An Interactive Approach for Satisfying Process Plan Generation

Xiaomin Li *, Subbarao Kambhampati & Jami Shah
College of Engineering and Applied Sciences
Arizona State University, Tempe, AZ 85287
Email: {xli, rao, jami.shah}@asu.edu

Abstract

Most existing approaches to computer aided process planning aim at full automation by searching for a plan that is optimal with respect to a pre-specified objective function. Such approaches are often infeasible in practice for three reasons: (i) the search space of potential plans is very large, (ii) optimality metrics are often context sensitive and can only be elicited through user interaction, (iii) because of the importance of process planning, organizations are more interested in process planning assistants that support human expert rather than autonomous planners.

In this paper, we will describe a new approach to handle the combinatorics of the search space of process plans and generate a satisfying process plan through interaction with the user. In our planning system, a *satisfying plan* is either a plan which minimizes the penalty cost associated with evaluation criteria violations, or one that satisfies the expert user. Knowledge about process plans is obtained from the Arizona State University Features Test Bed(ASUFTB), a comprehensive and systematic framework for recognizing and reasoning with features of machinable parts. Our approach can be seen as searching the space of interpretations for a design part as plans set up by ASUFTB. We will discuss the detailed algorithms and experimental results for satisfying process plan generation.

Introduction

Computer aided process planning (CAPP) is a key part to bridge the design and manufacturing. Process planning involves determining which sequence of operations to perform to manufacture a part given its description and the specification of the resources in the workshop. It should take into account both technological and economic considerations, some of which are hard constraints and others are preferences. This knowledge often represents both the experience and the know-how of engineers/specialists, which differ from one company to another.

Some of the existing approaches to CAPP (Britanik,1995; Gupta,1994; Hayes,1996; Kambhampati,1993) aim at full automation by searching for a plan that is optimal with respect to a pre-specified objective function. Such approaches suffer from three important limitations.

- First, the search space for process plans is too large to facilitate an efficient systematic search. This often necessitates restricting focus to a single interpretation of the current design and finding the best plan under this fixed feature set (which may not be the best plan globally).
- Second, and perhaps more important, these approaches assume the availability of a pre-specified objective function for evaluating process plans. In reality, the evaluation metrics for process plans are very much context dependent, and it is rarely the case that an accurate optimality metric is available *a priori*. Trade-offs among optimization objectives typically reflect user preferences and the presence of additional domain constraints not captured in the planning model. Moreover, the user may change the optimality criteria for process planning in the course of finding an optimal solution. Since the optimality metric is not completely known, we would like to make the user/process planning expert to be the final arbiter on the quality of the plan produced. Accordingly, if the expert is not satisfied, the planner should be able to resume its search for an improved process plan.
- A third and related shortfall of the current approaches is that they attempt to reach full automation in a situation where organizations are not comfortable delegating full process planning responsibilities to a computer.[1]

In this paper, we will describe a new approach to handle the combinatorics of the search space of process plans, and generate a satisfying plan through interac-

---

[1] Prof. Mantyla, a prominent process planning researcher, relates an anecdote about how when his research group offered their state-of-the art process planning system for use in a Finnish company, the company politely refused saying that process planning is too important an activity to be entrusted solely to a program.

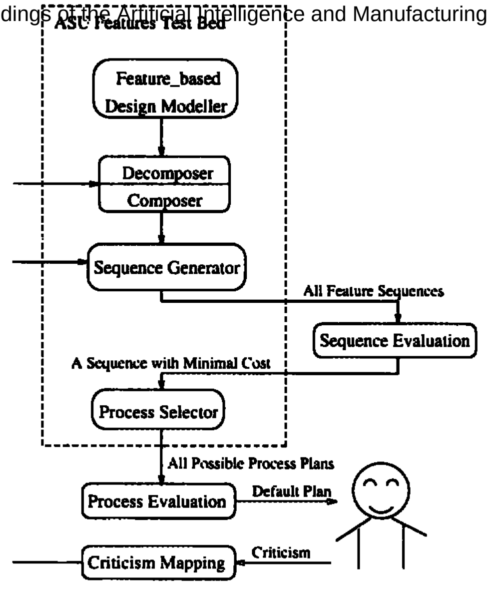


Figure 1: Planning System Architecture

tion with the user. In our planning system, a satisfying plan is either a plan which minimizes the penalty cost associated with violations of the evaluation criteria or one which satisfies the expert user. Knowledge about process plans is obtained from the ASU Features Test Bed(ASUFTB), a comprehensive and systematic framework for recognizing and reasoning with features in machinable parts. Our approach can be seen as searching the space of interpretations for a design part as plans set up by ASUFTB.

