A Model of Planning for Plan Efficiency: Taking Advantage of Operator Overlap.

Caroline Clarke Hayes
The Robotics Institute, Carnegie Mellon University
Pittsburgh, Pa. 15213

Abstract*

Many realistic planning problems (such as those in manufacturing) place a high premium on plan efficiency. However, classical planning theory does not offer much insight into ways of obtaining efficient plans. The objective of this paper is to present a model of planning for plan efficiency that has been drawn from a case study in the machining domain. Some of the important features of this domain are that problems are stated as sets of conjunctive goals, and operators that achieve those goals have a high degree of *overlap.* Operators can be said to overlap when they can share work. Because of this overlap, the cost of the operators is dependent on their order in the plan, (for example, it is less time consuming to buy vegetables if your last action was also done at the grocery store) However, looking for a near optimal set of overlapping operators can lead a very expensive search. The methods that human machinists were found to be using reduced the complexity of the search for good operators by using cues and patterns in the problem specification, gained from experience, to tell them when it might be useful to explore an operator.

1 Introduction

Many planners understand and exploit *negative interactions* between goals to organize a plan, but few exploit *positive interactions* to a large degree. (A negative interaction happens when accomplishing one goal hinders accomplishing another, and a positive interaction happens when accomplishing one goal helps to accomplish the other.)

One important use of positive interactions is to make plans more efficient. However, relatively little attention has been given to positive interactions. During the mid 70's, the primary planning problem was how to automatically generate plans at all, so although early "classical planners" [Fikes, 1972, Sacerdoti, 1974, Scacerdoti, 1975] considered efficiency to some extent, it was not their primary goal. During the late 70's and 80*s, plan efficiency was given a little more attention and a few researchers such as [Wesson, 1977], WilensJri [Wilensky, 1983], and Hammond [Hammond, 1988] have examined the use of positive interactions as a means of achieving it. The purpose of this paper is to focus on one particular type of positive interaction called *operator overlap,* the special difficulties that it presents, and to describe some behaviors observed in human machinists for making the search for near optimal plans more efficient. The real difficulty to be addressed in this paper is not in recognizing that operator overlap can be used to make plans more efficient, but in knowing what factors influence operator overlap, and how can to reduce the search in this very difficult problem of attempting to make near optimal plans.

2 Operator Overlap

There are many ways in which positive interactions can occur.[1] In particular, this paper focuses on *operator overlap,* positive interactions that occur when two operators share work. Wilensky's concepts of partial plan overlap and plan overlap "in which the execution of a single action fulfills a number of goals simultaneously,"[2] are very similar in spirit to operator overlap.

As an example of plan overlap, suppose one had four errands to run: pick-up cash at grocery store cash machine, drop off letters at the post office, pick-up ice cream at the grocery store, and buy stamps at the post office. The "buy stamps" and "drop off letters" operators can be said to overlap in terms of their destination: both are done at the post office. "Pick-up cash" and "pick-up ice cream" also share a destination, the grocery store. If one does the errands in the order they are listed above, one has to go through the effort of getting to the grocery store and the post office twice. However, if one groups "buy stamps" and "drop off letters" together, and "pick-up cash" and "pick-up ice cream" together, then one only has to go to each place once (assuming one has enough cash at the beginning to buy the stamps.)

Operator overlap can be further divided into two kinds: those with temporal dependancies, and those without It is this first type of operator overlap that we are concerned with in this paper, in which results disappear if not used immediately. For example, the condition of being at the post office goes away if you go to the grocery store between buying stamps and mailing the letters. However, the condition of "having money" will not *necessarily* change if you do some other operation between buying stamps and buying ice cream. This particular kind presents additional difficulties because the cost of accomplishing each goal is dependent not only on the operator used to accomplish it, but also on the order of the operators. Wilensky refers to this as

---

This project was funded by the Machinist Expert Consortium and Air Force Grant Contract #F33615-86-C-5-38.

Wilensky, "Planning and Understanding," Addison-Wesley, 1983, Chapter 10: Positive Goal Relationships, pp. 113-126.

[2] Wilenski, Planning and Understanding, Addison-Wesley, 1983, p. 119.

"partial plan overlap/' although a different name might capture the nature of the problem better.

