Process planning with conditional and conflicting advice

András Márkus (2), József Váncza
Computer and Automation Research Institute, Hungarian Academy of Sciences,
1518 Budapest, P.O.B. 63, Hungary

Abstract

Due to the vast amount and intricacies of technological knowledge, the process planning problem, even for relatively simple workpieces, can hardly be solved by a straightforward algorithm. The paper shows that this problem is more readily tractable as satisfying constraints that represent pieces of advice taken from experts. We define a generic constraint-based model for process planning that handles precedences and setup formation, as well as the mutual effects of operation sequencing and resource assignment. We demonstrate the use of our model with examples of machining prismatic workpieces and bending sheet metal parts.

Keywords: process planning, sequencing, constraints, optimization

1 INTRODUCTION

Due to the vast amount and the intricate relations of technological knowledge the use of computer-aided methods is quite limited in solving process planning problems. Actually, process planning (PP) has two sides: on one hand, human expertise is indispensable in the interpretation of the product model and in weighing the technical and economical factors relevant in the manufacturing environment. On the other hand, the consolidation of the far-reaching consequences of such concerns is a task far more suitable for computers. In order to find the proper division of work, the computer should offer its best facility with the efficient generation and evaluation of a variety of (partial) solutions. The decision tradeoffs in the practical applicability of such systems are as follows:

- If the model does not capture enough details of the problem, feasible solutions may be missed and the offered solution variants may need manual repair.
- If the model is too fine-grained, computing requirements may be prohibitive. A well-chosen, limited number of alternatives is better tractable.
- If the model applies a general purpose heuristic, its success depends on whether the experts find suitable drivers for the heuristics (e.g., estimates for the cost of the unknown part of the solution).

Taking into account these difficulties, we offer a new model for the consolidation of PP advice, specifiable in a variety of forms, generated either by experts or by program modules. The paper shows how large scale *constraint satisfaction in finite domains*, an artificial intelligence approach matured in these years, has been used in solving representative PP problems.

The paper is organized as follows: Sect. 2 offers an overview of related work and discusses key initiatives of the application of constraints. Sect. 3 outlines the problem setting of our investigations; Sect. 4 gives a detailed account of the new approach. Sect. 5 exposes open problems, and discusses why constraint satisfaction is a promising approach of solving PP problems.

2 RELATED WORK

PP had early been recognized as a hard problem that did not yield to problem decomposition without substantial simplifying assumptions. Computer-aided PP became possible when the informal concept of *engineering features* met the methods of conceptual modeling and knowledge representation. Features divide the problem and make it amenable for efficient, automated problem solving [3, 4, 6]. However, *dividing* a problem is not enough for *conquering* it. Features do interact, and the pieces of knowledge, consistent if limited to the local context of a particular feature, may contradict each other in the wider context of the workpiece. In addition, planning has to account for overall optimization objectives as well.

In the production engineering research community, constraint based approaches started more than a decade ago. Research focused on variational product design [5], the modeling of part families [7], and on geometrical reasoning. In the context of PP, interest centered around task precedences in assembly [8], and bending of sheet metal parts, [5, 9], with optimization aspects included [1]. Since these works used special-purpose constraint processing modules added to available CAD/CAPP systems, their experience was mostly limited to specific problems. However, they exposed requirements toward constraint-based methods: the need of

- working with disjunctive and conditional constraints;
- coping with constraint sets that may be conflicting;
- linking constraint satisfaction with optimization.

3 PROBLEM SETTING

Since the aim of our research was the definition and experimental verification of a broadly applicable, constraint-based model for PP problems, we did not focus our attention to a single kind of parts or production processes. Problems addressed here belong to two kinds of PP: the machining of prismatic workpieces, and the bending of sheet metal parts. (Inspection planning could be another application of our model.) The problem setting includes operation sequencing, grouping of operations into setups and the assignment of resources. The planning problem is decomposed as follows:

1. The interpretation of the product model yields a set of features, and for each of them a sequence of feature states (FSs). For bending, the FS sequence is just *flat* and *bent*; for prismatic parts the FS sequences consist of elements such as *raw*, *medium* and *fine*.