The general idea of our approach is given in Figure 1. The intended user is an experienced human process planner, who is knowledgeable about both the products and the manufacturing facilities of the factory. If the user is satisfied with the current plan, he may terminate the process; otherwise the user criticism is specified through the planning system interface, and is mapped by the planning system to make a new seed plan and the process continues. The real issue in applying this approach is operationalizing the user criticism as advice about how to re-navigate the search space of possible plans.

More specifically, our approach involves the following steps:

1. For a given part, evaluates all corresponding feature sequences, and pick up one with least sequence level penalty cost;
2. Generates all possible process plans for the chosen feature sequence, evaluates them, searches greedily the sequence of machining operations with minimum process level penalty cost, and presents it to the user;
3. If the user is satisfied with the plan, exits with success. Otherwise, goes to step 4.
4. Allows the user to specify the criticism based on the evaluation metrics;
5. Maps the criticism to make a new seed plan, and goes back to Step 3.

The rest of this paper is organized as follows: We start by briefly reviewing ASUFTB and the representation of process plans within it. Then we discuss the details of the search algorithms – including how plans are evaluated, how the total penalty costs are calculated, how the default plan is generated, and how the user criticisms are mapped by the planning system to improve the plan. Finally, we discuss an example, and present the conclusions and future work.

The ASU Features Test Bed (ASUFTB)

As mentioned earlier, knowledge about process plans is obtained from ASUFTB, which was developed by Shah et al. (Shah,1994). ASUFTB is a design by feature system which can recognize all possible interpretations of the machining features of a given part. These interpretations are used to systematically enumerate all candidate machining plans. We will now briefly describe the operations of ASUFTB with the help of Figure 1, and the example in Figure 2.

The feature_based design modeller is used to describe the given part and its stock. The decomposer decomposes the total removable volume (R), obtained by subtracting the part (P) from the stock (S), into minimum convex cells called atomic cells using a method called "halfspace partitioning" (Shah,1994). Figure 3 illustrates 12 minimum convex cells obtained by decomposing R (see Figure 2) using the half space partitioning method. Two atomic cells are considered to be adjacent if they share a face on the same halfspace. Some atomic cells such as cell 8 and cell 1 need special consideration because these cells serve as crossroads - they signify that there exist alternative ways of composition. These special cells are called *joining cells*. The composer composes atomic cells in different ways to generate Feature Based Models (FBM) by starting at each joining cell and continuously concatenating adjacent cells (unless the volume becomes concave). At this stage the concatenated volume is maximally convex and is assumed to be machinable, so the process begins again at some other joining cell. The sequence generator generates various machining sequences by traversing each FBM.

Every machining sequence is an input to the process selector where process selection is done for individual features in the sequence on the basis of shape capability of the process (Shirur,1994). A machining expression, based on both the tool-workpiece interaction and the tool motion characteristics, is generated for each feature. Machining processes in ASUFTB are modelled in terms of constraints on the tool shapes, cutting motion types between the tool and the workpiece, and the possible directions of feed motion. Feasible processes for each feature are determined by matching their process constraints with the machining expression of the feature. Any process whose constraints completely match

