From: AAAI-88 Proceedings. Copyright ©1988, AAAI (www.aaai.org). All rights reserved.

Geometric Reasoning and Organized Optimization for Automated Process Planning

Yasuyuki Maeda and **Katsuya Shinohara**
C&C Systems Research Laboratories
NEC Corporation
1-1, Miyazaki 4-chome, Miyamae-ku,
Kawasaki, Kanagawa 213, JAPAN

Abstract

In order to contribute toward the realization of a practical computer-automated process planning system, this paper discusses two essential subjects: geometric reasoning and optimization capabilities. ESPER provides a solution for these two problems. Geometric reasoning mediates between symbolic reasoning and geometric modeling. It implies recognizing various features in geometric data as well as manipulation of these geometries. Optimization requires effective problem solving strategy and, in addition, cooperation between system inference and users' interaction. For problem solving, knowledge is defined so as to avoid divergent search by pruning. Furthermore, users' interactions can be incorporated into the optimization process to obtain a better solution. Through developing the system, it is shown that the methodology proposed here is effective for realizing practical process planning systems.

1 Introduction

As computer use in design and manufacturing increases, process planning, which connects design and manufacturing activities, becomes more important. It seems promising to apply knowledge-based techniques to process planning. Therefore, there has been much research carried out in this application field (e.g. [Descotte and Latombe, 1985; Eliyahu *et al.*, 1987]). From the practical use viewpoint, however, coverage and flexibility available in existing systems are insufficient to deal with more complex objects and more varied operations. In order to bridge this gap between AI techniques and their practical use, it is necessary to find a solution more closely adapted to realities.

This paper describes a new practical knowledge utilization method in the field of process planning. In process planning, planners transform geometric information to manufacturing process information. In order to carry out this transformation activity automatically, the authors have developed a new method which enables automatic transformation from the geometry to its manufacturing process, and it was applied to build an expert system called ESPER (*E*xpert *S*ystem for *P*roduction *E*nginee*R*s). The method is characterized by two features: geometric reasoning and optimization strategy.

The geometric reasoning technique offers reasoning power about geometries, which is the crucial issue in applying knowledge processing to such planning fields. Though the necessity of geometric reasoning facilities has been recognized recently [Hirschtick and Gossard, 1986], knowledge processing and geometric reasoning are still separated in existing systems. The geometric reasoning facilities in ESPER can be used not only to recognize significant features in geometry, but to manipulate the geometry directly.

Since process planning is essentially an optimizing problem with complex factors, an efficient search strategy is required, especially when the problem size becomes large. The factors are so complicated that backtracking mechanisms, such as those used in TMS [Doyle, 1978; de Kleer, 1986], seem insufficient. ESPER enables accomplishing efficient planning by the use of task modules and, additionally, makes possible solution improving by incorporating user actions.

In the next section, problems in existing knowledge processing applications to the process planning field are discussed. To solve these problems, the authors propose two techniques: one is about geometric reasoning and one is about optimization. In Section 3, the geometric reasoning technique is described in detail, while the optimization technique is discussed in Section 4. Application implementation, using these two techniques, is described in Section 5. Section 6 concludes with a short summary.

2 Process Planning

2.1 Problems in Process Planning

In process planning, manufacturing process information is generated from the geometrical information extracted from design information regarding mechanical components. Manufacturing information consists of the order in which manufacturing machines and tools are to be used, fixture information, tool path, and cutting conditions.

It is well-known that a machining plan is not uniquely determined from geometric information. Therefore, the solution must be optimized according to several pertinent criteria (e.g. total machining cost or machining time) [Davies and Darbyshire, 1984].

Searching for every possible solution exhaustively and evaluating the results to obtain the best solution would not be realistic, since the search field is very large. Therefore, it is natural that planning should operate as a breadth-first search, along with pruning unpromising alternatives. Usually, the planning proceeds by steps, according to the search space hierarchy including machines, fixtures, tools, cuttings and cutting conditions.

Optimization problem is common in all planning subjects.

