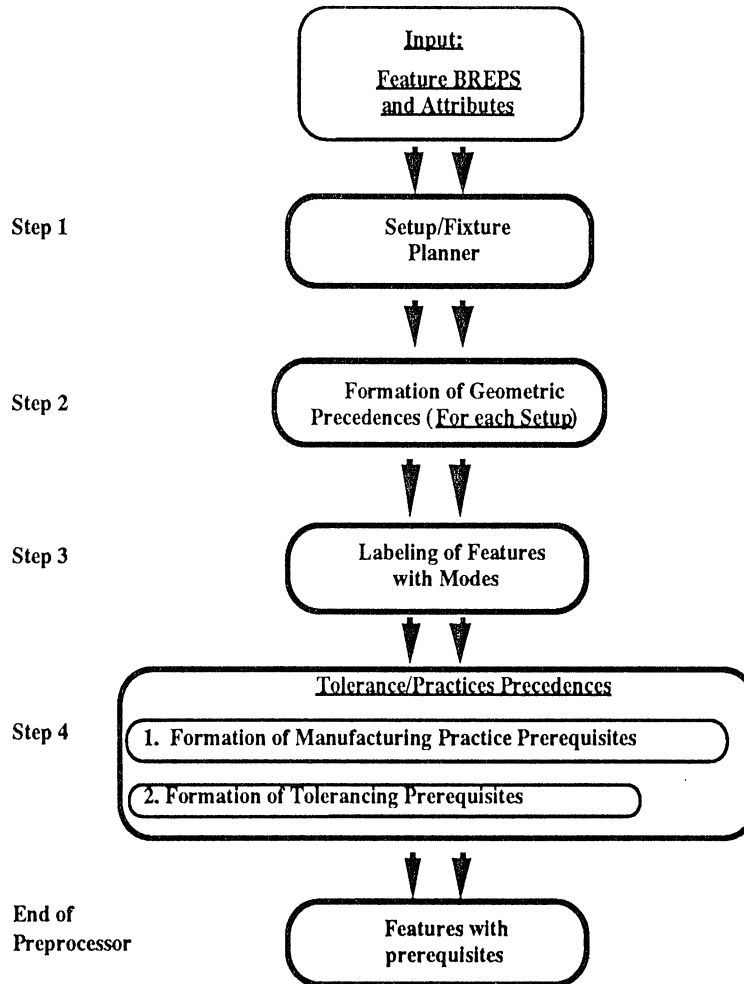


Figure 6: Structure of Preprocessor Module

4.2.2 Geometric Precedences

The next module of the preprocessor calculates the geometric precedences between the features within each setup. Formation of geometric precedences helps to avoid accidental collisions between the tool and portions of the workpiece that have yet to be machined. After each volume is machined, new volumes are exposed “to the outside” and become eligible for machining. This step is performed for each setup, and uses face adjacency information from the boundary representation of the workpiece (as it exists at the beginning of the setup) and the approach directions of the features which were determined in the setup/fixturing phase.

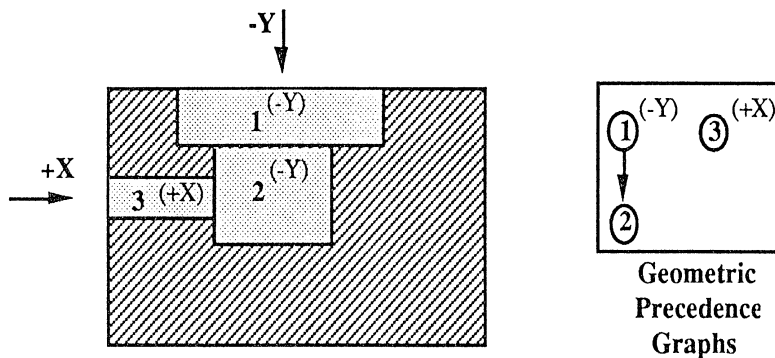


Figure 7: Showing features 1, 2, and 3 with approach directions and resulting precedences.

Although precedences between features at this stage can be thought of as existing in a graph-like format, the precedences are actually represented as prerequisite attributes in the feature data structures. Using this convention, prerequisites of different types can be added later on without worrying about the maintenance of the graphs, since the planning module will treat all prerequisites in a similar fashion. An abbreviated data structure for a feature which includes prerequisites might be represented in C as:

```
typedef id char[20];
typedef ad char[4];
struct feature {
    id name;
    ad approach_dir;
    feature *prereq[15];
    .
    .
};
```

Pointers to the prerequisite features reside in the null-terminated array `prereq[]`, and mechanisms for adding and deleting prerequisites are utilized as the planning proceeds. For the example in Figure 7, features 1 and 3 would have no prerequisites, and feature 2 would have a pointer to feature 1 in its `prereq[]` array.