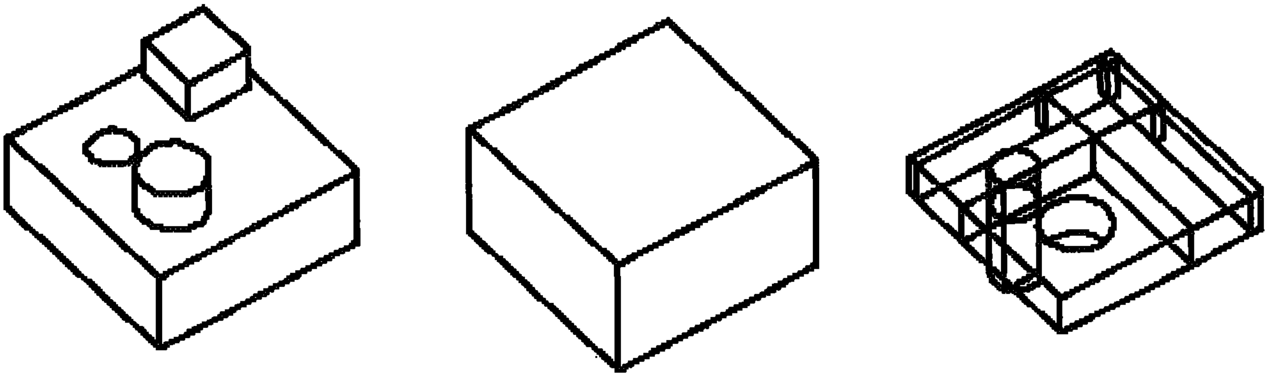


Figure 2: Example: part (P) and stock (S), the removal volume R (S-P) is decomposed (using halfspace partitioning) into 12 atomic cells

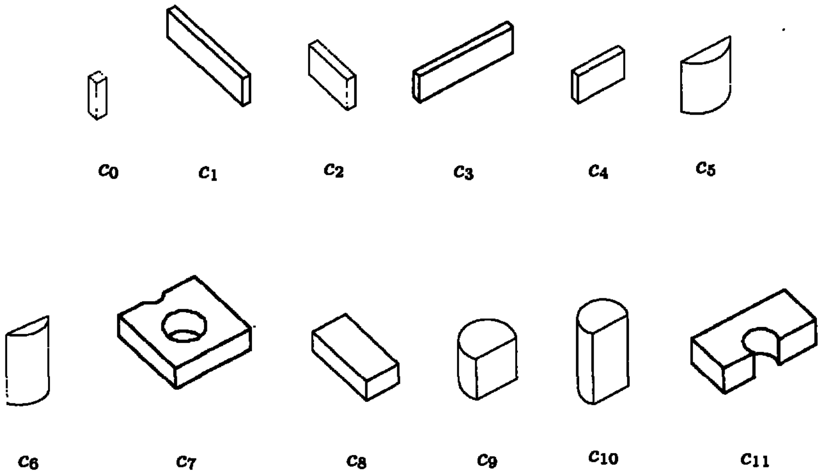


Figure 3: Atomic cells for the removal volume R (c_0 to c_{11})

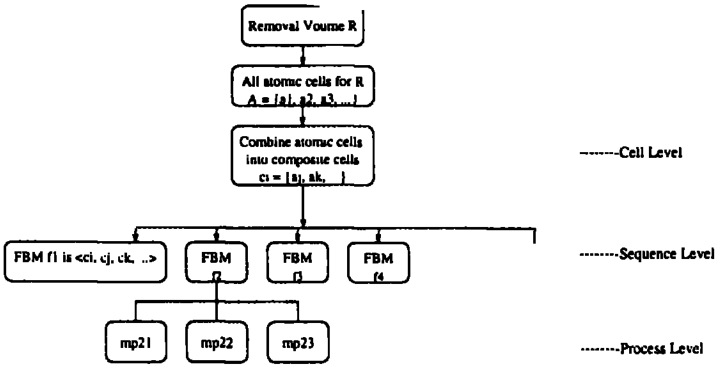


Figure 4: Process Plan Search Space Setup By ASUFTB

the terms in the machining expression is added to the list of feasible processes for a given feature. Since a feature can have multiple machining expressions and each of them maps to several alternate processes, an exhaustive list of alternative processes for each machining feature is generated. A detailed description about ASUFTB can be found in (Li,1997).

In summary, support for process planning in ASUFTB comes at three different levels. The first level, called the cell level, involves the Decomposer and the Composer which split the removal volume into atomic cells and combine them into FBMs. The second level, called the sequence level, involves the Sequence Generator which picks a FBM and makes a traversal to generate a machining sequence. The third level, called the process level, involves the Process Selector which chooses feasible machining processes for each of the features in the chosen FBM. Thus, ASUFTB implicitly sets up the search space shown in Figure 4 consisting of all potential process plans for machining the part. The objective of our approach is to navigate this search space guided by the evaluation metrics and the user feedback.

Satisfying Process Plan Generation

As we have discussed in the last section, ASUFTB builds up the search space of process plans, and the generation of a satisfying process plan consists of heuristically searching this huge space. In this section, we will discuss the details of the search algorithms, focusing on how plans are evaluated, how the total penalty costs are calculated, how the planning system generate a default process plan, and how the user criticisms are mapped by the planning system to find a new plan.