Because the cost of an operator is dependent on its order, near optimum integer programming techniques such as those proposed by [Bard, 1989] and [Kusiak, 1985] cannot be adapted to this problem. In those models, it is assumed that the cost of the operation is not dependent on its order in the plan.

3 The Structure of the Machining Problem

In the type of machining problems discussed in this paper, the objective is to start with a rectangular block of metal and to use a three axis vertical computer numeric controlled (CNC) machining center to cut a variety of shapes or *features* into the block (see figure 1). The part shown has five features: two angles, a slot and two holes. Each hole is tapped (threaded) and chamfered (beveled) on the bottom side (side 4). The taps and chamfers are considered to be sub-features of the holes. The two angles form a 90 degree corner where they meet.

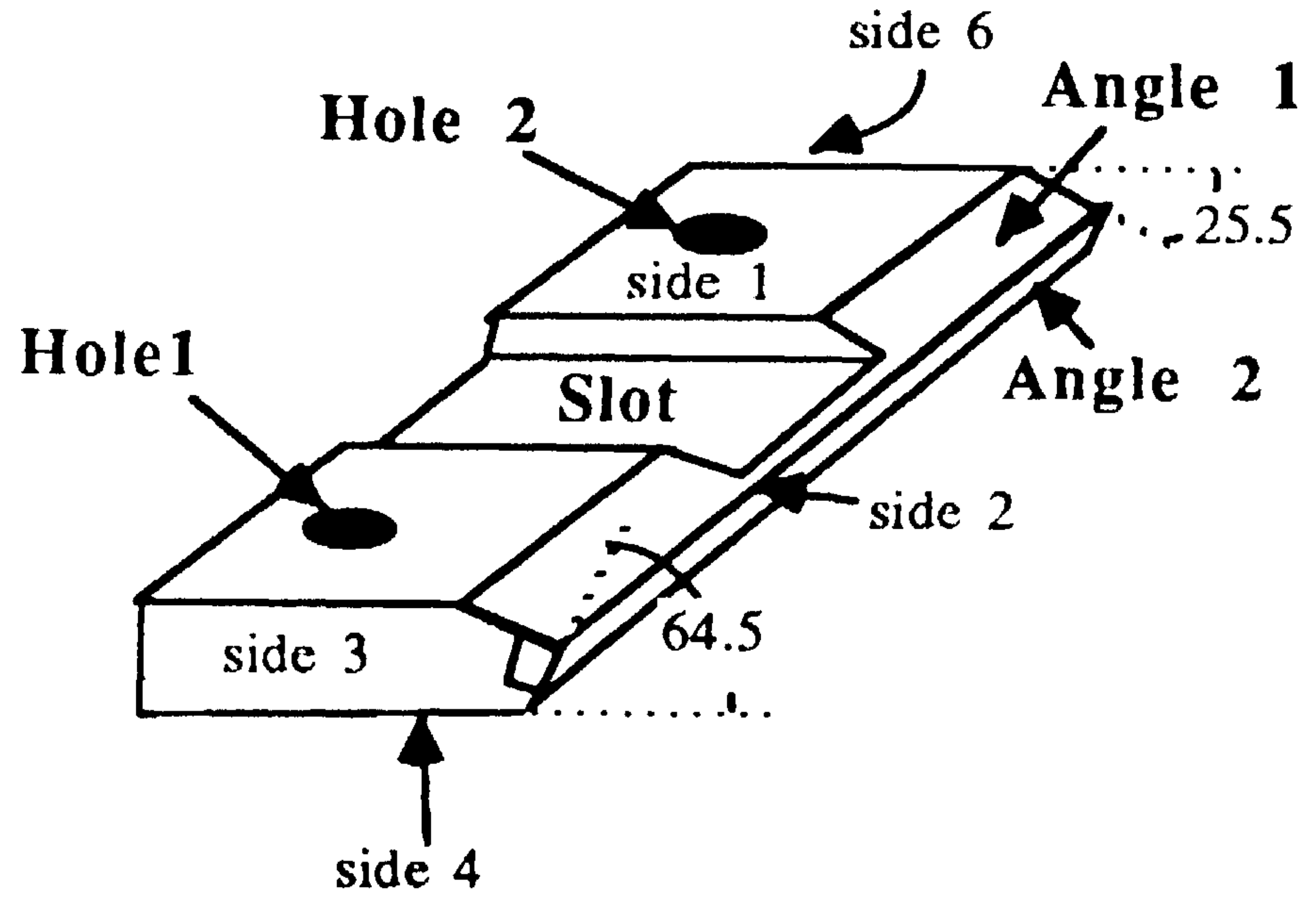

Figure 1: A part with a slot, two tapped and chamfered holes, and two angles.