Treating all prerequisites in a similar fashion may not be sufficient if “precedence cycling” occurs. An example of precedence cycling is when a feature A has a feature B as a prerequisite, feature B has a feature C as a prerequisite, and feature C has feature A as a prerequisite. In this case, conflict resolution measures will be needed, and it may be necessary to rank the types of precedences to ensure that the higher ranked precedences are obeyed when breaking such ties.

4.2.3 Labeling of Features With Initial Machine Mode

The next step is to label each feature with the appropriate machine mode for the first operation to be initiated in the machining of the feature. This step is important later on, since the planning module only considers features with a single machine mode at any time (at each PML). The machine mode can be determined from the manufacturing template of the feature. The mode can be included as another field in the feature data structure. For example:

```
typedef struct feature feature;
typedef name char[20];
typedef ad char[4];
struct feature {
    name id;
    ad approach_dir;
    feature *prereq[15];
    int mode;
    .
    .
};
```

There are some feature producing operations which may have more than one machine mode. A feature such as a hole through the center of a rotational part, oriented the same as the axis of symmetry of the part, has multiple machine modes; it can be machined when the part is rotating and the drill bit is stationary, or when the part is stationary and a rotary powered tool is used. In this case, Hinduja and Huang [13] have suggested that these holes should be machined *first*,

in order to obey machining practices. Therefore, the machine mode for holes of this type can be defined to be the mode in which the part is rotating⁴.

4.2.4 Tolerance/Manufacturing Practice Precedence Creation

After the initial machine mode has been assigned to each feature, additional precedences are formed by incorporating tolerances and manufacturing practices. Although some of the tolerances may have been taken into account in the setup planning stage, it may be necessary to assign some more precedences, as a result of the dimensioning of the part. The information necessary for precedences caused by tolerances is input by the user, either as part of the CAD model or in a separate step, and is attached to the feature BREP as a set of attributes.

Manufacturing practices fall into two categories:

1. *Within a feature*, additional operations may be required in order to machine the feature. For example, to machine a pocket with a flat end milling cutter, a hole must be drilled prior to milling to allow the entrance of the milling cutter.
2. *Between features*, alteration of precedences may be necessary. An example of this situation occurs when a hole opens onto a slanted surface (see Figure 8). If the face was perpendicular to the axis of the hole, the geometric precedences would require that the hole be drilled after the face was machined. However, in the case of a slant entrance face, it is desirable for the hole to be drilled before the face is machined (to avoid slippage of the drill bit).

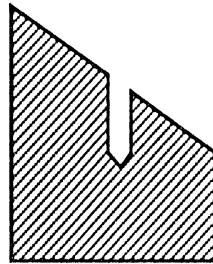


Figure 8: Hole feature with axis not perpendicular to entrance face

The first kind of manufacturing practice is treated by the inclusion of additional operations in the manufacturing templates. In the template for a pocket, an additional drilling operation is included prior to the milling operations. The second kind of manufacturing practice is handled in this module, and requires searches through the feature precedences for patterns that are at odds with manufacturing practices. Once such a pattern is found, precedences are rearranged to rectify the situation. Tolerance information can be used to form precedences in a similar manner. The rules are applied in the form of condition/action pairs. Two examples are:

```
IF feature A is dimensioned from feature B,
THEN make feature B a prerequisite for feature A.
```

```
IF the approach direction for a hole is not normal to
the entrance face for the hole,
THEN make the hole feature a prerequisite for the face
feature.
```

⁴The mode in which the part is rotating is usually considered to be the “first” mode of any setup (if it is required) because roughing operations for turned parts are performed first. In addition, material removal costs are lower for turning than for milling. [4]

At the conclusion of the preprocessor, the setups, the PMLs, and the order in which the workpiece will assume these locations are determined. In addition, all precedences between the features are established. The features are labeled with their initial machine mode. Next, we describe the planning module which accepts the features and develops the process plan.

4.3 Planning Module

In the planning module, the following tasks are performed:

- tools are selected for each operation;
- machining parameters are calculated for each operation, taking into account the effects of simultaneous machining operations;
- the sequence of operations is established, taking into account the capabilities of the machining units and the potential for collision;
- tool paths and intermediate motions (such as tool changes, rapid tool traversals, and workpiece transfers between setups or PMLs) are determined for each operation.

