

Constraint-based process planning in sheet metal bending

András Márkus¹ (2), József Váncza¹ and András Kovács²

¹ Computer and Automation Research Institute, Hungarian Academy of Sciences

² Faculty of Electrical Engineering and Informatics,
Budapest University of Technology and Economics

Abstract

The majority of research dealing with computer-aided process planning of sheet metal bending approaches this problem as heuristic search. Since relevant engineering knowledge mostly consists of declarations that prohibit collisions and just a handful of generative rules, there are no useful means to drive these heuristics. In order to find a new way, we have made experiments with a constraint-based approach: using predefined constraint types and geometric constraint satisfaction, complex bending problems have been solved. By returning not just a single solution but a Pareto-optimal set of solutions (i.e., operation sequences, with appropriate part orientations and tools assigned) we have left the engineers freedom to apply further, not yet modeled parts of their domain knowledge.

Keywords: process planning, constraints, bending

1 INTRODUCTION

Computer aided methods in engineering call for the *efficient use of deficient knowledge* [1]: while the engineer must not try to conceal if his knowledge is not formal enough to be filled into some predefined framework (such as standard optimization methods), the computer expert must not renounce the innovation of problem solving frameworks. One of the promising new research directions in CAX is the integration of optimization and constraint-based techniques for solving complex, loosely defined design, planning and scheduling problems [2].

Earlier we have developed a generic constraint-based model and planning engine for manufacturing process planning [3, 4]. By exploiting the expressive power of constraint programming (CP), the relevant, sometimes conflicting pieces of domain knowledge were represented. The planner applied standard satisfaction techniques and customized search to find cost-optimal solutions in the presence of hard, soft and conditional constraints. The present paper describes our next steps taken from this basis:

- refined modeling in the bending domain, with a new integration of reasoning over geometry and technology,
- refinement of the proposed set of constraint types,
- development of a new solution strategy for balancing multiple evaluation criteria in a user-friendly way,
- evaluation of a mainstream optimization and constraint programming engine.

The paper is organized as follows: Sect. 2 is an overview of related work with constraints in process planning. Sections 3 and 4 outline the problem setting and our planning model for sheet metal bending. Sect. 5 gives details of the geometry module. Sect. 6 outlines branch-and-bound search for Pareto sets. Sect. 7 discusses experiments with the new constraint engine.

2 RELATED WORK

In the production engineering community, constraint based research started with works focused on geometric constraints for assembly and variational product design [5, 6], on the modeling of part families [7] and geometric reasoning in parametric design [8]. Constraint reasoning was applied in [9] to the problem of designing universal sheet metal bending tools for part families.

In the field of process planning, more specifically in operation sequence planning, interest centered around task precedences in assembly [10], and bending of sheet metal parts [11, 12]. Special emphasis was put on dealing with tolerances of bent sheet metals in [13]. Work reported in [3] introduced constraint types that were shown to cover the needs of a wide variety of CAPP problems.

These works explored that (1) the constraint-based methods should be ready to deal with a variety of logical structures and cope with over-constrained problems, and (2) optimization and constraint satisfaction should be seamlessly integrated.

3 PROBLEM SETTING

3.1 Sheet metal bending

Sheet-metal parts are typically produced by a sequence of bending operations. The bending process starts with a flat part and ends up with a three-dimensional object of interconnected planes (see Fig. 1). The bending operations are executed on a bending machine (press brake), using various tool and holding resources. Tools consist of dies and punches of different shape and length. There is also a need of grippers that hold the part during and in-between the operations.

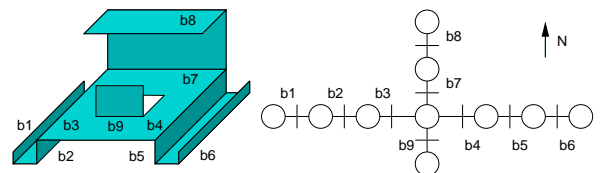


Figure 1: A sample part and its connectivity graph.

The bending operations should be sequenced so as to avoid part-tool, part-machine and part-part collisions (see Fig. 2). Although bending operations are local, they often make global changes in the geometry of the part. Hence, all of their effects can hardly be specified in advance. Process engineers developed various rules of thumb to support the generation of executable sequences [11, 14, 15]. For instance, outside bends should be done first to avoid “rolling

up” the part that would prohibit tool access to outside bend lines later. Tall flanges most likely interfere with the bending machine, hence their bends should be postponed as far as possible. It is suggested to make internal bends as early as possible, whereas bends determining the shape of the part should be left to the end of the plan. However, almost each rule has its exception. For instance, on our sample part bends b_1 , b_2 and b_3 form together a so-called channel that should be made from inside toward outside. There are of course some hard rules as well: to compensate for the spring back of the sheet, overbends have to be made. In a corner, to avoid part-part interference, the outside bend must be done strictly before the inside bend. Hence, domain knowledge in bending is unanimously captured by soft knowledge representation methods: by fuzzy rules, preference rules, and/or weighted constraints.

