IMACS: A Case Study in Real-World Planning

Satyandra K. Gupta*
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
skgupta@ri.cmu.edu

Dana S. Nau
Computer Science Department and
Institute for Systems Research
University of Maryland
College Park, MD 20742
nau@cs.umd.edu

William C. Regli
Department of Mathematics and Computer Science
Drexel University
Philadelphia, PA 19104
regli@mcs.drexel.edu

For submission to *IEEE Expert and Intelligent Systems*.

Abstract

This article illustrates the complexities of real-world planning and how we can create AI planning systems to address them. We describe the IMACS Project (Interactive Manufacturability Analysis and Critiquing System) from the University of Maryland, College Park. IMACS is an automated designer's aid to evaluate the manufacturability of machined parts and suggest design modifications to improve manufacturability.

Over the course of our efforts on IMACS the manufacturing domain continually challenged us to come up with working solutions that would scale to realistic problems. This paper compares and contrasts IMACS's planning techniques with those used in classical AI planning systems and describes (1) how some of IMACS's planning techniques may be useful for AI planning in general, and (2) what challenges need to be overcome by AI planners so that they can be successfully used in manufacturing process planning.

Similarities between AI planning techniques and IMACS planning techniques indicate the large unrealized potential of AI planning techniques in solving real-world manufacturing problems. On the other hand, differences seem to indicate the need for domain-specific planning techniques. In particular, our experience suggests that process planning for complex machined parts would not be accomplished very well by populating a general purpose planner with domain-specific knowledge. Instead, we needed to integrate the domain-specific knowledge into the planning algorithms themselves.

1 Introduction

How to generate manufacturing process plans automatically is a challenging research problem that has attracted the attention of both the AI and engineering communities. To AI researchers, process planning offers a real-life testbed for AI search and planning techniques. At the same time, economic advantages associated with the automation of process planning make this problem attractive to engineers. Previous applications of AI planning technology to manufacturing planning, however, generally have had little impact on manufacturing practices [9, 15].

Many issues arising in manufacturing process planning are similar to issues investigated in AI planning; others are distinctly different. Some of the former may be amenable to the use of existing AI planning

*Author to whom all correspondence should be addressed.

techniques—and some of the latter may lead to new principles useful in AI planning. To investigate such issues, AI researchers will need a better understanding of manufacturing problems and concerns, so as to get better ideas of what the interesting generalizations are, and which techniques from AI might best be applied to realistic manufacturing problems.

IMACS is an automated designer's aid to evaluate the manufacturability of machined parts and suggest design modifications to improve manufacturability while maintaining the functional intent. IMACS works by generating and evaluating alternative operation plans. Our primary goal in developing IMACS was to handle real-life mechanical artifacts. In retrospect, we found that techniques used in IMACS have both striking similarities and significant differences with those of AI planning techniques.

The similarities between AI planning techniques and IMACS's planning techniques indicate the large unrealized potential of AI planning for solving real-world manufacturing problems. On the other hand, the differences seem to indicate the need for increased use of domain-specific planning techniques. In particular, our experience suggests that process planning for complex machined parts is not accomplished very well by populating a general purpose AI planning system with domain-specific knowledge: in order to get IMACS to work well, we needed to integrate the domain-specific knowledge into the planning algorithms themselves.

2 The Process Planning Problem

Over the last two decades, many attempts have been made to apply computers to the task of generating *process plans*—i.e., detailed descriptions of how to manufacture an artifact. Most of these attempts have resulted either in systems that solved an overly simplified version of the problem and thus were not capable of handling real-life parts, or in systems that required vast amounts of domain-specific knowledge and thus could only operate on a very limited set of parts.

The input to most process planning systems is a description of the part (attributed solid models or CAD designs in most modern systems). Manufacturing knowledge is modeled as the set of available manufacturing operations. To match the operations against the part shape, most previous approaches adopted the following underlying formulation:

- Each manufacturing operation is capable of creating a certain type of primitive shape. The primitive shapes are called **features**.
- Many times, more than one manufacturing operation can result in the same primitive shape. So, there is one-to-many mapping from **features** to **manufacturing operations**.

Using this formulation, the following two-step approach is used to generate plans:

Step 1. Decompose the given part into a set of primitive shapes (a **feature-based representation**). In most of the literature this step is referred to as feature extraction or recognition [14, 22]. A number of different techniques have been developed to identify features on the part. A detailed survey on these techniques can be found in [20].

Step 2. After decomposing the part into a set of primitive shapes, map each shape into an operation or sequence of operations that can create it [10, 11]. Since more than one operation or sequence of operations may be capable of creating the shape, various optimization techniques can be used to select which operation or sequence of operations to use. Finally, the parameters associated with each operation are optimized and the detailed process plan is created.

While the above formulation is a convenient and intuitive way to specify the problem, it is fundamentally limited: many times, it is possible to machine the same volume using many alternative collections of primitive shapes. In such cases, there is no unique way to decompose a given part into a unique feature-based representation (i.e., collection of primitive shapes), and attempts to identify a single decomposition will leave out other possible choices.

Since each choice corresponds to a different collection of process plans, the usual approach of considering only one choice and ignoring the others may generate a sub-optimal process plan or no plan at all. On the other hand, using brute force methods to identify all possible decompositions and thereby generate all possible process plans is computationally infeasible.

In previous approaches, geometric reasoning was not integrated with planning and was limited to the task of **feature recognition**—the identification of features directly from the CAD data. After that step, feature-to-operation mapping was used to select operations for particular features. Unfortunately, whether or not a feature will have a feasible operation associated can not be determined in isolation. Feasibility of an operation depends on the state of the workpiece when an operation is being carried out. For example, tooling accessibility or fixturability conditions can only be verified with respect to the intermediate workpiece geometry. In most previous approaches, such evaluation was not performed—therefore, the plan generated by such a system was not guaranteed to be correct.

3 IMACS: Project Overview

IMACS (Interactive Manufacturability Analysis and Critiquing System) is a computer system for analyzing the manufacturability of machined parts, in order to help designers produce designs that are easier to manufacture. Because of pressing demands to reduce lead time for product development, it is becoming increasingly important to analyze the manufacturability of proposed products during the design stage. By analyzing the manufacturability of machined parts once the geometry and tolerances have been specified, IMACS will help in creating designs that not only satisfy the functional requirements but are also easy to manufacture.

3.1 A Collaborative Interdisciplinary Team

The team of researchers that developed IMACS was a highly interdisciplinary one that included both engineers and computer scientists,[1] with a very broad background covering AI search and planning, manufacturing, mechanical design, algorithms, and solid and geometric modeling. The group members shared offices—this led to true interdisciplinary training at the most fundamental level: engineers learned how computer scientists think and vice-versa. By operating in a truly collaborative environment, we developed a shared vocabulary and learned to appreciate and understand the views and constraints brought by the various disciplines.

We approached the process planning problem by taking techniques for AI planning, process modeling, and solid modeling, and bringing them together in a unified framework. Since our goal was to build a sound and practical solution to the process planning problem, we did not want to simplify the problem in favor of achieving mathematical elegance—nor did we want to restrict ourselves to a single technique or approach in cases where a combination of approaches might work better. However, to keep the problem tractable (and within reach of a limited development staff), we decided to focus on drilling and milling operations performed on a vertical machining center.