To put the machining problem into classical planning terms: each geometric feature or sub-feature: hole, angle, pocket, etc., is a goal. The initial state is a prismatic block of metal. The machining plan must successfully achieve a conjunct of all these goals to make the part. For each goal, there may be a number of methods (operators) that can be used to accomplish that goal. For example, some operators which could be considered for making the angles are shown in figure 2. Each method specifies a method for *fixturing* of clamping the part, an orientation of the part, and a tool for cutting it. Each of these methods can be accomplished on the machining center, the machining center can perform a large variety of different types of operations such as drilling, milling, etc., but it can only do one operation at a time.

4 Overlapping Machining Operators

In order to form an efficient machining plan, one must overlap the operators that compose the plan as much as possible. As implied earlier, the idea is to shorten the plan by choosing for each goal the operator that overlaps the most with the other operators in the plan. Goals (features) that overlap are grouped together. The length of the plan is roughly minimized when one has formed a minimum number of such overlapping groups.

In any problem domain there is usually a number of ways in which operators can overlap; the primary way in which it happens for CNC machining operators is by sharing fixture set-ups. A *fixture set-up* refers to the way in which the part is oriented and clamped before it is cut. Set-up usually accounts for a large portion of the time cost of each operator: sometimes as much of 90% of the time may be spent in setting-up for one machining operation. It is assumed that all other costs are relatively small compared to set-up time, so that minimizing the number of set-up will approximately minimize the total cost of the plan. Although this assumption does not always hold, it is a fairly good rule of thumb. Human machinists use this same heuristic for minimizing the plan cost.

CNC machining operators may also overlap in other ways, such as by sharing the same tool. One can also shorten the plan by minimizing the number of times the tool must be changed: operators that use the same tool can be grouped together in the plan to avoid having to change the tool. However, it is much more time consuming to change a set-up than to change a tool, so these other types of overlap are typically considered only after set-up overlap has been considered. In any domain, it is important to know when the cost of one type of operator overlap consistently dominates another. If such a situation occurs, the problem can be decomposed nicely into two nearly independent sub-problems; first the plan is optimized for the bigger savings (i.e. fixture set-up), second, within each fixture group formed the plan is optimized for the lesser saving (i.e. tool changes) with in each group.

5 A Machining Example of Operator Overlap

The example will be draw from a study of the part shown in figure 1. Let us consider the two features Angle 1 and Angle 2. Angle 1, shown in figure, can be made by any the methods shown in figure 2. Each of these methods represents one possible machining operator that could be used to make Angle 1. (There are many other methods not shown, which will also work.)

