TECHNICAL RESEARCH REPORT

Reducing Setup Cost by Automated Generation of Redesign Suggestions

by D. Das, S.K. Gupta, and D.S. Nau

T.R. 94-12

ISR
INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Reducing Setup Cost by Automated Generation of Redesign Suggestions*

Diganta Das[†] Satyandra K. Gupta[‡] Dana S. Nau[§]

University of Maryland
College Park, MD 20742 USA

Abstract

All mechanical designs pass through a series of formal and informal redesign steps, involving the analysis of functionality, manufacturability, cost and other life-cycle factors. The speed and efficacy of these steps has a major influence on the lead time of the product from conceptualization to launching.

In this paper we propose a methodology for automatically generating redesign suggestions for reducing setup costs for machined parts. Our approach is based on interpreting the design as a collection of machinable features. Our methodology generates alternate machining features by making geometric changes to the part, and adds them to the feature set of the original part. The designer may provide restrictions indicating that certain surfaces and volumes should not be changed, in which case all redesign suggestions generated by our approach honor those restrictions. Using features from the extended feature set generated above, one or more new designs may be found that need fewer setups than the original part.

Keywords: redesign, manufacturability analysis, generation of alternative features

*This work was supported in part by NSF Grants NSFD CDR-88003012, IRI9306580, and DDM-9201779.
[†]Institute for Systems Research and Department of Mechanical Engineering. Email: diganta@src.umd.edu
[‡]Institute for Systems Research and Department of Mechanical Engineering. Email: skgupta@src.umd.edu
[§]Institute for Systems Research and Department of Computer Science. Email: nau@cs.umd.edu

1 Introduction

In all component design procedures, the design goes through a design cycle consisting of analysis and review of the design for cost effectiveness and quality. Ideally, the design review would take into account the capabilities and costs of the production processes to be used. However, it is not always possible to do this for all facets of the production process, particularly for complicated methods such as machining. After the component enters the production cycle, experienced process planners and machinists may discover that alterations in the design would be beneficial—but few companies have organizational structures that enable the design team to take advantage of this information. If tools were available at the design stage to suggest design revisions for cost containment, this would help in reducing the product realization cost. This paper describes a first step toward the development of such a tool.

The production cost of a machined component comes from many factors—but one of the biggest factors is the number of setups it takes to machine the component. Reducing the number of setups will reduce the machine's idle time, and will require fewer work-holding devices. Furthermore, reducing the number of setups will result in better machining tolerances. In this paper, we describe a structured methodology for generating possible modifications for reducing the number of setups it takes to machine the component.

The basic steps of the redesign scheme are as follows:

Initial Step: Preprocessing. Get the design of the part P from the designer. The designer may also provide restrictions indicating that certain surfaces and volumes should not be changed, in which case all redesign suggestions generated by our approach will honor those restrictions. This step in described in Section 4.

Step 1: Analyze the current design.

- **Step 1a.** Find all the possible machining features in the original part P which can be removed from the stock S to produce P (for details, see [5]). Put all these features into the set $\mathcal{F}$.
- **Step 1b.** Find the precedence constraints on the order in which the features in $\mathcal{F}$ can be machined, as described in Section 5.
- **Step 1c.** As described in Section 5, find the lowest number of setups in which P can be machined from S using the features in the set $\mathcal{F}$. This involves examining *Feature-Based Models* (FBM's) in $\mathcal{F}$; these are subsets of $\mathcal{F}$ that contain no redundant features and are sufficient to create P.

Step 2: Generate possible feature modifications. For each feature $f \in \mathcal{F}$, use feature modification operators (see Section 6) to generate alternate features for f. These alternate features will have different geometry from f, but will satisfy the designer's restrictions. Let $\mathcal{F}'$ be the set of all of the old and new features.

Step 3. Generate and present design alternatives.

- **Step 3a.** Determine precedence constraints among the features of $\mathcal{F}'$ (see Section 5.3).
- **Step 3b.** If FBM's can be found in $\mathcal{F}'$ that require fewer setups than the original part, then present them to the designer as redesign suggestions (see Section 7).

The paper is organized as follows. Section 2 contains definitions. Section 3 reviews related work. Sections 4 through 7 describe the details of our approach, with an example to explain how the procedure works. Finally, Section 8 includes concluding remarks and ideas for future work.

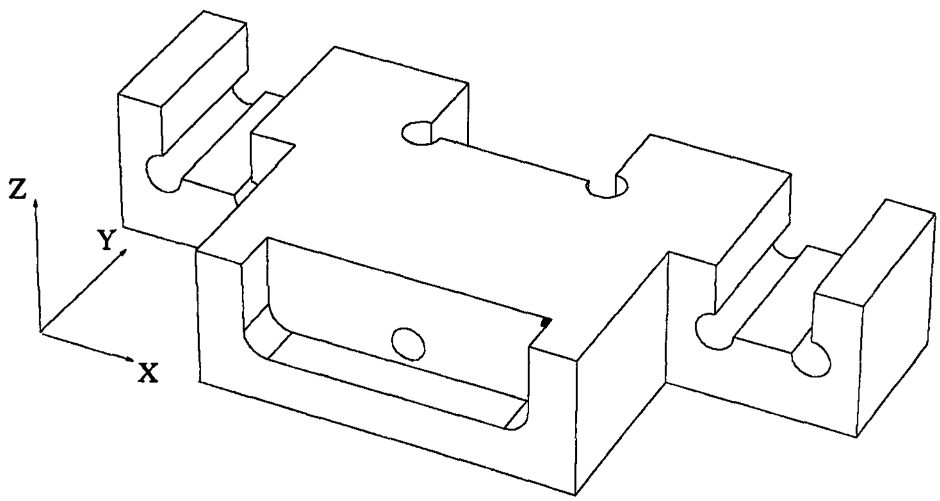


Figure 1: An example part, which we will call P1.

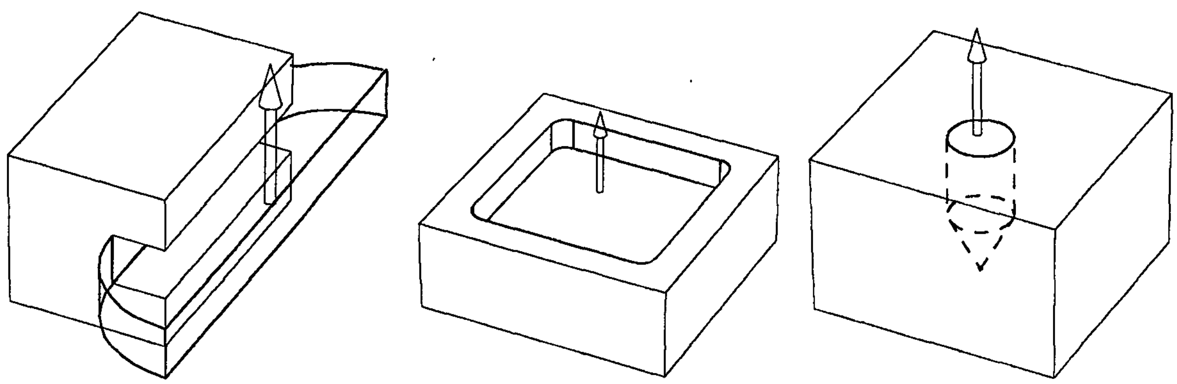
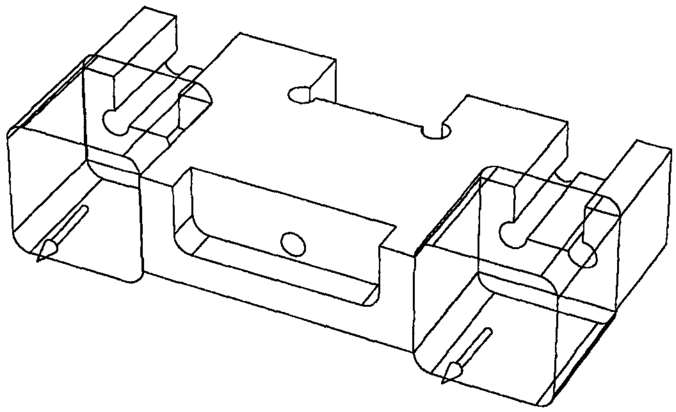


Figure 2: Machining features: slot milling, end milling and drilling.