3.2 Overview of the IMACS Approach

As shown in Figure 1, IMACS evaluates the manufacturability of a proposed design by generating and evaluating operation plans. The fundamental components of this system include:

[1]Over the course of the project, IMACS included contributions from two faculty in Computer Science, two faculty in Mechanical Engineering, 3 engineering graduate students, 4 computer science graduate students, and 2 undergraduates, as well as the input of many other colleagues.

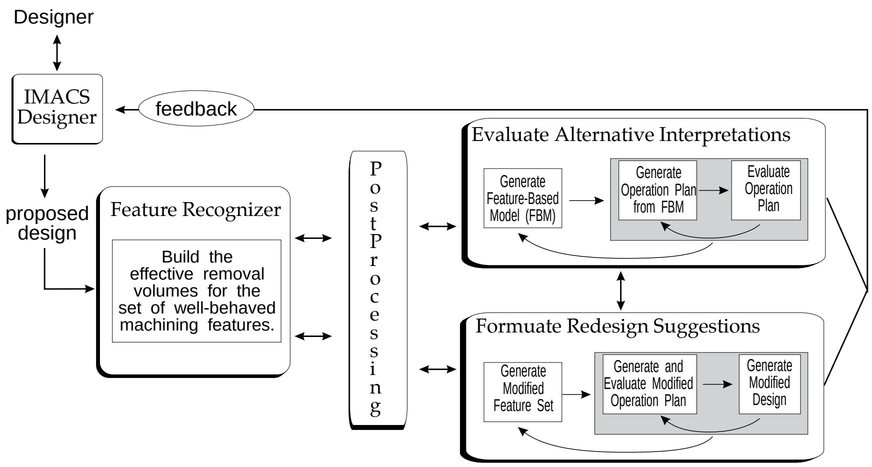


Figure 1: Basic approach used in IMACS.

- A module for recognition of machining features from a CAD model;
- A module for generating and evaluating operation plans from the features returned by the feature-recognition module;
- A module for suggesting combinations of design modifications to the designer to improve manufacturability while maintaining the functional intent.

Figure 2 shows the user interfaces for the major IMACS components.

IMACS can identify manufacturing difficulties interactively, as the designer creates a design, by generating and evaluating alternative operation plans. In this way, it checks whether an operation plan exists that can create the design, what the best operation plan is, and what kinds of machining problems might occur. Since many such manufacturability problems can only be detected at the planning level, where existing rule-based commercial systems often have difficulty.

3.3 Novel Aspects of IMACS

The formalism behind IMACS is based on the notion that multiple feature-based representations of the part exist and must be considered. Our approach overcomes the limitations of earlier work in three significant ways:

- **Accurate Process Modeling Using Machining Features.**

  Traditionally, **form features** have been used to represent the relationship between shape primitives and manufacturing operations. Most existing feature definitions for the machining domain use geometric entities such as a collection of faces or a parameterized volume to represent this relationship. These definitions lack several key pieces of information related to the parameters of the machining operation, such as its accessibility, cutting tool, dimensional constraints, and so forth.

  We used **machining features** to represent the geometric capabilities of processes. Our features not only have the shape information but also allow us to capture a wide variety of process constraints imposed by the workpiece shape. Examples of these constraints include accessibility and maximum tool diameter. Our features allow us to prune those choices which might lead to infeasible plans. For

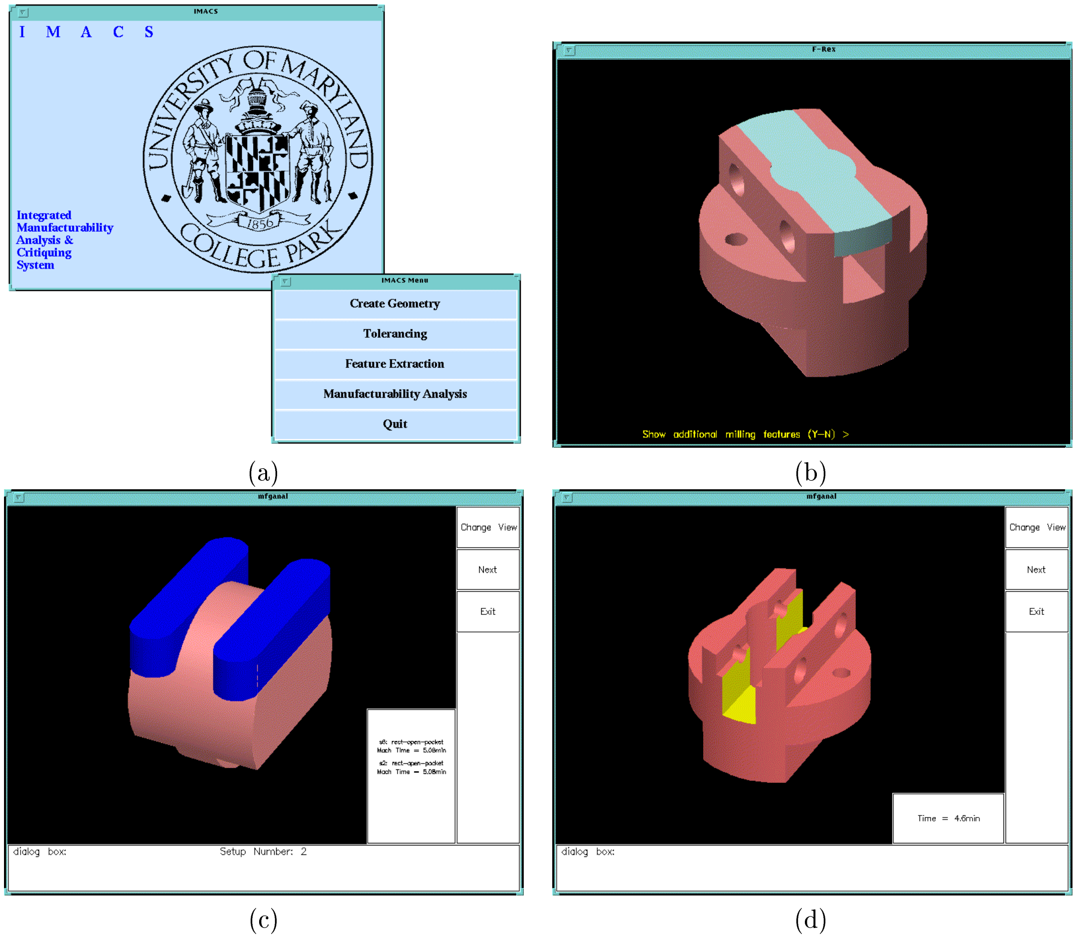


Figure 2: The IMACS interface (a), feature recognition module (b), and manufacturability analysis system (c-d). (d) shows a time estimation for the machining of the highlighted surfaces.

example, if some portion of a workpiece resembles an end-milled slot but the slot is not accessible for machining, then it is of little value to recognize this portion of the workpiece as a end-milled slot.

- **Closed Loop Feature Recognition.**

  In general, there may be several alternative interpretations of the design as different collections of machinable features, each corresponding to a different way to machine the part. Determining which of these alternatives is most preferable requires considering the part's dimensions, tolerances, and surface finishes, along with the availability and capabilities of machine tools and tooling, and fixturability constraints. In most previous approaches, feature recognition was done separately from process planning. Whenever alternative interpretations are present, in absence of any machining knowledge, these systems make an arbitrary choice among various possible alternatives.

