Sheet Metal Bending: Generating Shared Setups

Satyandra K. Gupta
Mechanical Engineering Department and Institute for Systems Research
University of Maryland
College Park, MD 20742
skgupta@eng.umd.edu

David Alan Bourne
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
db@ri.cmu.edu

Abstract

Contemporary process planners for sheet metal bending solve the process planning problem for individual parts. Quite often, many different parts can be produced on shared setups. However, plans generated by current process planning systems fail to exploit this commonality between setups and try to generate optimal setups for individual parts. In this paper, we present an algorithm for multi-part setup planning for sheet metal bending. This algorithm takes a set of parts and operation sequences for these parts, and tries to find a shared setup plan that can work for every part in the set. Setup changes constitute a major portion of the production time in batch production environments. Therefore, multi-part setup planning techniques can be used to significantly cut down the total number of setups and increase the overall through-put.

1 Introduction

Increasing emphasis on more personalized products and shrinking product lives is resulting in major changes in manufacturing practices. As we move towards mass customization, we need ways to handle a wider variety of products on the shop floor. A number of new technologies such as flexible manufacturing systems, modular fixturing, etc. are being deployed to handle the increase in product variations. These new technologies focus on achieving process and material handling flexibility. So far, very little attention has been paid in process planning systems to exploit commonality in tooling and fixturing across multiple parts. Contemporary process planning systems handle one part at a time, attempting to find the best plan for every part [2, 4, 12]. Such planners fail to identify commonality among parts and cannot select common tooling and shared fixtures that work for multiple parts. This limitation results in more frequent setup changes and reduced overall throughput time.

In this paper, we describe a two-step approach to setup planning of sheet metal bending operations for all the parts in a part family. The first step is to identify the setup constraints imposed by various bending operations. These setup constraints describe spatial constraints on the sizes and locations of various tooling stages in the setup. The second step is to generate setup plans which satisfy all setup constraints. Any setup plan that can satisfy all setup constraints is capable of accommodating every part in the part family. We use constraint propagation techniques combined with sheet metal bending domain heuristics to identify compatible setup constraints and create setup plans. Our algorithm takes a set of parts and operation sequences for these parts, and tries to find a shared setup plan that can work for every part in the set.

2 Background

2.1 Definitions and Preliminaries

Sheet Metal Bending: In sheet metal bending, a flat part is bent using a set of punches and dies. Figure 1 shows three example parts and corresponding starting flats. For a detailed description of sheet-metal bending processes, readers are referred to [1, 8, 13].

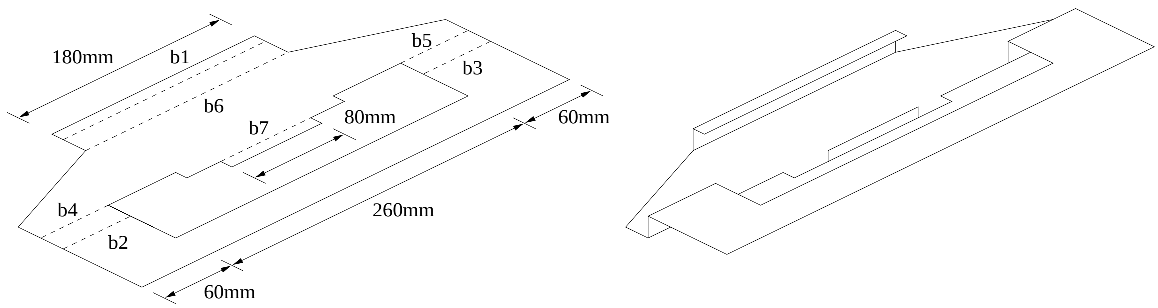
Operation Sequences: An operation sequence is an *ordered set* of bending operations. For example, $[(b7)(b2, b3)(b4, b5)(b1)(b6)]$ is an operation sequence for the part shown in Figure 1(a). Please note that a bending operation can include more than one bend-line. Whenever, a bending operation includes more than one bend line, it implies that all bend lines in that operation will be processed simultaneously.