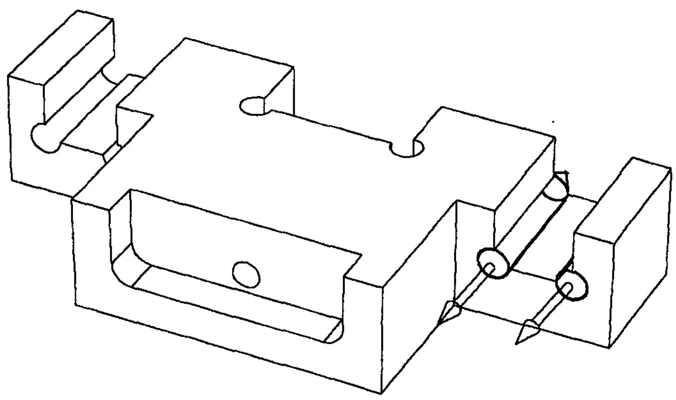
2 Definitions and Notation

A *part*, P, is the final component created by executing a set of machining operations on a piece of *stock*, S. A *workpiece* is the intermediate object produced by performing zero or more of the operations needed to create P. To represent P and S, we use geometric solids (cf. [14]). For example, Figure 1 shows an example part which we will call P1; this part would typically be machined from a rectangular piece of stock.

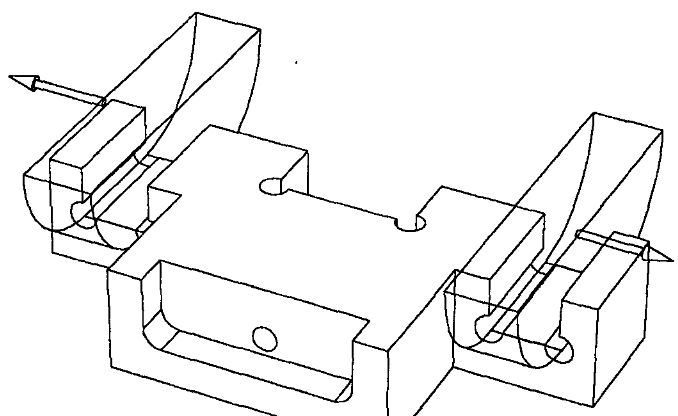
A *machining feature* is a portion of the workpiece affected by a particular machining operation. A machining feature consists of three components: the volume removed, the *approach direction* (the direction from which the operation is performed), and the type of operation. In this paper, the only types of operations we will consider are end milling, slot milling, and drilling, in a vertical machining center. Each machining operation is capable of creating certain types of surfaces: drilling produces cylindrical and conical surfaces, and end milling and slot milling produce planar and cylindrical surfaces. The basic three types of machining features used in this paper are shown in Figure 2. As an example, Figure 3 shows some of the end milling, slot milling, and drilling features for the part P1.



a) Some end-milling features



b) Some drilling features



c) Some slot-milling features

Figure 3: Examples of machining features for the part P1.

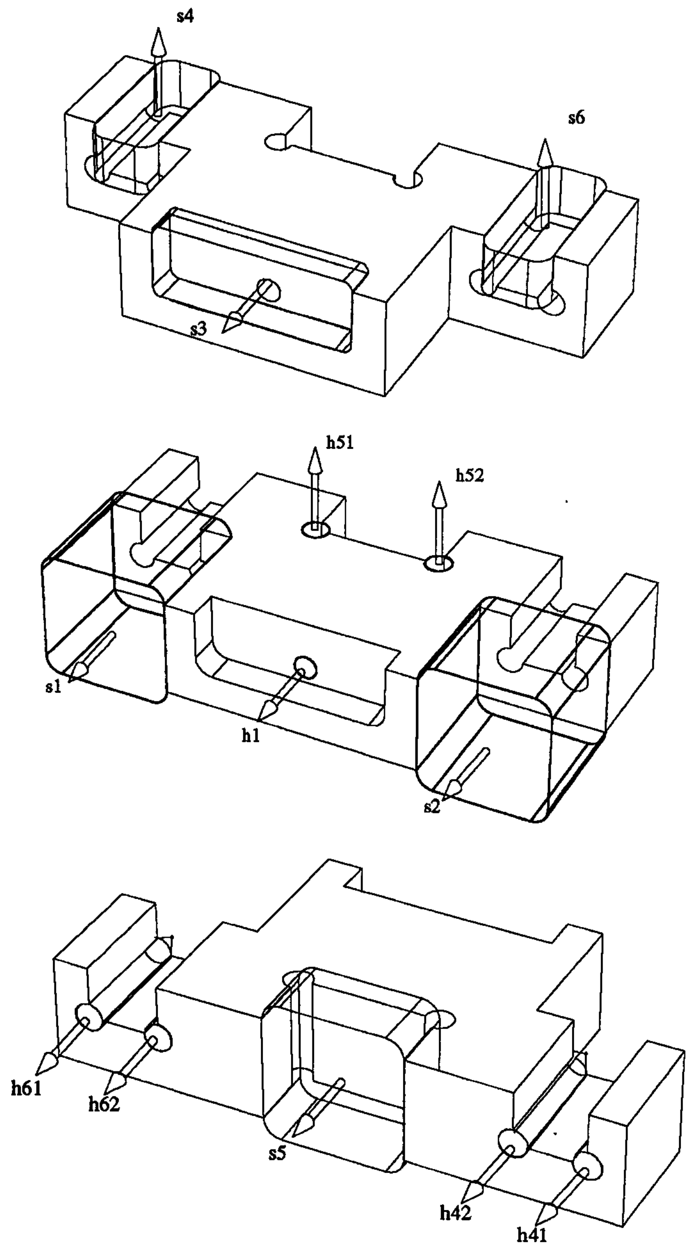


Figure 4: The feature-based model FBM 1 = $\{s1, s2, s3, s4, s5, s6, h1, h41, h42, h51, h52, h61, h62\}$ for the part P1.