Plan Evaluation and Cost Calculation

First, we explain how a plan is evaluated, and how the penalty cost for the process plan is calculated. A plan is evaluated from two levels: sequence level and process level.

There are five evaluation operators:

- Accuracy operator

  In the sequence level, accuracy operator measures whether the order of features in a machining sequence is correct. In the process level, accuracy operator checks whether the assigned processes meet the form and finish specifications.

- Consistency operator

  In the sequence level, the consistency operator measures the ability of process plans to repetitively produce within the specified tolerances. In the process level, the consistency operator focuses on the repetitive capability of processes to produce intrinsic dimensions of a feature within the specified tolerances.

- Operation time operator

  In the sequence level, the operation time operator evaluates the air time, when no cutting is done but the tool is being moved from one position to another; and the tool change time, which depends on the number of feature types. In the process level, the operation time considers only the machining time, which is the amount of time taken for removing the feature volume by the machining process.

- Feasibility operator

  The feasibility evaluation operator only applies to the process level evaluation. It compares the size capability of the processes with the intrinsic dimensions of features, and checks whether they are feasible candidates for machining the feature volume.

- Setup operator

  The setup evaluation operator compares the specified minimal setup number (default is one) with that of the current plan. It only applies to the process level evaluation.

  In our planning system, the minimum number of setups for a plan is estimated by calculating the tool approach directions of the features and the precedence constraints among them. The profile of each process is associated with a marker that can be used for positioning and orientation (Shirur,1994). The marker defines a local coordinate frame which is used to specify both the direction of interference and the direction of sweep of the tool. The x-direction of the marker points in the direction of the interference, and the z-direction points in the direction of the instantaneous sweep of the profile. The y-direction is chosen so as to maintain a right-handed orthogonal. To estimate the number of tool approach directions, the planner calculates the different x-directions of the marker under the precedence constraints among the processes.

For each evaluation criteria, the penalty cost of a violation is one unit. The total cost for each level evaluation will be the sum of the product of the weight and the violation cost of each criteria in the level. The planning system also supports changing the default weights if necessary. Detailed information about evaluation operators can be found in (Kartheek,1996).

Default Plan Generation

We now describe how the default plan is generated. To choose a FBM which is more likely to generate a good default process plan, the planning system evaluates all the FBMs using the sequence evaluation operators, and picks up the FBM with the least penalty cost. All the possible process plans for the chosen FBM are then generated by the process selector, and evaluated using process level evaluation operators. The planning system finally uses a greedy approach to find one with the least process level penalty cost, and presents it to the expert user.

Plan Search Space Renavigation Through Interaction with the Expert User

The planning system supports the user to critique current process plan in terms of evaluation metrics. Based on each type of criticism, the system will map it into modifications to current plan.

- Criticism to the current process plan in either sequence level accuracy or operation time (air time) is mapped to find a feature sequence with a different order from the previous one but having the same feature set since only the order of features has effects on these two metrics.
- Criticisms in process level accuracy, feasibility, and setups are mapped to find new processes to replace some of the old ones because the characteristics of the process is closely related to these metrics.
- Criticism for operation time in process level (actual machining time) is mapped to split the largest feature in the current plan since the machining time is proportional to the amount of material being removed and has nothing to do with the feature or operation orderings. The algorithm used to split the feature is illustrated in Figure 5, where F is a set of features consisting of the machining sequence, and f is a feature belonging to F which needs to be split. It results in a different FBM without feature f and including new required features and as many old feature as possible. This method can be considered as a splitting and merging of the composite cells in the FBM.

As we know, if there is no feasible process for one of the features in the FBM, such FBM can be pruned from the plan search space immediately. This basic mechanical knowledge is incorporated in the planning system to reduce the FBM branch (see Figure 4 for the search space setup by ASUFTB) when the user criticisms are mapped to renavigate the process plan space.

```
procedure SPLIT_MERGE(F, f)
1. Find all joining cells in $f$.
2. For each joining cell $jc \in f$,
    2.1 Check whether $jc$ can be combined with each feature $f' \in F - \{f\}$. For any such $f'$, do the following :
        2.1.1 Combine $jc$ with $f'$ and generate a new feature $nf$.
        2.1.2 If $nf$ is concave, split it into two convex features $nf1$ and $nf2$. The planning system then searches the machining sequences which include $nf1$ and $nf2$.
        2.1.3 Otherwise, the system searches the machining sequences which include $nf$.
    2.2 For all the machining sequences generated in 2.1, find the one with the maximum number of other old features. If there exists more than one of such sequences, evaluate them using the sequence level criteria, and pick up the machining sequence F' with the least penalty cost.
3. Return F' with success.
```