Tooling Stages: A tooling stage consists of a pair: a punch and a die. A tooling stage is specified by its length and location on the press-brake. Intermediate workpiece shapes impose restrictions on the tooling stage length that can be used to perform a bending operation. We refer to constraints on the shape and sizes of tooling as setup constraints (see Section 5 for details).

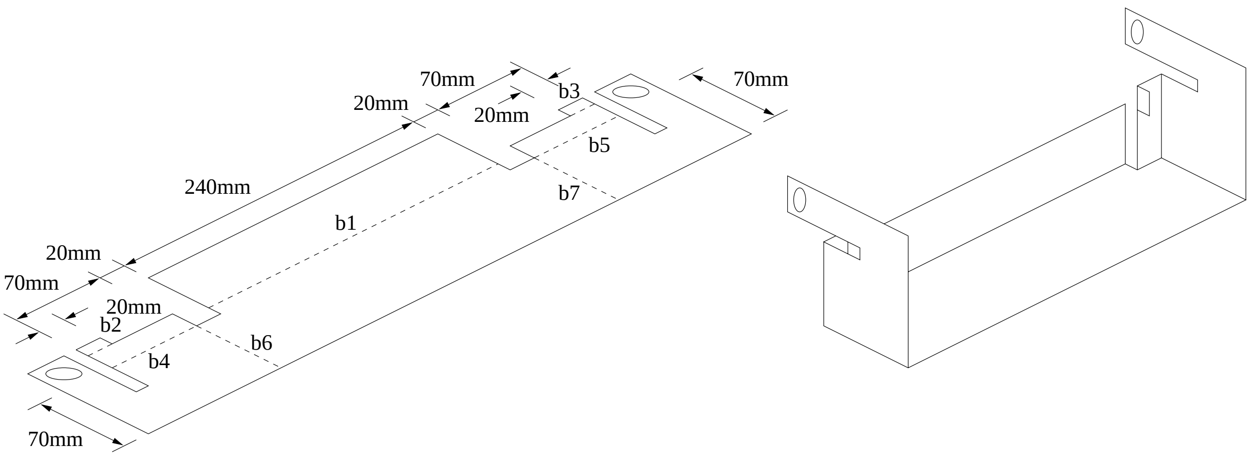
Colinear Bends: Colinear bends are bending operations which have colinear bend-lines separated by gaps. There are two types of colinear bends: interrupted colinear bends and uninterrupted colinear bends. Interrupted colinear bends cannot be done on a single tooling stage. Interruptions among bends require gaps among tooling stages. The *interruption index* of a colinear bend refers to the number of required gaps in tooling stages. For uninterrupted colinear bends, the interruption index is 0.

Press Brake Setups: A press brake setup describes the arrangements of various tooling stages on the bending press-brake. A press-brake setup can consist of one or more tooling stages.

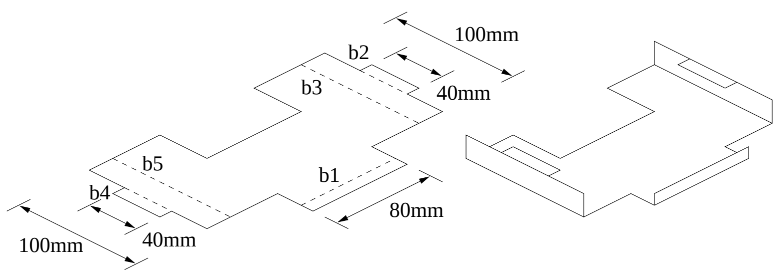
Setup Plans: A setup plan describes the press-brake setup and the assignment of various bend lines to tooling stages in the setup. Each assignment is a triple of the following form: *(bend, stage, position)*. Where, *bend* is a bend line, *stage* is a tooling stage, *position* is the relative position of the bend line with respect to the left edge of the tooling stage.



(a): Part 1



(b): Part 2



(c): Part 3

Figure 1: Example Parts (dashed lines denote bend-lines)

2.2 Operation Planning for Sheet Metal Parts

In our previous research, we have implemented a large process planning system to perform a wide variety of process functions for robotic sheet metal bending press-brakes. This system is based on generative process planning and currently handles one part at a time. This system consists of a number of specialized planners such as grasping, tooling, and moving, which solve the problems in their respective domains. . Technical descriptions of various components of this system can be found in [7].