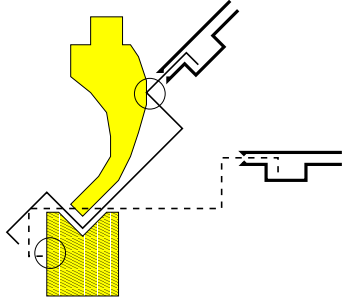


Figure 2: A bending operation on the sample part. Potential places of collisions are encircled.

When selecting (or designing) tools, geometric dimensions, tolerances and bending forces must be taken into consideration. However, the bending operations executed so far determine the intermediate shape of the part and constrain, indirectly, the applicable resource sets of the forthcoming operations. E.g., if bends b_3 and b_4 are performed before b_7 , then b_7 can be made by a tool of exact length only.

The same tool can often make several bends, some of them may be even shorter than the length of the tool. Holdings are also usually appropriate for performing subsequent bending operations. Certain operations may be even merged by using one tool to perform them at the same time. All this gives a chance for optimization. Important optimization criteria are – as in all process planning domains – the minimal tool and holding changeovers. In sheet metal bending a further, specific criteria is that the number of unbalanced operations (when the center of gravity of the part does not fall between the tool and the holding device) should be as small as possible [12]. Note that this criterion is in conflict with that of having the minimal repositioning of the part.

Beyond optimization criteria, realistic CAPP models should provide means to cover and utilize *best practice* by expressing characteristics that the experts anticipate in “good” plans. Such plans are not only feasible but close to optimal – although there might be no proof that good plans must really have these features.

3.2 Our approach to the planning problem

Of course, the basic criterion for the adequacy of a plan is that it must be executable; i.e., it should use the appropriate resources and should be collision-free. In our previous constraint-based experiments in bending, instead of making a complete representation of the geometry of the part and the tools, we have prepared a rich set of constraints over the sequencing and the resource assignments of the operations. However, these constraints did not contain the actual geometric data, just the *results* of such geometrical

reasoning, inter-mixed with some rules of thumb presented above. Obviously, such a representation is incomplete, but, in other cases, could easily turn into an over-constrained plan specification. So it happened that we did not succeed in filtering out all illegal plans, or the constraints turned out contradictory even if the problem was well solvable by a human expert. As a matter of fact, the need of using an intricate, carefully tuned system of soft and hard constraints was due to the above mentioned difficulties.

As the next step in our constraint-based bending project, now we have developed a model with much closer integration of geometry and process planning. We extracted crisp knowledge related to the geometry of bent sheet metals from the engineering expertise. With this decision we have lost some of the generality of our earlier model, but, on the other hand, this extension offered a test whether those constraint types are relevant in this setting as well. In a similar way, such a development was a new test against our solution strategies, too.

All in all, our statement of the process planning problem in the bending domain is as follows:

Given are (1) the geometric model of the sheet metal part and the applicable tools, (2) the various optimization criteria, and (3) domain knowledge on some global properties of good plans.

Find a set of solutions that are executable, and optimal in the Pareto sense.

4 CONSTRAINT-BASED PROCESS PLANNING

4.1 Part and tool representation

The frame of the part is described by a connectivity graph, a graph with *pane* and *bend* nodes, where each pane node is connected to nodes corresponding to the adjacent bends. Rectangular plates, each fixed to the corresponding pane node, build up the solid structure of the part. E.g., see Fig. 1, where panes and bends are shown as circles and lines, respectively.

Pairs of dies and punches are referred to as tools. They are characterized by their length, as well as front and back profiles. We do not deal explicitly with grippers and specify each holding with the orientation of the part.

4.2. Operations, resources and plans

There is a set of irreversible operations, one associated to each bend. By executing an operation, the state of the corresponding bend changes from undone to ready. The fabrication process of the workpiece can be modeled as a permutation of these operations, with resources assigned. Tools and orientation the part should be selected from pre-defined finite sets.

4.3 Constraints and criteria

Our CAPP model represents domain knowledge both by hard and soft constraints. Hard constraints describe characteristics that are required in order to achieve an executable plan, hence, they must be satisfied by the solution. Soft constraints consist of base predicates describing certain pieces of advice, and attached weights that express the importance of the that advice. The constraints belong to the following types:

- operation precedence and neighborhood;
- resource assignment;
- resource sharing (setup formation);
- conditional constraints, where one of the above properties is conditioned by operation precedence and/or resource assignment.

The constraints express general expertise in bending, as well as the geometry of the actual sheet metal part.

Preferences defined over executable plans are incorporated in the optimization criteria. The following three measures of plan quality were applied:

- number of tool changes,
- number of grasping changes, and
- number of unbalanced bending operations.