Since we use the concepts "feature state" and "machining operation to produce the feature state" interchangeably, we suppose that there is a one-to-one correspondence between FS's and these operations (notwithstanding the initial states of the features). In most cases a FS can be made with different resource assignments (e.g., in different part orientations and by using various tools), that specify, for each operation, the set of available *manufacturing alternatives* (MAs) of that FS.

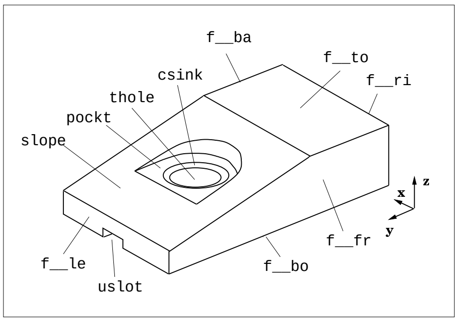

Figure 1: Example of a prismatic workpiece.

2. The exploration of the manufacturing alternatives results in a set of MAs attached to each operation. Depending on the details of the PP model, members of the MA sets may have an internal structure. Typically, the components represent holding and tool; for prismatic parts holding means part orientation and bases, for sheet metal parts it means orientation and gripping. Usually, some combinations of the component values are unfeasible, e.g., assuming a vertical machine tool, in orientations where the u-slot of the part on Figure 1 is accessible from the $+z$ direction of the machine, the slot can be machined by an end-milling tool. However, in orientations where the bottom face ($f\_\_bo$) is parallel with the z axis of the machine, it can be machined by a side-milling tool.

3. Manufacturing constraints (MCs) are the results of technological and economical considerations: they are related to operation precedence and grouping, and point out the available manufacturing alternatives (MAs) and their relations to operation sequencing. Sect. 3.1 gives further details.

Some of the manufacturing constraints are considered *hard*, i.e., the plans must always satisfy them. The other, so called *soft* constraints represent advice with *weights* attached to them: the planner's aim is to obey the soft constraints as far as possible (i.e., to collect a high sum of the weights of fulfilled constraints). While the sum should be *as high as required* in planning with an *aspiration level*, it should be *as high as possible* when both plan *correctness* and *cost* are considered in a multi-objective setting. In addition, soft MCs help us to follow more detailed, perhaps conflicting, technological and economical advice, and to relax the over-constrained formulations. (N.B., the weights are attached to constraint *instances*, not to constraint *types*.) Finally, there are introduced *conditional* constraints: these are hard constraints with relevance limited by specific conditions on the sequencing and resource assignment details of the plan.

4. Plan generation and evaluation makes *acceptable plans*, i.e., sequences of operations, together with the selected manufacturing alternatives, so that all hard constraints are satisfied; considering the role of soft constraints, see the description above. Finally, we determine the cost of executing the process plan with the manufacturing alternatives actually chosen; this opens up the way toward optimization.

Concerning tasks **1**, **2** and **3**, we suppose that the available PP tools (and, as the last resort, user interaction) generate a constraint-based description of the planning problem at hand, *if the underlying, constraint-based PP model is expressive enough.*

3.1 The representation of the PP problem

Since each operation has to be executed once, the plan contains a permutation of the operations; for each operation, a set of the available manufacturing alternatives are given.

Operation	Orientation	Tool
$f\_\_to$	$-x-y+z$	$face\_mill$
$f\_\_fr$	$-x-y+z$	end_mill
$f\_\_ri$	$-x-y+z$	end_mill
$f\_\_bo$	$+y+x-z$	$face\_mill$
$f\_\_ba$	$+y+x-z$	end_mill
$f\_\_le$	$+y+x-z$	end_mill
h_cen	$+y+x-z$	$center\_drill$
h_drl	$+y+x-z$	$twist\_drill$
$uslot$	$+y+x-z$	end_mill
$slope$	$+x+y+z$	$face\_mill$
$pockt$	$+x+y+z$	end_mill
$csink$	$+x+y+z$	$countersink$
h_thr	$+x+y+z$	$thread\_mill$

Figure 2: Plan generated for the prismatic workpiece, optimized for the number of setups. Operations h_cen, h_drl and h_thr belong to the feature $thole$.