A number of systems have been developed to automate various process planning functions for a broad variety of sheet metal operations [3, 9, 11, 14]. These systems attempt to handle a wide variety of sheet metal processes (such as stamping, blanking, punching, sheering, bending) and attempt to sequence various operations based on high level interactions among them.

We are aware of two main research efforts that attempt to address sheet metal bending operation planning in detail. De Vin *et al* [4, 5] have developed a process planning system for finding a feasible operation sequence. Their system addresses the part-tool collision and tolerance constraints. In addition, they use heuristics for minimizing material handling time to guide the search. Radin *et al* [10] have developed a variation of branch and bound search technique to find the optimal operation sequence. They first try to find a feasible solution and then try to improve it in subsequent iterations. Their cost criteria is based on the number of tool changes and material handling time.

3 Problem Formulation for Multi-Part Setup Planning

Generation of operation sequences for individual parts itself is a challenging research problem. Therefore, we have decoupled multi-part setup planning problem from operation sequence generation problem. For the purposes of multi-part setup planning, we assume that we have been given an operation sequence for every part. In practice, we use our process planning system (described in [7]) to select tools and find near-optimal operation sequences for each individual parts independently. Our planner uses state-space search techniques to find near optimal operation sequence for individual parts (i.e., operation sequence that minimizes the number of tooling stages in the setup).

Given a set of parts and a set of operation sequences for parts, the goal of multi-part setup planning is to find a shared setup plan. In order to find a setup plan, the setup planning algorithm needs to compute the following.

1. Number of tooling stages in the setup.
2. For each tooling stage, compute its length and its location on the press-brake.
3. Assignment of each bending operation to a tooling stage. For a bending operation, the tooling stage assignment is specified by selecting a tooling stage and the relative position of the bend-line with respect to the left most edge of tooling stage.

We try to minimize the total number of tooling stages in the press-brake setup subjected to the following two constraints:

1. There should be no interference between tooling stages and workpiece shapes during any bending operation. Such interferences distort the workpiece and may damage the tools and are strictly forbidden.

2. Every tooling stage should fit inside the press brake. Each press brake has a predetermined finite tool holding space in which various tooling stages need to be installed. Therefore, total required length for various tooling stages should be less than the available tooling space on the press brake.

In addition to the minimization of the number of tooling stages, we select individual tooling stage lengths such that each tooling stage length is minimized.

4 Setup Planning Algorithm

For every bending operation, the workpiece geometry and the tool geometry impose constraints on the tooling stage that will be used to perform the bending operation. These constraints restrict the maximum tooling stage length and require certain minimum gaps between tooling stages. These constraints determine if more than one operation can be done on the same tooling stage.

The setup planning algorithm tries to minimize the total number of tooling stages in the setup and create setups that fit within the tool holding space of the press-brake. We identify bending operations with compatible setup constraints, and then generate setup plans by assigning bending operations with compatible setup constraints to the same stages.

Main steps in our setup planning algorithm are described below:

Step 1: Identify the setup constraints for every bending operation in the given set of operations.

Step 2: Find the most expensive bending operation from the setup point of view, using the following heuristics:

- Find the colinear operation with the maximum number of required interruptions.
- If there is no colinear operation, then find the bending operation with the maximum bend length.

Step 3: Find the set of operation among the remaining operations which have compatible stage constraints with the operation selected in Step 2.

Step 4: Construct composite setup constraints by performing:

- a union operation on various obstructions in the setup constraints;
- a intersection operation on various gaps in the setup constraints.

Step 5: Select the minimum length stage (or set of stages in case of colinear operations) which satisfy the composite setup constraints. Assign operations to stage(s) by computing relative locations of operation on the stage.

Step 6: Remove the assigned operations from the set of operations being considered. If there is any unassigned operations, then go to Step 2.