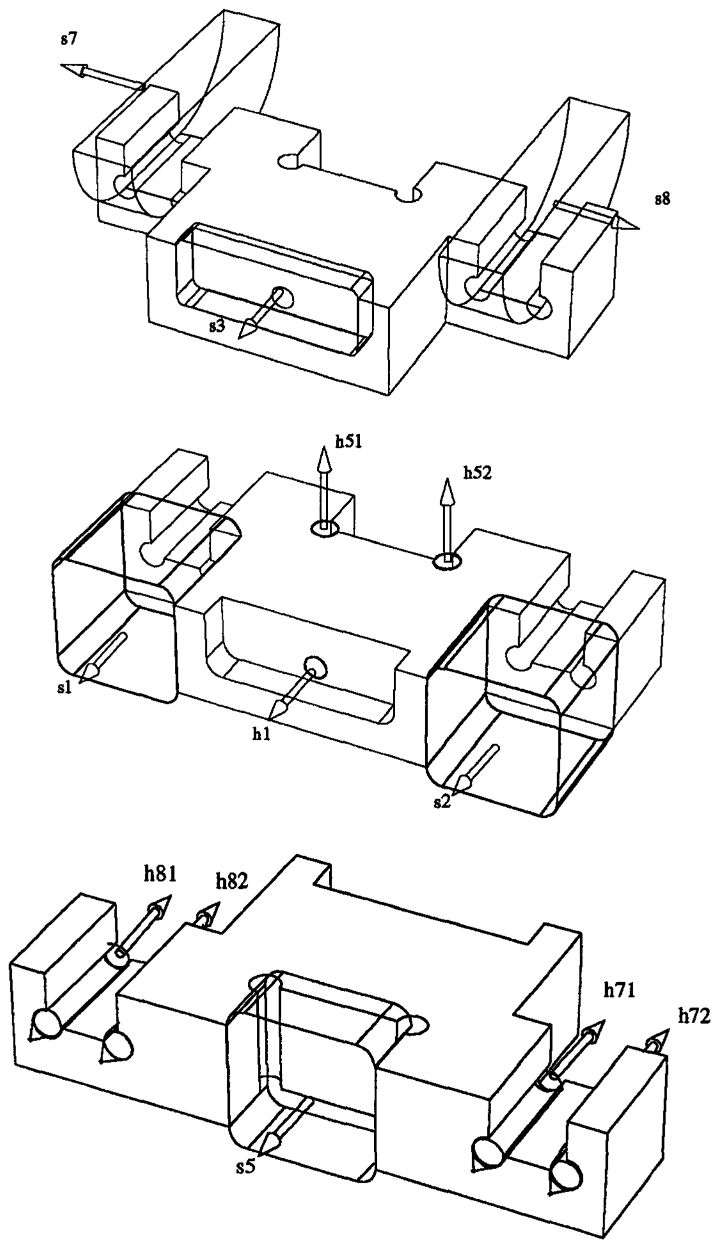


Figure 5: The feature-based model FBM 2 = $\{s1, s2, s3, s5, s7, s8, h1, h81, h82, h51, h52, h71, h72\}$ for the part P1.

$\mathcal{F}$ is the set of all machining features f that could potentially be used in generating P from S; for details see [6]. For the part P1, Figure 3 shows some of the features in $\mathcal{F}$. An FBM (feature-based model) $F \subseteq \mathcal{F}$ is a set of machining features such that subtracting F from the stock S produces the part P. For example, Figures 4 and 5 show two different FBM's for the part P1. In general, a single part may have several different FBM's, and thus there may be several different ways to machine the part.

3 Related Work

One of the first attempts at generating redesign suggestions was by Jakiela et al. [11, 10, 12]. Their work concentrated on automating the Boothroyd and Dewhurst [1] design for assembly methods. The redesign suggestions are made at the design stage as and when new features are added to the design.

Hsu, Lee and Su [8] reported redesigning of components for assembly from three major criteria: parallelism, assemblability and redundancy. They also defined some functions to modify the parts on the basis of these analysis. These functions work on the basis of splitting, combining or perturbing components.

Hayes, Desa and Wright [7] reported some advances in the direction of making redesign suggestions based on process planning knowledge. They did a protocol study to analyze the working methodology of experienced process planners, and used the results of this analysis to formulate modifications in mechanical designs.

For net shape manufacturing operations such as stamping, injection molding, and sheet metal working, several works on manufacturability evaluation and modification have been reported in the literature. For example, Lazaro *et al.* have developed a methodology for finds violations of design-for-manufacturing rules for sheet-metal parts [4]. From a library of suggestions, it displays hints for modifying the design. Similar methods are also used by others [13, 21, 19, 9]. Complete redesigned parts are not suggested in any of these cases, but suggestions are provided for avoiding manufacturability problems detected by the domain specific manufacturability evaluator.

Mantyla [15, 16] *et al.* proposed the novel concept of *feature relaxation.* This concept was developed to support the idea of "design by least commitment". The feature relaxation groups proposed by them are pairs of geometric features which are interchangeable. During process planning one of the members of the group can be chosen to get the overall approach direction for machining to be minimum. This allowed them to use different operations and build the final part geometry based on the operations which would be more convenient to use. Their feature relaxation groups are similar to our operators for generating alternate features (see Section 6) for local modification. However, the feature relaxation groups did not consider the restrictions on geometry which may be imposed by the function of the design. In certain cases, the feature relaxation might result in part geometry which is not compatible with the functional requirement of the part. Moreover, the objective of their work was only to minimize the the number of approach direction for machining the part, but as discussed in Section 5.2 in presence of precedence constraints the number of setups may exceed the number of approach directions to machine the part.

4 Representing Functional Requirements as Geometric Constraints

Designs are created by designers in order to satisfy various functional requirements. Thus, to be effective, a redesign scheme needs information about these functional requirements. However, these requirements can be both complex and disparate in nature. For example, a thermal scientist may be interested in the rate of heat dissipation from a surface, but a mechanist may be interested in the surface's load-bearing capacity. It is not clear how to represent these different kinds of requirements in a geometric CAD model in such a way that they could be used by a design analysis procedure.

To avoid this difficulty, we do not attempt to represent the functional requirements in a detailed manner. Instead, our approach is based on the observation that the part to be manufactured is typically a component of of a larger assembly, and most functional requirements will involve how the part interacts with other portions of the assembly [20, 17]. The faces of a part, and the volumes of space adjacent to the part, are where the part interacts with other portions of the assembly. Thus, we ask the designer to attach *functional requisite labels* to various surfaces and/or volumes in the CAD model. These labels can be used to specify geometric constraints based on the engineering purposes those surfaces or volumes are expected to serve. If the designer does not attach a label to a surface or volume in the design, then we assume that this surface or volume is not important to the function of the part.

Functional Requisite Labels on surfaces. The purpose of attaching a functional requisite label to a surface s is to state that some portion p of s is a *functional face*, i.e., p must remain as a face in the part design, even if the design is modified. To indicate this, the label allows the designer to specify the following information:

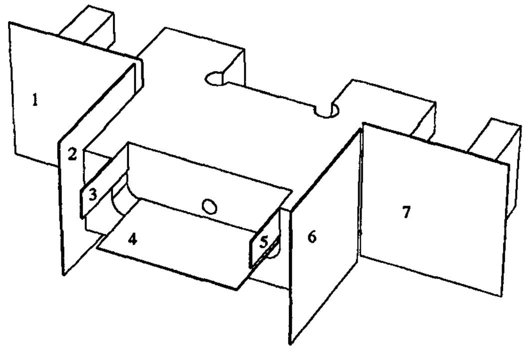
1. the minimum permissible surface area for p;
2. the maximum permissible surface area for p;
3. a region (specified using a point and a radius) within which p's centroid must be located;
4. a region (specified by giving the perimeter) of s within which p itself must be located;
5. as a special case, the designer can mark a face as unchangeable, to specify that the face should not be modified at all.

As an example, in Figure 6 we have specified functional requisite labels ff1 through ff16 on some of the surfaces associated with the part P1. Suppose that for both the part P1 and the stock from which it is made, the length (the X direction shown in Figure 1) is 160 mm, width (the Y direction) is 75 mm and the height (the Z direction) is 30 mm. In Table 1 we show the restrictions put by the designer on the faces ff1 through f16.