At the start of the planning module, the clock time is initialized to zero. The PMLs are set to assume the conditions at time zero that were determined in the setup/fixtures planner. Next, for each free MU, the accessible PMLs are found, and the features without prerequisites at these locations are retrieved. For each such “active” feature, the next operation and machining parameters are generated taking into account the conditions on the machine. The operations are assigned to the machining units, a collision avoidance check is performed, and the clock time at which the next machining unit becomes free is calculated. This loop is then repeated. The steps in the planning module are shown in the flow diagram of Figure 9 and explained next.

4.3.1 Determine Accessible Locations

For each available MU, the accessible PML are determined through the use of a 0-1 matrix defined *a priori*. The rows of the matrix correspond to the MUs and its columns to the PMLs. If the *i*-th MU can access the *j*-th PMU, then element $(i,j) = 1$; otherwise $(i,j) = 0$. The clock times at which the MUs become available are determined from the machining parameters. An example is shown below. Various (parallel) machine configurations can be distinguished by such matrices.

	PML 1	PML 2	PML 3	...
MU 1	1	0	0	...
MU 2	1	1	0	...
MU 3	0	1	1	...
...

Planning Module

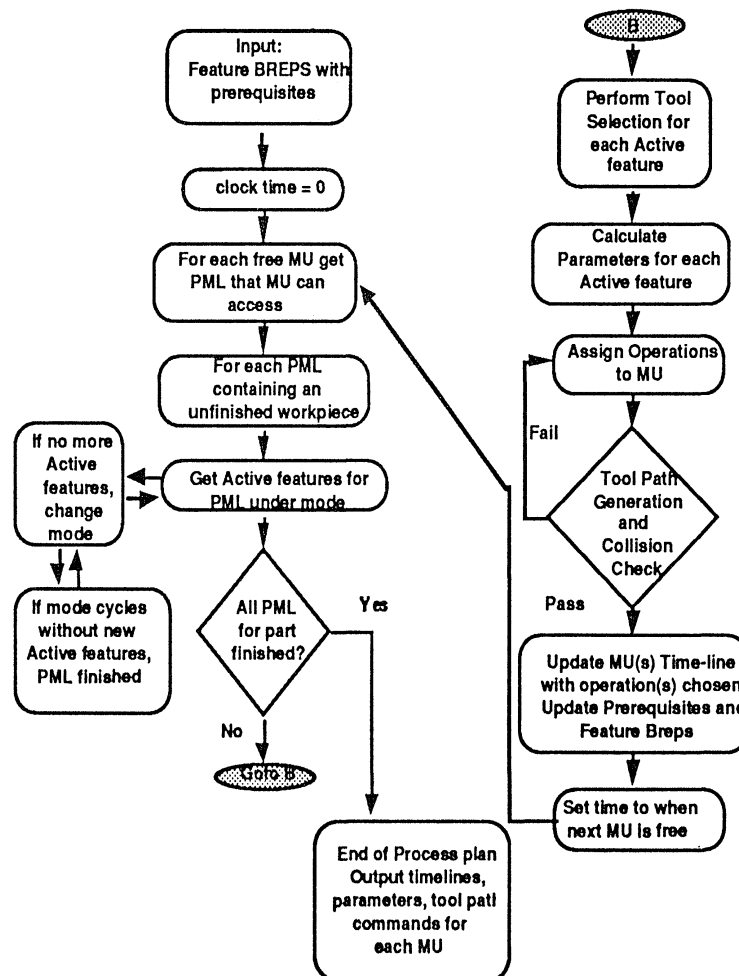


Figure 9: Structure of Planning Module

4.3.2 Retrieve Active Features

The “active” features are acquired for each PML in the next step of the algorithm . A feature F_i is considered active when

- (i) F_i has no unfulfilled prerequisites (i.e., operations)
- (ii) F_i is designated to be machined at the current PML
- (iii) the machine-mode at the current PML is the same for F_i

If F_i does not meet all of these conditions, then it is considered inactive and is eliminated from contention (for the current iteration). The active features are obtained by first finding all the features to be machined at the specific PML with the proper machine mode, and then by detecting which of these features have no remaining prerequisites. If no active features are found at a specific PML, then the machine mode is changed and the process is repeated. The modes will cycle in order, first with the part rotating, then with the part stationary, and then with synchronized motions between the fixture and the tool. Note that not all of these modes may be possible for every kind of fixture or PML.