During each pass, the above algorithm assigns at least one operation to a stage (or a set of stages). This algorithm does not allow back tracking. Therefore, this algorithm runs in polynomial time. If the given set of operation sequences do not involve any interrupted colinear bends, then the optimal solution is guaranteed (i.e., we find the setup with the minimum number of tooling stages). This algorithm can be easily converted into a back tracking heuristic search algorithm to find the optimal answer in every case. However, for more than a thousand real-life test cases, we found sub-optimal solutions only in two cases. This success is based on the the heuristics used in Step 3 to identify the most expensive bending operations. In most cases, colinear operations with the largest interruption indices and the bending operations with the longest bend-lengths are the most constraining operations (i.e., have least flexibility in number of stages and stage length) and the most expensive from the setup point of view.

5 Generating Setup Constraints

Various bending operations impose constraints on tooling stage lengths. In order to do setup planning, we need to compute setup constraints resulting from various bending operations. Setup constraints are generated by analyzing any potential interference problem between the geometric models of the tool and the intermediate workpiece. These constraints describe the length restrictions on tooling stages and also identify the required gaps between tooling stages. Our approach for generating setup constraints consists of the following main steps:

Step 1. Construct the geometric model of the workpiece at the time of the bending operation. This model is referred to as the intermediate part model.

Step 2. Perform a geometric intersection of the intermediate part model with the model of a tooling stage spanning the entire press-brake tooling space.

Step 3. Analyze the part-tool intersection regions to determine setup constraint parameters.

For each bend-line, we use six parameters to capture the setup constraints associated with the bend-line. Figure 2 graphically shows these parameters. These parameters define the following setup constraints:

- *A tooling stage should fit within the gaps around the bend* (i.e., $Gr + Gl + L \geq S \geq L$).
- *A bend should be placed on the stage such that there is no part-tool collision on the left side of the bend* (i.e., $Gl \geq P$).
- *A bend should be placed on the stage such that there is no part-tool collision on the right side of the bend* (i.e., $Gr \geq S - P - L$).
- *A bend should be placed on the stage such that there is no part-tool collision with the left adjacent stage* (i.e., $Sl \leq P + Dl$).
- *A bend should be placed on the stage such that there is no part-tool collision with the right adjacent stage* (i.e., $Sr \leq S - P - L + Dr$).

where, Dl is the distance between this stage and the left adjacent stage. Dr is the distance between this stage and the right adjacent stage. L is the length of bend-line. S is the length of

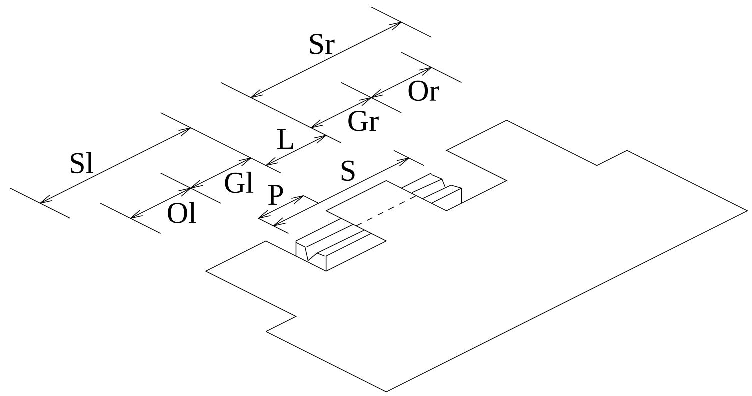


Figure 2: Setup constraint parameters.

tooling stage. P is the relative position of the bend line with respect to the left edge of the tooling stage.

Setup constraints for colinear bends are generated by combining setup constraints for various individual bends in the colinear bend. In cases of colinear bends, there are additional constraints on placement of stages with respect to each other. Relative positions of various stages with respect to each other in a colinear stage group should be such that for every individual bend b_i in the colinear bend there exist a relative position P_i with respect to S_i (stage to which b_i is assigned) that satisfies the setup constraints for b_i.

