Automated Process Planning for Sheet Metal Bending Operations

S.K. Gupta, D.A. Bourne, K.H. Kim, and S.S. Krishnan

Rapid Manufacturing Laboratory
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

In this paper, we describe a generative process planning system for robotic sheet metal bending press-brakes. This process planning system employs a distributed planning architecture. Currently, our system consists of a central operation planner and three specialized domain specific planners: tooling, grasping, and moving. The central operation planner proposes various alternative partial sequences and each specialized planner evaluates them based on its objective function. The central operation planner uses state-space search techniques to optimize the operation sequence. Once a CAD design is given for a new part, the system automatically determines: the operation sequence, the tools and robot grippers needed, the tool layout, the grasp positions, the gage and the robot motion plans for making the part. The distributed architecture allows us to develop an open-architecture environment for doing generative process planning and encapsulate the specialized knowledge in specialized planners.

Keywords: Process Planning, Sheet Metal Bending, Automated Manufacturing, State Space Search, and Distributed Systems.

1 Introduction

In order to offer flexibility, better quality control, higher degree of automation, and improved productivity, machine tool manufacturers are combining material processing, material handling, and part positioning systems into single integrated manufacturing cells. Programming such integrated cells manually is a time consuming task and can become a major bottleneck in effectively using such cells. Process planning and part programming time directly affect the lot sizes that can be economically produced on these cells. We believe that automated

process planning systems can significantly enhance the throughput of such integrated cells and dramatically lower the economic lot sizes [6].

Depending upon the level of process plan details, the process planning systems can be divided into two different types: macro planners and micro planners. Macro planners deal with the higher level process planning decisions such as selection of machines, selection of operation types, ordering operations, selection of tools etc. Micro planners deal with the lower level planning decisions such as selection of operation parameters, NC code generation etc. To create a completely automated process planning system, we need both capabilities. Traditionally, these two types of planners were developed independently and were interfaced later. Due to strong interactions among various components of an integrated manufacturing cell, macro planning and micro planning functions need to be tightly integrated into a single system.

In this paper, we describe an automated process planning system for a robotic sheet-metal bending press-brake. Our system is based on the generative approach and performs both macro as well as micro planning. Once a CAD design is given for a new part, the system determines: the operation sequence, the tools and robot grippers needed, the tool layout, the grasp positions, the gage and the robot motion plans for making the part. These plans are sent to the press-brakes controller, which executes them and then returns gaging information back to the planning system for plan improvement. A second plan is then formulated, which reduces the gaging time by incorporating the reduced uncertainty in the part location.

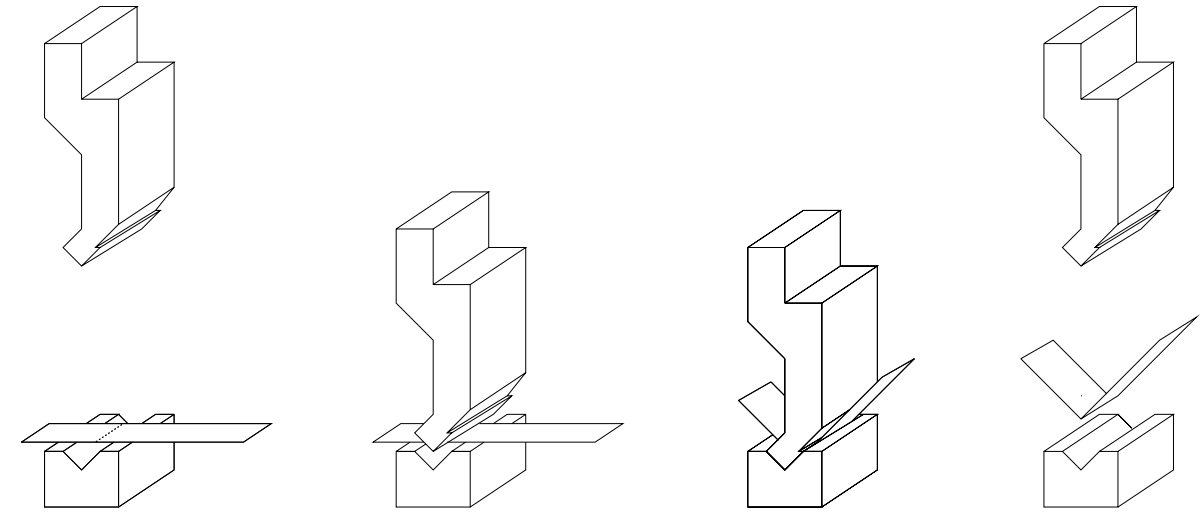
Our system is based on a distributed architecture. We have a separate planner for each specialized component of the robotic press-brake. These specialized planners collaborate with a central operation planner to perform the process planning. Currently, our system consists of a central operation planner and three specialized planners: tooling, grasping, and moving. The central operation planner proposes various alternative partial sequences and each specialized planner evaluates them based on its objective function. A distributed architecture allows us to encapsulate specialized planning knowledge of each component into a separate module and provides an opportunity for using a different representation and problem solving technique for each planning module. This architecture also provides a highly modular environment for adding more specialized planners to the system.

Our system presents a significant improvement over the state-of-the-art [2]. After the release of final CAD file, using our system, we can produce the first part in less than an hour. For full production automation of sheet metal bending, the resulting planning and execution time is reduced for the first part by an order of magnitude (comparing the system to trained human experts using the best available computer automation tools at this time).

2 Overview

2.1 Sheet Metal Bending

In sheet metal bending, a flat part is bent using a set of punches and dies. The punch and the die are mounted on a press-brake, which controls the relative motion between the punch



(a): position part on the die (b): position punch on the part (c): perform bending (d): take out the part

Figure 1: Sheet Metal Bending Process

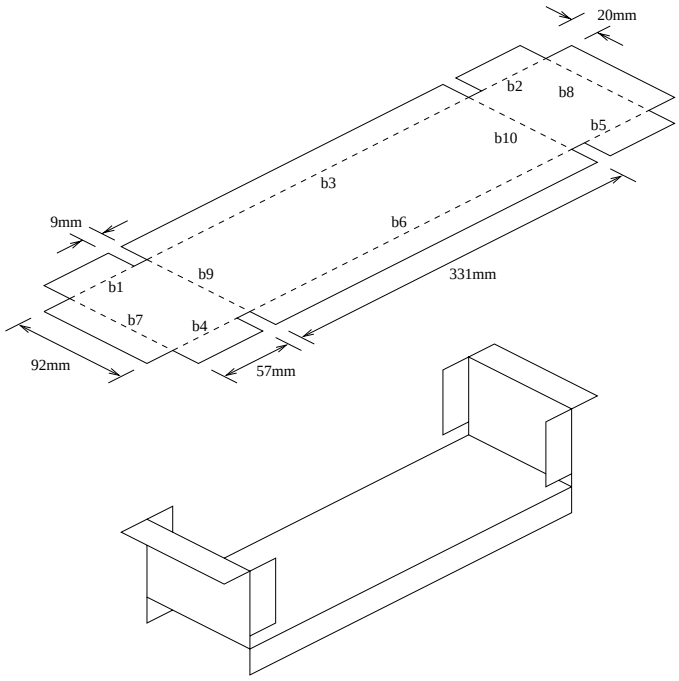


Figure 2: A Sheet Metal Part Example (bn are the names of dashed bend lines)

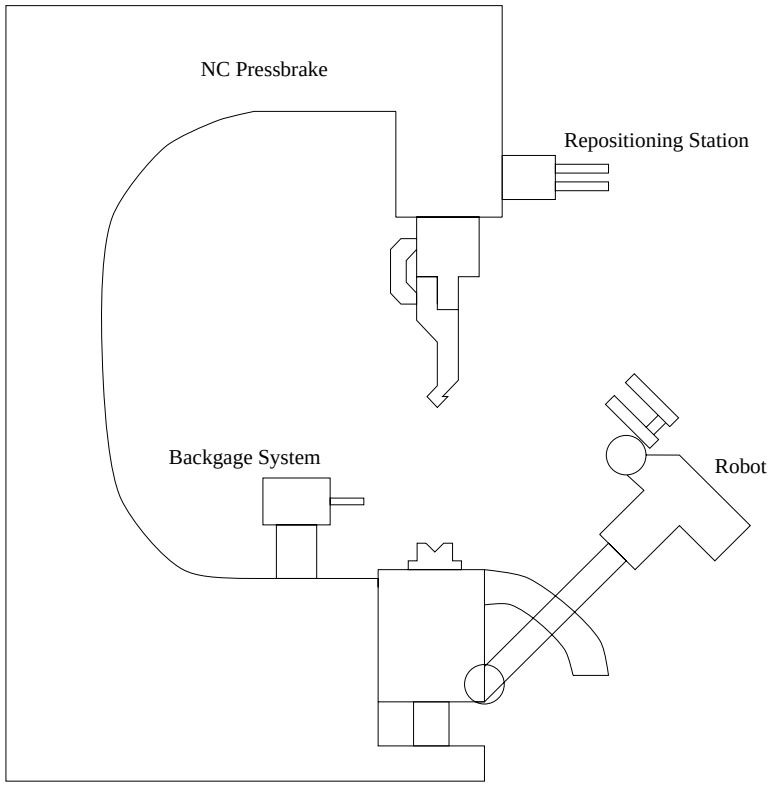


Figure 3: Bending Workstation

and die, and provides the necessary bending pressure. Figure 1 illustrates the sheet-metal bending process. For a detailed description of sheet-metal bending processes, readers are referred to sheet metal handbooks [4, 3, 28]. In a typical problem, we are given a final part and a starting flat part. The flat part is bent along the bend lines to create the final part. Figure 2 shows an example part and the corresponding starting flat. It should be noticed that each bending operation can be performed in two different ways. Each bend line connects two faces. Any one of these two faces can be kept outside the press-brake, resulting in two different possibilities for orienting the part in the press-brake. Therefore, in order to specify a bending operation, we need both the bend line and the part orientation. For the sake of brevity, we will represent bending operations only by identifying the associated bend lines. Many times the intermediate workpiece geometry is such that only one of these choices will work.