In order to balance the expressiveness of the MCs with the efficiency of the constraint satisfaction mechanism, we do not support a free, unlimited variety of constraints: instead, all the MCs have to be expressed by using some predefined forms. Below we list those kinds of MCs that are supported by our model. The central claim of the paper is that they are really sufficient and enough for representing the manufacturing constraints found in the literature or met in our PP praxis:

(A) Precedence type constraints:

- *Operation precedence:* $op1$ has to be made before $op2$.
- *Immediate operation precedence:* $op1$ has to be made *immediately* before $op2$.
- *Operation combination:* $op1$ and $op2$ have to be neighbors, so as to allow their combination.

(B) Grouping (setup) type constraints:

- *Shared manufacturing alternative of two operations:* no matter where the operations $op1$ and $op2$ are in the plan, they must have the same MA attached. (An example: centering and twist drilling of a through hole have to be made from the same direction.)
- *Shared manufacturing alternative within a plan segment:* $op1$ and $op2$ must have the same MA value, and, in addition, all the operations that fall in-between $op1$ and $op2$ must have the same MA value, too. (An example: neither re-fixturing nor tool change are allowed in-between the centering operations of strictly tolerated holes.)

(C) Constraints on the machining alternatives:

- *The list of alternatives allowed as components of the* MA*s for an operation.*
- *The list of alternatives conditioned by other components of the* MA*:* Given a single value of one of the components of the MAs for an operation, the constraint specifies the list of allowed values of the other components. (An example: for each tool allowed at an operation, the constraint specifies the list of feasible part orientations.)

(D) Conditional constraints for precedences and machining alternatives:

- *Operation precedence conditioned by* MA*s:* If the selected MA of $op1$ belongs to the set of alternatives given in the constraint, then $op2$ has to precede $op3$. (An example: if the hole on Figure 1 is drilled from the top face, then this operation has to be executed before making the slope.)
- MA*s conditioned by operation precedence:* If $op1$ precedes $op2$, then the MA of $op3$ has to belong to the set of alternatives given in the constraint.

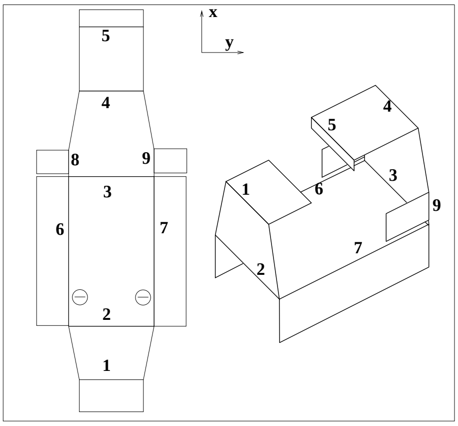

Figure 3: Example of a sheet metal part.

- *Operation precedence conditioned by operation precedence:* If $op1$ precedes $op2$, then $op3$ has to precede $op4$. (An example: for the sheet metal part, if $b3$ precedes $b7$ then $b7$ must precede $b9$.)

Constraints of type (A) and (B) exist both in the hard and soft versions. The other types of constraints are all hard since soft constraints of these types would call for reasoning over fuzzy sets that are incompatible with the mechanisms of the constraint engine.

3.2 The cost elements of the plans

Since we are interested in the *relative* costs of the plans, costs are attached to those details that may be varied: additive, independent cost elements have been attached to the MA selected and to the change of MAs in-between consecutive operations. Some of the cost elements could be estimated for partially defined plans as well: e.g., as soon as we know that two consecutive operations have no MAs shared, a minimal setup change cost may surely be added to the total cost.

3.3 Do usual planning methods work for this model?

Before presenting the details of our constraint-based PP method, first we list the factors that impede the application of more usual planning methods with our model:

Mathematical programming formulations work well for generating sequences with precedence constraints. Conflicting precedences can be resolved in a preprocessing phase [2]. Further elements of our model would explode the size of an LP formulation of the PP problem. Anyway, the 0/1 nature of these models calls for branch-and-bound techniques and domain specific, combinatorial heuristics: these could be nicely incorporated into the constraint-based approach.

Methods with incomplete search (simulated annealing, genetic algorithms, taboo search) offer more flexibility, but since there are few general principles of their use, each application should start almost from scratch: they offer very few as for a *problem solving engine*.