Problem solving techniques, developed in AI research (e.g. [Stefik, 1980; Sacerdoti, 1977]), are also effective in process planning to a certain extent. However, some customizations are necessary, when these techniques are applied to process planning. GARI [Descotte and Latombe, 1985] incorporated selective backtracking mechanisms geared to weighted knowledge in order to make it possible to compromise among preferential rules. [Smith and Ow, 1985] realized a constraint-directed reasoning mechanism in job-shop scheduling. [Murphy *et al.*, 1987] adopted the blackboard approach to access the process planning problems.

However, these techniques are insufficient for application to practical problems for the following reasons:

1. Since knowledge representation does not fully reflect the nature of geometry, planning is executed in a symbolized world, which differs to some degree from the physical world.
2. These approaches aim at establishing a single inference mechanism applicable to the overall process planning activity. As operations coverage widens and empirical knowledge increases, the system performance efficiency is reduced. This is especially true when the problem size becomes large.

2.2 Solutions

In order to compensate for the former issue about the physical world, geometric reasoning capability has been introduced. For the latter issue about efficiency, a new optimization strategy and an interactive optimization method have been introduced.

The geometric reasoning capability enables feature extraction from geometric data in the process planning viewpoint. Additionally, it allows direct manipulation of pertinent geometry while reasoning is going on.

On the basis of the optimizing strategy, process planning is broken down into several tasks. Individual tasks are realized as a knowledge module, which consists of a candidate generation part and an optimization part. Cooperation with system inference and user interaction is also achieved to obtain a better solution.

Details of the two approaches are described in the following Sections 3 and 4.

3 Geometric Reasoning

One of the most remarkable characteristics of process planning, which differs from other applications for knowledge-based techniques, is its treatment of three dimensional geometry. Many solid modeling systems have been developed for dealing with three dimensional objects in a computer. However, an attempt to use solid modeling techniques, with the intention of completely re-creating the world of three-dimensional geometry, encounters many difficulties. A more suitable method for knowledge-based reasoning is needed.

3.1 Feature Recognition

Most of the existing process planning expert systems require that designers should describe shapes and cutting areas in terms of entities [Descotte and Latombe, 1985; Eliyahu *et al.*, 1987]. That is, interpretation and judgment about pertinent geometries are left to the designers. On the other hand, process planning systems themselves handle geometric information in the same way as other symbolic data.

There have been some attempts to extract feature information from geometric data, using pattern matching techniques. In the STOPP system [Choi and Barash, 1985], users have only to define the cross-sectional contours of holes to be machined. A complex hole is resolved into primitive feature patterns that one drill can generate. However, this feature extraction is limited only to hole machining, because a hole contour can be easily described as a sequence of vectors. With regard to slot machining, only simple patterns, such as rectangle slots or pockets, are allowed, and no complex shapes can be handled [Henderson and Anderson, 1984; Jared, 1984].

From the practical use viewpoint, handling complex shapes is indispensable. Figure 1(a) shows an example of a practical pocket's contour, defined in a CAD system. Representing these shapes as they are, namely, representing them in terms of lines and arcs, requires a large amount of data. Moreover, such data alone cannot be used to reason from geometric information (e.g. to judge what tool can generate this shape and what tool is most suitable for the shape). Consequently, an interpreting process is required for connecting knowledge processing with a geometric model.

The recognition process in ESPER is composed of two phases. In the first stage, rough features are extracted from the component shape, without regard to details. Next, each extracted feature is divided into primitive elements. In regard to slots and pockets, contours are divided into rectangles. Mating information about details (technological information, such as a radius of a corner and corner protrusion distance) is extracted and described by attributes for each element.