6 Compatibility of Setup Constraints

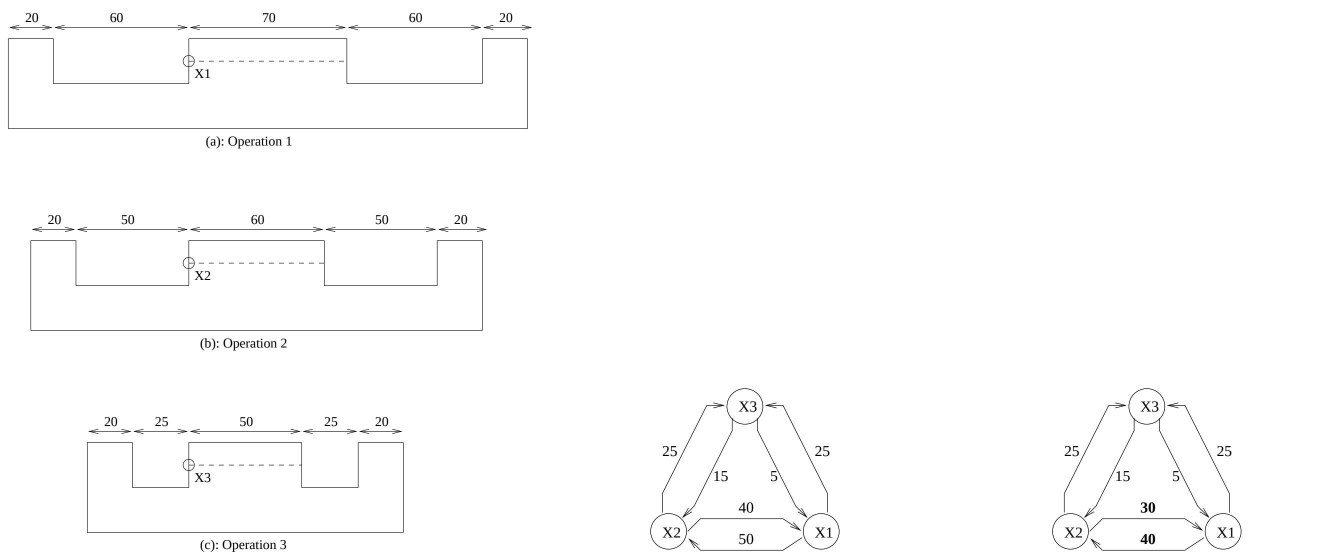
Two bending operations will have *compatible* setup constraints, if there exists a stage (or, a set of stages for colinear bends) which can accommodate both bending operations. We determine compatibility of two bending operations by establishing the feasibility of shared stages. If two operations can be overlayed on each other such that obstructions for one operation do not overlap bend-lines for another operation and vice versa, then we can create shared stages which can accommodate both operations.

Compatibility of Two Operations: Let i and j be two bending operations. Let o_i and o_j be reference points (i.e., left most points on the bend-lines) for these operations. Let X_i and X_j be the positions of o_i and o_j in an arbitrarily defined one dimensional world coordinate system. If setup constraints for two operations are compatible, then there will exist a *range of relative positions* of these operations in which obstructions for operation i will not overlap with the bend line for operation j, and vice versa. This condition can be mathematically expressed as follows:

$$X_j - X_i \leq XR_{j,i}$$

$$X_i - X_j \leq -XL_{j,i}$$

where, $XL_{j,i}$ is the left most position of operation j with respect to operation i, and $XR_{j,i}$ is the right most position of operation j with respect to operation i.



(a): three bending operations (b): Initial constraint graph (c): Reduced constraint graph

Figure 3: Example of Constraint Propogation.

For simple bends, $XL_{j,i}$ and $XR_{j,i}$ can be computed directly from setup constraints parameters in the following manner:

$$XL_{j,i} = \max(-Gl_i, L_i - L_j - Gr_j)$$

$$XR_{j,i} = \min(Gl_j, L_i + Gr_i - L_j)$$

If no feasible relative position range exists (i.e., $XL_{j,i} > XR_{j,i}$), then the two operations are considered incompatible.

For colinear bends, $XL_{j,i}$ and $XR_{j,i}$ are computed in the following manner. We first compute position range for each individual bend that compose the colinear bend. The position range for the colinear bend is computed by taking the intersection of position ranges for individual bends (e.g., the colinear bend can only be placed on the positions which fall within the positions ranges of every individual bend). Our current algorithm has the following limitation. For interrupted colinear bends, the interruption index of the shared stage is not allowed to exceed the maximum interruption index of the two colinear bends.