  In many cases, there might be very large number of interpretations, making it infeasible to generate all possible interpretations. In such cases, planning information can be used to prune unpromising alternatives. Thus, in order to improve computational efficiency, the feature recognition, plan generation, and plan evaluation steps should be integrated. We use a **closed loop** approach, incorporating plan evaluation into the feature recognition loop. This closed loop architecture allows us to efficiently recognize a select number of feature-based representations likely to produce good process plans. The approach takes advantage of already evaluated feature-based representations in generating new ones.

- **Simulation-Based Planning.**

  Manufacturing knowledge had previously been expressed in terms of the parameters of the manufacturing operation. Decisions made without accounting for the workpiece shape require verification steps. Intermediate part shapes also play a large role in determining the feasibility of plans. Unfortunately, developing closed form formulas that account for arbitrary part shapes is not possible; therefore, we need to perform a detailed simulation of machining operations to verify their feasibility. For example, a number of conditions such as fixturability, accessibility, achievability of tolerances, etc., need to be verified through simulation.

  Most previous approaches that did not perform simulation and planning decisions were based on rigid knowledge-bases. Such systems fail to account for the effect of the intermediate work-piece shapes on feasibility of machining operations.

  IMACS uses a simulation-based planning technique in which it verifies various applicability conditions for machining operations [8]. IMACS's simulation works at multiple levels of abstraction and allows it to reason with incomplete (partial) workpiece shapes.

4 IMACS: A Technical Description

4.1 Terminology and Nomenclature

A machined **part**, P, is the final component created by executing a set of machining operations on a piece of **stock**, S. For example, Figure 3 shows a socket P_0 and the stock S_0 from which P_0 is to be produced. Note that the goal to be achieved (i.e., the part to be produced) is represented not as a set of predicates (as is often done in AI planners), but instead as a CAD model (which IMACS represents using ACIS, a solid modeling system from Spatial Technologies Inc.).

An **operation plan** is a sequence of machining operations capable of creating the part P from the stock S. Since it would be physically impossible to produce P's *exact* geometry, designers give **design tolerance** specifications (e.g., see Figure 4) to specify how much variation from the nominal geometry is allowable in any physical realization of P. A plan is considered capable of achieving the goal if it can create an approximation of P that satisfies the design tolerances.

A **workpiece** is the intermediate object produced by starting with S and performing zero or more machining operations. Currently, the machining operations considered in IMACS include end milling, side milling, face milling and drilling operations, on a three-axis vertical machining center. Each machining operation creates a **machining feature**. Different researchers use different definitions of machining features; as shown in Figure 5, we consider a machining feature to include knowledge about the type of machining operation, the material removal volume (the volume of space in which material can be removed), and the accessibility volume (the volume of space needed for access to the part).

4.2 Feature Recognition

Although much past work on integrating design with manufacturing planning has involved **feature-based design** techniques in which users specified designs directly as sets of form features [3], most researchers have become convinced that a single set of features cannot satisfy the requirements of both design and process planning—instead, some form of **feature extraction** or **feature recognition** is needed.

For IMACS, we had to develop new algorithms to extract machining features directly from the CAD model [17, 18]. Previous work on feature recognition and extraction fell roughly into two categories: (1) specialized geometric algorithms for finding volumetric decompositions of a part to be used for sequencing

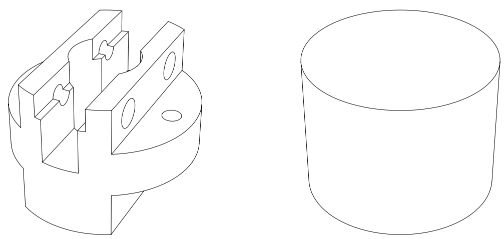


Figure 3: The socket P_0 and the stock S_0.

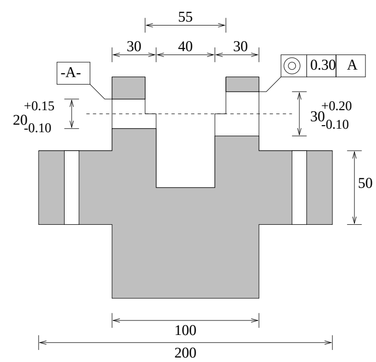


Figure 4: Dimensions and tolerances for the socket P_0

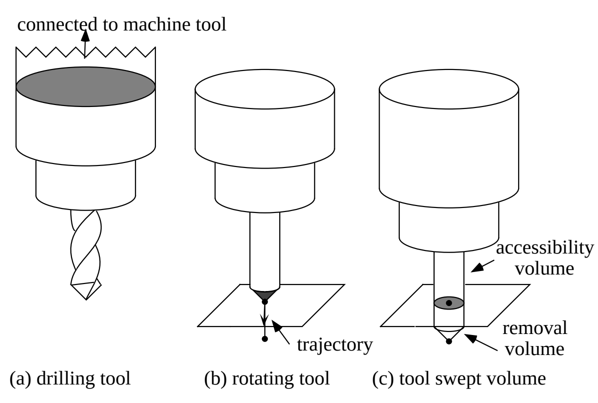


Figure 5: Example of a machining operation.

of machining operations, and (2) pattern analysis techniques that attempt to identify characteristic configurations representing features by examining the data structure for the boundary representation of the part at hand. The majority of this earlier work addressed recognition of **neutral form features**—domain-independent shape features not tightly associated with specific manufacturing and machining knowledge.

We formalized a method for **trace-based** feature recognition, building on earlier ideas of [22]. A **trace** comprises geometry and topology, design features, tolerances, and other forms of design knowledge associated with the CAD model. In this way, traces can be viewed as the information in the solid model of a part that is produced by an instance of a feature. Our two significant contributions in this area were the use of process-specific machining features (and not just form features) and the tight integration of feature extraction with the needs of process planning.

We note that there can be many—sometimes infinitely many—different machining features capable of creating various portions of a given part. Of these, we define a **primary** feature to be a feature that contains as much of the stock as possible without intersecting with the part, and as little space as possible outside the stock. Figure 6 shows examples of primary and non-primary features.

In every operation plan that IMACS will ever want to consider, each machining operation will create either a primary feature or a truncation of a primary feature—and the number of primary features for a part is always finite (in fact, polynomial). Thus, IMACS's first step is to find the set $\mathcal{F}$ of all primary features for P and S. For example, for the socket P_0 the set $\mathcal{F}$ contains 22 primary features, a few of which are shown in Figure 7.

In AI terms, machining operations are elementary actions and machining features are tasks. $\mathcal{F}$ is the set of all tasks that might ever be relevant for achieving the goal (i.e., creating the part). Unlike most AI planners, IMACS finds this set in advance before it begins to generate plans via feature recognition—as we discuss later, this technique may be useful in a number of AI planning problems.

4.3 Generating Incomplete Plans

Figure 7 shows that the features in $\mathcal{F}$ may overlap in complicated ways, and not all of them are needed to create the part (for example, we do not need to machine both $s1$ and $s2$). A **feature-based model** (FBM) is any irredundant subset of features $F \subseteq \mathcal{F}$ such that subtracting those features from S produces P. For example, Figure 8 shows an FBM, FBM1, for the socket P_0.