5 GEOMETRIC REASONING

In our approach, geometric information is processed in two ways:

1. Working with data in the part model, a geometrical pre-processor generates constraints by recognizing some common technological characteristics of the part, such as the so-called channel features, or the parallel, adjacent bends that are likely to be processable immediately after each other.
2. Since the above set of constraints is far from complete, the executability of bending operations has to be strictly checked.

Computational aspects require that failures should be recognized as soon as possible, hence, the complete collision detection should be executed each time after one more operation has been specified. In order to facilitate geometrical inference, the exact knowledge of the geometry is required. Accordingly, the sequence of the decisions has to follow the temporal sequence of the operations, either from the beginning to the end (bend direction) or vice versa (unbend direction). Since bending experience tells that the most troublesome operations are those at the end of the plan, the plans are generated in the unbend direction.

Collision detection is performed as follows: given the selected tool and grasping, the coordinates of the currently bent edge are calculated. This information is propagated along the arcs of the connectivity graph. Finally, collision check is performed between each rectangular solid and the tool, and each pair of solids belonging to the two opposite wings of the workpiece.

6 PLANNING WITH PARETO OPTIMIZATION

In process planning – as it is common in engineering problems – there is no single, most relevant evaluation criterion of the solutions, but there are several, competing aspects of their quality. Although the weighed sum of such qualities could be seen as a single indicator of the goodness of the plans, there are a number of arbitrary factors in such an approach. The other way is to accept that, indeed, there are several relevant but incomparable dimensions of the evaluation, and look for the Pareto set of the most preferable solutions. The Pareto set (P) is defined as a subset of all solutions (S) such that all the solutions in $S - P$ are dominated by some element of the set P , and none of the solutions in the set P is dominated by any other element of P . A solution s_1 is said to dominate another s_2 , if s_1 is at least as good as s_2 , from all points of view, i.e., in all dimensions, and it is strictly better at least in one dimension.

Branch-and-bound is a widely used search method that does not explore those parts of the search tree where the best solution can not be better than the best solution found so far. We extended this into a Pareto framework as follows:

1. Initialize search with the empty Pareto set.
2. Continue exploring the search space, until a new solution is found, so that the new solution is not dominated by any member of the current Pareto set.
3. If no such solution has been found, then return the Pareto set and quit search.

4. Insert the new solution into the Pareto set, and remove those members of the Pareto set that became dominated by the new solution. Go to 2.

This way, the search tree is pruned by the set of the previously found solutions. This strategy is sound and complete – ensures to find all the Pareto-optimal solutions – but in case of many or contradictory aspects, the search can be very time-consuming: the hope of a solution outstanding in (at least) one aspect can often be given up only near to the leaves of the search tree. Hence, it may result in a close-to-exhaustive tour over the valid solutions.

In order to prune the search space more efficiently, further, so called *redundant dimensions* are introduced. These dimensions incorporate best practice, such as “*try to execute parallel bends of the same lengths immediately after each other, using the same resources*”. These pieces of advice are described by soft neighborhood and resource sharing constraints. Our experiments have shown that these concerns – technically speaking, soft constraints used in cutting mode – with appropriately set threshold values are highly efficient in reducing the search space and lead to good solutions in a reasonable time. However, the setting of threshold values is quite critical: too loose bounds result in needlessly high search time, while too tight bounds risk losing some – potentially all – good solutions.

Hence, a two-phase search has been realized:

- In the first phase, striving for an early result, the threshold values are set tight. This usually leads to some nice-looking solutions – usually, but not necessarily members of the final Pareto set. If no solution is found, the bounds are relaxed. Solutions found in the first phase will make an initial guess of the Pareto set.
- In the second phase, search is started with this initial guess of the solutions. The search space is constrained only by the solutions found so far, and a complete Pareto branch-and-bound search is run.

7 IMPLEMENTATION AND EXPERIMENTS

The planner system has been implemented in the Optimization Programming Language (OPL) – a modeling language designed for combinatorial optimization and constraint programming [16] – with using a professional, general-purpose constraint solver.

Our system was tested with several parts of different complexity. For the sample part presented on Fig. 1, with 4 tools of different length and profile available, the following results were obtained:

The pre-processor generated 20 different constraints on operation precedence and resource assignment. These constraints gave a good but incomplete representation of the part geometry: although the plans generated with using these constraints only – soft constraints and collision detection switched off – showed similarity to the correct solutions, the constraints were not strong enough to exclude all collisions.

In the first run of the sound Pareto search (with collision checking switched on), tight bounds in the redundant dimensions were applied. Live up to our expectations, two members of the Pareto set were found in 25 seconds. The second run provided further two Pareto-optimal solutions that were sorted out in the first run because of their weakness in the redundant dimensions. However, the complete search took over 6 minutes to conclude. All the four members of the Pareto set were essentially different: while the first three were outstanding in different single dimensions, the fourth plan found a balance between the contradictory aspects. Table 1 compares the different solutions while Table 2 presents one of them.