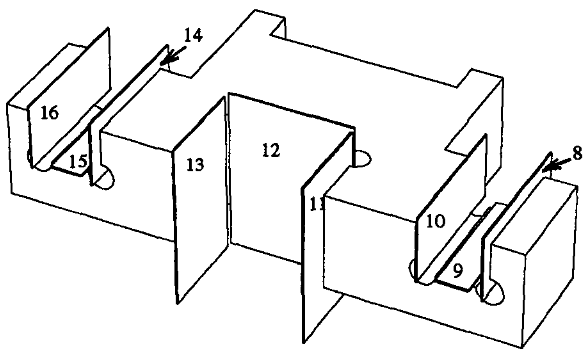
Functional Requisite Labels on volumes. The purpose of attaching a functional requisite label to a volume is to state that it is a *functional volume*, i.e., the entire volume must be left empty, either for mating or for clearance. Mating volumes act as enclosures or guides for other components. These volumes may include necessary allowances for thermal expansion or shrinkage, deformation under load, etc. Clearance volumes exist to allow proper fitting of mating components, as lubrication ducts or access areas, or to serve other functions. Even though v itself must be left empty, in some cases the designer may want to specify that it is permissible to enlarge v, producing a modified volume w. To indicate this, the functional requisite label allows the designer to specify the following information:

1. the maximum permissible volume for w;[1]

[1] We do not include a way to specify w's minimum volume, because this would be the same as the volume of v.



Front side functional surfaces ff1 through ff7



Back side functional surfaces ff8 through ff16

Figure 6: The functional faces for the part P1.

Table 1: Functional Requisite Labels on surfaces of the part P1.

Functional Requisite Label	Surface	Centroid Location	Minimum area	Maximum area
	Equation	Center and Radius (mm)	Sq mm	Sq mm
ff1	Y = 40	20,40,12, r=10	500	850
ff2	X = 40	40,20,15, r=7	850	1250
ff3	X = 50	50,5,20, r=2.5	100	200
ff4	Z = 10	80,5,10, r=1	350	400
ff5	X = 110	110,5,20, r=2.5	100	200
ff6	X = 120	120,20,15, r=7	850	1250
ff7	Y = 40	140,40,12, r=10	500	850
ff8	X = 10	10,57.5,22.5, r=3.5	300	800
ff9	Z = 15	20,57.5,15, r=1.5	300	700
ff10	X = 30	30,57.5,22.5, r=3.5	300	800
ff11	X = 60	60,65,15, r=3	450	800
ff12	Y = 55	80,55,15, r =0	700	1200
ff13	X = 100	100,65,15, r=3	450	800
ff14	X = 130	130,57.5,22.5, r=3.5	300	800
ff15	Z = 15	140,57.5,15, r=1.5	300	700
ff16	X = 150	150,57.5,22.5, r=3.5	300	800

2. the maximum permissible distance between the centroids of v and w;

3. surfaces that the volume w cannot violate.

For example, in figure 7 we have specified functional labels fv1 through fv10 on some of the volumes for the part P1. In this part, there are 26 functional requisite labels.

When a functional requisite label is attached to a surface or volume , that surface or volume becomes functionally requisite. Those surfaces and volumes along with the restrictions placed on those by the designer is called the functional requisites. A valid design needs to satisfy all these functional requisites. Once the CAD model and functional requisite labels have been obtained, this completes the preprocessing of the part and now the part is ready for analysis.

5 Analyzing the Design

As discussed in Section 1, our method for analyzing the design consists of three steps:

- find all possible machining features for the part;
- find precedence constraints among these features;
- find the lowest number of setups in which the part can be machined.

In this section we describe these steps in more detail, and show how they would be carried out on the part P1.

5.1 Extracting Machining Features

In machining the part, the faces of the part that are not faces of the stock will be created by machining operations that correspond to the machining features defined in Section 2. To determine how many setups will be required, we need to have some idea what these features are.

In [18, 5], we describe an algorithm for extracting from the solid model of the part all the machining features which can be used to create these faces. In some cases, faces can be created by more than one machining feature—and in such cases, the algorithm finds all the features which can create these faces.

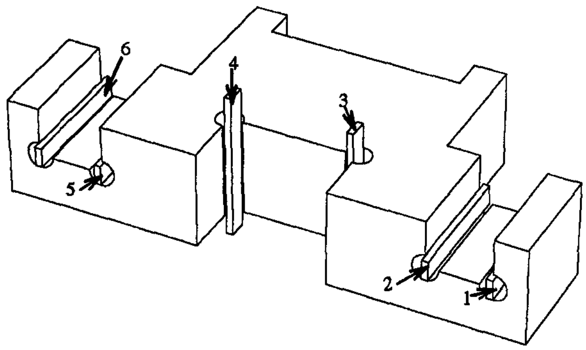
The feature recognition algorithm handles a large class of solids composed of features corresponding to drilling and milling operations, and its time complexity is quadratic in the number of solid modeling operations. Furthermore, the algorithm is provably complete over the set of all solids in our class, even if the features intersect with each other in arbitrarily complex ways.

As an example, Figure 3 shows some of the machining features extracted from the example part P1.

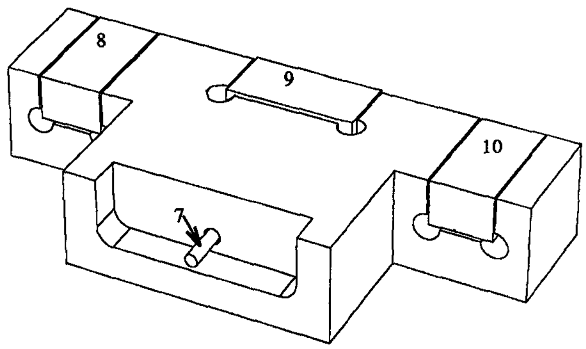
5.2 Finding Precedence Constraints

Due to various types of interactions (accessibility, setup, and so forth) among the features used to machine a part, the features cannot be machined in any arbitrary order. Instead, these interactions introduce precedence constraints requiring that some features be machined before or after other features. We are interested in finding these precedence constraints because, as we will see later, the number of setups required to machine the part will depend on them.

Here are two examples of precedence constraints:



Clearance volumes fv1 through fv6



Mating volumes fv7 through fv10

Figure 7: The functional volumes for the part P1.

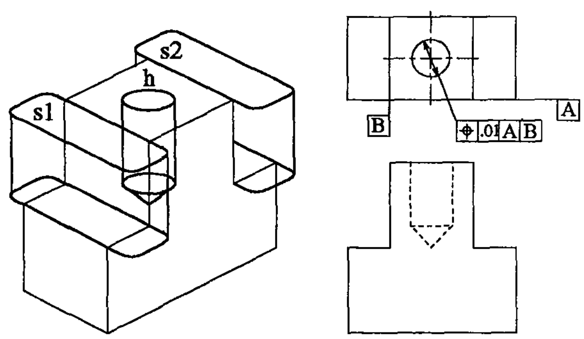


Figure 8: Part with precedence constraints.

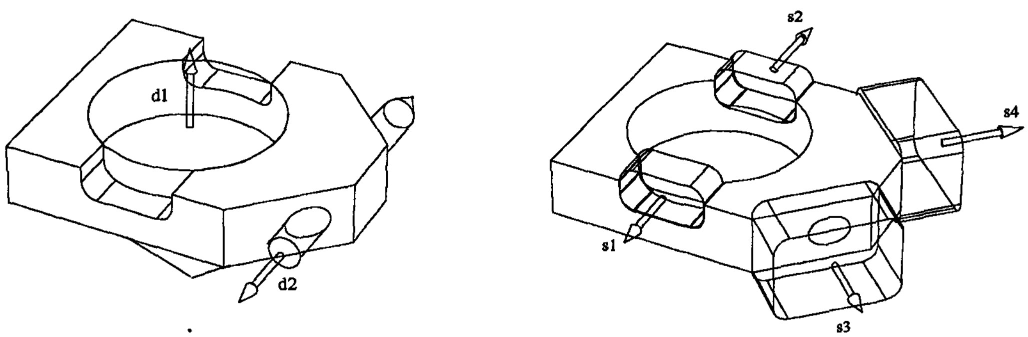


Figure 9: A part in which some of the features must be machined before others.