In AI planning terminology, an FBM is an **incomplete plan**: if we can machine the features in it, this will create the part. Since each FBM is a subset of $\mathcal{F}$, FBMs can be generated using set-covering techniques, but there can be exponentially many FBMs. As an example, for the socket P_0, $\mathcal{F}$ contains 22 primary features from which one can create 512 FBMs. In general, we usually will not want to generate *all* of these FBMs, because only a few of them will lead to good operation plans. Thus IMACS does a depth-first branch-and-bound search to generate and test FBMs one at a time, pruning unpromising FBMs as described in Section 4.7. For example, IMACS generates only 16 of the 512 FBMs for the socket P_0.

In many of the early generative process planning systems [2] the input was a symbolic representation of P as a set of machining features analogous to a single FBM, with no way to recognize or handle many of the geometric interactions among the features. This prevented such systems from generating realistic process plans for complex parts in which geometric interactions can make it quite difficult to decide what sets of features and machining operations to use, which operations to do when and in which setups, and how to hold the workpiece during each setup.

In one way or another, most recent work on generative process planning (both by manufacturing researchers and AI researchers) has tried to address these difficulties (e.g., [13, 22, 4, 1]).

4.4 Resolving Goal Interactions

An FBM represents a totally unordered plan. To resolve goal interactions, IMACS adds ordering constraints according to the following steps:

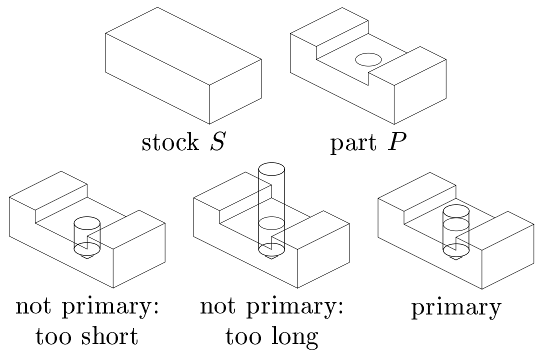


Figure 6: Non-primary and primary drilling features.

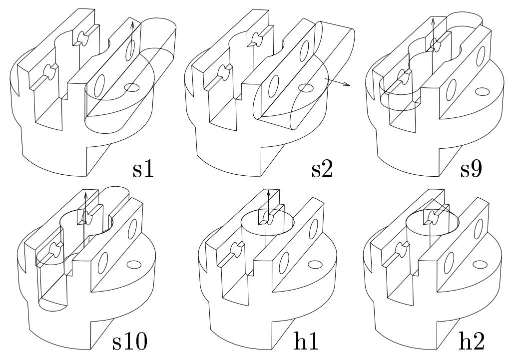


Figure 7: A few of the 22 primary features for the socket P_0. $s1$, $s2$, $s9$, and $s10$ are end-milling features; $h1$ and $h2$ are drilling features.

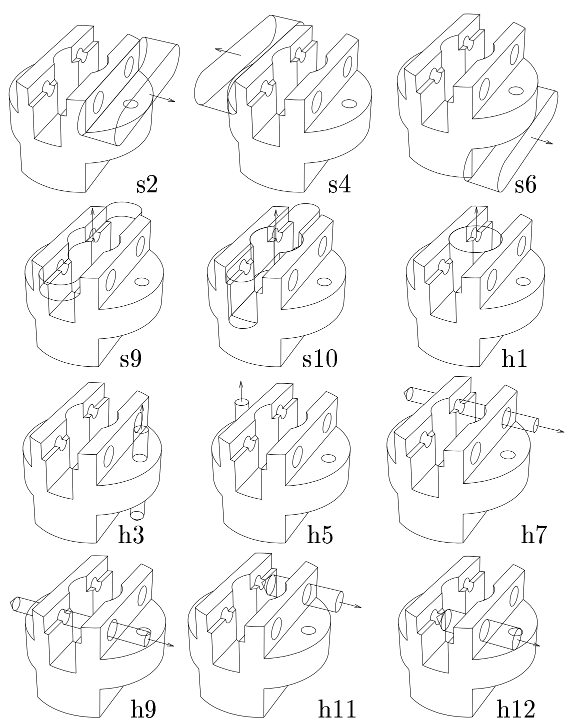


Figure 8: Feature-based model FBM1 for the socket P_0.

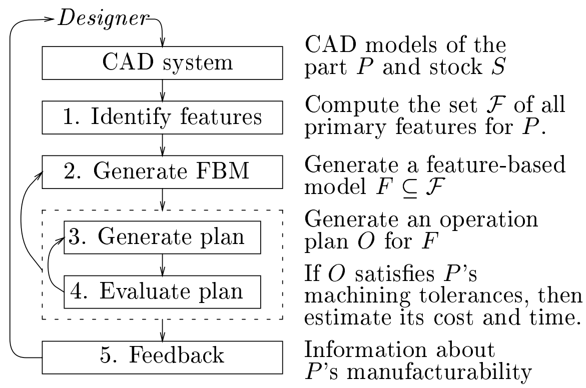


Figure 9: Detailed view of the operation plan generation and evaluation loop in IMACS.

1. **Identify ordering constraints.** Due to complex geometric interactions (accessibility etc.), some features must precede others. For example, in Figure 10, the hole $h1$ must be machined before the slot $s9$ in order to achieve reasonable machining tolerances and avoid tool breakage.

2. **Linearize.** Next IMACS generates all total orderings consistent with the precedences. If no such total ordering can be found, IMACS considers the FBM F to be unmachinable and discards it. Unlike the typical approaches used in AI planners, there would be no point in adding additional operators: they would just create redundant features, and if there is a feasible way to machine the part it will be found among the other FBMs.

3. **Modify goals.** Suppose features f and g overlap, and f precedes g in some total ordering. Then when we machine f, we are also machining part of g. We don't want to machine that same portion of g again later in the sequence, because we would merely be machining air. Thus, IMACS truncates g to remove the portion covered by f. As an example, several of the features shown in Figure 10(a) were produced by truncating the corresponding features in FBM1.

4. **Unlinearize.** Once the truncated features have been produced, several of the resulting FBMs may have identical features but different precedence constraints. In such cases the precedence constraints that differ can be removed, translating the total orders into partial orders. For example, Figure 10(b) shows the partial order for the FBM of Figure 10(a).

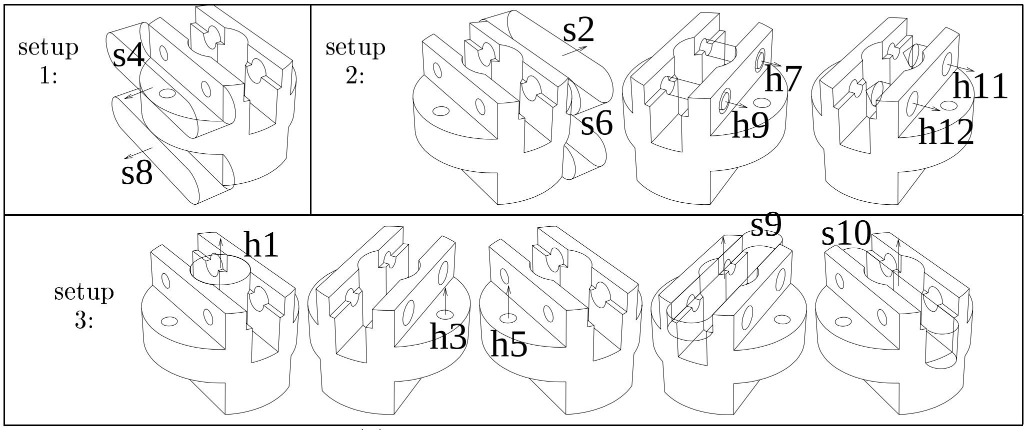
4.5 Additional Steps

To obtain an operation plan from the partially ordered FBM, IMACS uses these additional steps:

- **Incorporate finishing operations.** For faces with tight surface finishes or tolerances, IMACS adds finishing operations, with precedence constraints to make them be completed after the corresponding roughing operations. Currently one finishing operation per face is allowed.