Criteria	Plan1	Plan2	Plan3	Plan4
tool changes	1	1	2	2
holding changes	2	3	1	2
unbalanced bendings	3	0	4	1

Table 1: Members of the Pareto set.

Operation	Tool	Holding
b9	tl10	N
b5	tl50	W
b4	tl50	W
b6	tl50	W
b2	tl50	E
b3	tl50	E
b1	tl50	E
b8	tl50	S
b7	tl50	S

Table 2: Pareto optimal solution with one tool change, three holding changes and no unbalanced bendings.

8 CONCLUSIONS

We reported on the recent developments of our constraint-based CAPP model and planning engine. For a comparison of the past and present work, see Table 3 below.

	Past	Present
Domain	generic	bending
Geometric reasoning	separate step	integrated
Constraints made	before planning	before and during planning
Hard constraints	yes	yes
Soft constraints	many	few
Evaluation	single	multiple criteria
Search strategy	B&B	B&B with Pareto
Implementation	Mozart/Oz	OPL Studio

Table 3: Comparison of our CAPP systems.

CAPP, especially in the bending domain, have been considered for a long time an ill-structured problem whose solution required the utilization of deficient domain knowledge and the application of approximative and/or heuristic search methods. Our experiences with a constraint-based planning system suggest, however, that there is no strict borderline between hard and soft domain expertise. In bending, due to the integration of geometric reasoning into a constraint-based planner, almost the whole body of the inherently inconsistent pieces of domain knowledge could be bypassed. Computing resources relieved this way were utilized in solving complex multi-criteria optimization problems. However, there is still a need to express and use best practice of experts. In our model, soft constraints are just for this purpose; they make the search for solution sets more efficient.

ACKNOWLEDGMENTS

This work has been supported by the grants OM 02407/2000 and NRDP 2/040/2001.

REFERENCES

- [1] Hatvany, J., Lettner, F.: The efficient use of deficient knowledge. *Annals of the CIRP*, **32**(1), 423–426, (1983)
- [2] Wallace, M.G.: Practical Applications of Constraint Programming. *Constraints*, **1**(1), 1996.
- [3] Márkus, A., Váncza, J.: Process Planning with Conditional and Conflicting Advice. *Annals of the CIRP*, **50**(1), 327–330, (2001)
- [4] Váncza, J., Márkus, A.: A Constraint Engine for Manufacturing Process Planning. In: Walsh, T. (ed.), *Principles and Practice of Constraint Programming – CP2001*, Springer, 745–759, 2001.
- [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] Sata, T., Kimura F., Suzuki, H., Fujita T.: Designing Machine Assembly Structure Using Geometric Constraints in Product Modelling. *Annals of the CIRP*, **34**(1), 169–172, (1985)
- [7] Mäntylä, M., Lagus, K., Laakko, T.: Application of Constraint Propagation in Part Family Modeling. *Annals of the CIRP*, **43**(1), 129–132, (1994)
- [8] Shpitalni, M., Lipson H.: Automatic Reasoning for Design under Geometrical Constraints. *Annals of the CIRP*, **46**(1), 85–88, (1997)
- [9] Alva, U., Gupta, S.K.: Automated Design of Sheet Metal Punches for Bending Multiple Parts in a Single Setup. *Robotics and Computer Integrated Manufacturing*, **17**, 33–47, (2001)
- [10] Nof, S.Y., Rajan V.N.: Automatic Generation of Assembly Constraints and Cooperation Task Planning. *Annals of the CIRP*, **42**(1), 13–16, (1993)
- [11] 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)
- [12] Shpitalni, M., Saddan D.: Automatic Determination of Bending Sequence in Sheet Metal Products. *Annals of the CIRP*, **43**(1), 23–26, (1994)
- [13] de Vin, L.J., Streppel, A.H., Kals, H.J.J.: The Accuracy Aspect in Set-up Determination for Sheet Bending. *Int. Journal of Advanced Manufacturing Technology*, **11**, 179–185, (1996).
- [14] Gupta, S.K, Bourne, D.A., Kim, K.H., Krishnan, S.S.: Automated Process Planning for Sheet Metal Bending Operations. *Journal of Manufacturing Systems*, **17**(5), 338–360, (1998)
- [15] Wang, Ch.-H., Bourne, D.A.: Design and Manufacturing of Sheet-Metal Parts: Using Features to Aid Process Planning and Resolve Manufacturability Problems. *Robotics and Computer Integrated Manufacturing*, **13**(3), 281–294, (1997)
- [16] Van Hentenryck, P.: *The OPL Optimization Programming Language*. The MIT Press, 1999.