General purpose, symbolic planning methods of artificial intelligence offer a rich conceptual background. However, the conditional structures and contradictions of our model would be difficult to handle with these methods. Efficient generation of plans, especially if cost concerns are relevant, calls either for specialized methods (such as hierarchical task decomposition, case-based planning with domain specific similarity and modification concepts), or for the translation of the problem case into a general combinatorial problem.

Operation	Orientation	Tool
$b8$	$up + y$	$t4$
$b9$	$up - y$	$t4$
$b1$	$up + x$	$t4$
$b5$	$up + x$	$t4$
$b6$	$down + y$	$t4$
$b7$	$down - y$	$t4$
$b4$	$up + x$	$t3$
$b3$	$up + x$	$t3$
$b2$	$up - x$	$t3$

Figure 4: Plan generated for the sheet metal part, optimized for tool change.

4 PP WITH CONSTRAINT SATISFACTION

Since our PP model and the plan generation programs have been built with using the concepts and facilities of an advanced constraint programming environment, the Mozart system. It is available for research purposes from `http://www.mozart-oz.org`. Most of its facilities can be translated into other constraint systems, but we have carried out no investigations in this matter.

In a nutshell, the domain of each constrainable variable is either a finite set of nonnegative integers or a finite set with no specific structure. Main differences between integer and set variables are in their constraint handling methods. For programming convenience, usual data structures (such as records or lists) may be built of the constrained variables. Some types of the constraints can be checked efficiently (e.g., whether the value of an integer variable belongs to an interval, or whether the cardinality of a set variable is less than a number), and their implications are easy to find. These, so called *basic constraints* are stored in the *constraint store*. Further, so called *non-basic constraints* can not be checked in an efficient way (e.g., (in)equalities over the sum of several integer variables). The mutual satisfiability of these constraints is handled by *propagators* that automatically trigger each other while running in quasiparallel threads. Since the propagation machinery is incomplete (not *all* the implications of the constraints are generated) another kind of reasoning is applied: this, so-called *distribution* introduces a new, artificial constraint C and divides the original problem into two alternatives:

$$OldConstraints \cup \{C\}$$

and

$$OldConstraints \cup \{\neg C\}$$

If the constraint store becomes inconsistent in one of the alternatives, work continues with the other alternative, and (in the last resort) the system backtracks. Since finding a useful C is a crucial question, a number of built-in strategies are available, such as “divide at the midpoint of a given constraint domain” or “pick up the lowest value from the smallest domain”; further distributors can be programmed, too.

The search may run for a *single solution* (or for proving that there is no solution) and for *all solutions*. Satisfiability and optimization can be coupled by *branch-and-bound*: here the cost of the best solution found imposes a new constraint on the cost of further solutions.

Below, in the discussion of the representation of our PP constraint types, all the technical details are omitted and no more than the basic ideas are presented:

Operation sequencing by integer constraints: for each operation, its position in the plan is represented by an integer variable whose range is initiated from 1 to the total number of operations. Precedences are represented as inequalities between these variables. These variables must have pairwise distinct values; such a built-in constraint is provided by the system.

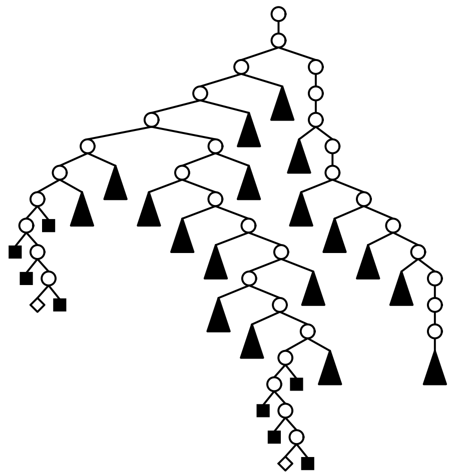

Figure 5: Distribution tree of the plan generation for the prismatic example part: white diamonds are solutions, black squares are subspaces with no solution, black triangles are hidden subtrees with no solution. The tree contains more than 10000 nodes.

Machining alternatives by set constraints: For this purpose, the finite set constraint mechanism is the suitable representation tool: the alternatives give an upper bound of the set, and the PP constraints are expressed as set inclusion relations and limits on to the cardinality of the sets.

Building soft and conditional constraints: The so-called *reified constraint*s of Mozart reflect the validity of a constraint in a new, 0/1-valued integer variable that could be further used within another constraints. This mechanism is well suited to calculate the total weight of the satisfied soft constraints as well.