2.2 Terminology and Nomenclature

Bending Workstations: A bending workstation consists of the following elements: a NC press-brake, a robot for material handling, a reposition station for regripping the part, a back-gage system for locating the part in the press-brake. Figure 3 shows a bending workstation.

Tooling Stages: A tooling stage consists of a contiguous set of punches and dies. In most cases, the length of punch and die stages are the same and are aligned with respect to each other. Typically, bending punches and dies are available in a variety of segment lengths. These segments can be placed next to each other to create new tooling lengths. For example, a tooling stage of length 85mm can be created by combining segments of sizes 50mm, 20mm, and 15mm.

Operation Sequences: An operation sequence is an ordered set of bending operations. For example, $[(b1, b2)(b8)(b7)(b3)(b4, b5)(b6)(b10)(b9)]$ is an operation sequence for the part shown in Figure 2. 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 created simultaneously.

Ordering Constraints: A part cannot be bent in an arbitrary sequence. The geometry and tools being used impose constraints on the way a part can be repositioned. Such constraints are discovered by various specialized planners in our system. These constraints are simple precedence constraints that are all gathered together and unified. If contradictions are found between the constraints, the planner can either announce that the part cannot be made or it can make a determination that some constraints are more critical than others and then throw out the offending ones. The constraint language used by our system is a simple regular language that constrains the operation sequence in several ways: bends before other bends, the existence of simultaneous bends, and position (e.g., bend 1 is last). The language also uses uses wild cards "*" and "?" to represent any number of operations (including none) and exactly one operation, respectively. A simple constraint stating that $b1$ proceeds $b2$ with any number of bends before $b1$, before $b2$ or after $b2$ is written: (* $b1$ * $b2$ *).

2.3 System Architecture

Figure 4 illustrates the architecture of the process planning system. There is one central planner that sends out queries to specialized planners. The central planner keeps track of the query results and develops a near optimal plan. Specialized planners act as servers, which solve problems in grasping, tooling and moving for given partial operation sequences. Note that all of the planners communicate in the Feature Exchange Language (FEL) [5], which is a human readable, extendible language. This FEL syntax is regular, human readable and easily processed by each module.

In our system, the part's design is presented to the planning system, which automatically plans all aspects of the setup and the execution steps for making the part. A person is then guided step-by-step in the setup process and the plan is sent to the controller. The controller has a built-in interpreter for executing the plan on the bending machine. The part is loaded by a separate loading-unloading robot, and the bending robot starts to bend the part bend-by-bend. At some point, the robot may interfere with a bend-line, and as a result the robot hands the part to a repositioning gripper, so that the robot can alter its grasp position. The

bending, and regrasping are continued as needed until the part is complete, at which time the unloading robot grasps the finished part and stacks it. The results of this production run are used to produce a better and faster plan, since most of the gage information can be reused making successive parts (i.e., the robot is not accurate but it is repeatable). This modified plan is then used to make the rest of the parts in the batch.

2.4 Related Work

Automated Process Planning: Over the last two decades, many different automated process planning systems have been developed. Surveys of various systems and techniques can be found in [1, 15]. Chang's book on process planning [7] provides an excellent introduction to main issues, challenges, and techniques being used in automated process planning systems.

Most existing process planning systems primarily focus on machining (e.g., milling, drilling, and turning operations). Though the process planning problem is conceptually similar for machining and sheet-metal bending, the relative importance of various issues differ significantly. The following two examples illustrate these differences. For machining, recognizing machinable shapes is a major challenge. On the other hand, for sheet metal bending, features are collections of bend-lines and can be easily identified. However, there are situations in which due to interactions between bend-lines several alternative interpretations might be possible. For sheet metal bending, selecting the most appropriate punch profile shape is a challenging problem. On the other hand, cutting tools for milling and drilling are selected based on their dimensions (diameters and lengths).

AI Based Process Planning Systems: Many different AI techniques have been used for process planning. Wang and Wysk describe a knowledge based technique for process planning [27]. Hayes and Wright describe a rule based technique for for process planning [16]. Nau describes a frame-based technique for process planning [19]. Increasingly, heuristic search is being used as a problem solving technique [13, 17, 25] in macro planning tasks. The exact nature of search algorithm and heuristics depends on the particular planning problem being solved. In our opinion, defining the state-space for search, selecting the most appropriate search algorithm, and developing accurate geometric simulations are key factors behind the success of a process planning system. In addition, efficient heuristics are needed for guiding the search process and to maintain the tractability of the computational problem.

Minimizing the tool and setup changes is one of the key functions of macro planning. Sarma and Wright [23] describe many different algorithms for minimizing tool and setup changes for machining operations. For sheet metal bending, the nature of setup minimization problem (described in Section 4) is quite different from machining. Therefore, we needed different algorithms for solving this problem.

Distributed Architecture for Process Planning: Distributed problem solving techniques have been used in in the following two process planning systems. NEXTCUT [9, 10] is

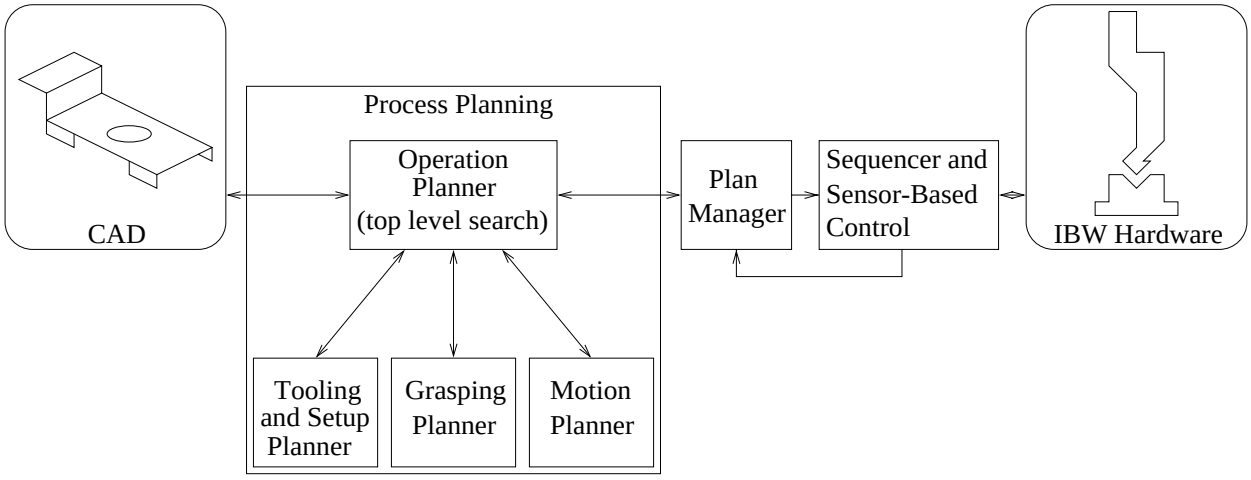


Figure 4: System Architecture

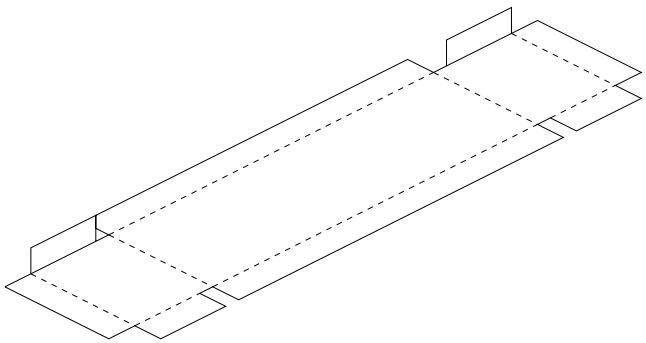
a process planning system for machining and it uses specialized agents to solve the pieces of the planning problem. Bourne [6] has developed a system for process planning for machining based on a distributed architecture. Our current system is based on an improved version of this architecture.

Process Planning for Sheet Metal: A number of systems have been developed to automate some aspects of the macro planning problem for sheet metal parts [24, 21, 29, 8]. These system attempt to handle a wide variety of sheet metal processes and attempt to order various operations based on high level interactions among them. While each of these systems has its strong points, it is difficult to assess their capabilities in dealing with the fine nuances of integration of macro and micro planning for robotic press-brakes. For example, an operation sequence that appears to be an optimal sequence during high level planning may actually turn out to be an infeasible sequence due to inadequate grasping area.

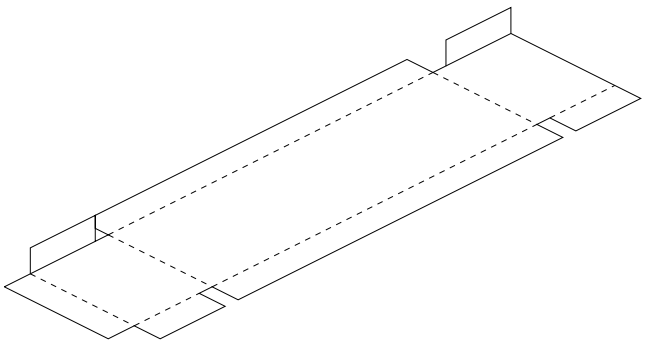
Finding an Optimal Operation Sequence for Bending: De Vin et al [11, 12] have developed a process planning system for finding a feasible operation sequence. Their system addresses the part-tool collision and tolerance constraints. In addition, they use heuristics for minimizing material handling time to guide the search.