Compatibility of n Operations: In case of n bending operations, we get two inequalities for every pair $i, j (i \neq j)$. If there exists a vector $\{X_1, X_2, \ldots, X_n\}$ which satisfies above inequalities, then n operations are considered compatible.

7 Incremental Constraint Propagation

A bending press-brake can accommodate multiple tooling stages. Therefore, not all the operations (and the associated setup constraints) need to be compatible in order for parts to share a setup.

Unfortunately, we do not know a priori which sets of constraints are compatible. Moreover, in case of colinear bends there are alternative ways in which operations can be considered compatible. For example, consider a colinear bending operation o_1 with interruption index 1, and a simple bending operation o_2. If we want to do these operations on shared stages, operation o_2 can be located either on the left or the right side of the interruption for operation o_1, leading to two different possibilities.

To identify setup-compatible bending operations, we are using a graph-based incremental constraint propagation algorithm. This algorithm allows us to: (1) exploit special form of inequalities (i.e., all inequalities in this problem only involve algebraic differences of two variables), (2) incorporate sheet metal bending domain heuristics, and (3) explore various alternatives with very little backtracking.

This constraint propagation algorithm explicitly constructs the feasibility region described by the set of linear inequalities. In our formulation, we represent operation positions (X_i's) as nodes in the constraint graph. We add edges between all pairs of nodes. We assign weights to edges based on linear inequality constraints in the following manner. For a constraint $X_i - X_j \leq W$, we assign weight W to the edge from node X_j to node X_i. Similarly, for the constraint $X_j - X_i \leq W'$, we assign weight W' to the edge from node X_i to node X_j. Figure 3(a) shows three bending operations. Following are the constraints associated with the positions of these operations with respect to each other.

- Pair-wise analysis of Operation 1 and Operation 2: $X_1 - X_2 \leq 40$ and $X_2 - X_1 \leq 50$
- Pair-wise analysis of Operation 2 and Operation 3: $X_2 - X_3 \leq 15$ and $X_3 - X_2 \leq 25$
- Pair-wise analysis of Operation 1 and Operation 3: $X_1 - X_3 \leq 5$ and $X_3 - X_1 \leq 25$

Figure 3(b) shows a constraint graph for these constraints. Once constraints are expressed in a graph form, constraint propagation techniques are used to compute the minimal constraint graph. The minimal graph describes the feasibility region of the original constraints. A minimal representation of a given constraint graph can be found by finding shortest paths between various pairs of nodes in the constraint graph. Please refer to [6] for detailed description of conditions which state that for given a constraint graph G, the equivalent constraint graph G' defined by the shortest paths among various nodes gives the feasible values for various node variables. A given set of constraints is *inconsistent* if there is a negative cycle in the constraint graph. Figure 3(c) shows the minimal graph for the constraint graph shown in Figure 3(b). This minimal graph defines the feasibility region of the constraints described above. This minimal graph represents the following set of constraints:

$$X_1 - X_2 \leq 30;\ X_2 - X_1 \leq 40;\ X_2 - X_3 \leq 15;\ X_3 - X_2 \leq 25;\ X_1 - X_3 \leq 5;\ X_3 - X_1 \leq 25.$$

We use an iterative version of constraint propagation method to explicitly generate the feasibility region. We instantiate a constraint graph to keep track of the range of the possible positions of various operations with respect to each other. As indicated in Step 3 of our setup planning algorithm (please see Section 4), we start with the most expensive operation in the current set of operations. We refer to this operation as the seed operation. We evaluate various remaining operations for compatibility with the seed operation. In case of multiple choices, we select the compatible operation that is most expensive from the setup point of view (we use the same criteria as in Step 3 of our setup planning algorithm). The seed operation and its compatible operation are