- **Determine setups.** On a three-axis vertical machining center, features cannot be machined in the same setup unless they have the same approach direction. This knowledge, along with the partial ordering constraints, can be used to determine which features can be machined in the same setup, as shown in Figure 10(b). Although the specific computations are different, the problem is a special case of what is known to AI researchers as the plan-merging problem [6, 1].

- **Determine process details.** To select cutting parameters such as those shown in Figure 10(c), IMACS uses the recommendations of the Machinability Data Center's handbook. The maximum recommended cutting parameters are used, rather than attempting to select optimal cutting parameters; thus IMACS's estimates involve considerable approximation.



(a) features to be machined

(c) process details

Feature name	Feature type	Tool diam (mm)	Feed rate (mm/min)	Number of passes	Pass length (mm)
$s4$	end-milling	50	166	2	225
$s8$	end-milling	50	166	2	225
$s2$	end-milling	50	166	2	225
$s6$	end-milling	50	166	2	225
$h7$	drilling	20	244	1	106
$h9$	drilling	20	244	1	106
$h11$	drilling	30	203	1	39
$h12$	drilling	30	203	1	39
$h1$	drilling	75	108	1	172.5
$h3$	drilling	20	244	1	56
$h5$	drilling	20	244	1	56
$s9$	end-milling	50	166	1	250
$s10$	end-milling	40	207	3	240

(b) ordering constraints

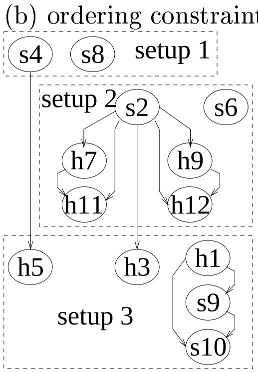


Figure 10: An operation plan derived from FBM1. This plan is the optimal one for making P_0. Note that each feature is either a primary feature from FBM1 or a truncation of a primary feature from FBM1.

As shown in Figure 11, these steps correspond to a task decomposition somewhat analogous to that used in Hierarchical Task Network (HTN) Planning [19, 21, 23, 25, 12].

Since each FBM can lead to several different operation plans, IMACS does the above steps inside a depth-first branch-and-bound search, evaluating the plans as described in Section 4.6 in order to find the optimal operation plan. Figure 10 shows the operation plan IMACS finds for the socket P_0.

4.6 Operation Plan Evaluation

Once IMACS has found an operation plan, it evaluates whether the plan can achieve the design tolerances. To verify whether a given operation plan will satisfy the design tolerances, IMACS must estimate what tolerances the operations can achieve. Typical approaches for computer-aided tolerance charting are computationally very intensive, and only consider limited types of tolerances. Therefore, IMACS simply evaluates the manufacturability aspects of a wide variety of tolerances without getting into optimization aspects. As an example, the operation plan shown in Figure 10 satisfies the tolerances shown in Figure 4, and thus is an acceptable way to make P_0 from S_0.

If the plan can achieve the design tolerances, then IMACS estimates the plan's manufacturing time.

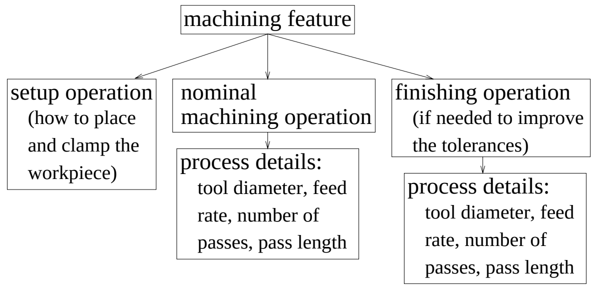


Figure 11: Task decomposition in IMACS.

Operation	Time (min)	Operation	Time (min)
drill h1	2.3	mill s2	5.0
drill h3	0.3	mill s4	5.0
drill h5	0.3	mill s6	5.0
drill h7	0.6	mill s8	5.0
drill h9	0.6	mill s9	4.0
drill h11	0.3	mill s10	4.2
drill h12	0.3	3 setups	6.0

Total Time: 39 minutes

Table 1: Estimated production time for the operation plan shown in Figure 10.

The total time of a machining operation consists of the cutting time (when the tool is actually engaged in machining), plus the non-cutting time (tool-change time, setup time, etc.). Methods have been developed for estimating the fixed and variable costs of machining operations; our formulas for estimating these costs are based on standard handbooks related to machining economics, such as [24]. As an example, Table 1 shows the estimated production time for the operation plan of Figure 10.

4.7 Efficiency Considerations

IMACS uses a depth-first branch-and-bound search to generate and evaluate FBMs and plans one FBM at a time. By evaluating them as they are being generated and keeping track of the best one it has seen so far, IMACS can discard FBMs and plans that look unpromising, even before they have been fully generated. For example, from the 22 primary features shown in Figure 7 one can form 512 FBMs for the socket P_0, but IMACS generates only 16 of these FBMs. The search of the space for the socket example is shown in Table 2.

Below are some of IMACS's pruning criteria, which can be thought of as similar to notion of *critics* in HTN planning:

- IMACS will discard an FBM if it contains features whose dimensions and tolerances appear unreasonable. Examples would include a hole-drilling operation having a length-to-diameter ratio that is too large; a recess-boring operation having a ratio of outer diameter to inner diameter that is too large; and two concentric hole-drilling operations with tight concentricity tolerance and opposite approach directions.

- IMACS will discard an FBM if it appears that there will be problems with work-holding during some of the machining operations. Currently, IMACS's work-holding analysis is based on the assumption that a flat-jaw vise is the only available fixturing device [4].

- IMACS will compute a quick lower bound on the machining time required for an FBM or plan, and will discard the FBM or plan if this lower bound is above the time required by the best plan seen so far.

```
Generating and Evaluating FBMS...
 found a FBM: [ s9 h12 s10 h11 s3 s5 h9 h4 s7 h2 h6 h7 s1 ]
              PLAN Evaluation->*setup problems* unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h9 h4 s7 h1 h6 h7 s1 ]
              PLAN Evaluation->*setup problems* unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h9 h3 s7 h2 h5 h7 s1 ]
              PLAN Evaluation->*setup problems* unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h9 h3 s7 h1 h5 h7 s1 ]
              PLAN Evaluation->*setup problems* unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h10 h4 s7 h2 h6 h8 s1 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h10 h4 s7 h1 h6 h8 s1 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h10 h3 s7 h2 h5 h8 s1 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s3 s5 h10 h3 s7 h1 h5 h8 s1 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h9 h3 s8 h1 h5 h7 s2 ]
              PLAN Evaluation->*updating current-best plan*
 found a FBM: [ s9 h12 s10 h11 s4 s6 h9 h3 s8 h2 h5 h7 s2 ]
              PLAN Evaluation-> unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h9 h4 s8 h2 h6 h7 s2 ]
              PLAN Evaluation-> unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h9 h4 s8 h1 h6 h7 s2 ]
              PLAN Evaluation-> unpromising PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h10 h3 s8 h1 h5 h8 s2 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h10 h3 s8 h2 h5 h8 s2 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h10 h4 s8 h2 h6 h8 s2 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
 found a FBM: [ s9 h12 s10 h11 s4 s6 h10 h4 s8 h1 h6 h8 s2 ]
              FBM check ->*toler. incomp.* among h8 and h11 PRUNED
Best FBM: [ s9 h12 s10 h11 s4 s6 h9 h3 s8 h1 h5 h7 s2 ]
Total number of evaluated FBMs: 8
```

Table 2: The search of the space of operation plans for the socket in Figure 3.