Radin [22] et al have developed a variation of branch and bound search technique to find near-optimal operation sequences. 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.

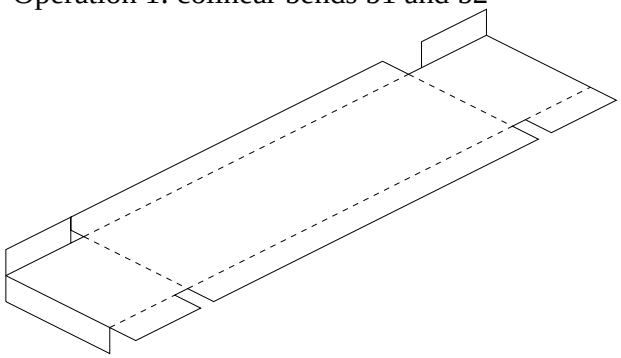
Both of these systems do not perform detailed planning for grasping and do not account for tool changes due to constraints on the stage lengths. Therefore, the operation sequence generated by these systems may turn out to be sub-optimal or may even be infeasible on a robotic press-brake.



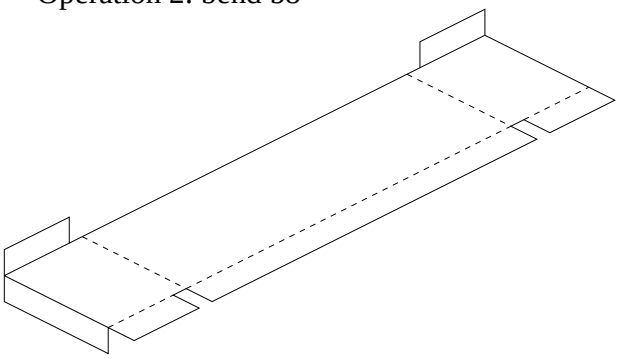
Operation 1: colinear bends b1 and b2



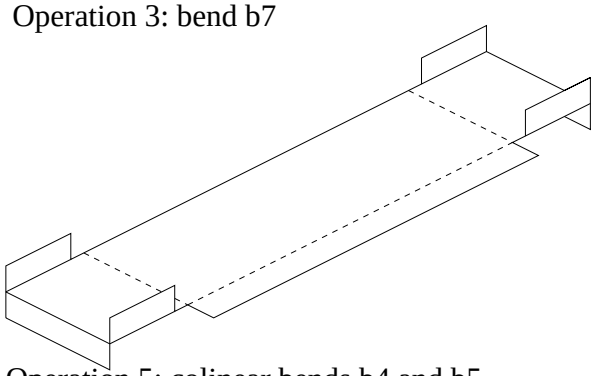
Operation 2: bend b8



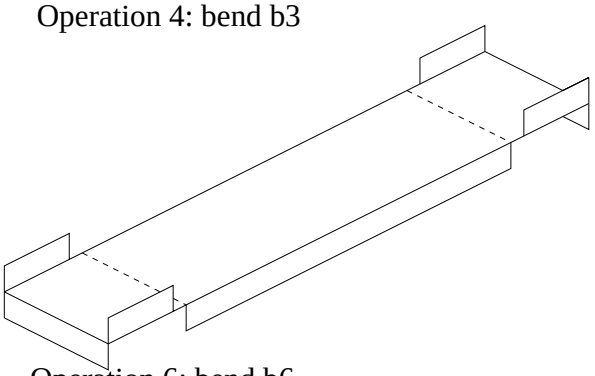
Operation 3: bend b7



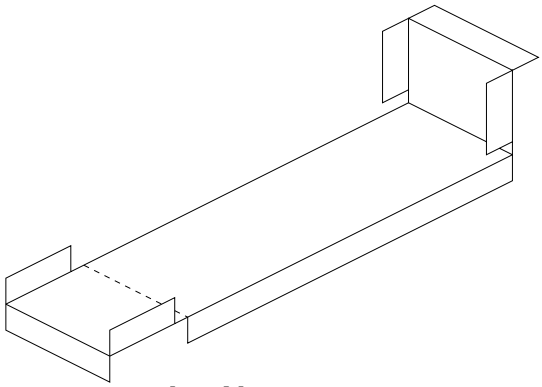
Operation 4: bend b3



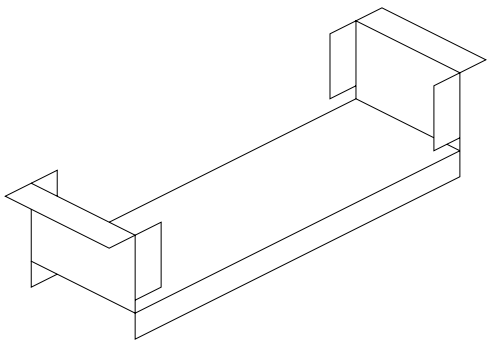
Operation 5: colinear bends b4 and b5



Operation 6: bend b6

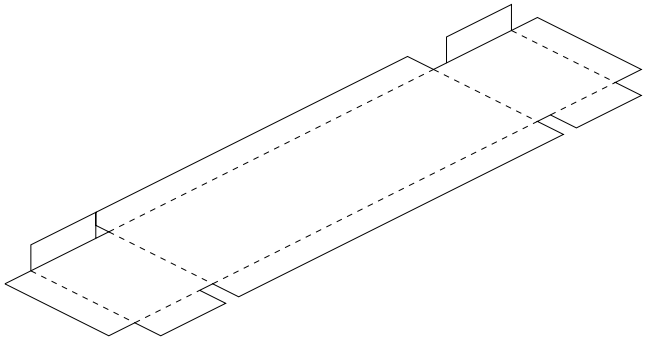


Operation 7: bend b10

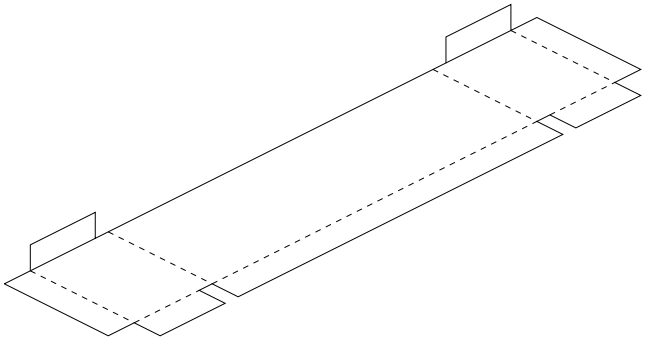


Operation: bend b9

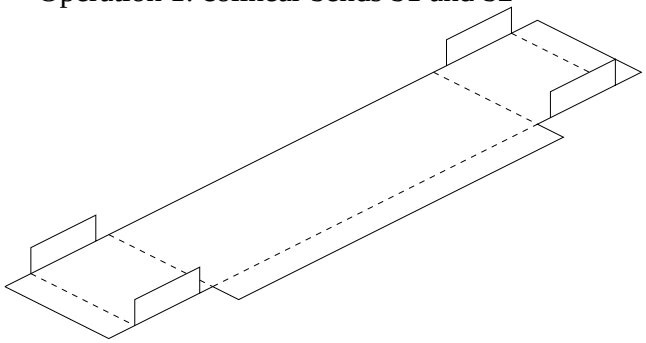
Figure 5: Operation Sequence 1



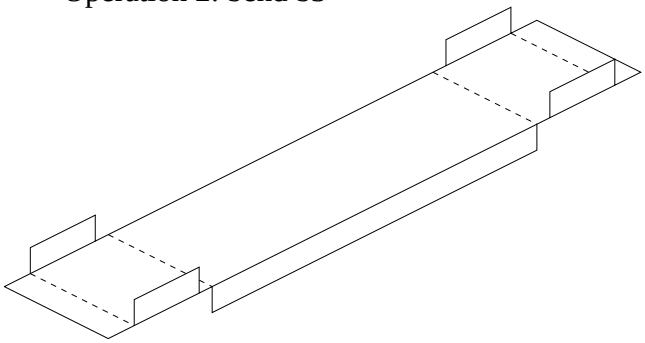
Operation 1: colinear bends b1 and b2



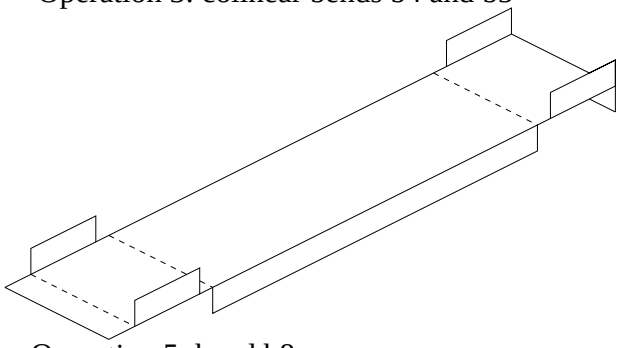
Operation 2: bend b3



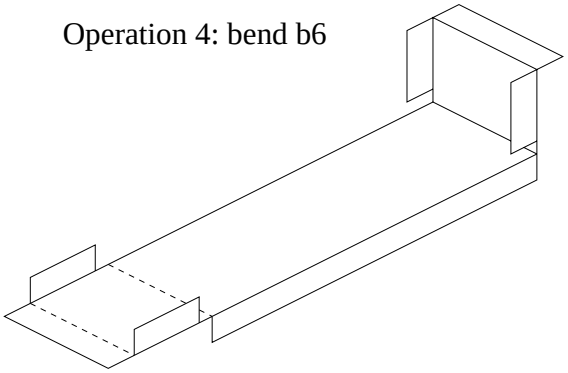
Operation 3: colinear bends b4 and b5



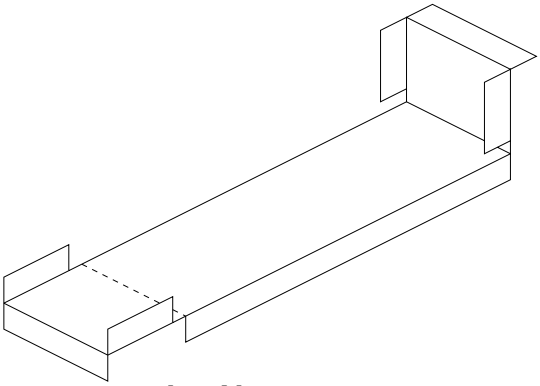
Operation 4: bend b6



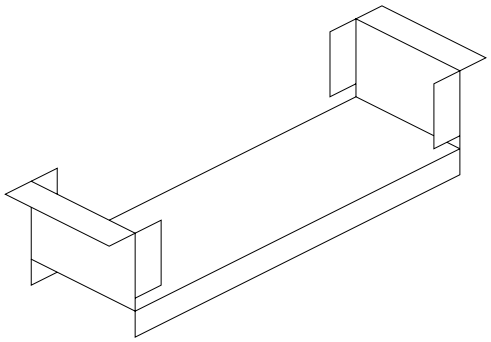
Operation 5: bend b8



Operation 6: bend b10



Operation 7: bend b7



Operation: bend b9

Figure 6: Operation Sequence 2

3 The Central Operation Planner

The central operation planner itself is a governing module that keeps track of the potential alternatives and optimizes the operation sequence. It proposes alternative partial operation sequences to various specialized planners. Each specialized planner evaluates the proposed sequence based on its own cost criteria and returns the evaluation to the central operation planner. The central operation planner uses state-space search techniques to optimize the operation sequence.

3.1 State-Space Search Formulation

We have formulated the process planning problem as a state-space search problem. In our formulation, the starting blank (e.g., the flat sheet) is considered the initial state and the final bent part is considered the goal state. Various bending operations act as the search operators that transform one search state to the other search state. The operation planning problem is defined as the problem of finding a near-optimal sequence of operations that transform the initial state into the goal state.

In our formulation, each partial operation sequence is a possible state in the search tree. State evaluation is performed incrementally, i.e., a new state is generated by adding a single bending operation to an existing state. Incremental evaluation allows specialized planners to evaluate the given state by only reasoning about the last operation and caching the results for the remaining operations from the parent state.

3.2 Search Algorithm

The operation planner uses an A* algorithm[1] coupled with a constraint solver. The A* uses the aggregate costs from the specialized planners to attempt plan optimization. However, to avoid trying impossible sequences, the planner first verifies that a given operation sequence is consistent with its available constraints. Then following the A* algorithm, the planner chooses the cheapest sub-sequence at each step and queries all of the specialized planners about how expensive it would be to add one operation to the current operation sequence. This process continues until all of the bends have been completed.

A* algorithm produces an optimal solution if the predicted cost is less than or equal to the actual cost. Prediction heuristics that satisfy this condition are called admissible prediction heuristics. In most cases, perfect prediction is impossible without exploring the complete search space. Therefore, one can either use admissible heuristics that tend to under-predict, or one can use inadmissible heuristics that tend to over-predict. Using inadmissible heuristics leads to sub-optimal solutions. However, the solution found by A* is sub-optimal only by the over-predicted amount. In our case, using admissible heuristics leads to exploration of

[1] A best-first search algorithm with prediction heuristics for estimating cost to the goal state from the current state (for details please see Nilsson's book [20])

large search space. Therefore, we are using inadmissible heuristics. Our current heuristics predict very close to the actual cost and as a result we get near-optimal solutions.

Each specialized planner uses algorithms that are greedy about satisfying their own objectives, and if those objectives cannot be met then the entire system performance is sacrificed (both in terms of the required search time and the quality of the final plan). To compensate for this dilemma, we have developed a scheme to share the part specific constraints before the search for a plan begins. Sharing constraints among various specialized planners allows each specialized planner to account for other's constraints in the prediction of cost and reduces the number of conflicts.

3.3 Communications with Specialized Planners

The central operation planner uses the following three types of queries to communicate with the specialized planners:

Preparatory Queries: The planner starts by sending the part name as well as other locations for various databases that describe the manufacturing domain: machine description, tool library and gripper library. Then each specialized planner makes an initial assessment of what tools (e.g., punches, dies and grippers) are required to make the part. In addition, each planner makes a prediction of how expensive it will be to make the part according each planner's own cost criteria. Finally, each planner is given the opportunity to describe constraints that must be followed for that planner to succeed at all.