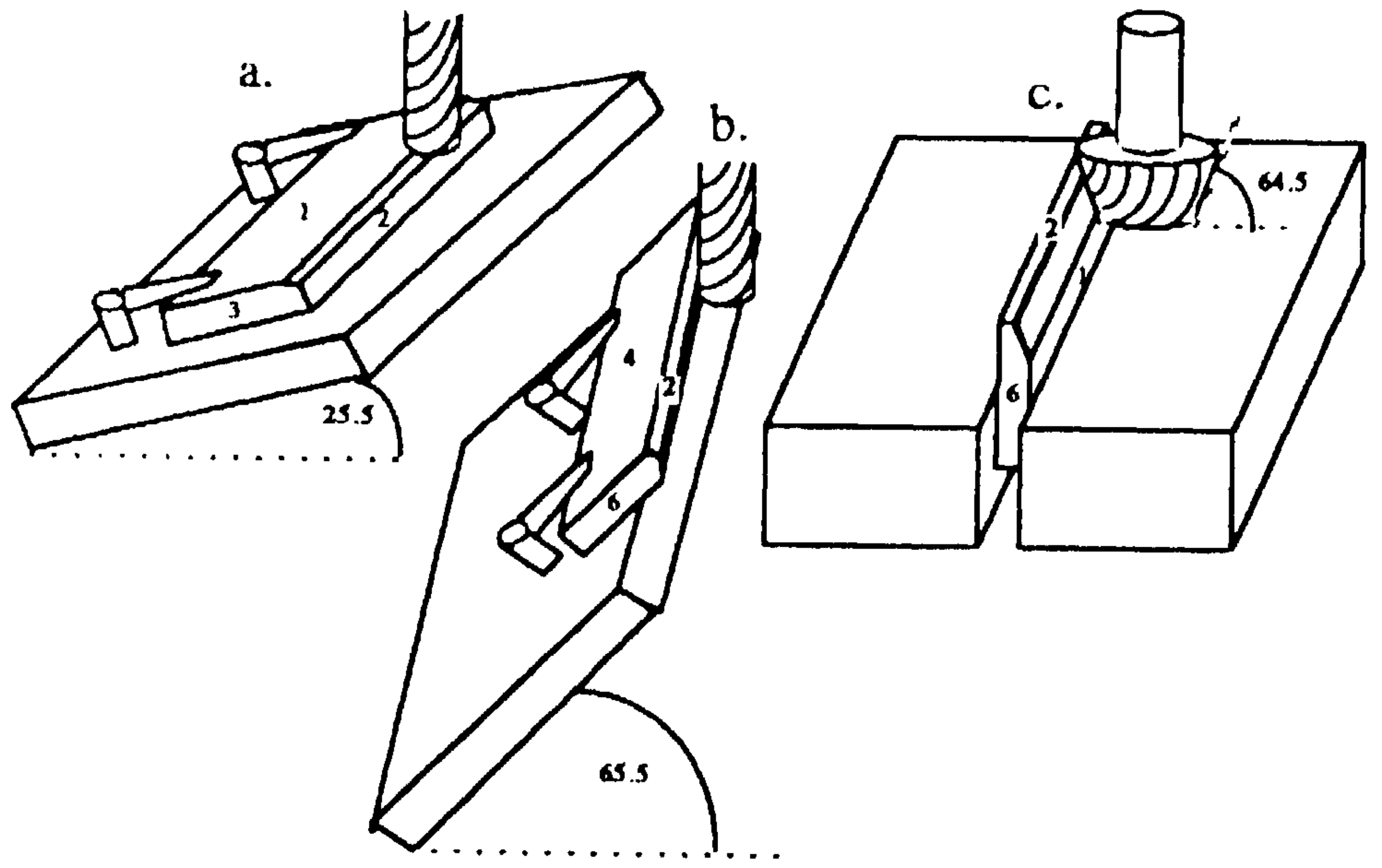

Figure 2: Some alternative methods for making Angle 1.

Each machining operator is described by an orientation of the part, a set of fixtures, and a set of tool types. The first method, (figure 2. a) the fixture is a sine table tilted at 25.5°, the tool is a face mill, and the orientation with respect to the sine table is: side 2 facing right, and side 1 facing up. Fixtures are devices used to clamp work pieces or hold them steady. All of the operators shown in figure 2.a, b and c can be described respectively as:

	feature:	orientation:	fixtures:	tool types:
op I.a	(Angle1	side 1 up side 2 right	sine table tilted 25.5°	face mill)
op I.b	(Angle 1	side 4 up side 2 right	sine table tilted 644°	side mill)
opI.c	(AngleI	side 2 up side 1 or 4	vise right	64.5° angled cutter)

When forming a plan one must choose only one of these operators for making Angle 1. Op I.c, which uses the vise, turns out *not* to be a good choice for this situation, but 1 .a and I.b, which both use the sine table, are both reasonable choices. In fact, if the decision was based only on local information, op I.a would be preferred slightly over op I.b However, the best operator choice should be based on how it affects the whole plan, not just on what is best for one individual feature.