Assume, for example, that the pocket contour, as shown in Figure 1(a), is given to the system. At first, the contour is approximated with an XY-polygon (Figure 1(b)). Detailed information about corners is registered to each vertex at this moment. Next, the polygon is divided into rectangles so as to minimize the total length of dividing lines (Figure 1(c)). Since this division problem is NP-complete, the system employs a greedy algorithm [Rivest, 1982; Imai and Asano, 1984]. After this division, primitives (rectangles

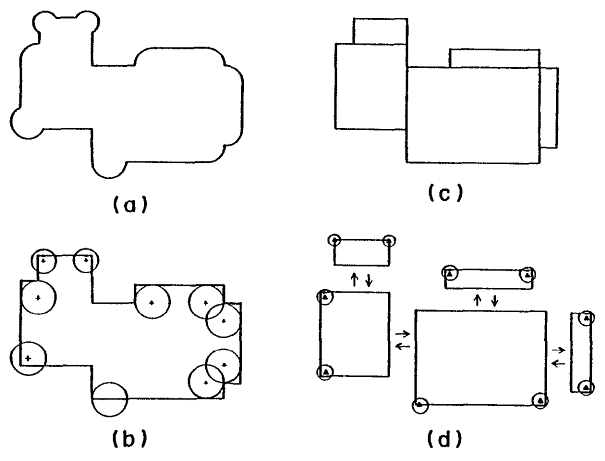

Figure 1 : Feature recognition of a pocket's contour

in this case) are generated and relations between primitives are examined, as shown in Figure 1(d). Detailed characteristics, such as the radius of each corner, are also examined at the same time and connected with pertinent primitives.

In this way, the geometry is reproduced in terms of a network representing its features. An individual node corresponds to an individual primitive, and relations between nodes represent geometrical relations between primitives (e.g. adjacency or inclusion). By the above transformation, it becomes possible to recognize complex features, using empirical knowledge described in the form of heuristic rules.

3.2 Geometry Manipulation

In process planning, manufacturing process information is generated from the geometrical information extracted from design information regarding mechanical components. Planning usually begins with the geometry of a finished component and progresses in the inverse direction of manufacturing. That is, it works backward from the finished component until the blank material is reached. Therefore, in order to treat a component with a complex shape, geometry manipulation capability is indispensable.

Examples of geometry manipulations are breakdown and reorganization of volumetric elements. Since complex shapes can be machined in many ways, in which volume breakdown may also vary, conversion of one volumetric representation to another is important. This kind of conversion cannot be carried out without extensive knowledge about geometry.

- *Division and Merging*
  In determining shapes for each cutting area, extracted features are divided or merged, if necessary. For example, the feature is divided when there are two kinds of corner radii (e.g. R6 and R10) in a feature and it is possible to divide the feature into two areas, in which corner radii are common.
- *Subtraction*
  When determining shapes of finishing areas, shapes of the corresponding rough cutting areas should also be considered at the same time. Finishing shapes are determined so as to avoid redundant cutting. That is, the area which has been already removed by rough cutting is excluded from the finishing operation.
- *Reduction*
  For cutting efficiency, a spiral tool path is preferable to a simple two-way tool path. To generate a spiral tool path from the cutting area shape, reduction is applied to the shape repeatedly and circular paths are separated from the original shape, until the entire shape is removed. Next, these circular paths are connected, so as to form a tool path. Figure 2 shows this process of iterative reductions and circular path connection to generate a cutting tool path.

The above manipulations are applied to the network representing the geometry. Each manipulation on a shape is resolved into operations on its primitives (such as rectangles). Through the use of these manipulations, heuristic rules concerning geometrical features can handle the geometry immediately.

Figure 3(a) shows a cross section of two adjoining pockets A and B, differing in depth. If some rules suggest that these pockets should be finished at the same time, the adjacency relation and their individual depths are examined, and pertinent merging and division occurs in order to determine cutting areas P and Q, as shown in Figure 3(b). The underlying knowledge about these manipulations is as follows.

(1) If A adjoins B and B is deeper than A, finishing A can be merged with finishing B.
(2) If B is deeper than A, and finishing A is merged with finishing B, finishing B should be divided according to the depth for A.

In the case of Figure 3(c), similarly, three finishing areas are generated (Figure 3(d)). In Figure 3(e), on the other hand, two finishing areas are generated (Figure 3(f)).