Search Queries: After the preparatory phase, the state space search is performed to find a near-optimal solution. During this phase, the central planner proposes various alternative partial sequences and sends them to specialized planners for evaluation. Each specialized planners evaluates the given partial sequence according its own cost criteria. Each specialized planner computes the actual cost of the sequence and the predicted cost of performing the remaining bends. If the given partial sequence is infeasible for a specialized planner, then that planner returns infinity as the cost for that particular partial operation sequence. The central operation planner adds the actual cost and the predicted cost returned by various specialized planners and uses this information to guide the search.

Finalize Queries: After a near-optimal operation sequence is found, every specialized planner is given a chance to provide the detailed plan information during the final round. During this phase specialized planners perform computation intensive portions of their task. Specialized planners tend to perform feasibility tests during search. However, sometimes computing controller interpretable instructions is a computation-intensive task. Such tasks are performed during this phase. This phase is also used to perform computation-intensive second order optimization within a given bend sequence, for example, a good robot motion plan.

3.4 Examples

For a given part, there is no universal best plan. Depending upon the batch size, the best sequence may change. Each repositioning takes on the average 15 seconds. Setting up a tooling stage on the press-brake takes on the average two minutes [2]. Please note that repositioning needs to be performed for every part in the batch. On the other hand, the stage setup needs to be done only once for every batch. Figures 5 and 6 show two different operation sequences for the part shown in Figure 2. Operation Sequence 1 requires one repositioning operation and four tooling stages. On the other hand, Operation Sequence 2 requires two repositioning operations and two tooling stages (please refer to Sections 5 and 4 for details). Let us assume that execution time (excluding repositioning time) for both sequences is one minute. Now let us consider the following two cases:

- *Case 1:* batch size is one.
  - *Time for Operation Sequence 1:*
    total time = execution time + repositioning time + setup time
    $1 \times 1.0 + 1 \times 0.25 + 4 \times 2.0 = 9.25$ minutes
  - *Time for Operation Sequence 2:*
    $1 \times 1.0 + 2 \times 0.25 + 2 \times 2.0 = 5.50$ minutes
- *Case 2:* batch size is one hundred.
  - *Time for Operation Sequence 1:*
    $100 \times 1.0 + 100 \times 0.25 + 4 \times 2.0 = 133$ minutes
  - *Time for Operation Sequence 2:*
    $100 \times 1.0 + 200 \times 0.25 + 2 \times 2.0 = 154$ minutes

If we are producing only one part, Operation Sequence 2 is preferred over Operation Sequence 1. On the other hand, if we are producing one hundred parts, Operation Sequence 1 is preferred over Operation Sequence 2. Our system can automatically account for the batch size and can produce the near-optimal plan for the given set of resources.

4 The Tooling and Setup Planner

The objective of the tooling and setup planner is to construct a press-brake setup for the given operation sequence (partial or complete). In order to determine the best press-brake setup, several considerations must be made: (1) selecting bending tools (i.e., punches, dies, punch holders, and die holders); (2) finding tool stage layouts inside the press-brake bounds

[2] This setup time does not include the time to establish the machine coordinate system. This time is usually independent of the number of states. On the average, it takes ten minutes to establish the machine coordinate system for a new setup.

(i.e., which tool should be positioned where); and (3) assigning each bending operation to a tooling stage in the press brake setup.

The tooling and setup planner must create compact setups that utilize the press-brake space effectively. This is needed not only to minimize setup efforts, but also to reduce material handling times. Any setup where the tooling stages are not maximally compact results in increased transfer time from one stage to the other, and risks not being able to make the part at all because some stages may not fit on the machine.

The tooling and setup planner influences the planning process by communicating estimated setup efforts (and feasibility). In the current scheme, each new tooling stage requires a constant setup effort. Therefore, the setup effort is directly proportional to the number of stages. For each partial operation sequence, the tooling system provides two cost estimates to the A* algorithm. The first estimate is the setup cost for the partial operation sequence. The second cost estimate is the estimated setup cost for the remaining operations. The first cost estimate is done by creating a press-brake setup for the partial operation sequence and counting the number of tooling stages in the setup. For the second estimate, we predict the required number of tooling stages based on the predicted intermediate part shape for the remaining bends.

4.1 Tool Selection

The tooling and setup planner selects punches, dies, punch-holders and die-holders, before the search for the operation sequence begins. The tool selection process is a two-step process. The first step is to match the available tools with the desired features of a given bend, such as bend-radius and bend-angle. This step results in a list of potentially feasible tools which can be used to perform the bend. The next step is to determine whether the tool and part will not interfere with each other while bending. This requires that we select the tool whose profile does not collide with the part during bending.