Let us examine Angle 2 to see how it affects operator choice for Angle 1. The operators that can produce Angle 2 are roughly the same as those used for angle 1, except that either the tool type or the orientation is different.

	feature:	orientation:	fixtures:	tool types:
op2.a	(Angle2	side 1 up side 2 right	sine table tilted 25.5°	side mill)
op2.b	(Angle2	side 4 up side 2 right	sine table tilted 64.5°	face mill)
op2.c	(Angle2	side 4 up side 2 or 5	vise right	64.5° angled cutter)

As with Angle 1, the most preferable operator choice for Angle 2 based on local information only is to face mill the angle on a sine table: op 2.b. Thus, if we use only local preferences, we would make a plan in which Angle 1 and Angle 2 are each made in separate set-ups: Angle 1 would be made with op I.a by face milling with the sine table tilted at 25.5°, and Angle 2 would be made with op 2.b by face milling with the sine table tilted at 64.5 degrees.

However, we can make a more efficient plan by taking operator overlap into account. To overlap, the set-ups must be exactly the same: both must use the same fixture and the same part orientation. Note that two pairs of operators overlap; I.a overlaps with 2.a, and I.b overlaps with 2.b. Each of these overlapping pairs forms a group of two features which share a set-up. All the operators in a group can be combined into a new single operator. If operators I.a and 2.a were combined they would form the following new operator:

	feature:	orientation:	fixtures:	tool types:
op3.a	(Angles 1 and 2	side 1 up side 2 right	sine table tilted 25.5°	face mill and side mill)

If operators I.b and 2.b were combined they would form the following new operator:

	feature:	orientation:	fixtures:	tool types:
op3.b	(Angles 1 and 2	side 4 up side 2 right	sine table tilted 64.5°A	face mill side mill)

If we use either of these new combined operators, we have to side mill one of the angles, which is locally sub-optimal. However, the advantage of being able to combine two set-ups into one far outweighs the disadvantage of side milling one of the angles. So a better (i.e. more efficient) plan than the one proposed above, would use op 3.a: face mill Angle 1, and side mill Angle 2 on a sine table tilted at 25.5°. So the local operator preference to face mill Angle 2 turned out to be incorrect.

6 Additional Complexities

In the previous section we presented a basic model of operator overlap. However, we note two problems. First, the picture presented was a much simplified one; there are additional temporal constraints that must be considered when operator overlap is integrated into the planning problem as a whole. Secondly, searching for overlapping operators and the minimum number of set-ups could be a very expensive process. The real problem is how to do the search efficiently.

Addressing the first problem: when looking for operator overlap, in addition to checking that operators can share the same set-up, one must also check that temporal constraints do not interfere with grouping of operators. For instance, going back to our errand example where we had to get stamps and mail letters at the post office, and get cash and ice cream at the grocery store, what if you needed to get money before you buy stamps, and you need stamps to mail the letters, and you have to do everything else before buying the ice cream because it will melt, then it may not be possible to group get cash and get ice cream into one trip to the grocery' store. In other words, in order to overlap, not only must two operators share work, but it must also be possible to execute them in one contiguous sequence. The point is that any temporal restrictions on the problem must be considered before deciding what operators can overlap.

Temporal constraints come from at least two sources: operator preconditions and negative interactions between operators. Having money is a precondition to buying stamps or ice cream, and the fact that the ice cream will melt while you are at the post office is a negative interaction between the buying ice cream and the the time required to execute the other operators. Thus, one needs to look for operator preconditions and negative interactions before doing the overlap analysis shown in the previous section. Finding negative interactions for machining problems is discussed in more detail in [Hayes, 1987a] and [Hayes, 1987b].

Addressing the second problem, "how can the search for overlapping operators be done efficiently?" Let us first explore why searching for operator overlap can be expensive. An obvious but rather inefficient way to look for overlap would be to generate *all* possible operators for making *all* features, then look for *all* pairs of operators that can share set-ups, and then find a minimum set of operator groups. This last step can be quite expensive if there are many features could possibly be placed in many groups. The first step, generate all possible operators, can also be quite expensive; there are often many ways to make one particular feature In fact, new operators can be created during the planning process so the number of ways of making a particular feature are potentially infinite. For example, machinists often design new tools or fixtures in the course of planning, which is essentially creating a new planning operators. Designing new planning operators is a very time consuming process, but it is often key in generating a short plan; the longer a machinist thinks about the problem the more likely it is that he will be able to think up a clever operator that results in a large amount of overlap with other operators. However, it is clearly important to limit the generation of new operators. (A detailed examples of operator design and a more extended discussion can be found in [Hayes, 1989])

Observations of human machinists suggest ways around this problem. First they, do not necessarily take the time to find the plan with the absolute minimal number of set-ups. Secondly, it seems that human machinists rarely generate *all* of the alternative methods (operators) for producing any particular feature. Most of the applicable operators that could be generated are not useful much of the time. However it is difficult to know in advance which operators to generate and which not to. Going back to our angle example from figure 1, typically it is not useful to side mill an angle, so one might think that one should not generate that operator. However, in this particular case, because Angle 1 and Angle 2 were 90° apart, it turned out to be useful.