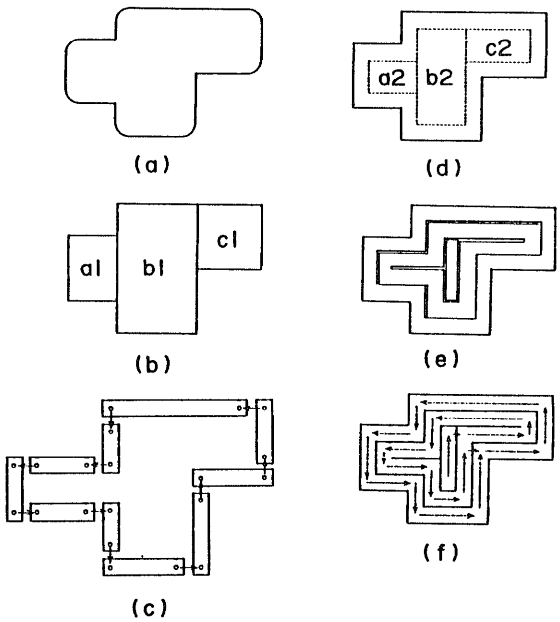

Figure 2 : Reduction to cutting primitives

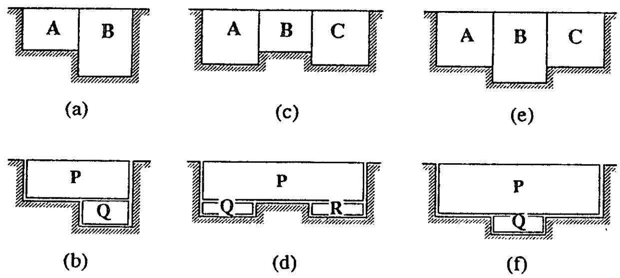

Figure 3: Cutting area generation

4 Optimization

4.1 Problem Solving

Process planning can be considered to be an optimization problem, which minimizes manufacturing cost or time, while satisfying given technical conditions, such as tolerances and allowable surface roughness.

4.1.1 Task Division

Process planning consists of such operations as feature extraction from the defined shape, tool selection, fixture design, ordering machining sequences, tool path generation, and determining cutting conditions. Because of the size and complexity of the problem, it is reasonable to divide it into several tasks.

Each task can be considered as an optimization problem, which determines one category of items, in order that it optimizes certain criteria (e.g. cutting time) under some constraints. For example, tool selection requires minimizing cutting time under limitations regarding the number of attachable tools.

In the ESPER system, these tasks are realized as "task modules". Each task module is activated when its starting conditions are satisfied and determines a particular category of items for each feature.

It is true that there are some sub-problems among these tasks that can be solved mathematically. Practically, however, a number of factors, some of which influence one another, must be taken into consideration, before making the appropriate decision. Figure 4 shows the relationship between factors in determining cutting areas and selecting tools. In this case, for instance, tool selection depends on cutting depth, cutting depths are determined by the cutting order, and the cutting order is affected by tool selection.

In order to control the planning process efficiently, standard optimization techniques are insufficient. Some temporary concepts should be introduced as intermediates. That is, before the optimization begins, candidates for items are selected and partial constraints on the values are generated. These intermediates narrow the search space and make the planning process efficient. As shown in Figure 5, introducing intermediate concepts removes loops in dependence among items.

4.1.2 Task Realization

In ESPER, the above-mentioned task module is composed of two parts: candidate generation stage and optimization stage.

(A) *Candidate Generation Stage*

For each feature, candidates for values to be determined and any constraints concerning the values are generated. This is accomplished by heuristic rules stored in a rule-base.

In the tool selection module, for example, tool candidates are enumerated for each feature with preferential weight in consideration of cutting efficiency. In the machining ordering module, constraints between two machining elements are generated as relations.

(B) *Optimization Stage*

Values are selected from the above candidates in order that no constraints are broken. This optimization is accomplished by algorithms specialized for each task.

For example, in the tool selection module, a tool is selected for each particular feature from out of the available candidates. Because the number of tools to be attached to one machine is limited, tools are selected in consideration of their commonality, as well as their preferential weight. In the machining ordering module, machining elements are ordered under the above constraints, so as to minimize the total length of tool paths.