4.8 Generating Suggestions for Redesign

Given an interpretation of the design as a collection of machinable features, IMACS can generate alternative machining features by making geometric changes to the original features. These features are added into the feature set of the original part to create an extended feature set. The designer may provide restrictions on the design indicating the type and extent of modifications allowed on certain faces and volumes. All redesign suggestions generated by our approach honor those restrictions provided by the designer.

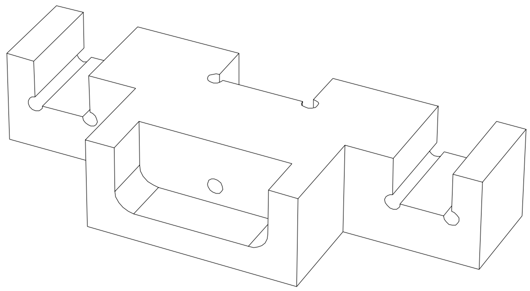
By taking combinations of features from the extended feature set generated above, we can generate modified versions of the original design that still satisfy the designer's intent, such as the design shown in Figure 12. Figure 12 (a) shows the design of a bracket, and Figures 12 (b) and (c) show variations on (a) that reduce manufacturing cost.

By considering precedence constraints and approach directions for the machining operations as well as simple fixturability constraints, we can estimate the setup time that will be required for each design. Any modified design whose setup time is less than that of the original design can be presented to the designer as a possible way to modify the original design.

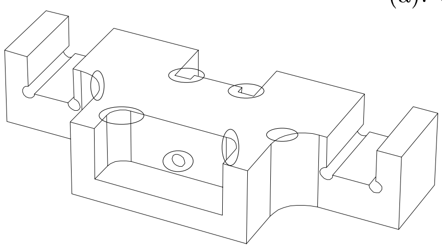
5 IMACS Compared to Classical AI Planning

An abstract version of the process planning problem appears to be a special case AI planning problem. Consider the following formulation:

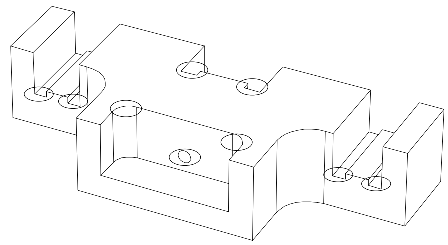
- the manufacturing workpiece is a state in the planning search space;
- each manufacturing operation is a planning operator which transforms a given state to some other state.



(a): a bracket



(b): a modified version of the bracket with the modifications circled.



(c): a second modified version of the bracket with the modifications circled.

Figure 12: An example part and two versions redesigned to minimize setup cost.

Using this formulation, one can assume that the stock material is the initial state and the desired part is the goal state. The process planning problem can now be defined as the problem of finding a sequence of operations which transform the initial state to the goal state. So in an abstract sense, IMACS planning techniques are very similar to that of classical AI planning. However, there are several important differences between the techniques used in IMACS and those used in classical AI planning systems. For example, Figure 1 reveals two immediate differences between IMACS and many AI planning systems:

1. Unlike most AI planners, IMACS generates more than one plan and evaluates the merit of each plan it generates, to find an optimal plan. To measure plan merit, IMACS uses an estimate of the plan's manufacturing time, as described in Section 4.6. However, it is feasible to incorporate estimates of production cost as well.

2. We have developed algorithms for IMACS to suggest changes in the design to improve its manufacturability while still fulfilling the designer's intent [4], as illustrated in Figure 12. In AI terms, this means that IMACS can automatically suggest changes to the goal to make it easier to achieve.

In retrospect, however, we see in IMACS numerous manifestations of fundamental AI problems. Some of the techniques we developed in the manufacturing domain may prove valuable in analyzing more general AI planning problems.

5.1 State-Space Search in IMACS

IMACS computes a set of possible state transitions in advance by recognizing features. The technique of finding all primary features before beginning to generate plans can be generalized as follows:

- Enumerate the set of all tasks that might ever be relevant. Call this set $\mathcal{F}$.

- Loop:
  - Generate an incomplete plan F as a subset of $\mathcal{F}$
  - If the plan F has a goal interaction that can't be resolved via machining operation precedence constraints, discard it. (If a promising plan exists, it will be generated in another loop iteration.)
  - Flesh out the plan (using task decomposition, critics, plan merging, etc.)

This technique should be useful whenever it is feasible to enumerate in advance the set $\mathcal{F}$ of all relevant tasks. More specifically, suppose that we can construct $\mathcal{F}$ in polynomial time, and that each task in $\mathcal{F}$ will need to be achieved at most once. Then every plan we will care to consider is a subset $F \subseteq \mathcal{F}$, and we can generate these plans nondeterministically in polynomial time. If each goal interaction involves at most a constant number of tasks, then we can determine in polynomial time whether there are ordering constraints sufficient to make F a successful plan.

This idea helps to explain a puzzling theoretical problem. In the worst case, planning with STRIPS-style operators is PSPACE-complete [5], but the best known example of STRIPS-style planning is blocks-world planning, which is only NP-complete [7]. This discrepancy can be explained by noting that in a blocks-world problem containing n blocks there are only at most $2n$ possible relevant tasks: for each block b, we might want to move b to the table, and if the goal state contains $on(b, c)$ for some c, then we will want to move b to c.

5.2 The Frame Problem in Process Planning

The *Frame Problem*, as it is known in classical AI planning, is the problem of specifying which conditions in a state description should change (and which should not change) after applying a given operator [16].

Manufacturing planning is a highly interconnected domain where small choices can have significant, and often unpredictable effects. Traditional process planning systems avoided these intricacies by modeling the world in a *STRIPS-style* with purely geometric *form features*[2] and process rules—for example, the occurrence of a *hole* feature in the solid model of a part implies that one or more drilling operations will go into the process plan.

The problem arises that simple STRIPS-style rules do not capture the complexity of manufacturing process planning. To address this, we employed features based on machining process models. Interactions among the features can then be more efficiently handled at both geometric and operation-planning levels.

6 Future Research Challenges

6.1 Computational Challenges

Traditional AI planning and search problems (such as blocks-world planning, 15-puzzle, chess playing etc.) involve extensive symbolic computation. For example, the search tree for games of even moderate complexity (such as chess) might contain many millions of nodes, each with its own representation of the internal state of the world (a board, location of pieces, etc.). Expanding a given state in the search to examine others involves the application of symbolic operators (moving pieces) and the application of heuristics (comparing the relative quality of nodes) to determine which path to follow through the search space. For manufacturing planning domains, the search space can be orders of magnitude larger and good operators hard to find.