It seems that what the machinists are doing is considering a set of commonly used alternatives first. Additional more esoteric operators such as side milling an angle on the sine table, are considered only when "cues" in the problem suggest that they might be useful. In this case, the cue was that the angles were 90° apart. Like esoteric operators, it seems that new operators are not designed unless a cue in the problem or a higher level goal indicates that it might be fruitful to do so.

Observations of human planners support this hypothesis. Three protocols were taken of different expert machinists (each subject in this study had more than 18 years of experience) making plans for the part shown in figure 1. Only one subject noticed that Angle 1 and Angle 2 were 90° apart; and he was the only one to produce a plan in which both angles were done in one set-up. His statements imply that combining the steps was a direct result of noticing the 90° cue:

> Subject: "Then because you have me stuck on a vertical machine — are these two faces 90 degrees from each other? ... Sure looks like 90 degrees to me."
>
> Experimenter: "Yeah. Yeah, 'cause I remember that as being 90 degrees."
>
> Subject: "Then I finish the part in one more setting. I'm gonna put it up on a sine plate ... clamp it down."

The other two machinists, who did not notice the 90° angle, only considered making the angles by more standard methods. They each produced a plan in which the angles were made in two separate set-ups. These observations seem to indicate that noticing the cue was crucial in leading the machinist to consider non-standard methods.

In some ways this strategy is similar to one employed by Hammond's TRUCKER, "TRUCKER avoids the hard work of conjunctive goal planning unless execution time cues indicate that such planning will be fruitful."[3] However, the strategies employed here are not used to avoid planning entirely, but to reduce the search that needs to be done.

These observations can be summarized into the following method:

- For each goal (feature) generate a few commonly used, standard operators.
- Use cues in the problem specification to direct search for additional, less standardly used operators, or to indicate when it is useful to design new ones.
- Add temporal restrictions between operators, produced by operator preconditions and negative interactions.
- Compare across goals (features) for positive interactions: look for operators that can be grouped together into set-ups. The temporal constraints found in the previous step will sometimes prevent two operators from being grouped together.
- Choose operator groupings that result in the fewest possible number of set-ups. *(Machinist* uses a greedy algorithm to do this.)

This method is implemented in the *Machinist* program.

7 Discussion

First, it is important to note that the cues used positive interactions are quite different than those used for finding negative interactions described in [Hayes, 1987a]. The cues used in negative interactions identify an interaction, and suggest an operator ordering that will avoid the interaction, while the cues used in positive interactions are used to to limit the number of operators that need to be considered.

Second, the program's performance was compared against 4 humans ranging in experience from 2 years to 5 years. Each of the machinists and the program were presented with the same 3 parts and asked to produce a plan for it. The average plan length for each machinist and the program are shown below in figure 3. There is no average listed for the two second year machinists because they did not manage to make feasible plans for all three parts. Since the plans were of different lengths, a partial average would not have been meaningful.