- In the part shown in Figure 8, the hole h is referenced to two surfaces A and B, which are contained respectively by the features s1 and s2. For this reason, there are precedence constraints s1→h and s2→h.
- Figure 9 shows a part in which the slot-hole interactions create strict precedence constraints for machining of that part. The large vertical hole h1 must precede the two end-mill features s1 and s2 on its side. Also, to get a flat entry face for drilling, the horizontal hole d2 must precede the end-mill features s3 and s4. The precedence constraints for this part are shown in Figure 10.

More generally, a pair of features f and f' will have precedence constraints under the following conditions:

1. If f contains a face that is datum for a face in f' then there will be a precedence constraint $f \to f'$s.
2. Features f, and f' will have precedence constraint $f \to f'$ if f' is not accessible until f is machined. Some examples are given below; more such cases are enumerated in [2, 3].
   (a) if f is an end-milling feature with at least one side open and f' a drilling feature then machine f before f';
   (b) if f is a blind pocket and f' is a drilling feature, then machine f' before f;
   (c) if f is a slot-milling feature and f' is a drilling feature, then machine f before f';
   (d) if two drilling features f and f' are collinear then drill the smaller of the two first.

The procedure to find the precedence constraints is straightforward. We simply check, for each pair of features, whether the above conditions hold.

5.3 Finding the Minimum Number of Setups for the Original Part

In the previous section, we found precedence constraints on all of the features in $\mathcal{F}$. However, to machine the part, one will not machine all of these features. Instead, one will machine some subset $F \subseteq \mathcal{F}$ that is sufficient to create P. The subset F is called a Feature-Based Model (FBM). The number of setups required to machine an FBM is determined by the precedence constraints among the features in the FBM, and the approach directions for the features in the FBM.

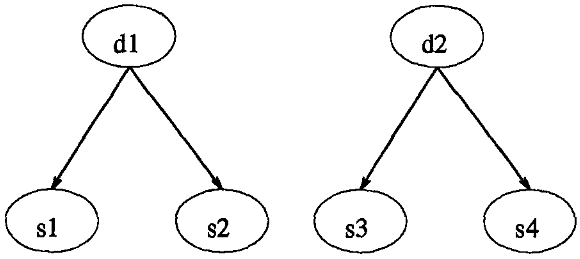


Figure 10: Precedence constraints among the features for the part shown in Figure 9.

Finding the minimum number of setups required to create P is basically a set-covering problem. The minimum number of setups needed to create P, is the minimum, over every FBM F that satisfied certain requirements, of the number of setups required for F. In particular, we will be interested in FBM's that satisfy the following requirements:

1. the features in the FBM satisfy all the functional requisites set by the designer;
2. none of the features in the FBM is redundant., i.e., every feature satisfies some functional requisite not covered by other features.

There may be a large number of FBM's that satisfy the above requirements, and the problem is how to compute the minimum number of setups without enumerating all of them. To do this, we we use the procedure FIND-FUNCTIONAL-COVER described below. FIND-FUNCTIONAL-COVER is a branch-and-bound procedure that finds FBM's, one by one, and computes the number of setups for the ones that appear promising.

FIND-FUNCTIONAL-COVER takes two arguments: a list FL of the functional requisites provided by the designer, and a partial FBM G (which is initially empty).

procedure FIND-FUNCTIONAL-COVER(FL, G)[2]

1. If the number of approach directions of the features in G is greater than or equal to the lowest number of setups computed so far, then return, because G will not result in a FBM which needs fewer setups.
2. Otherwise, if G covers all of the functional requisites in FL, then call FIND-BEST-SETUP to find the number of setups for G. FIND-BEST-SETUP is defined in Section 5.3.1.
3. Otherwise, do the following:
   (a) Choose a functional requisite fl in FL that is not already covered by G.
   (b) For each feature $g \in \mathcal{F} - G$ that satisfies fl, do the following:
   Let $G' :=$ CLEANUP($G \cup g$). (Thus, G' is $G \cup g$, with redundant features removed.)
   Call FIND-FUNCTIONAL-COVER(FL, G').

[2]In many practical machining parts including the example being presented here the features in $\mathcal{F}$ can be divided into unique equivalent sets where members of same groups completely satisfies a set of functional requisites. In those cases the problem of finding FBMs reduces to picking one feature from each group. In this case, the computation can be made much more efficient.

The procedure CLEANUP one arguments: a partial FBM G. If G' contains any feature g' such that both G' and $G' - \{g'\}$ satisfy the same set of functional requisites, then g' is redundant, so the procedure subtracts returns $G' - \{g'\}$. Otherwise, it returns G'.

At step two of FIND-FUNCTIONAL-COVER, if the FBM G covers all of the functional requisites in FL, then normally it will also cover the total removal volume of the part design (i.e., subtracting G's features from the stock S will create the part P). If this this condition ever is not satisfied, it could be for one of the following reasons:

1. The designer did not put all the functional requisite labels necessary for the performance of the part. If the designer does not correct this problem, then the modifications suggested by the system will not reflect the designer's intent.

2. There is some removal volume which does not serve any useful purpose. In this case, the part design should be modified to eliminate that removal volume.

In either case the designer would be notified of the discrepancy, and given the choice to either edit the functional requisite labels or to modify the design itself before proceeding further.

In step two, the procedure FIND-BEST-SETUP finds the lowest number of setups required to machine the FBM G. This procedure is described in Section 5.3.1.

5.3.1 Finding the Number of Setups Needed to Machine an FBM

Once we have an FBM, it is a specific set of machining features all of which needs to be machined to get the final part. Each feature f has a specific approach direction $\vec{v}(f)$, and some of these features may have some precedence constraints among them. Each operation sequence for machining the part corresponds to a sequential ordering $\{f_1, f_2, \ldots, f_m\}$ of the features that is consistent with the precedence constraints. In a vertical machining center, the number of setups required by this operation sequence is one more than the number of times the approach direction changes when we scan the sequence from start to finish.

The number of setups needed to machine the FBM will be the minimum, over all operation sequences satisfying the precedence constraints, of the number of setups required by the operation sequence. As described below, this can be computed by calling FIND-BEST-SETUP(F, C), where F is the FBM, and C is the set of precedence constraints. This depth-first branch-and-bound algorithm will return the minimum number of setups needed to machine F.

procedure FIND-BEST-SETUP(F, C)

1. Initially, $n := \infty$
   n is a global variable which contains the size of the best solution found so far.

2. call EXTRACT-SETUP$(F, C, 0)$
   This is a recursive algorithm which searches the features to find the operation sequence that requires the fewest setups.

3. return n
   n is the minimum number of setups required to machine that FBM, this value is returned by the procedure EXTRACT-SETUP

end procedure

procedure EXTRACT-SETUP(F, C, i)

1. If $i \geq n$, then return, because the number of setups exceeds the best solution so far.
2. Otherwise, if $F = \emptyset$, then we have found a better solution, so set $n := i$ and return.
3. Otherwise,
   (a) Let **READY** be the set of all features in F that have no predecessors.
   (b) Let V be the set of all approach directions of features in **READY** (i.e., $V = \{\vec{v}(f) : f \in \textbf{READY}\}$).
   (c) For every $\vec{v} \in V$, let $\textbf{SETUP}(\vec{v})$ be the set of all features f in F such that
      i. f has $\vec{v}$ as its approach direction;
      ii. either f has no predecessor in F, or all predecessors in F have $\vec{v}$ as their approach direction.