4.3.3 Perform Tool Selection and Parameter Calculation

In this stage, the tools and parameters for the next machining operation for each active feature are determined.

Tool Selection

The procedures used to select tools are different depending on whether planning for a batch of parts or planning for an individual part is taking place. Since the time required for an operator to manually install tools on the machining units can be distributed over the full run, the installation costs become inconsequential. The tools can then be chosen specifically for the part from a list of all tools available (for the machine), instead of limiting the tool selection to what is currently on the machining unit or in the tool magazine.

In our algorithm, the MUs are empty at the beginning of the sequencing of operations. As the planning proceeds, the MUs are filled with the appropriate tools for the part. This allows the left over space on the MUs to be filled with replacement tools that might be required for the machining of the specific part if desired. The algorithm used to select a tool for machining feature F_i consists of the following steps:

- Step 1: Check the manufacturing template of feature F_i to retrieve the next operation C .
- Step 2: From the BREP of F_i (and attached attributes) retrieve the dimensions, orientation, and surface characteristics for C .
- Step 3: List all remaining features F_k requiring machining operations similar to C which will be performed at a PML that the free MU can access.
- Step 4: Choose a tool T from the tool library such that T is valid for C and as many operations on F_k as possible.
- Step 5: Assign tool T to one MU during the assignment phase of the algorithm.

Parameter Calculation

We use a forward planning methodology for calculating the machining parameters. Forward planning is characterized by the natural progression of the workpiece from *raw stock* to *finished part*, as a result of the various machining operations. In backward planning methodology, the process is conceptualized to begin from the finished part and machining operations are used to *fill in* material until the raw stock is achieved. The final process plan is formed by reversing the sequence of operations that were used to fill in the workpiece. An advantage of backwards planning is that for each feature the final operation producing the desired surface finish is chosen first in the planning algorithm. Unfortunately, this method is ill-suited for the process planning of parallel machines because the effects of time, the sharing of machining parameters, and collision avoidance prevent such (backward) process plans from being reversed without severe implications.

The machining parameters for the next pass are calculated based on:

- the current machining conditions resulting from current operations, such as spindle speed, cutting force, machine power, etc.;
- the desired surface characteristics to be produced by the current operation (i.e., whether the operation is a final pass, or an intermediate one);
- the tool chosen for the operation;

- established machining guidelines, such as in the Machining Data Handbook [33].

4.3.4 Assign Operations to MUs

Once the parameters and tools have been calculated, it is necessary to assign machining units to operations. There may be more MUs than active features, or vice versa. Therefore, a ranking of operations, and/or a ranking of the MUs, according to some performance index is required. The best measure of performance may be to consider the amount of tool idle time in each operation, which is made up of tool change time, tool traverse time, and the increase in machining time caused by the use of “non-optimal” parameters. At this point, the capabilities of the MUs to perform the operations must also be taken into account.

The performance indices must be updated to reflect whether a machining unit can access a specific feature at a particular time. This is a local accessibility measure, as opposed to the global accessibility of machining units discussed earlier--PML vs. MU matrix. In addition, the capabilities of the MUs must be taken into account when determining the performance indices; machining operations may require feed rates or resolutions that a particular MU cannot achieve. Once the performance indices are computed, the “best” operations are then chosen using any assignment method, such as the Hungarian Algorithm [12].

4.3.5 Tool Path Calculation and Collision Check

After the operations have been assigned to the machining units, it is necessary to make sure that the particular combination of operations chosen will not lead to any collisions. Since the collision check is computationally very expensive, it is not a good practice to carry out a collision check for each of the possible combinations of operations during the assignment phase. Therefore, tool paths are calculated after the assignment step. The tool paths (along with the intermediate motions generated in the parameter calculation stage) should be calculated in terms of G- and M-codes, so that the path can be stored in terms of motion commands, and not just as a record of the location of the MUs versus time.

As mentioned in Section 3, dynamic collision checks will be necessary. One method is to compute tool swept hypervolumes in 4-dimensions, time being the fourth dimension [14]. A method for such collision check between tools T_A and T_B is as follows:

1. Beginning at the current clock time, compute tool swept volumes for T_A and T_B .
2. If the swept volumes do not intersect, current tool paths are valid. Else compute tool swept hypervolumes for T_A and T_B .
3. If the hypervolumes do not intersect, the tools paths are valid. Else generate and evaluate new tool paths.