In the discrete domains of classical AI, computational activity concerns purely symbolic actions, computed with integer representations[3] modeling the state of the world and the transitions in it: comparing two states in a search can often be performed as a short fixed-length sequence of comparison operations on these integers; applying operators to generate new states is accomplished by a series of discrete arithmetic

[2] A *form feature* is a general shape template characterizing local geometric configurations on the model of the final part.

[3] Note that these integer representations include strings and text, which are represented in ASCII as sequences of integers.

operations; evaluating heuristic functions is also primarily an integer processing task, although the values calculated might be floating point numbers.

Automated (or semi-automated) manufacturing-operation planning involves both symbolic computation and extensive geometric reasoning. Geometric reasoning operations range from simple queries to a solid modeler (e.g., what is the location of this point?) to complex calculations requiring extensive processing (e.g., list all planar surfaces with normal vectors parallel to v.).

Simple application of classical AI strategies to these domains creates several serious ramifications. Because of these issues, AI techniques have proven difficult to scale to complex, real-world, parts—those having thousands of geometric and topological entities and possibly hundreds of interacting feature instances.

Size of the search space. In process planning for complex mechanical parts, the search spaces can be quite large, with even the most basic manufacturing domains being on the order of complexity of games such as chess. Planning the selection of tooling, tool holders, and manufacturing operations creates a huge set of possibilities, and sequencing the operations can introduce combinatorially many possible orderings.

This problem is compounded by the fact that the amount of memory needed to represent each state in the state space can be quite large. In process planning problems, a state would normally be represented by a solid model with symbolic attributes attached to the geometry. For non-trivial product models, explicitly storing even a few of these states requires tremendous amounts of memory: solid models for designs of realistic mechanical parts and assemblies can run into gigabyte size (and worse)—resulting in search spaces requiring terabytes of memory.

To address this, one requires a dual representation for planning states. In IMACS, a feature-based representation (which is relatively compact) is used to store states. During planning, an attributed boundary representation (larger) is used.

Floating point calculations. Floating point calculations are used extensively in geometric reasoning applications. Consider the relatively commonplace operation of testing equality. For purely symbolic systems, comparing integers or strings is a matter of some fixed number of calls to the CPU—e.g., comparing two strings of length n requires $O(n)$ operations.

For geometry-centric domains, the data elements are far more complex. Even the most basic task, such as comparing whether or not two points in 3-dimensional space are equal, can be computationally difficult. Geometric entities are represented with floating point numbers (typically of double precision) and, because floating point computation is notoriously unstable and inconsistent, comparisons require use of epsilon values. Comparing two lists of points still requires an $O(n)$ algorithm, but one with a constant factor perhaps an order of magnitude larger than the algorithm integer comparison. Comparing more complex geometric entities, such as faces and solid bodies, requires algorithms that are even more elaborate.

Speed. Speed is a serious issue for process planning systems. After one has accounted for all of the inconsistencies introduced by floating point numbers, most geometric queries are executed through calls to geometric routines in a solid modeling kernel. Such computation is much more time consuming than the integer-driven symbolic computation performed in more traditional AI planning systems. This computational complexity results in a node generation time which is two to three orders of magnitude larger.

6.2 System Requirements Challenges

Most automated process planning systems are used in environments that are different from traditional planning problems and this imposes a number of unique requirements on the builders of process planning systems.

Techniques for selecting plans. In most situations, a human process engineer has to be the final judge of the quality of the plan generated by an automated planning system. This raises a very interesting research issue in case of complex planning problems: how to design an objective function that can distinguish among very good plans. For example, if two holes were to be drilled on a block, it is very difficult to decide which hole should be drilled first based on the machining time alone.

It is often the case, however, that because of other factors (such as proximity of one of the holes to the side wall of the block) a human planner might have a strong bias towards one of the two choices. Human planners often use many ill-defined notions such as probability of successfully completing the plan, minimizing risks of resource failure, etc., in making their selections. The experience of the human planner is very hard to capture in a quantitative model for use in a computerized system. There are two ways of augmenting AI planning systems that will help address these problems:

1. **Use of multiple objective functions.** An initial set of quantitative, cost-based objective functions can be used to discard undesirable plans. Another set of objective functions is needed to perform comparisons among good plans. Such objective functions can be used to rate plans based on probability of success, risk to manufacturing resources etc. Moreover, additional customizable objective functions should be available that can be modified to enforce user preferences.

2. **User guided search.** Users must participate at some level in the search process. For example, a user should be able specify some strategic decisions such as "prefer face milling over end milling" for a given part. When a system cannot distinguish among two good alternatives, it should allow the user to break the tie.

Improved planning strategies. One main difference between the process planning problem and other general AI planning problems is node processing time (i.e., the CPU time needed to generate a new state from a given state). Each state in the process planning problem includes one or more geometric models and computing new states requires significant amounts of geometric computation. This makes process planning problems computationally intensive, implying that planning for a complex part requires that we incorporate domain-specific knowledge to ensure that we are not generating unpromising nodes which will later be discarded.

Some of the techniques which can be used in a variety of manufacturing domains (most especially machining) are described below:

- **Utilizing geometric symmetry.** Geometric symmetries in the part can be used to reduce the number of nodes that need to be processed. For example, symmetric portions of the part should be created by similar operations.

- **Utilizing geometric patterns.** Many parts have patterns of geometric features (e.g., a rectangular array of holes). Such patterns can be treated as a single manufacturing operation and significant numbers of nodes can be eliminated from the search space.

- **Applying a cluster of operators.** This allows generation of a state which is more than one operator away from the given state without explicitly generating intermediate states. For example, consider a hole that needs to be drilled using three different drilling operations. In this situation, it is better to treat these three holes as a single cluster of operators without ever needing to generate two intermediate states explicitly.

Integrating interactions with world. Most classical AI planning systems operate under a **closed world** assumption: it is implicitly assumed that these systems will be used solve planning problems in a stand-alone mode and will have little need to interact with rest of the world. In the development of such systems, the emphasis has been on how to implement the best possible algorithm for solving the particular planning problem.

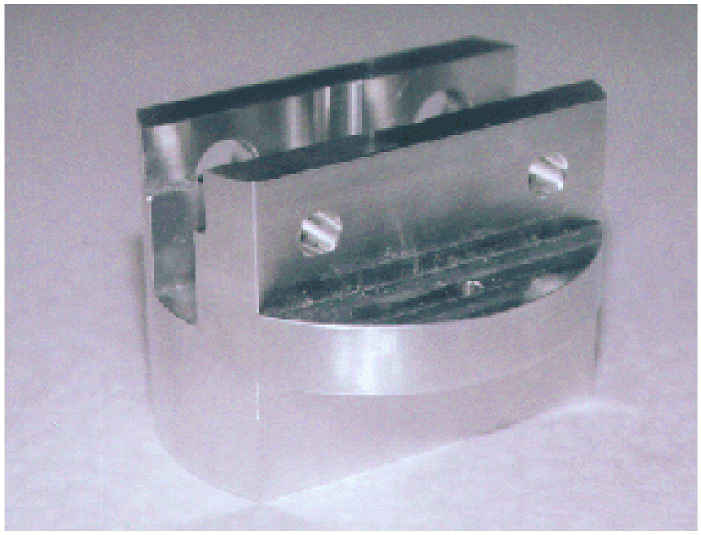


Figure 13: The final machined socket.