Figure 5: Algorithm for Plan Modification in the Cell Level

Example

In this section, we will give an example which has been tested using our approach. Figure 6 illustrates a feature sequence chosen by ASUFTB which has the least sequence level penalty cost for the machining part in Figure 2. The corresponding operation set generated by the greedy approach which has the minimum process level penalty cost is listed as follows:

- 000
  Turning
  10
- 000
  Plunge Milling
  5
- 00-1
  Plunge Milling
  39
- 000
  Plunge Milling
  5
- 000
  Plunge Milling
  5

The number below each process indicates the penalty cost of the process for violating the evaluation criteria. The number above each process such as 001 is the z-direction of the local coordinate frame associated with the profile of the process, and it points to the tool approach direction of the corresponding process. The total penalty cost for a process plan is the sum of the cost for each process and the cost of the tool approach directions when it is greater than the specified one. For current plan, its total penalty cost is 71.

Figure 7 illustrates a new feature sequence generated by the algorithm SPLIT_MERGE when the previous plan is presented to the user, and the user is not satisfied with the operation time in the process level. The second feature $\{c_0, c_3, c_4\}$ in Figure 6 is the feature to be split. The new features generated by the SPLIT-MERGE procedure are the feature $\{c_3, c_7, c_{11}\}$ and the feature $\{c_0, c_4\}$ in Figure 7. The corresponding best process plan is:

- 000
  Turning
  10
- 00-1
  Plunge Milling
  15
- 00-1
  Shaping
  10
- 00-1
  Shaping
  5
- 00-1
  Broaching
  10

As we have seen, the cost of the new process plan is 50, which is 30% less than the previous one(71). This example shows that our approach in handling the combinatorics of the search space of process plans is very efficient.

Conclusion and Future Work

In this paper, we proposed a new approach for handling the combinatorics of the search space of process plans, and for generating a satisfying process plan. In our planning system, a *satisfying plan* is the plan which either minimizes the penalty cost associated with the evaluation criteria violation, or satisfies the expert user. We described and illustrated the implementation and operation of our approach on top of the ASU Features Test Bed, concentrating in particular on how a plan is evaluated, how the total penalty costs are calculated, how the default plan is generated, and how the user criticisms are mapped by the planning system to find a new plan. The detailed algorithm has been given, and the experimental results were also illustrated. As we have seen, our approach handles the combinatorics of the search space of process plans by guiding the search direction through self-evaluation as well as the criticism specified by the expert user. We will focus on exploring the issues involved in improving the user-planner interface so that a larger variety of user criticisms can be handled. We will also analyze the method we use in process planning to see if it can be generalized to AI planning.

References

J. Britanik, and M. Marefat. 1995. Case-Based Manufacturing Process Planning with Integrated Support for Knowledge Sharing. *In Proceedings of the 1995 International Symposium on Assembly and Task Planning (ISATP'95), 107-112,* Pittsburgh, PA.

Chang, T. -C. 1985. *An Introduction to Automated Process Planning Systems,* Englewood Cliffs, New Jersey: Prentice Hall

John R. Dixon, Adele Howe, Paul R. Cohen, Melvin K. Simmons. 1986. DOMINIC I : Progress Towards Domain Independence in Design by Iterative Redesign. *In Proceedings of the ASME Computers in Engineering Conference,* Chicago, IL.

S. K. Gupta, Dana S. Nau, William C. Regli and Guangming Zhang. 1994. A Methodology For Systematic Generation And Evaluation Of Alternative Operation Plans.*In Advances in Feature Based Manufacturing, 161-184.*

Caroline C. Hayes. 1996. P3: A Process Planner for Manufacturability Analysis. *IEEE Tran. on Robotics and Automation* 12(2).