Note, if a collision is detected in step 3, alternate tool paths and parameters can possibly be generated, and the test repeated. If all possible tool paths have been attempted and a collision still occurs, either the next highest ranked group of operations can be chosen, or one of the operations causing the collision can be left out (which will mean that the machining unit originally chosen for this operation will remain idle until the next operation is finished).

4.3.6 Machining Unit Time-Line and Updating of Prerequisites

Once the group of operations that has been selected passes the collision check, the operations are added to the time line schedule for the machining units that are involved. Since the

parameters are already known, the amount of time needed to perform each operation is easily determined, as is the next clock time that a MU will become available for a new operation. In addition, revisions are made concerning the status of both the active and the inactive features. For the selected features, the manufacturing templates, the machine mode, and the feature BREPs are updated. If the machining operation is a concluding one (i.e., its completion finishes the feature) the prerequisites that are satisfied are removed from the inactive features. Next, the clock time is set to when the next MU becomes available, and the program flow returns to the beginning of the sequencing cycle.

The final process plan consists of:

- a time line schedule for each MU containing the machining operations to be performed, the PMLs for each operation, the tools used during each operation, and the values for the machining parameters;
- a record of the controller commands for each machining unit

5. CONCLUDING REMARKS

The advent of parallel machine tools has raised issues in computer-aided process planning that have not been addressed in previous research. In this paper, these issues have been identified, their implications discussed, and a high level architecture for a process planning system incorporating these issues has been proposed. A salient feature of the proposed system is that the process information is determined for each operation in a single iteration, rather than calculating one process parameter at a time for *all* operations. Furthermore, the architecture of the proposed system is for parallel machines in general, and can accommodate any number of MUs and PMLs.

Our ongoing work deals with the development of a prototype using the framework of the process planning system described herein. In addition, methods are being investigated to allow the user some input in setting up the system (before planning is initiated), and to allow subsequent editing of plans, once a process plan has been generated.

6. REFERENCES

- [1] Allen, D. K., "Computer Aided Process Planning: Software Tools," PED-21 Vol. 21, *Integrated and Intelligent Manufacturing*, ed. by Liu, C. R., and Chang, T. C., ASME, WAM, pp. 391-400, 1986.
- [2] Chang, T.C and Wysk, R.A. *An Introduction to Automated Process Planning Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.
- [3] Chang, T.C. *Expert Process Planning for Manufacturing*, Addison-Wesley Publishing Company, Inc., 1990.
- [4] Chang, T.C., Wysk, R.A., Davis, D.P., and Choi, B. "Milling Parameter Optimization through a Discrete Variable Transformation," *Int. J. Prod. Res.*, Vol. 20, No. 4, pp. 507-516, 1982.

- [5] Chou, Y.C., Chandru, V., and Barash, M.M., "A Mathematical Approach to Automatic Design of Fixtures: Analysis and Synthesis," Proc. of ASME Winter Annual Meeting, Boston, pp. 11-27, 1987.
- [6] Cutkosky, M.R., and Tenenbaum, J.M., "A Methodology and Computational Framework for Concurrent Product and Process Design," Mechanisms and Machine Theory, 1989.
- [7] Descotte, Y., and Latombe, J-C., "Making Compromises among Antagonist Constraints in a Planner," Artificial Intelligence, Vol. 27, pp. 183-217, 1985.
- [8] Egbelu, P.J. "Planning for Machining in a Multijob, Multimachine Manufacturing Environment," Journal of Manufacturing Systems, Vol. 5, No. 1, pp. 1-12, 1985.
- [9] Eversheim, W., Fuchs, H., and Zons, K. H. "Automated Process Planning with Regard to Production by Application of the System AUTAP for Control Problems," Computer Graphics in Manufacturing Systems, 12th CIRP International Seminar on Manufacturing Systems, Belgrade, 1980.
- [10] Hayes, C. and Wright, P. "Automating Process Planning: Using Feature Interactions to Guide Search," Journal of Manufacturing Systems, Vol. 8, No. 1, pp. 1-15, 1990.
- [11] Hayes, C. and Wright, P. "Automated Planning in the Machining Domain," Knowledge Based Expert Systems for Manufacturing, WAM, ASME, Dec. 7-12, 1986, Anaheim, CA, pp. 221-232.
- [12] Hillier, F. S., and Lieberman, G. J. *Introduction to Operations Research*, Holden-Day, Inc., 1967, pp. 199-204.
- [13] Hinduja, S. and Huang, H. "OP-PLAN: an automated operation planning system for turned components," Proc. Instn Mech. Engrs, Part B, (Journal of Engineering Manufacture), Vol. 203, pp. 145-158, 1989.
- [14] Hoffman, C., and Zhou, J. "Some Techniques for Visualizing Surfaces in Four-Dimensional Space," Computer Aided Design, Vol. 23, No. 1, Jan/Feb. 1991, pp. 83-91.
- [15] Iwata, K., Kakino, Y., Oba, F., and Sugimura, N. "Development of Non-Part Family Type Computer Aided Production Planning System CIMS/PRO", in *Advanced Manufacturing Technology* (Proc. 4th Int. IFIP/IFAC Conference, PROLOMAT, 1979), ed. by P. Blake, Elsevier North-Holland, New York, 1980.
- [16] Joshi, S., Vissa, N. N., and Chang, T.C. "Expert Process Planning System with Solid Model Interface," in *Knowledge Based Systems in Engineering*, ed. by Kusiak, A., Taylor & Francis, Philadelphia, PA, pp. 111-136, 1989.
- [17] Kanumury, M. "AMPS--An Automatic Manufacturing Process Planning System," M.S. Thesis, School of Industrial Engineering, Purdue University, 1988.
- [18] Kusiak, A. "Integer Programming Approach to Process Planning," International Journal of Advanced Manufacturing Technology, Vol. 1, No. 1, pp. 73-83, 1985.
- [19] Kusiak, A. *Intelligent Manufacturing Systems*, Prentice Hall, 1990.