4.1.1 Predicting Intermediate Part Shapes

In order to select tool profiles to perform a certain bend, we need to know what the geometry of the part will be while performing the bend. We have observed that the critical constraints during punch profile selection, usually are a set of closely related parallel bends forming recognizable "features". Depending upon how many bends are involved and the direction of their bend angles, we have grouped them into two-bends, z-bends, channels and hat-bend features.

The first step in the tool selection process is to recognize the existence of various features in the part. Once these features are extracted from the part, we use them to predict what the parts will look like while doing a bend which participates in any of the features. This is possible because we know the operation sequence within a feature. For example, in a two-bend feature we need to first do the bend farthest from the gripper and then the other bend. Similarly, in certain channel features, we need to do the middle bend first, followed by

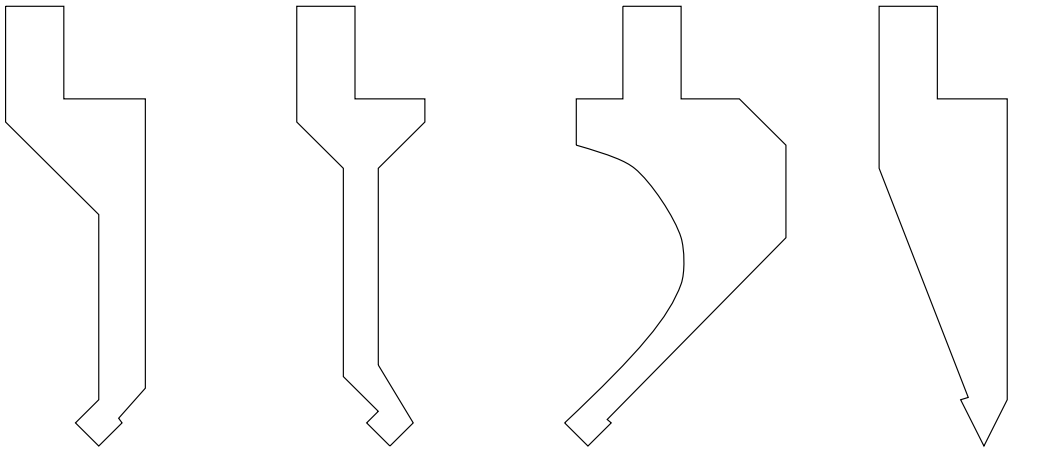


Figure 7: Various Types of Punches

the bend which is closer to the gripper and then the third bend. There are similar predicted operation sequences for each of our features.

We use the feature extraction process to find the various features in which this bend has interactions. This information is then used to generate precedence constraints between the current bend and the other bends with which it shares features. These precedence constraints are used by the tooling and setup planner in two ways: (1) constraints are sent back to the central operation planner to guide and prune its search process; (2) constraints are also used to build predicted intermediate part models for each bend. A predicted intermediate part model consists of bending up all the bends which are likely to precede the current bend, on the basis of our feature recognition process.

4.1.2 Punch and Die Selection

For each bend, the designer specifies the desired bend-angle and the bend-radius. This information is used along with the thickness of the material is used to select tools. The dies and punches are considered feasible if their geometric parameters (radius, angles) are suitable for the design parameters of a given bend. Combinations of various tool parameters and their matching bend parameters are are described in handbooks of bending techniques [4, 3, 28].

We first select a least constraining die from among a database of dies. The selected die should match the desired V-width and be capable of withstanding the tonnage required to make the bend. In addition, the bend should not have a flange smaller than the minimum needed for the selected die.

The selected die is then used to prune the list of punches in the database to come up with those punches which match the die. Figure 7 shows various types of punches). The punch tip-angle is required to be the same as the die V-angle. In addition, the punch tip-radius is required to be as close to bend radius as possible. The punch should be capable of handling

the tonnage needed to perform the bend. The list of feasible punches is then used to select the one whose profile will be feasible for making as many bends as possible.

Predicted intermediate part models are used to perform intersection tests with candidate punch profiles in order to select a profile which is found feasible for all bends in the part. If any one profile is not feasible for all bends in the part, then a minimal set of punch profiles are selected such that they cover all the bends in the part. The selection of the punch profiles can be adapted to satisfy various optimality criteria.

We have obtained good results with the feature extraction method (described in previous subsection) of building intermediate part models. The drawback of this method is inherent to feature-recognition algorithms and is very sensitive to the boundary conditions since we do not have a fuzzy concept of features. This method has worked well in a variety of test parts which we have analyzed and has resulted in picking the best punch profiles for making the part.

4.2 Setup Planning

For every bending operation, the intermediate 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. Every feasible press brake setup needs to respect these constraints. The intermediate workpiece shape is determined by the bending sequence. Therefore, the bending sequence and the type of tools being used have a strong influence on setup constraints. Main steps of our setup planning approach are described below:

- Identify the setup constraints for every bending operation in the given set of operation sequences.
- Partition the set of bends into various compatibility sets (i.e., sets of bends having compatible setup constraints). Bends which are in the same compatibility set can be performed on the same tooling stages.
- Combine compatible constraints into composite constraints. These composite constraints are used to generate tooling stages that satisfy these constraints. Finally, bending operations are assigned to tooling stages by specifying their relative locations with respect to stages.

4.2.1 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

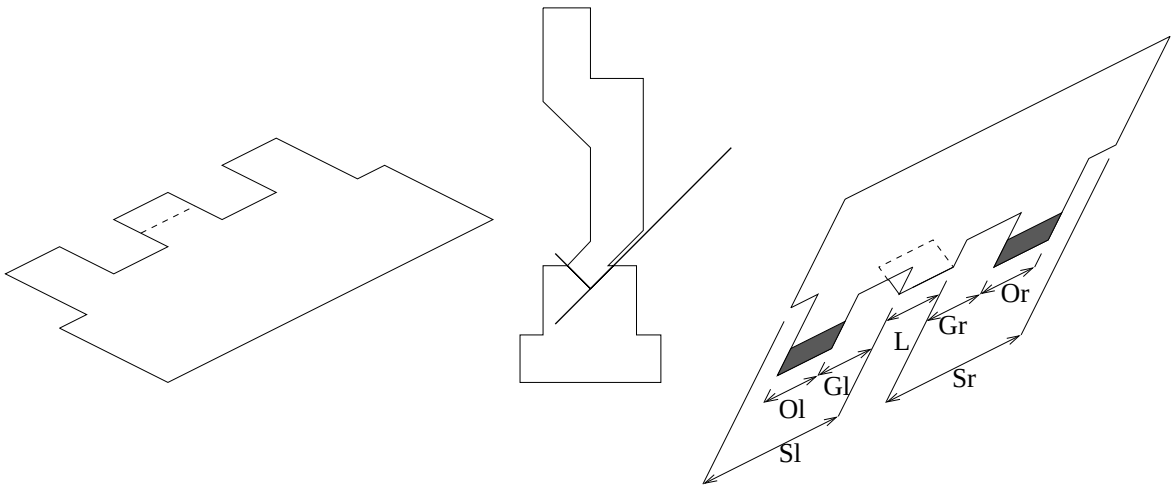


Figure 8: Generating setup constraints.

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:

- Construct the geometric model of the workpiece at the time of bending operation in the bend sequence. This model is referred as the intermediate part model.
- Perform a geometric intersection of the intermediate part model with the model of a tooling stage spanning the entire press-brake tooling space.
- Analyze the part-tool intersection regions to determine setup constraint parameters.

Figure 8 shows an example of setup constraint generation. In this case, tabs on both side of the bend intersect with the die during intersection test. Therefore, this bend cannot be performed on an infinitely long tooling stage. The minimum tooling stage length for this operation is $L -$ tolerance[3]. In order to avoid interference between the tool and the intermediate workpiece, the maximum allowed tooling stage length is $Gr+Gl+L-$clearance[4]. Besides these restrictions, adjacent stages should also clear the safety margins.

4.2.2 Generating Setup Plans

Once we have computed the setup constraints for a partial or a complete operation sequence, we can proceed with the setup planning. We are interested in creating setups that involve the minimum number of tooling stages and fit on the die rail of the press-brake.

[3]Minimum allowed tooling stage length is slightly smaller than the bend length. Recommended tolerance is around 2 mm. Reducing the tooling stage length by more than this value results in poor bend quality.

[4]Maximum allowed tooling stage length is slightly smaller than the overall gap around bend. Recommended clearance is around 2 mm. In practice, this value depends on the accuracy of part placement.

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 composite stages. If two operations can be overlayed on each other, such that obstructions for one operation do not overlap bend-lines for the other, and vice versa, then we can create *composite stages* which can accommodate both operations.

Compatibility of Setup Constraints Associated with Bending Operations: Let i and j be two bending operations. Let o_i and o_j be reference points for these operations. Let X_i and X_j be the location 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 range of relative positions can be computed from examining the setup constraint parameters shown in Figure 8. This condition can be mathematically expressed as follows:

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.

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

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

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

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

Compatibility of n Operations: In case of n bending operations, we get the following two inequalities for every pair $i, j (i \neq j)$.

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

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

If there exists a vector $\{X_1, X_2, \ldots, X_n\}$ which satisfies these inequalities, then n operations are considered compatible. We use an iterative constraint propagation method to identify the relative position range of every operation.