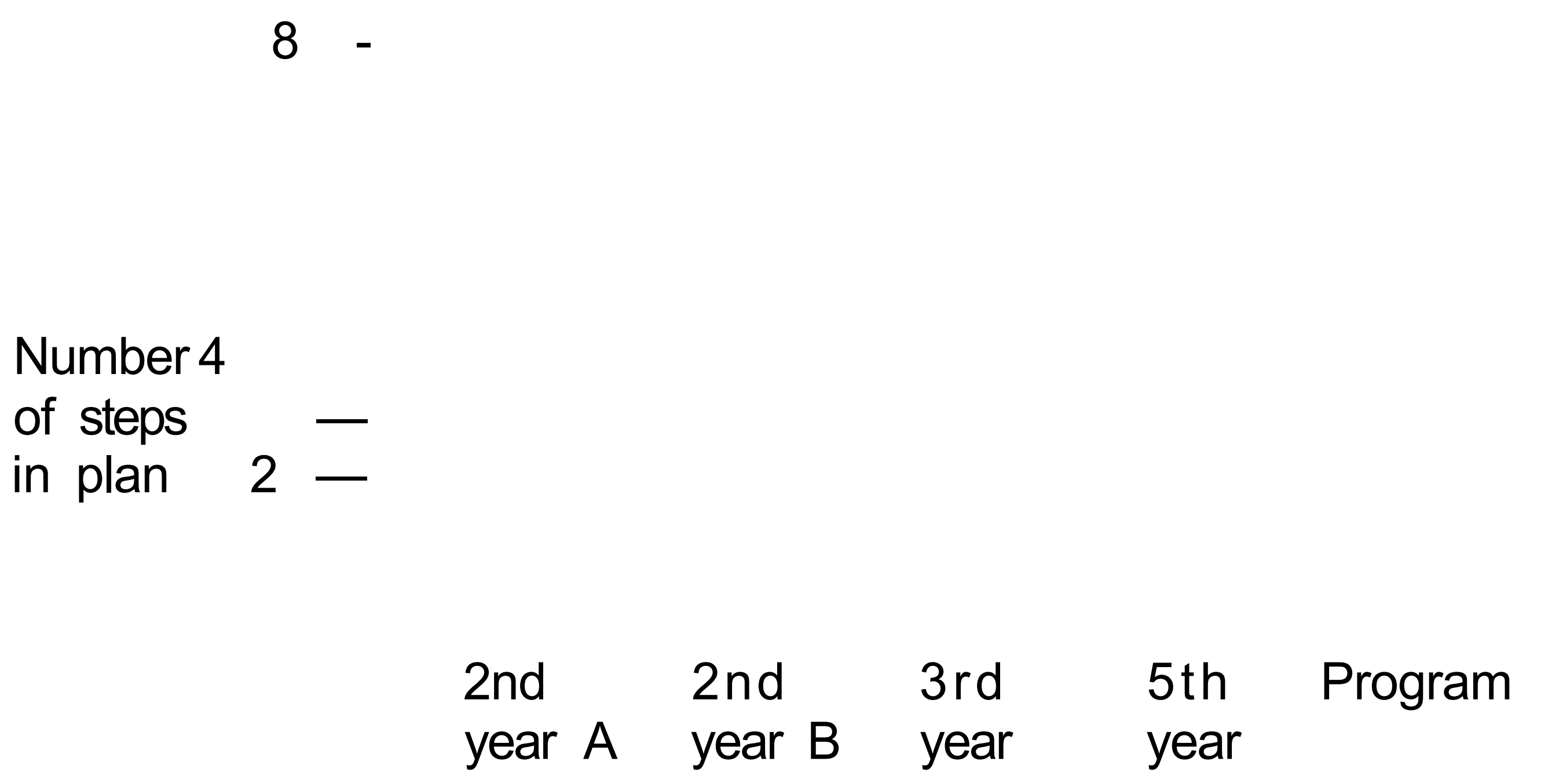

Figure 3: The average plan length produced by four machinists and the *Machinist* program.

There are several interesting things to note about this comparison. It is not shown on the graph but the least experienced machinists (the two second year apprentices), tended to make the shortest plans. However, they were short because they left out important steps, and even when they did manage to make a feasible plan it was usually judged by expert machinists to be of poor quality. Most likely it was good luck rather than good design when their plans did work.

The apprentice of mid-level experience, the third year apprentice, tended to make the longest plans. He was consistently able to make a feasible and reliable plan for all three parts. If the *Machinist* program contained no efficiency

---

[3] Hammond, et al., Proceedings of AAAI-88, vol. 2, p. 537.

strategies, it would make plans that look similar to those of the third year apprentice; He made no attempt to to combine or merge steps. At his stage of development it was difficult enough for him to make correct plans; it was too complex for him to worry about efficiency issues in addition.