   Note that all of these features can be machined in the same setup.
   (d) Let $C_s(\vec{v})$ be the set of precedence constraints that are associated with the features in $\textbf{SETUP}(\vec{v})$.
   (e) If there exists a $\vec{v} \in V$ such that there is no feature in the set $F - \textbf{SETUP}(\vec{v})$ that has $\vec{v}$ as its approach direction, then call EXTRACT-SETUP($F - \textbf{SETUP}(\vec{v}), C - C_s(\vec{v}), i + 1$).
   (f) Otherwise, for every approach direction[3] $\vec{v} \in V$, call EXTRACT-SETUP($F - \textbf{SETUP}(\vec{v}), C - C_s(\vec{v}), i + 1$).

5.3.2 Result of the Algorithm on an Example Part

For the part P1, there are several FBMs that cover all of the functional requisites. Two such FBM's are FBM's 1 and 2, which are shown in Figures 4 and 5.

Among all FBMs created by the algorithm, FBM 1 can be machined in the lowest number of setups. FBM 1 has only three approach directions—but due to the precedence constraints among these features (see Figure 11), the minimum number of setups required to machine FBM 1 is four.

Which FBMs are found by the procedure FIND-FUNCTIONAL-COVER will depend on the order in which the functional requisites are picked for coverage, and the order in which features are placed in the FBM being generated. Any time a partial FBM G is generated whose number of approach directions exceeds the number of setups found by the FBM found so far, the algorithm will discard G. Since FBM 1 has the lowest number of setups of any possible FBM for the part, it will always be generated—but since FBM 2 has a larger number of setups, it may or may not be fully generated, depending on whether algorithm starts generating it before or after

6 Creating local modifications

In Section 5 we presented a method for finding the minimum number of setups required to machine a part. We are interested in improving on this setup cost, by considering modifications to the geometry of the existing design. To avoid having to go back to the conceptual design stage, we will only consider local modifications on the machining features already found in the original design.

Since the number of setups will depend on the approach directions for the features and the precedence constraints among them, our objective is to modify some of the features in such a way

[3]The efficiency of the algorithm depends on the order in which it examines the approach directions in V. Our heuristic is to iterate over the approach directions of V in order of decreasing cardinality of $\textbf{SETUP}(\vec{v})$.

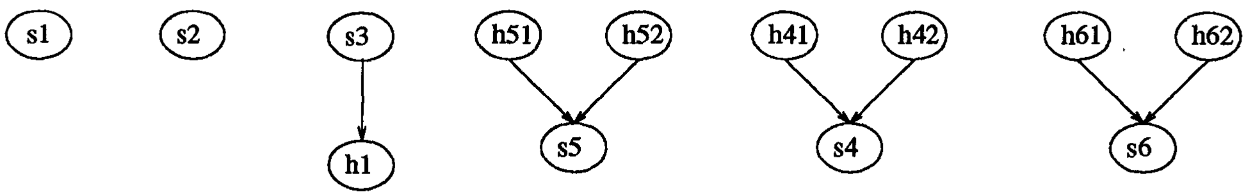


Figure 11: Precedence constraints among the features in an FBM of the example part of figure 1.

as to allow them to be machined from different approach directions. Each feature in the original design satisfies some of the design's functional requisites—so if we modify a feature, we want the new feature to satisfy the functional requisites satisfied by the old feature.

For doing this, we classify each feature according to the machining operation M that is used to create it, and define a set of redesign operators $O(M)$ for each machining operation M. We consider two different kinds of redesign operations, both of which will create features closely resembling the features that they are replacing:

1. convert one machining operation to another operation;
2. perform the same operation from a different direction.

The basic way these operators work is described below:

1. Pick a feature f from $\mathcal{F}$.
2. Find the class of the feature f, i.e., the machining operation which creates f.
3. Get the set of operators for that feature class.
4. Apply these operators, to get new features (there can be more than one feature created by one operator).
5. For each of the features created, find all the faces and volumes which would be altered or removed if the new feature replaces the old one.
6. If replacing the old feature with the new one would cause any of the functional requisites to be violated, then that modification is not valid.
7. Otherwise, if the new feature is not already in the set $\mathcal{F}'$ (which would happen if the new feature were previously generated by modifying some other feature), then display that feature to the designer as an alternative of the original feature.
8. If the designer accepts that feature as a possible alternative, then add it to the set $\mathcal{F}'$.

If the designer does not find the new feature to be acceptable in Step 8, this means that in some way or another, the new feature violates the designer's intent. One way that this can happen is if the functional requisite labels specified by the designer were not sufficient to represent all of the functional requirements that the designer had in mind. In that case, the designer can go back and modify the functional requisite labels, and restart the analysis procedure.[4]

[4]Ultimately, we would like to provide ways whereby the analysis procedure can take up where it left off, incorporating the modified functional requisite labels into its analysis. This is a topic for future work.

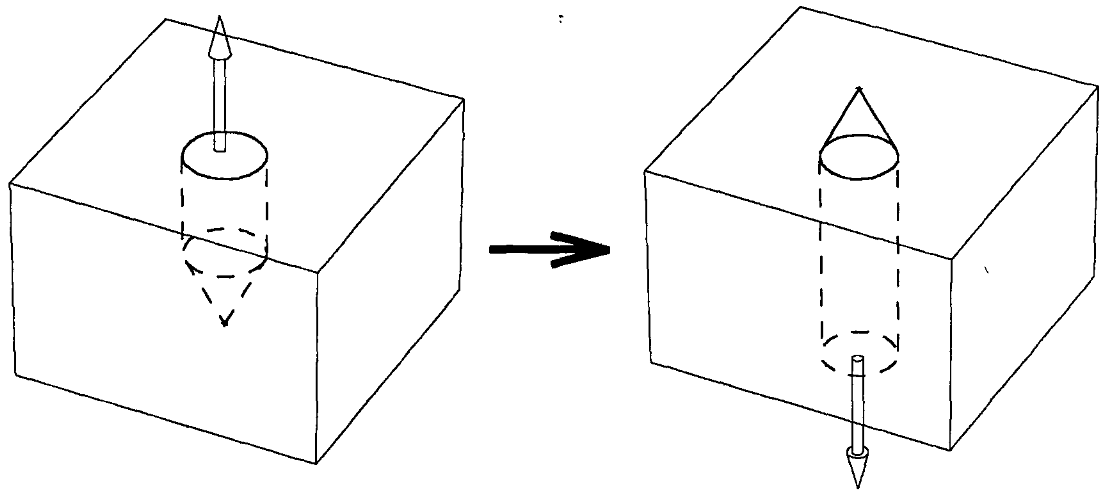


Figure 12: Converting drilling features.

These steps extend the feature set $\mathcal{F}$, to create a new set $\mathcal{F}'$. In addition to the features of the original part, $\mathcal{F}'$ contains those created by the local modification operators that do not violate the functional requisite labels and are acceptable to the designer as possible local modifications. As examples, Figure 12 shows one way of modifying a drilling feature, and Figure 13 shows a modification that changes an end-milling feature to a slot-milling feature.

7 Generating Design Alternatives

A *reconstructed* FBM$'$ is a subset of $\mathcal{F}'$ where all the features g in an FBM$'$ together cover the functional requisites for the original part and requires less number of setups than the least number of setups required to machine the original part. When all the features $g \in$ FBM$'$ is subtracted from stock S it creates a new part P' which is a valid redesign suggestion for the part P.

After the precedence constraints are set (see Section 5.2) among the features in the extended feature set $\mathcal{F}'$ we are in a position to attempt to extract possible alternative parts. As shown below, the procedure is similar (but not identical) the procedure FIND-FUNCTIONAL-COVER of Section 5.3.

procedure FIND-NEW-FUNCTIONAL-COVER(FL, G)

1. If the number of approach directions of the features in G is greater than or equal to the number of setups computed by FIND-FUNCTIONAL-COVER earlier, then return, because G will not result in a FBM which needs fewer setups.