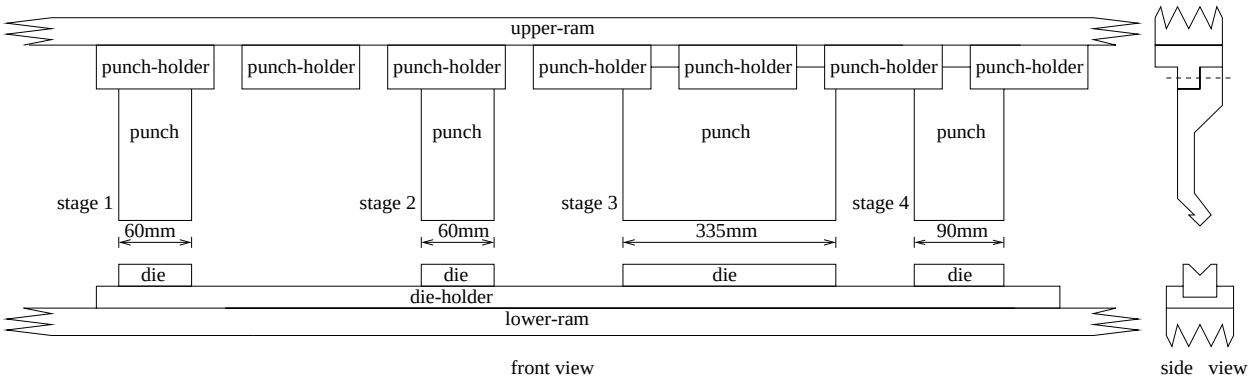


Figure 9: Setup for Operation Sequence 1 (shown in Figure 5)

Setup Planning Algorithm: Main steps in our setup planning algorithm are described below:

- Let O be the list operations that has not been assigned to any tooling stage.
- Find the most constraining bending operations o in O using the following heuristics:
  - Find the colinear operation with the maximum number of required gaps in O.
  - If there are no colinear operations, then find the bending operation with the maximum length in O.
- Let $c(o)$ be the operations in O which have compatible stage constraints with o.
- Build a stage (or a set of stages) s which satisfy stage constraints for o and $c(o)$. Assign o and $c(o)$ to s by computing relative locations of o and $c(o)$ with respect to s.
- Remove o and $c(o)$ from O.
- If O is not empty, then go to Step 2.

After identifying required tooling stages and assigning various bends to stages, we arrange stages on the bending machine to minimize the motion efforts in transferring part from one stage to other. Currently, this step is accomplished by using a greedy technique which involves identifying part transfer frequency among all pair of stages and placing stages with higher frequency next to each other. This algorithm also takes into account external stage positioning constraints that restrict certain stages to be positioned at certain locations due to robot-grasping or part-gaging requirements.

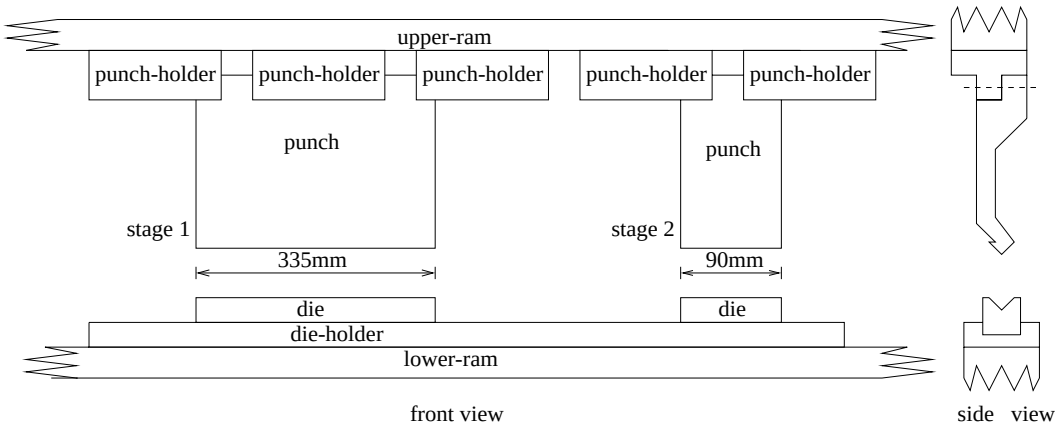


Figure 10: Setup for Operation Sequence 2 (shown in Figure 6)

4.3 Tooling and Setup Planning Example

For the part shown in Figure 2, the tooling and setup planner selects a regular punch and a die with a 6mm V-width.

Operation Sequence 1 (shown in Figure 5) requires at least four tooling stages. Bends b3 and b6 are the longest bends in the part. Therefore, we need a tooling stage for them. In this operation sequence, bends b9 and b10 are constrained on both sides in the intermediate workpiece. Whenever a bend is constrained on both sides in the intermediate workpiece, it cannot be done on a tooling stage which is longer than the bend. Therefore, the tooling stage used for bends b3 and b6 cannot be used for bends b9 and b10, requiring a different tooling stage for these bends. Similarly, colinear bends b4 and b5 are constrained on both sides in the intermediate workpiece. Therefore, this operation sequence requires at least four tooling stages. The press-brake setup for this operation sequence is shown in Figure 9.

Operation Sequence 2 (shown in Figure 6) requires at least two tooling stages. In this case, we need a tooling stage for bends b3 and b6. In this operation sequence, bends b7, b8, b9 and b10 are constrained on both sides in the intermediate workpiece. Therefore, we need a different stage for them. It should be noticed that in this operation sequence, no other bend is constrained on both sides in the intermediate workpiece. Therefore, this operation sequence requires at least two tooling stages. The press-brake setup for this operation sequence is shown in Figure 10. Therefore, Operation Sequence 2 is better from press-brake setup point of view.

5 The Grasping Planner

The grasping planner is responsible for determining the best place to grasp the part during the bending process. In addition, it must also determine how a second gripper can be used to temporarily hold the part, so that the robot gripper can regrasp it at a different position. We call this operation a *repo*, which is short for gripper repositioning. To accomplish these

goals successfully, the grasping planner must also choose the best robot gripper and repo gripper for the given part. Once a search for the operation sequence begins, the grasping planner receives a request from the central planner with a candidate operation sequence and the grip area (feasible grasp locations) that has already been determined for the previous bend(s) (if any). The grasping planner then generates a new grip area for the last bend in the sequence and identifies the common gripping area by intersecting the previous grip area. This intersected grip area is returned to the planner with a prediction for how many repos will be required to complete the part; production time is a function of this parameter. When the grasping planner cannot identify any grip area, it returns an infinite cost to the planner, meaning that the current operation sequence cannot be accomplished from the grasping perspective.

The success in finding a near optimal operation sequence in a reasonable time depends on how well the number of repos is predicted before and during the search. The method for predicting the number of repos is based on choosing a number of potential grasp points and determining how many bends can be accomplished without releasing the part. The bends that are feasible from a grasp point are called a bend set. Grasping then combines different bend sets until all of the bends are accounted for and the repo prediction is simply the number of bend sets combined minus one. The predicted cost for the A* algorithm is computed by multiplying the repo prediction and the repo penalty (or time), which is an estimate of how long it actually takes to repo the part. Typically, a part made with fewer repos is more accurate than a part made with more repos and it also saves production time. The accuracy problem arises because of compliance in the grippers grasp surface, mechanical slop in the gripper, robot positioning error and slip during the grasping process. The production time constraint is more critical when the batch size is large and the goal is to minimize cycle time.

5.1 Gripper Selection

Given the part and tool information, the grasping planner identifies the most suitable robot gripper and repo gripper. Once grippers are chosen, the search process can begin and the grasping planner can make accurate predictions about the number of repos that are required at any given point in the search. Our system has a variety of grippers from which to choose (about 50). In order to standardize and speed up the computing effort for selecting grippers, each gripper in the library is simplified by parametrizing it according to its critical dimensions. Figure 11 shows two types of grippers in a typical gripper library. In order to select a gripper that supports an operation sequence with the least number of repos, the grasping planner tries each gripper in the gripper library to see if it can be used to hold the part and then it estimates the required number of repos required for that gripper. It is usually easier to hold the 2D flat part than 3D finished part, therefore the number of bend sets required to form a closure (all bends) is usually less when the 2D geometry is used. We use the 3D geometry to generate a predicted worst case for the repo prediction, although this is not necessarily true. While intermediate part shapes can be used to predict a more accurate number of repos, it becomes computationally expensive to generate part geometries

according to all possible operation sequences. Given a robot gripper and a part, the bend sets are generated by:

- Selecting candidate grip locations discretized with a spacing along the boundary of the part.
- Filtering out invalid grip locations by checking for interference between the gripper and bent flanges, while avoiding poor contact between the gripper pads and the part.
- Filtering out invalid grip locations by checking them against the robot's workspace limits and to avoid collisions with other components of the bending machine.

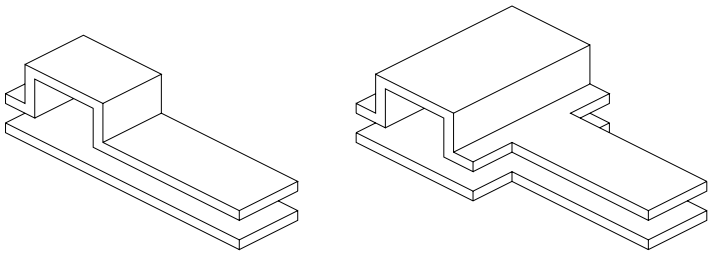
For a given gripper, the estimate for the number of repos is generally between the predictions made with the 2D and the 3D geometry. Thus, when these repo predictions are the same, we have a high confidence that the prediction is correct. By this analysis, every gripper in the library is evaluated and ranked. A gripper that requires the least number of repositioning is selected. When there are multiple grippers with the same ranking, heuristics based on the part and gripper size are used to pick a gripper in order to minimize part droop during handling. Typically, a gripper with a bigger width and a longer finger length is favored for this purpose.