Constraints given as data: All the constraint programs that represent the constraint types described in Sect. 3.1 are *automatically generated* from workpiece data. Accordingly, the PP engineers do not need to know anything about the underlying constraint handling modules that we have developed on the basis of the Mozart system: the engineers need to be familiar with a simple data format only.

5 CONCLUSIONS AND FURTHER WORK

The aim of this research was to show that the handling of PP problems can be greatly enhanced by a powerful, general-purpose constraint satisfaction mechanism, working on constraints built of *well-defined constraint types of a limited variety*. Main features of the new approach are as follows:

- Tight coupling of operation sequencing and resource assignment, with a variety of conditional formulations included.
- A new approach of dealing with over-constrained problems: we provide the ability to follow as much (conflicting) technological and economical advice as possible.
- Smooth shift of attention between feasibility and optimization of plans.
- Clear-cut separation of the description of the actual PP problem from a dedicated problem solving module, that, in turn, is driven by a general-purpose, powerful constraint engine.

There are several, interrelated points worth further investigations:

- What other constraint types and what kinds of refinements of the present types are needed for further types of PP problems?
- How can the efficiency of the plan generation process be improved with dedicated constraint processing algorithms (e.g., with better lookahead estimations, or with specific distribution strategies)?
- Can this constraint-based approach be extended to other fields related to the PP problem, e.g., to the design of modular fixtures or tool-kits?

All in all, we do not claim that constraint-satisfaction offers a panacea of manufacturing engineering problems. However, this method should be considered as a new *integrating tool* for problems that are hard to established techniques. In the future, constraint engines should be used to *consolidate data and knowledge coming from a variety of CAD/CAM programs*. Finally, our proposed method integrates the variant and generative approaches to PP: while the *variant-based* methods of the engineers' thinking should be distilled into particular types of constraints, the *generative* side of the engineers' activities would get powerful support from professional level constraint solvers.

ACKNOWLEDGMENTS

This work has been supported by the *Methodology for Emergent Synthesis* Project (96P00702) in the *Research for the Future Program of JSPS* while A. Márkus was visiting researcher at the Ueda Laboratory of Kobe University. Thanks are due to the Programming Systems Lab, Universität des Saarlandes, Saarbrücken, Germany for making the Mozart system available.

REFERENCES

[1] Duflou, J., Kruth J.-P., Van Oudheusden D.: Algorithms for the Design Verification and Automatic Process Planning for Bent Sheet Metal Parts. Annals of the CIRP, **48**(1), 405–408, (1999)

[2] Horváth, M., Márkus, A., Váncza, J.: Process Planning with Genetic Algorithms on Results of Knowledge-Based Reasoning. Int. J. of Computer Integrated Manufacturing, **9**, 145–166, (1996)

[3] Houten, F.J.A.M. van, Erve, A.H. van't, Kals, H.J.J.: PART, a Feature-Based CAPP System. Proc. of the 21st CIRP Int. Sem. on Manufacturing Systems, Enschede, 25–29, (1990)

[4] Inasaki, I., Tönshoff, H.K., Zahn, G.: Integration of CAD, CAPP and Fixturing Planning by a Knowledge-Based Data Model. Proc. of the CIRP Int. Sem. on Intelligent Computation in Manufacturing Engineering, Como, 317–323, (1998)

[5] Kimura, F., Suzuki, H., Ando, H., Kinosada, A.: Variational Geometry Based on Logical Constraints and its Applications to Product Modeling. Annals of the CIRP, **36**(1), 65–68, (1987)

[6] Lutters, D., Wijnker, T.C., Kals, H.J.J.: Information Management in Process Planning. Annals of the CIRP, **48**(1), 385–388, (1999)

[7] Mäntylä, M., Sohlenius, G.: Representation of Process Planning Knowledge for Part Families. Annals of the CIRP, **42**(1), 561–564, (1993)

[8] Nof, S.Y., Rajan V.N.: Automatic Generation of Assembly Constraints and Cooperation Task Planning. Annals of the CIRP, **42**(1), 13–16, (1993)

[9] Shpitalni, M., Saddan D.: Automatic Determination of Bending Sequence in Sheet Metal Products. Annals of the CIRP, **43**(1), 23–26, (1994)