2. Otherwise, if G covers all of the functional requisites in FL, then call FIND-BEST-SETUP to find the number of setups for G. If this number is less than the number of setups computed by FIND-FUNCTIONAL-COVER earlier, then store G so that it can be displayed to the user as described in Section 7.1. Otherwise, discard G.

3. Otherwise, do the following:

   (a) Choose a functional requisite fl in FL that is not already covered by G.

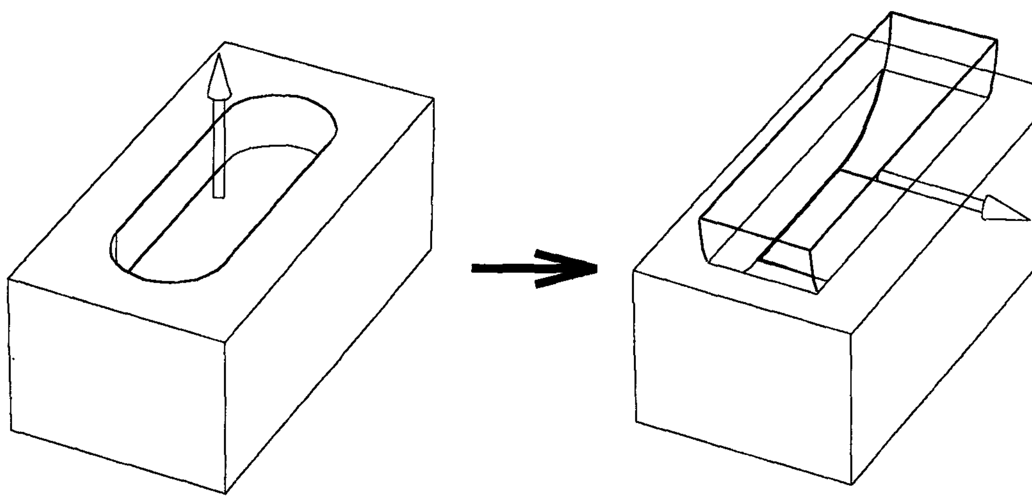


Figure 13: Converting an end milling feature to a slot milling feature

(b) For each feature $g \in \mathcal{F} - G$ that satisfies fl, do the following:
Let $G' :=$ CLEANUP$(G \cup g)$, where CLEANUP is as described in Section 5.3. (Thus, G' is $G \cup g$, with redundant features removed.)
Call FIND-FUNCTIONAL-COVER(FL, G').

As an example, suppose we apply this algorithm to the features in the set $\mathcal{F}'$ computed in Section 6. Then we get several reconstructed FBM's that satisfy all of the functional requisites, and also reduce the number of setups. For example, Figures 14 and 15 show the parts P2 and P3. These parts, which are modified versions of P1, each can be machined in two setups. Figure 16 shows which features of P2 are different from those of P1. Table 2 gives the properties of the functional faces; by comparing this information to the information in Table 1 it can be seen that these faces are in the range specified by the designer.

7.1 Presenting alternate designs

It becomes obvious from the analysis of this example (see figure 14 and figure 15) that there is a possibility of generating a very big set of new design suggestions. All these suggestions will fulfill the basic requirements set by the problem. Theoretically any one of these can be picked by the designer as an alternative. In practice however, designer will find many of these alternatives to be unpromising. We need to order these choices based on some other factors other than just the number of setups for presentation to the designer. Some of the factors possible to use as this secondary guideline are:

1. Machining time : this estimation can be done directly on the FBM as all the features are defined in terms of manufacturing processes.
2. Number of tools required : This also depends on the numbers and types of manufacturing features in the FBM.

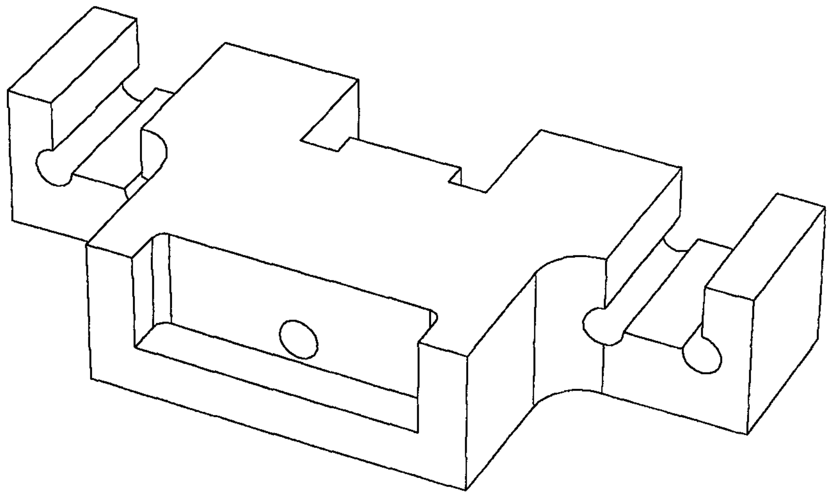


Figure 14: P2, a modified version of the part P1.

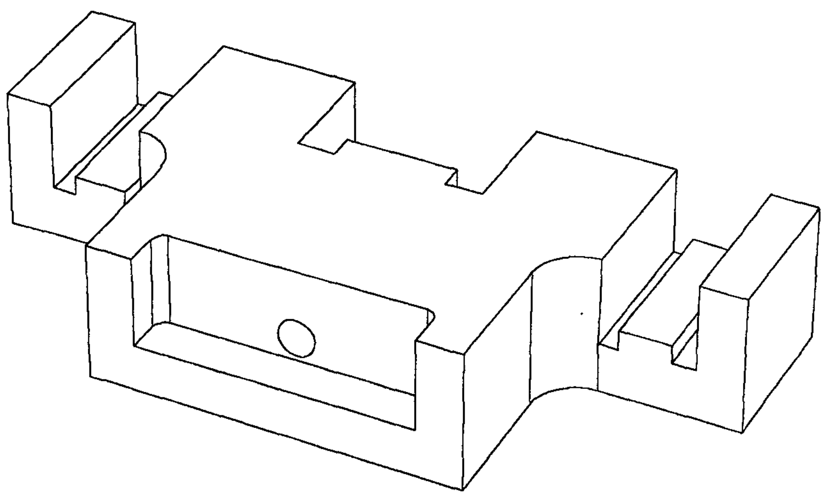


Figure 15: P3, another modified version of the part P1.

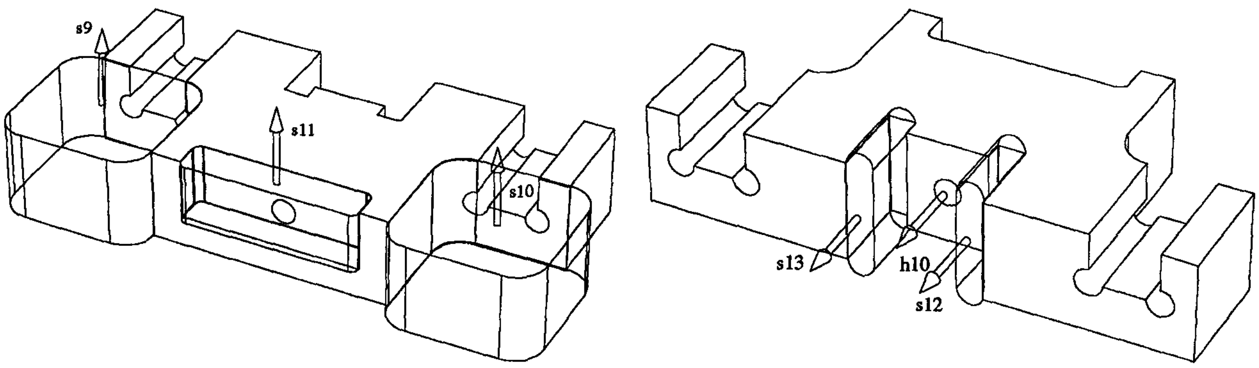


Figure 16: Features of P2 that are different from the features of P1.