5.2 Finding Robot Grip Locations

Many part geometries have complicated boundaries, and so we have found it easier to represent the grasp area as a finite number of grip line segments (normal to part's boundary) rather than grasp areas. A grip line segment represents valid grip locations with a varying depth. A gripper can hold the workpiece and make a given bend as long as the gripper's reference center is placed along this line segment. For a point (on a grip line segment) to be a valid grip location, the following constraints must be met:

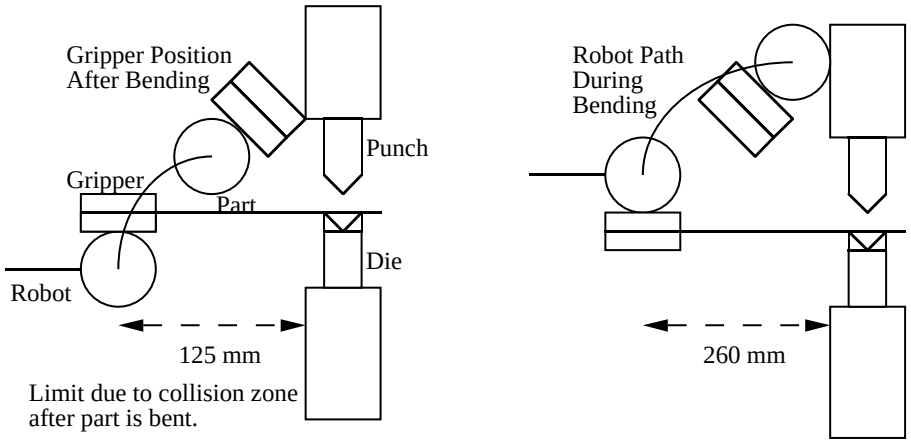
- The gripper pad(s) should have good contact with the workpiece.
- The gripper geometry should not interfere with the workpiece when the gripper is closed.
- The gripper should not interfere with the machine and tools during bend following.
- The robot's tool center point must remain within the workspace during all bending processes such as repositioning, loading/unloading and gaging.

Droop makes it difficult to load and unload the part from the press-brake, as well as performing the repo operation. Thus, it is important to choose a gripper location from the feasible grip area that minimizes droop. For this reason, we favor a grip location at the deep end of each grip line segment and as near to the center of the part as possible.



(a): IG gripper with up knuckle gap (b): TG gripper with up knuckle gap

Figure 11: Grippers with Simplified Geometry



(a) Gripper Right Side Up Limit **(b) Gripper Upsidedown Limit**

Figure 12: Robot Limit Problem Because of Gripper Offset-Axis

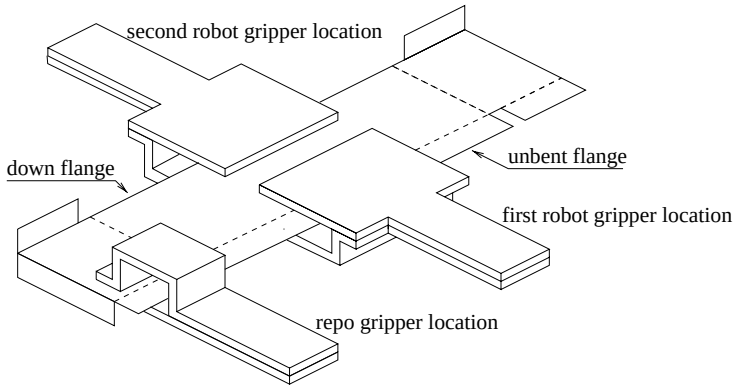


Figure 13: Robot and Repo Gripper Locations for Operation Sequence 1

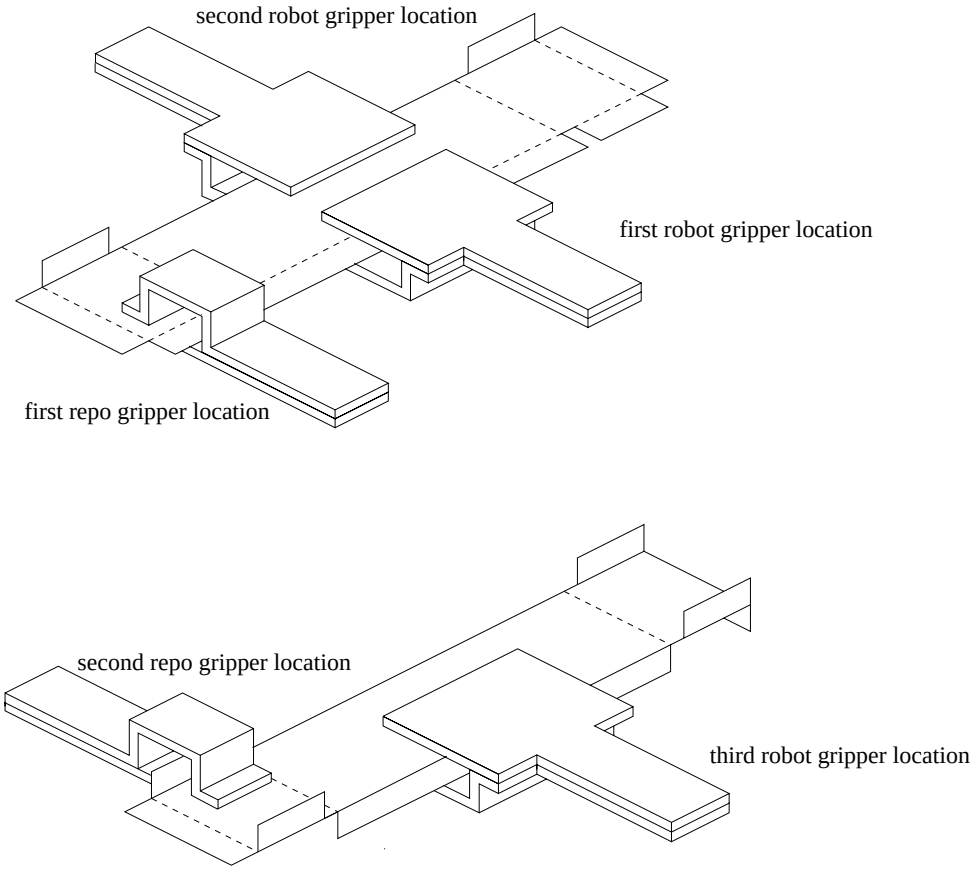


Figure 14: Robot and Repo Gripper Locations for Operation Sequence 2

5.3 Finding Repo Grip Locations

Planning the repo grip locations is similar to planning the robot grip locations, except for the constraints related to interference with the tools. However, it has to satisfy two other constraints: 1) the repo gripper should not interfere with the robot gripper during a part exchange, and 2) the part should not interfere with the machine when held by the repo gripper. When it is necessary for the robot gripper to regrip the part at the repo station, the grasping planner identifies and selects the grip location for the repo gripper according the constraints we have described. Since the early decision of repo grip location affects the robot grip locations for the subsequent bends, we postpone the decision of repo grip location until the next repo is required or the complete operation sequence is tried. This strategy of delaying the selection of a grip location means that the program can pick the location based on the operations to be performed, rather than not knowing which operations will be performed before the next repo.

5.4 Grasping Example

For the part shown in Figure 2, the grasping planner selects a TG-2025-D23 gripper from the gripper library to generate the most greedy solution with the least number of predicted repos. IG-0725-U332 gripper is selected for a repo gripper, because its knuckle is tall enough to grasp over the flanges and it can grasp on a small area of the part.

The bending machine we use has many limits that must be enforced or there will be a collision. For example, Figure 12 shows, (i.e., the gripper is parallel to the bending plane), the gripper cannot get any closer than 125 mm to the bending machine, when the gripper is right-side-up and side loading. When the gripper is upside-down, the limit is even worse (260 mm) because of the way the last robot axis is offset with respect to the gripper. These limits and many others determine what bends can be done given a particular grip location and the bend to be performed.

Operation Sequence 1 (shown in Figure 5) requires one repo after b3. Figure 13 shows two locations of the robot gripper and one location of the repo gripper on the part. For the 1st and 2nd robot grip locations, TG gripper knuckles are placed underneath the part and the 2nd robot grip location grasps the part over the down flange.

For Operation Sequence 2 (shown in Figure 6), a repo is required between b3 and colinear bends b4 and b5 because the distance between them is only 92 mm while the robot constraint for the side loading requires at least 250 mm (twice the limit of 125 mm) between them. Since the bend angles for b7 and b8 are negative, the robot axis has to be in an upside down orientation in order to load into the press-brake. The upside-down limit causes the safe distance to side-load both b9 and b10 without a repo to be at least 520 mm (twice the limit 260 mm for each), while the distance between b7 and b8 in the part is only about 452 mm. Thus, it is necessary to have a repo between b7 and b8. b9 and b10 can be made without repositioning since the distance between them (331 mm) is greater than the necessary safe distance for the side loading (250 mm). Thus, the large limit for upside-down side loading is the cause of the extra repo in this sequence. Figure 14 shows three computed

locations for the robot gripper and two locations for the repo gripper with respect to the part. The knuckle orientation for all robot grip locations are placed below the part. The IG repo-gripper knuckles are placed above the part for these two operations.