Figure 6 shows the basic architecture for one task module.

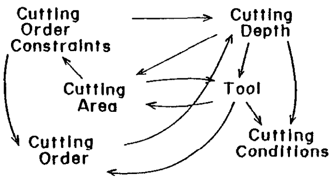

Figure 4 : Dependence among principal factors

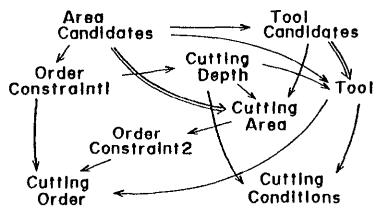

Figure 5 : Adjusted dependence

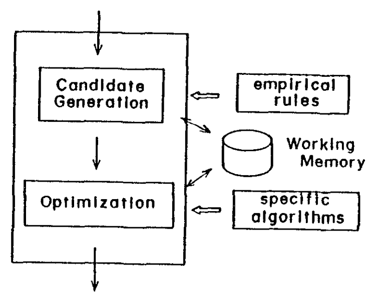

Figure 6 : Basic task module configuration

```
if   (&MEMBER Material (&GET ?FinishTool 'RecommMaterials))
     (&<= (&GET Pocket 'BottomRoughness) 2)
     (&>= (&GET ?FinishTool 'Diameter)
          (* (&GET Pocket 'MinCornerR) 1.6))
     (&<= (&GET ?FinishTool 'Diameter)
          (* (&GET Pocket 'MinCornerR) 2.0))
then (&GEN-REL FinishToolCandA Pocket ?FinishTool)
```

Figure 7 : Example of candidate generating rules

In the system, all data, including input data, intermediate results and final results, are stored in a working memory in the form of frames and relations among them. In the candidate generation stage, the following operations are applied to the working memory.

a1) **Instantiate** (*class*)
a2) **Delete** (*instance*)
a3) **Generate-relation** (*relation instance1 instance2*)
a4) **Change-relation** (*relation instance1 instance2*)
a5) **Delete-relation** (*relation instance1 instance2*)

Figure 7 shows an example of empirical rules used to search tool candidates. In the optimization stage, the following operations are applied.

b1) **Establish** (*instance attribute value*)
b2) **Change** (*instance attribute value1 value2*)

Each task module, also represented as a frame, determines the particular attributes for each feature. These modules are controlled using control rules. As task modules are activated in succession, details of the manufacturing plan are decided by degrees.

In this way, the search process can be controlled. However, since the search is not exhaustive, there is no guarantee that a globally optimal solution is always obtained. Consequently, user intervention is sometimes indispensable.

4.2 Interactive Optimization Strategy

In order to obtain a better solution, advice by a user is helpful. In ESPER, this advice comes from the user interface, by which the user can check intermediate results and operate them. This kind of operation by the user is called "user actions". ESPER can be used to investigate alternative plans, when any user action occurs.

4.2.1 Incorporating User Actions into Optimization

Since process planning goes on basically as a breadth-first search, combining system inference with user operation is not easy, for the following reasons:

(1) The influence of partial change spreads over a wide range. For example, a tool change in one area may influence the total number of tools to be used in one machine, so if there is no room for adding a new tool, one of the other selected tools must be canceled.

(2) The time when a user demand will appear cannot be specified in advance. Whenever users want to make changes, they are not allowed to contradict what has been already specified.

One of the important goals in developing ESPER was to provide flexibility to the user, regardless of the order in which the solution space is searched. To accomplish this, a method has been developed for allowing the user to specify any change, addition or deletion, which will automatically be incorporated into the next search process.

A typical interaction flow between the system and the user is as follows:

a. After planning has been finished, the results are presented by request to the user in the form of tables and graphics.
b. The user evaluates the results and changes them if needed. These actions are registered in preparation for later activation at the appropriate moment. At last, the user appoints a break-point where he/she wants to stop planning.
c. On the basis of user actions, it is possible to know from what level the planning should be carried out again, and the planning goes back there.
d. At each appropriate task, the user's instructions are added as restrictions, and then the pertinent task module is activated.
e. As the planning proceeds, the validity of each action is examined, and invalid actions are removed.
f. After the planning is finished, results and valid actions are presented to the user. He/she can see where user actions have been activated, as well as check the results in the form of tables and figures. He/she may add other actions and cancel some of the valid actions.