- [20] Lambert, B.K., and Walvekar, A.G., "Optimization of Multi-Pass Machining Operations," *Int. J. Prod. Res.*, Vol. 16, No. 4, pp. 259-265, 1978.
- [21] Link, C.H., "CAPP-CAM-I Automated Process Planning System," in *Proceedings of the 13th Numerical Control Society Annual Meeting and Technical Conference*, March 1976.
- [22] Mani, M., and Wilson, W.R.D., "Automated Design of Workholding Fixtures Using Kinematic Constraint Synthesis," *Proc. of 19th NAMRC*, SME, Univ. of Illinois, Urbana-Champaign, pp. 437-444, 1988
- [23] Miller, P. C. "Lathes Turn to Other Tasks," *Tooling & Production*, pp. 54-60, March 1989.
- [24] Miska, K. H. "Driven Tools Turn on Turning Centers," *Manufacturing Engineering*, pp. 63-66, May, 1990.
- [25] Nau, D., and Gray, M. "SIPS: An Approach of Hierarchical Knowledge Clustering to Process Planning," *Bound Volume*, ASME, WAM, Anaheim, California, Dec. 1986.
- [26] Phillips, R. H., X. D. Zhou, and C. B. Mouleeswaran. "An Artificial Intelligence Approach to Integrating CAD and CAM through Generative Process Planning," *Proceedings of the 4th ASME International Computers in Engineering Conference*, Vol. 2, Book No. G240, pp. 459-463.
- [27] Rich, E., and Knight, K. *Artificial Intelligence*, 2nd ed., McGraw-Hill, Inc., 1991, pp. 37-39.
- [28] Schaffer, G. "GT via Automated Process Planning," *American Machinist*, pp. 119-122, May, 1980.
- [29] Tempelhof, K.H. "A System of Computer Aided Process Planning for Machine Parts," in *Advanced Manufacturing Technology* (Proc. 4th Int. IFIP/IFAC Conference, PROLOMAT, 1979), ed. by P. Blake, Elsevier North-Holland, New York, 1980.
- [30] Tulkoff, J. "Lockheed's GENPLAN," *Proceedings of the 18th Numerical Control Society Annual Meeting and Technical Conference*, Dallas, Texas, May, 1981, pp. 417-421.
- [31] Wang, H. P. and Wysk, R.A. "Intelligent Reasoning for Process Planning," *Computers in Industry*, Vol. 8, pp. 293-309, 1987.
- [32] Watson, E. and Egbelu, P.J. "Scheduling and Machining of Jobs through Parallel Nonidentical Machine Cells," *Journal of Manufacturing Systems*, Vol. 8, No. 1, pp. 59-68, 1990.
- [33] *Machining Data Handbook*, 2nd ed., Machinability Data Center, Metcut Research Associates, Inc., Cincinnati, Ohio, 1972.



THE UNIVERSITY OF MICHIGAN

DATE DUE

8/16 1:39