Process planning is a bridge between computer-aided design (CAD) and production planning. In the modern manufacturing workplace, input for a process planning system is received from a CAD system and the output of a planning system is used by production planning and scheduling systems to generate production plans. Each of these systems must interact with the process planning system—for example, CAD activity can benefit from feedback on how to improve the proposed design. For this purpose, it may be desirable for the planning system to explore non-optimal plans because they may provide good clues for how to modify and improve the design. Hence, process planning systems need to support "what-if" situations.

During production planning, the status of various manufacturing resources can often introduce dynamic constraints that need to be met for a process to fit well within the factory schedule. For example, a process planning system might generate a plan which uses some resource A before using some resource B, but from the point of view of production planning, it might be better if these two operations were interchanged. The ability to re-plan based on such considerations is needed for process planning systems to work well in real-life situations.

7 Conclusions

Our past interaction with AI researchers and manufacturing researchers seem to indicate the following:

- Since **AI planning researchers** are usually more interested in general conceptual problems than domain-dependent details, the AI approach to manufacturing planning typically has been to create an abstract problem representation that omits unimportant details, and look for ways to solve the abstract problem. From the viewpoint of the manufacturing engineer, these "unimportant details" often are very important parts of the problem to be solved—and this can lead manufacturing engineers to view AI planning techniques as impractical.

- **Manufacturing process planning researchers** typically want to solve a particular manufacturing problem, and present their research results within the context of this problem, without discussing how the approach might generalize to other planning domains. For AI researchers, this makes it difficult to see what the underlying conceptual problems are, or whether the approach embodies a general idea that can be applied to other problems. This can lead AI planning researchers to view manufacturing planning as a domain full of ad-hoc, domain-specific programs rather than general principles and approaches.

Our goal in IMACS project has been to capitalize on the strengths of both domains and create a planner that is based on sound principles and yet can handle realistic parts. IMACS shows that it is possible to address manufacturing process planning both realistically and in a principled manner. During the course of this work we developed a new approach based on the existence of the multiple feature-based representations for mechanical artifacts, such as the UMD Socket pictured in Figure 13. The techniques developed in our research overcome the limitations of previous systems in the following ways:

1. **Accurate Process Modeling Using Machining Features** to represent the capability of the manufacturing processes. Our features not only have shape information but also allow us to capture a wide variety of process constraints imposed by the workpiece shape.

2. **Closed Loop Feature Recognition** to effectively integrate planning knowledge with geometric reasoning, thus incorporating plan evaluation in the feature recognition loop. IMACS's closed loop architecture allows us to recognize the select feature-based representations that are most appropriate for process planning.

3. **Simulation-Based Planning** to verify machining operations. IMACS's simulation works at multiple levels and reasons with approximate workpiece shapes.

We hope that the results presented in this paper will help AI researchers discover ways to effectively apply AI techniques to manufacturing process planning in a realistic manner, and possibly to discover issues arising in manufacturing that may be applicable to AI planning in general.

Acknowledgments

This work was supported in part by NSF Grants IRI-93-06580, DDM-92-01779, NSFD EEC-94-02384, the National Institute of Standards and Technology, National Research Council, and a forgivable loan from General Electric Corporation awarded to William Regli. Software support was received through a grant to the University of Maryland from Bentley Systems Incorporated.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, or the other supporting government and commercial organizations.

ACIS `.sat`, Bentley MicroStation `.dgn`, and STEP AP203 format solid models for the example parts in this paper can be obtained through the World Wide Web through the NIST Design, Process Planning, and Assembly Repository—please see URL `http://www.parts.nist.gov` for more details.

Further information about IMACS, including color images produced using it, are available at `http://www.cs.umd.edu/projects/cim/imacs/imacs.html`.

References

[1] J. Britanik and M. Marefat. Hierarchical plan merging with applications to process planning. In *IJCAI-95*, 1995.

[2] T. C. Chang and R. A. Wysk. *An Introduction to Automated Process Planning Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[3] M. R. Cutkosky and J. M. Tenenbaum. Toward a framework for concurrent design. *International Journal of Systems Automation: Research and Applications*, 1(3):239–261, 1992.

[4] Diganta Das, Satyandra K. Gupta, and Dana S. Nau. Generating redesign suggestions to reduce setup cost: A step towards automated redesign. *Computer Aided Design*, 1996.

[5] K. Erol, D. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 1995.

[6] David Foulser, Ming Li, and Qiang Yang. Theory and algorithms for plan merging. *Artificial Intelligence*, 57(2-3):143–182, 1992.

[7] Naresh Gupta and Dana S. Nau. On the complexity of blocks-world planning. *Artificial Intelligence*, 56(2-3):223–254, August 1992.

[8] S. K. Gupta and D. S. Nau. Systematic approach to analyzing the manufacturability of machi ned parts. *Computer Aided Design*, 27(5):323–342, 1995.

[9] Inyong Ham and Stephen C.-Y. Lu. Computer-aided process planning: The present and the future. *Annals of the CIRP*, 37(2):591, 1988.

[10] C. C. Hayes and P. Wright. Automatic process planning: using feature interaction to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[11] Caroline C. Hayes. P^3: A process planner for manufacturability analysis. *IEEE Transactions on Robotics and Automation*, 12(2):220–234, April 1996.

[12] S. Kambhampati and J. Hendler. A validation structure based theory of plan modification and reuse. *Artificial Intelligence*, May 1992.

[13] Subbarao Kambhampati, Mark Cutkosky, Jay Tenenbaum, and Soo Hong Lee. Integrating general purpose planners and specialized reasoners: Case study of a hybrid planning architecture. *IEEE Transactions on Systems, Man and Cybernetics (Special Issue on Planning and Scheduling)*, 1992.

[14] M. Marefat and R. L. Kashyap. Geometric reasoning for recognition of three-dimensional object features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):949–965, October 1990.

[15] James L. Nevins and Daniel E. Whitney. *Concurrent Design of Products and Processes: A strategy for the next generation in manufacturing*. McGraw-Hill Publishing Company, 1989.

[16] Nils Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers Incorporated, CA, 1980.

[17] W. C. Regli, S. K. Gupta, and D. S. Nau. Toward multiprocessor feature recognition. *Computer Aided Design*, 29(1):37–51, 1997.

[18] William C. Regli, Satyandra K. Gupta, and Dana S. Nau. Extracting alternative machining features: An algorithmic approach. *Research in Engineering Design*, 7(3):173–192, 1995.

[19] E. D. Sacerdoti. *A Structure for Plans and Behavior*. American Elsevier, 1977.

[20] Jami Shah, Martti Mäntylä, and Dana Nau, editors. *Advances in Feature Based Manufacturing*. Elsevier/North Holland, 1994.

[21] Austin Tate. Generating project networks. In *Proc. 5th International Joint Conf. Artificial Intelligence*, 1977.

[22] J. H. Vandenbrande and A. A. G. Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1269–1285, December 1993.

[23] David E. Wilkins. Domain-independent planning: Representation and plan generation. In James Allen, James Hendler, and Austin Tate, editors, *Readings in Planning*, pages 319–335. Morgan Kaufmann, 1990. Originally appeared in *Artificial Intelligence* 22(3), April 1984.

[24] W. Winchell. *Realistic Cost Estimating for Manufacturing.* Society of Manufacturing Engineers, 1989.

[25] Q. Yang. Formalizing planning knowledge for hierarchical planning. *Computational Intelligence*, 6:12–24, 1990.