Although the program made slightly longer plans than the fifth year machinist (6.33 steps compared to 6.00 steps on the average) the program's plans were judged to be of better quality by expert machinists. A quality comparison of these same plans was reported in [Hayes, 1987a]. In this study, two expert machinists rank ordered the plan form best to worst in a blind test. The program's plans ranked higher than any of the humans' plans on the average. The point is that length is not the only measure of quality in a plan. Efficiency is important, but a certain degree of robustness must be maintained as well.

Since then the *Machinist* program has undergone much development. Another study comparing the program's performance to that of human machinists (i.e. more than 10 years experience) using more complex parts, is currently underway.

8 Conclusions

Operator overlap represents one kind of positive interaction that can be used to improve plan efficiency. The machining domain offers a case study of operator overlap in realistic planing problems. The research reported here describes a particular kind of operator overlap in which the cost of operators are dependent on their order in the plan. Additionally, the overlapping portions of the operators must account for the majority of the cost of each operator. This situation arises in many manufacturing problems and in errand running problems where travel times are large.

The idea is to shorten the plan by grouping together operators that overlap. The main difficulty, however, is to find an efficient method of searching for a near optimal set of such groupings. Human machinists have been observed to use a method that reduces the size of the search problem: only commonly used operators are explored, unless a learned cue in the problem tells them that other methods are likely to lead to cost savings through overlap. For example, since the angles were 90° apart, it indicated that the work could be done in one set-up if side milling on the sine table was used.

Finding operator overlap is complicated by the fact that temporal restrictions, created by negative interactions and operator preconditions, must also be considered before deciding which operators can be overlapped.

Additional problem decompositions can be achieved if the problem contains several types of operator overlap where one type almost always gives you a bigger savings than the other type, such as fixture set-up changes and tool changes. Then the problem can be decomposed into two nearly independent subproblems; first the plan is optimized for the bigger savings (i.e. fixture set-up), second, with in each fixture group formed, the plan is optimized for the lesser saving (i.e. tool changes) with in each group.

The planning strategies described here are implemented in the *Machinist* program. In a comparison with humans it was shown to produce plans that were almost as efficient, and slightly more robust than those of a machinist with 5 years experience.

9 Acknowledgments

A special thanks to Gregg Lebovitz Paul Wright, Dan Barrish, David Boume, Tom Freund, Dan McKeel and Jim Dillinger, for their participation in this project.

References

[Bard, 1989] Bard, J. F. and T. A. Feo. The Cutting Path and Tool Selection Problem in Computer Aided Process Planning. *Journal of Manufacturing Systems,* 8(1): 17-26, January 1989.

[Fikes, 1972] Fikes, R. E., P. E. Hart, and N. J. Nilsson. Learning and Executing Generalized Robot Plans. *Artificial Intelligence,* (3):251-288, 1972.

[Hammond, 1988] Hammond, K. J. Learning from Opportunities: Storing and Re-using Execution-time Optimizations. *AAAI-88,* :536-540, 1988.

[Hayes, 1987a] Hayes, C. C. Using Goal Interactions to Guide Planning. *The Sixth National Conference on Artificial Intelligence,* 1:224-228, July 1987.

[Hayes, 1987b] Hayes, C. C. Masters Thesis: Planning in the Machining Domain: Using Goal Interactions to Guide Search. Technical Report, Robotics, Carnegie Mellon University, April, 1987.

[Hayes, 1989] Hayes, C. C. Designing New Planning Operators During Planning. Technical Report, Carnegie Mellon University, Robotics Institute, 1989. forthcoming.

[Kusiak, 1985] Kusiak, A. Integer Programming Approach to Process Planning. *International Journal of Advanced Manufacturing Technology,* l(l):73-83, September 1985.

[Sacerdoti, 1974] Sacerdoti, E. D. Planning in a Hierarchy of Abstraction Spaces. *Artificial Intelligence,* 5(2):115-135, 1974.

[Scacerdoti, 1975] Sacerdoti, E. D. The Nonlinear Nature of Plans. *IJCAI4,* :206-214, 1975.

[Wesson, 1977] Wesson, R. B. Planning in the World of the Air Traffic Controller. *IJCAJ-77, 'M^Al^,* 1977.

[Wilensky, 1983] Wilensky, R. *Planning and Understanding.* Addison Wesley, 1983, pages 113-126, Chapter 10: Positive Goal Relationships.