4.2.2 Controlling Planning Process

The actions taken by the user include the same operations as the system takes on the working memory, that is, a1 to a5 and b1 and b2 in 4.1.2. In addition to them, the following action is also allowed to the user.

c) **Cancel** *(action)*

The user actions are stored on the list called "action agenda". Each action is connected with a pertinent task module frame.

The task to be carried out first is determined by the control rules. After the candidate generation stage or optimization stage for each task is carried out, validity is examined for pertinent actions in the agenda. Only valid actions are fired and kept on the agenda. After the task module and the pertinent actions have finished, the next task to be carried out is selected by the control rules.

The above procedure is repeated until it comes to the task where the user has specified stopping. Next, user interaction begins again, in which the user can analyze the values determined by the system and those determined by user actions. Figure 8 shows this control architecture for a task module, including the reflection of user actions.

5 Implementation

Based on the above presented techniques, the ESPER system was implemented. ESPER was developed for automating process planning, that is, automatically generating manufacturing data, such as NC tapes, from CAD data. It also provides both graphical and tabular interface to a user, including cutting simulation, for comprehension ease. Figure 9 shows a general view of the system.

The system currently deals with 2.5 dimensional mechanical components made of aluminum or steel, which can be machined by a machining center. A sample component is illustrated in Figure 10.

The system is implemented in UTILISP and EXCORE, a knowledge representation language featuring frame, rule and

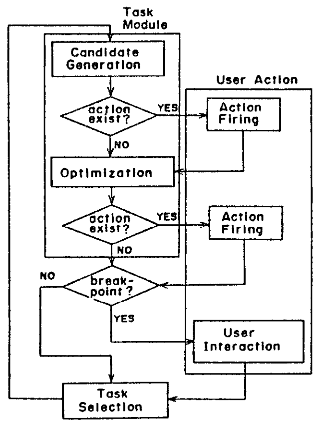

Figure 8 : Optimization incorporating user actions

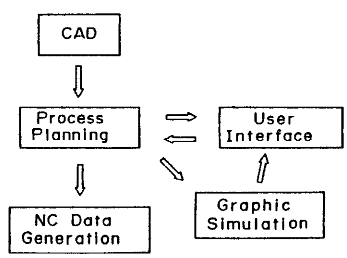

Figure 9 : Planning flow using ESPER

object-oriented descriptions, except for the user interface, which is in C. The system introduces frame representation to describe design objects and manufacturing environments, while empirical knowledge is represented in the form of rule blocks. It contains a control module and 13 task modules, composed of rule blocks and specific algorithms, as follows:

(1) Finishing areas and tools selection
(2) Roughing areas and tools selection
(3) Middle roughing areas selection
(4) Milling order determination
(5) Milling conditions determination
(6) Drilling element development and tools selection
(7) Drilling conditions determination
(8) Drilling order determination
(9) Tool sequence determination
(10) Cutting path generation
(11) Approach and retraction path generation
(12) Drilling cycle path generation
(13) Positioning path generation

6 Concluding Remarks

This investigation has been conducted in order to bridge the gap between AI techniques and their practical application. For this purpose, two concepts, essential in developing a practical process planning system, were proposed.

Geometric reasoning power realizes close ties between knowledge processing and geometry. ESPER has been interfaced with a CAD system by means of a feature extraction facility. Moreover, it can make delicate judgments concerning geometry and can accomplish geometric manipulations easily.

The proposed optimization strategy has been shown feasible in the ESPER system. ESPER involves the ability to incorporate user interactions into the optimizing process. They can be used later to find any deficiency in the planning process and improve it. A more sophisticated interface for ESPER, which shows the planning process itself graphically, is currently being developed.

The authors' goal is to develop a totally integrated environment, including design, process planning and manufacturing. Though further work is needed to take full advantage of these approaches, the results obtained demonstrate the viability of the presented ideas.