Table 2: Status of functional faces in the part shown in Figure 14.

Functional label	Centroid	Area
	Center location	Sq mm
ff1	12.25,40,12.15	550
ff2	40,15,15	900
ff3	50,3.75,20	150
ff4	80,4.9,10	390
ff5	110,3.75,20	150
ff6	120,15,15	900
ff7	147.75,40,12.15	550
ff8	10,57.5,24.5	385
ff9	20,57.5,15	420
ff10	30,57.5,24.5	385
ff11	60,63,15	660
ff12	80,55,15	720
ff13	100,63,15	660
ff14	130,57.5,24.5	385
ff15	140,57.5,15	420
ff16	150,57.5,24.5	385

/

8 Discussion and Conclusion

Our ultimate goal is to develop a methodology for generating redesign suggestions which will help reduce the overall production cost and the lead time between product conceptualization and launching. As a step toward this goal, in this paper we have proposed a methodology for automatically generating alternative designs for a mechanical part.

Our proposed approach is to use information resulting from analysis of the original part. The criterion used here for improving the design is the number of setups required to machine the part. Some of the salient features of our approach are described below:

1. We use a functional requisite labeling scheme for marking relevant faces and volumes in the original design. This scheme will help to reduce the number of alternatives generated.

2. We generate additional alternative features, and then select features that cover the functional requisite labels. In contrast to approaches (e.g., [7]) that analyze a single operation plan setup by setup to decide on which setups can be removed or combined, our approach works on a more global level, considering alternative operation plans and alternative setups.

3. We consider precedence constraints among features, to make sure we actually reduce the number of setups. Reducing only the number of approach directions would not necessarily reduce the number of setups.

4. When possible, we generate multiple alternatives for the same part. This allows the designer to run other types of analysis (such as structural or thermal analysis) before deciding which alternative design to use.

Currently, some of the elements of this approach have been implemented, and others are underway. For future work, we intend to complete our implementation, and in addition we plan to incorporate certain improvements and extensions, as described below.

We are interested in considering geometric and dimensional tolerances of the part while creating local modifications and while generating redesign suggestions. The functional requisite labeling scheme will be broadened using this information. We are looking into methods of ordering the various design suggestions generated by our method for presenting to the designer. At a later stage we want to incorporate other manufacturing cost factors as criterion for generating redesign suggestions. Along with these improvements we will do implementation and testing of the methodology towards building a prototype design advisory system.

References

[1] G. Boothroyd and P. Dewhurst. *Design for Assembly -A Designer's Handbook.* Department of Mechanical Engineering, University of Massachusetts at Amherst, 1983.

[2] James G. Bralla, editor. *Handbook of Product Design for Manufacturing.* McGraw Hill Book Company, 1986.

[3] Tien-Chien Chang. *Expert Process Planning for Manufacturing.* Addison-Wesley Publishing Co., 1990.

[4] Anthony de Sam Lazaro, David T. Engquist, and Dean B. Edwards. An intelligent design for manufacturability system for sheet-metal parts. *Concurrent Engineering: Research and Applications*, 1(2):117–123, 1993.

[5] S.K. Gupta, T.R. Kramer, D.S. Nau, W.C. Regli, and G. Zhang. Building MRSEV models for CAM applications. *Advances in Engineering Software,* special issue *on Feature-Based Design and Manufacturing*, 1994. To appear. Also available as Institute for Systems Research Technical Report TR 93-84.

[6] S.K. Gupta and D.S. Nau. A systematic approach for analyzing the manufacturability of machined parts. *Computer Aided Design*, 1994. To appear. Also available as Institute for Systems Research Technical Report TR 93-76.

[7] C. C. Hayes, S. Desa, and P. K. Wright. Using process planning knowledge to make design suggestions concurrently. In N. H. Chao and S. C. Y. Lu, editors, *Concurrent Product and Process Design*, pages 87–92. ASME, 1989.

[8] Wynne Hsu, C. S. George Lee, and S. F. Su. Feedback approach to design for assembly by evaluation of assembly plan. *Computer Aided Design*, 25(7):395–410, July 1993.

[9] K. Ishii, L. Hornberger, and M. Liou. Compatibilty-based design for injection molding. In N. H. Chao and S. C. Y. Lu, editors, *Concurrent Product and Process Design*, pages 153–160. ASME, 1989.

[10] M. Jakiela and P. Papalambros. Concurrent engineering with suggestion-making cad systems: Results of initial user tests. In B. Ravani, editor, *Advances in Design Automation.* ASME, 1989.

[11] M. Jakiela, P. Papalambros, and A. G. Ulsoy. Programming optimal suggestions in the design concept phase: Application to the boothroyd assembly charts. *ASME Journal of Mechanisms, Transmission, and Automation in Design*, 107:285–289, June 1985.

[12] Mark J. Jakiela. Intelligent suggestive cad systems : Research overview. Technical report, Massachusetts Institute of Technology, 1989.

[13] Prashant V. Mahajan, Corrado Poli, David W. Rosen, and Michael J. Wozny. Design for stamping: A feature-based approach. In P. J. Guichelaar, editor, *Design for Manufacturability*, pages 29–49. ASME, 1993.

[14] M. Mantyla. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, MD, 1988.

[15] M. Mantyla, J. Opas, and J. Puhakka. Generative process planning of prismatic parts by feature relaxation. In B. Ravani, editor, *Advances in Design Automation*, pages 49–60. ASME, 1989.

[16] M. Mantyla, J. Opas, and J. Puhakka. Generative process planning of prismatic parts by feature relaxation. Technical report, Helsinki Institute of Technology, Laboratory of Information Processing Science, Finland, Feb 1989.

[17] G. Pahl and W. Beitz. *Engineering Design.* Design Council, London, 1984.

[18] William C. Regli and Dana S. Nau. Building a general approach to feature recognition of material removal shape element volumes (mrsevs). In Jaroslaw Rossignac and Joshua Turner, editors, *Second Symposium on Solid Modeling Foundations and CAD/CAM Applications*, New York, NY 10036, USA, May 19-21, Montreal, Canada 1993. ACM SIGGRAPH, ACM Press.

[19] J. M. Schmitz and S. Desa. The application of a design for producibility methodology to complex stamped products. In N. H. Chao and S. C. Y. Lu, editors, *Concurrent Product and Process Design*, pages 169–174. ASME, 1989.

[20] David G. Ullman. *The Mechanical Design Process.* McGraw-Hill, Inc., 1992.

[21] R. V. Welch and J. R. Dixon. Extending the iterative redesign model to configuration design: Sheet metal brackets as an example. In W. H. Elmaraghy, W. P. Seering, and D. G. Ullman, editors, *Design Theory and Methodology*, pages 81–88. ASME, 1989.

Printed	Thu Feb 10 11:00:04 1994
Started	Thu Feb 10 10:59:58 1994
Submitted	Thu Feb 10 10:59:58 1994
Name	/tmp/pstex1988
Owner	diganta at tulgey

Jamresistance
Jobheader on
Pagereversal
language ultrascript

System Version TURBO UltraScript 6.0T Rev. A IP/II, Serial #91:7:11
Page images processed: 23
Pages printed: 23

Paper size (width, height):
2560, 3328
Document length:
457554 bytes
%%[Starting Server]%%