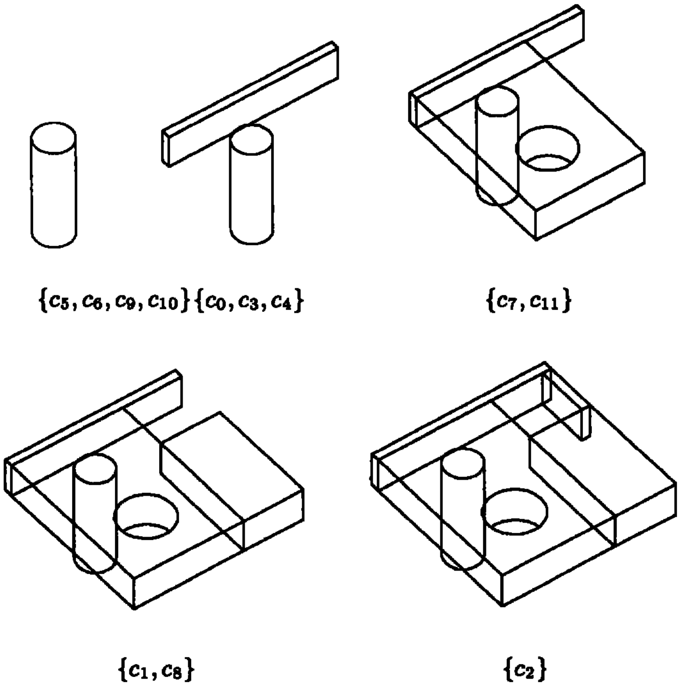
Kartheek V. Hirode. 1996. Automated Evaluation and Interactive Refinement of Machining Process Plans. M.S thesis, Dept. of Mechanical Engineering, Arizona State University.

Subbarao Kambhampati, Mark Cutkosky, Marty Tenenbaum and Soohong Lee. 1993. Integrating General Purpose Planners and Specialized Reasoners: Case Study of a Hybrid Planning Architecture. *IEEE Trans. on Systems, Man and Cybernetics, Special issue on Planning, Scheduling and Control* 23(6).

Xiaomin Li, Subbarao Kambhampati, Kartheek Hirode, and Jami Shah. 1997. Process Planner's Assistant : An Interactive and Iterative Approach to Automating Process planning. *In Proceedings of 1997 ASME Design Engineering Technical Conferences,* Sacramento, California.

Jami J. Shah, Yan Shen and Arvind Shirur. 1994. Determination of Machining Volumes from Extensible Sets of Design Features. *In Advances in Feature Based Manufacturing, 129-157.*

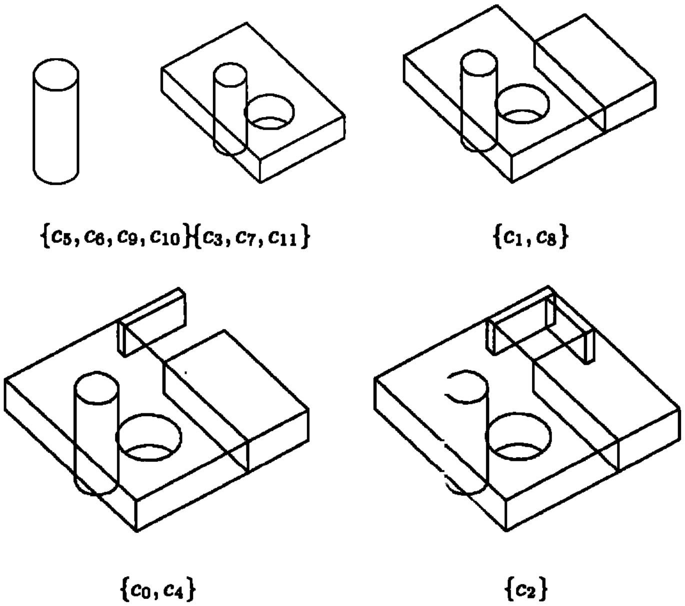
Arvind Shirur. 1994. Automatic Generation of Machining Alternatives For Machining Volumes. M.S thesis, Dept. of Mechanical Engineering, Arizona State University.



$\{c_5, c_6, c_9, c_{10}\}\{c_0, c_3, c_4\}$ $\{c_7, c_{11}\}$

$\{c_1, c_8\}$ $\{c_2\}$

Figure 6: A machining sequence generated by ASUFTB. The features comprising the FBM are removed one by one starting at the top left. The atomic cells corresponding to each removed feature are labelled under the respective subfigures.



$\{c_5, c_6, c_9, c_{10}\}\{c_3, c_7, c_{11}\}$ $\{c_1, c_8\}$

$\{c_0, c_4\}$ $\{c_2\}$

Figure 7: Cell-level modification of plans, showing how the features comprising the FBM are removed one by one starting at the top left. The atomic cells corresponding to each removed feature are labelled under the respective subfigures.