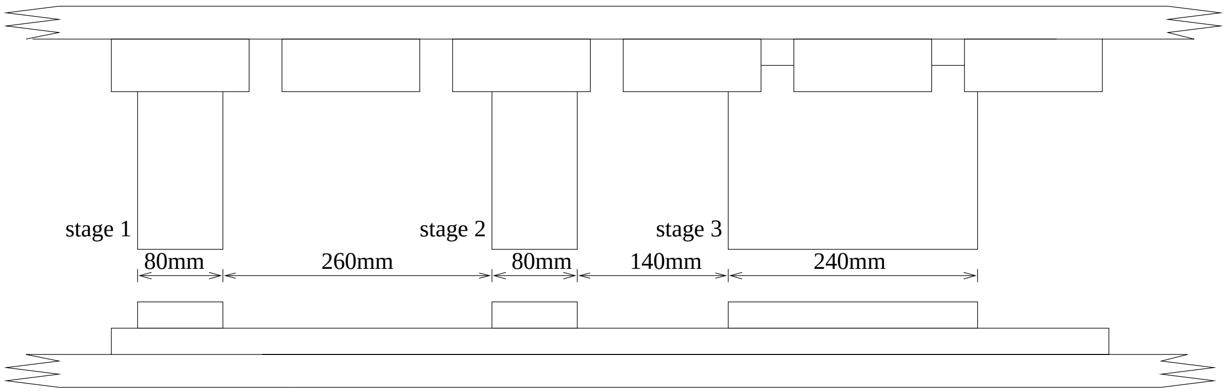


Figure 4: Shared setup for Parts shown in Figure 1.

used to initialize the constraint network. After adding these operations to the constraint graph, we calculate the position range for these two operations. We proceed by adding additional operations to the constraint graph one at a time. When multiple choices are present, we select the operation that is most expensive from the setup point of view. Each time we add an operation to the constraint graph, we update the possible position ranges of all the operations in the graph to account for the new operation by updating the shortest paths between various nodes. If adding a node results in a negative cycle, we do not add that node to the constraint graph. When it is not possible to add additional operations to the graph, we stop. As the next step, we compute positions of various operations with respect to each other from the possible position ranges. Position values are computed to minimize the tooling stage length. After computing position values of various operations in the graph, we compute composite setup constraints by performing union of various obstructions values and intersection of various gaps values.

For the three bending operation shown in Figure 3(a), the following position values are computed: $X_1 = 0$, $X_2 = 5$, and $X_3 = 10$. Composite setup constraints for these position values are described by the following parameters: $L = 70$, $Gr = 15$, $Gl = 15$, $Or = 65$, $Ol = 65$, $Sr = 65$, and $Sl = 65$. The composite setup constraint defined by above parameters implies that the minimum tooling stage length to perform all the three operations is 70mm.

8 Discussion and Conclusion

8.1 Implementation and Examples

The algorithm described in this paper has been implemented by using C++ . For geometric modelling and reasoning we have used NOODLES geometric kernel (developed at Engineering Design Research Center, CMU). For graphical interface we have used HOOPS graphics library (available from Autodesk Inc).

Figure 4 shows the shared setup for parts shown in Figure 1(a)-(c). The shared setup plan for these three part is shown in Table 1. The following operation sequences were used in generating this setup plan:

- Part 1: $[(b7)(b2, b3)(b4, b5)(b1)(b6)]$

Table 1: Shared Setup Plan for Parts shown in Figure 1.

part	*bend*	*stage*	*position*
Part 1	$b7$	stage 1	0
Part 1	$b2$	stage 2	0
Part 1	$b3$	stage 1	20
Part 1	$b4$	stage 2	0
Part 1	$b5$	stage 1	20
Part 1	$b1$	stage 3	30
Part 1	$b6$	stage 3	30
Part 2	$b1$	stage 3	0
Part 2	$b2$	stage 1	0
Part 2	$b3$	stage 2	60
Part 2	$b4$	stage 1	0
Part 2	$b5$	stage 2	10
Part 2	$b6$	stage 1	12
Part 2	$b7$	stage 1	-2
Part 3	$b1$	stage 3	80
Part 3	$b2$	stage 3	100
Part 3	$b3$	stage 3	70
Part 3	$b4$	stage 3	100
Part 3	$b5$	stage 3	70

- Part 2: $[(b1)(b2, b3)(b4, b5)(b6)(b7)]$
- Part 3: $[(b1)(b2)(b3)(b4)(b5)]$

In this case, we have assigned all colinear bends to the same group of stages. Please notice that lengths of colinear stages (stage 1 and stage 2) are not equal to lengths of any colinear bends. Instead, lengths of colinear stages have been derived from composite constraints and are suitable for accommodating all colinear bends.