Acknowledgements

The authors would like to thank Dr. S. Goto, Mr. T. Yoshimura and Mr. K. Kawagoe for their encouragement in this work. Further, the authors would like to thank Mr. S. Kawabata, Mr. N. Uemura, Mr. K. Kobayashi and Mr. M. Edahiro for their valuable suggestions and discussions.

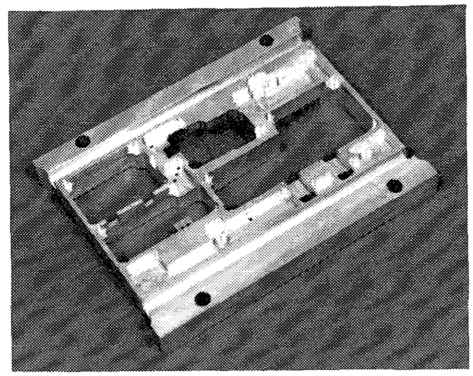

Figure 10 : Target component

References

[Choi and Barash, 1985] B. K. Choi and M. M. Barash : "STOPP - An Approach to CADCAM Integration", *Computer-Aided Design, Vol.17, No.4*, 1985

[Davies and Darbyshire, 1984] B. J. Davies and I. L. Darbyshire : "The Use of Expert Systems in Process Planning", *Annals of CIRP, Vol.33, No.1*, 1984

[de Cleer, 1986] J. de Kleer : "An Assumption-Based Truth Maintenance System", *Artificial Intelligence, Vol.28, No.1*, 1986

[Descotte and Latombe, 1985] Y. Descotte and J. C. Latombe : "Making Compromises among Antagonist Constraints in a Planner", *Artificial Intelligence, Vol.27*, pp.183-217, 1985

[Doyle, 1978] J. Doyle : "Truth Maintenance Systems for Problem Solving", *MIT AI Lab Technical Report 419*, Cambridge, 1978

[Eliyahu *et al.*, 1987] O. Eliyahu, L. Zaidenberg and M. Ben-Bassat : "CAMEX - An Expert System for Process Planning on CNC Machines", *Proc. AAAI-87*, pp.794-798, 1987

[Henderson and Anderson, 1984] M. R. Henderson and D. C. Anderson : "Computer Recognition and Extraction of Form Features - A CAD/CAM Link", *Computers in Industry, Vol.5, No.4*, 1984

[Hirschtick and Gossard, 1986] J. K. Hirschtick and D. C. Gossard : "Geometric Reasoning for Design Advisary Systems", *ASME International Computers in Engineering Conference & Exhibition*, 1986

[Imai and Asano, 1984] H. Imai and T. Asano : "Dynamic Segment Interaction Search with Applications", *25th Annual IEEE Symposium on Foundations of Computer Science*, pp.393-402, 1984

[Jared, 1984] G. E. Jared : "Shape Features in Geometric Modeling", in M. S. Picket and J. W. Boyse(eds.): *Solid Modeling by Computers - From Theory To Application*, Plenum, 1984

[Murphy *et al.*, 1987] A. Murphy, V. Jagannathan and S. Goodrum : "Blackboard Approach to Process Planning Problems", in D. Sriram, R. A. Adey(eds.): *Knowledge Based Expert Systems in Engineering: Planning and Design*, Computational Mechanics Pub., 1987

[Rivest, 1982] R. L. Rivest : "THE *PI* (Placement and Interconnect) System", *ACM 19th Design Automation Conference*, pp.475-481, 1982

[Sacerdoti, 1977] E. D. Sacerdoti : "*A Structure for Plans and Behavior*", American Elsvier, 1977

[Smith and Ow, 1985] S. F. Smith and P. S. Ow : "The Use of Multiple Problem Decomposition in Time-constrained Planning Tasks", *Proc. 9th IJCAI*, 1985

[Stefik, 1980] M. J. Stefik : "Planning with Constraints", *Rep. No.80-784*, Dept. of Computer Science, Stanford University, 1980