8.2 Summary

We have described a new approach to setup planning for sheet-metal bending. This approach also works well for multi-part setup planning. The basic idea behind this approach is to first identify the setup constraints imposed by various operations and then generate setup plans which can satisfy these constraints. This approach allows us to encapsulate part specific information from the setup planning algorithm. Therefore, the setup planning algorithm can deal with one or more parts without any changes in its structure.

Setup changes constitute a major portion of the production time in the batch production environment. Multi-part setup planning can be used to significantly cut down the total number of setups and increase the overall throughput.

8.3 Current Limitations

Currently, we have the following three main limitations:

- The current implementation independently generates operation sequences for every part in the part-family. This may result in operation sequences with conflicting setup constraints and we may not be able to find compatible setups.
- Current implementation is based on a greedy technique (i.e., does not allow back-tracking) for fast response and interactive uses. It works satisfactorily for a moderate number of parts (10 or less). In order to aggressively find shared setups for large number of parts (greater than 20), we will require back-tracking capabilities and much more sophisticated optimization techniques.
- Given a multi-part setup planning problem, currently we either find a shared setup or return a failure. In case of failure, we do not partition the given problem into subproblems. Such a capability will allow us to divide unsolvable part-families into two or more smaller part-families and generate shared setups for these smaller part-families.

We intend to address these limitations in our future research.

Acknowledgments. This work was supported in part by NSF Grant DMI-9713781 and Amada America Inc. We would like to thank the people at Amada for their support and their thorough expertise in bending, in particular, Ken Hazama.

References

[1] Amada Sheet Metal Working Research Association. *Bending Technique*. Machinist Publishing Company Limited, first edition, 1981.

[2] Tien-Chien Chang. *Expert Process Planning for Manufacturing*. Addison-Wesley Publishing Co., 1990.

[3] L. Cser, M. Geiger, W. Greska, and M. Hoffman. Three kinds of case-based learning in sheet metal manufacturing. *Computers in Industry*, 17:195–206, 1991.

[4] L.J. de Vin, J. de Vries, A.H. Streppel, and H.J.J. Kals. PART-S: a CAPP system for small batch manufacturing of sheet metal components. In *Proceedings of the 24th CIRP International Seminar on Manufacturing Systems*, pages 171–182, Copenhagen, 1992.

[5] L.J. de Vin, J. de Vries, A.H. Streppel, E.J.W. Klaassen, and H.J.J. Kals. The generation of bending sequences in a CAPP system for sheet metal components. *Journal of materials processing technology*, 41:331–339, 1994.

[6] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[7] S.K. Gupta, D.A. Bourne, K. Kim, and S.S. Krishanan. Automated process planning for sheet metal bending operations. *Journal of Manufacturing Systems*, 17(5):338–360, 1998.

[8] T.M. Hancock. A systems approach to sheet metal processing. *Computers in Industry*, 13:245–251, 1989.

[9] B.O. Nnaji, T.S. Kang, S.C. Yeh, and J.P. Chen. Feature reasoning for sheet metal components. *International Journal of Production Research*, 29(9), 1991.

[10] B. Radin, M. Shpitalni, and I. Hartman. Two stage algorithm for rapid determination of the bending sequence in sheet metal products. In *ASME Design Automation Conference*, Irvine, CA 1996.

[11] J.S. Smith, P.H. Cohen, J.W. Davis, and S.A. Irani. Process plan generation for sheet metal parts using an integrated feature-based expert system approach. *International Journal of Production Research*, 30(5):1175–1190, 1992.

[12] H.P. Wang and J.K. Li. *Computer Aided Process Planning*. Elsevier Science Publishers, 1991.

[13] C. Wick, J.T. Benedict, and R.F. Veilleux, editors. *Forming*, volume 2 of *Tool and Manufacturing Engineers Handbook*. Society of Manufacturing Engineers, fourth edition, 1983.

[14] G. Yut and T.C. Chang. A study of automated process planning for sheet metal products. In *NSF Design and Manufacturing Systems Conference*, January 1993.