6 The Motion Planner

The motion planner performs three main tasks: (1) generating gross motion moves to move part from one location to another location, (2) generating fine motion moves for unloading the part after bending, and (3) back gage planning for locating the part inside the press-brake.

6.1 Gross Motion Planning

Gross motion planning has the task of looking for a near-optimal solution in a 5 DOF system (3 translational and 2 rotational axes). While there are some features of this problem that make it easier, there are also features that make it surprisingly difficult. The machine geometry is largely simple (few obstacles), which makes this a relatively tractable problem. On the other hand, many of the parts that are being manipulated do not fit comfortably within the workspace. The other difficulty arises because the axis limits of the robot are not uniform in the world coordinate system of the machine. For example, it is possible that simply turning the robot's pitch axis 180 degrees would cause a Z axis over-travel.

The goal of the primary problem solved by motion is to compute a series of 5-tuples that allow the robot to move smoothly and without collision or limit violation between critical points (i.e., the start and end points). Simple interpolated values between the computed coordinates must also be valid points. This requires a large number of calls to a collision and limit checking algorithm, which determines if an intermediate point is valid or not.

One of the first steps in gross motion motion planning, is to build a model of the bending machine and its environment. This model is used by the collision checking module and building the model requires compromise. The kinematics of the robot also need to be modelled with a required degree of accuracy. The model is kept geometrically simple to avoid unnecessary checking. But there is no loss in the quality of plan, since minor details of the machine do not offer realistic motion regions.

The collision checking needs to decide correctly and quickly whether a robot path would cause a collision on a real machine. The step size of the collision checking algorithm should be limited as a function of the resolution of the bending machine model. A larger step size may result in no collisions being recognized in the model, while the path may result in a collision between the robot and the real machine.

An A* algorithm is used to search for feasible paths between a start and goal position of the robot. The general strategy is to achieve goal pitch and yaw positions as quickly as possible. The cost of rotations are given higher priority since, these moves require more time which is undesirable. Starting with the initial point, the algorithm expands each node using several intelligent built-in strategies. These strategies have been designed around the robot

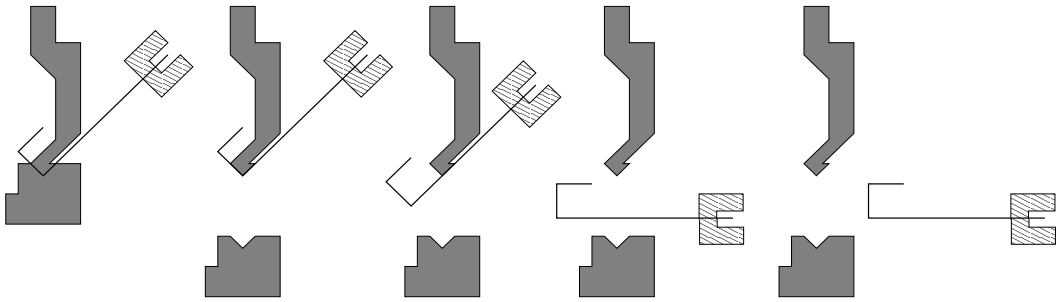


Figure 15: Fine Motion Under Punch Contact

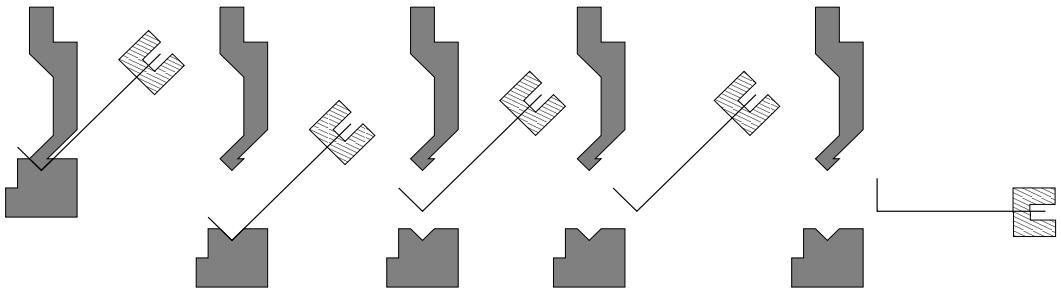


Figure 16: Fine Motion Under Die Contact

and bending machine environment and they help the A* algorithm in advancing towards the goal by quickly identifying the most promising moves. The expansion of a node involves collision checking and points with no collision are saved for future expansion, along with the cost involved in reaching that point and the expected cost of reaching the goal. The A* search progresses in this manner, until it finds the cheapest path from the initial point to the goal, if one exists.

Gross motion planning assists the central operation planner in optimizing the operation sequence to minimize the required motions. For example, flipping a part over can be difficult for either a robot or a person to accomplish. Therefore, it may be desirable to develop a plan that minimizes the required part flips.

6.2 Fine Motion Planning

After a bending operation, it is necessary to unload the part from the machine while the part is being held by the robot gripper. Geometrically, the part model consists of the partially bent-up part and the model of the gripper. The tool model represents the entire tool layout along with the models of the relevant portions of the bending machine.

The fine motion planner first decides if the the part can be allowed to come down with the die-table as the press-brake is opened. Sometimes, the part is wrapped-around the punch tip in such a way that the part cannot come down with the die-table without a collision. In such cases, the fine motion planner decides that the robot should move the part up (+Z) to

compensate for the press-brake's opening. The part continues to remain in contact with the punch-tip after the press-brake is opened and this mode is called "punch-contact" (shown in 15). In other cases, where there is no danger of a collision, the part is allowed to come down with the die-table and this mode is called "die-contact" (shown in 16).

The fine motion planner computes a sequence of moves such that the partially bent-up part can be successfully unloaded from the bending machine, while satisfying the following conditions:

- The part should not collide with any part of the tool during the entire sequence of moves.
- The robot should stay within its kinematic limits during the execution of the fine motion plan.
- The planner should generate a detailed sequence of robot moves which can be executed by the robot.
- The fine motion computation time needs to be kept to a minimum, since fine motion feasibility is tested whenever any node is expanded during the operation sequence search process.

The fine motion path is integrated with the gross motion plan in a seamless manner. The fine motion plan unloads the part and leaves it outside the machine from where the gross motion planner begins its motion sequence. Our approach to the problem of computing a fine motion plan is structured into three tiers of complexity. We describe each of the three tiers of geometric reasoning below.

Simple Geometries: The fine motion planner starts by quickly analyzing the geometry of that portion of the part which is inside the machine. Let us refer to this portion as the inside-part. Our analysis is used to rapidly compute the geometric dimensions of the inside-part along the X, Y and Z axes. This information is compared with the tool model parameters such as the current die opening and the shape and length of the tool.

If it turns out that the geometry of the inside-part is such that the part can be unloaded in a straight-forward manner, then we recognize this fine motion problem as a "simple" case. In such a case, we do not explicitly search for an unloading path but use the geometric analysis to enable us to automatically generate a valid fine motion path. This path is then translated into a sequence of valid robot moves, after inserting adequate safety margins into the moves. These safety margins are user-definable through a database file.

We have found that a large percentage of all fine motion queries are trapped at the first tier itself. Since the computation time at this level is very low (under 1s.), this improves the overall performance of system.

Known Strategies: The next tier of fine motion planning involves a more detailed analysis of the inside-part. This analysis recognizes the existence of up-flanges, down-flanges and their sizes. This geometric data is again compared with the geometric parameters of the tool model and the result of this comparison is used to select one out of a handful of intelligent unloading moves. These known moves are stored in the fine motion planner and have been computed by us based on a detailed study of the interaction between tool geometries and complex part geometries.

This tier of reasoning sometimes needs to carry out minor modifications to the basic motion plan which is selected. The feasibility of these modifications are tested by performing intersection tests between the tool model and appropriately-displaced part models. The use of these geometric boolean operations result in higher computation times at the second tier when compared to the first tier. These times have been found to be under a few seconds, and have been acceptable in our planning system.

A* search: The first two tiers of reasoning have successfully dealt with all of the fine motion planning problems, in our testing on over two hundred parts. But, we have added a last tier of reasoning for the sake of completeness of our algorithm. The last tier uses a form of A* search to exhaustively search for a path if one exists. This search is done by incrementally displacing the part along different axes, testing for collisions and by trying to steer the part towards the outside of the machine. This tier provides a form of completeness to the fine motion planner, in the sense that it will find a path which may not fall into any of the simple paths, nor into any of the known paths. At the same time, we have noted that all test parts have been successfully unloaded using one of the first two tiers of reasoning, and that some complex path found by the A* search may not be practical from a application view-point.

6.3 Gage Motion Planning

In our bending workstation, gaging is an essential component to achieve a specified bend accuracy. In each bend, a bend angle and a flange height need to be within the design specifications for the assembly and the finished product. While the bend angle is achieved with proper punch and die, the accuracy of flange height in each bend is achieved with the aid of gages. The press brake is equipped with two gages. We refer to gaging for the flange height as Y gaging and refer to gaging for bend line position along the die as X gaging (see Figure17).

In manual sheet metal bending, a human operator determines the gage positions for each bend so that the specified flange height can be achieved when the part edges parallel to the bend line touches the gages at the same time. A bend deduction compensation formula based on the part thickness and the bend angle is used to determine the accurate gage positions in Y. Once the production begins, the operator performs both Y gaging and X gaging for each bend simultaneously by properly positioning the part on the required die and pushing it against the two gages. This could be a tedious repetitive operation for a large lot size.

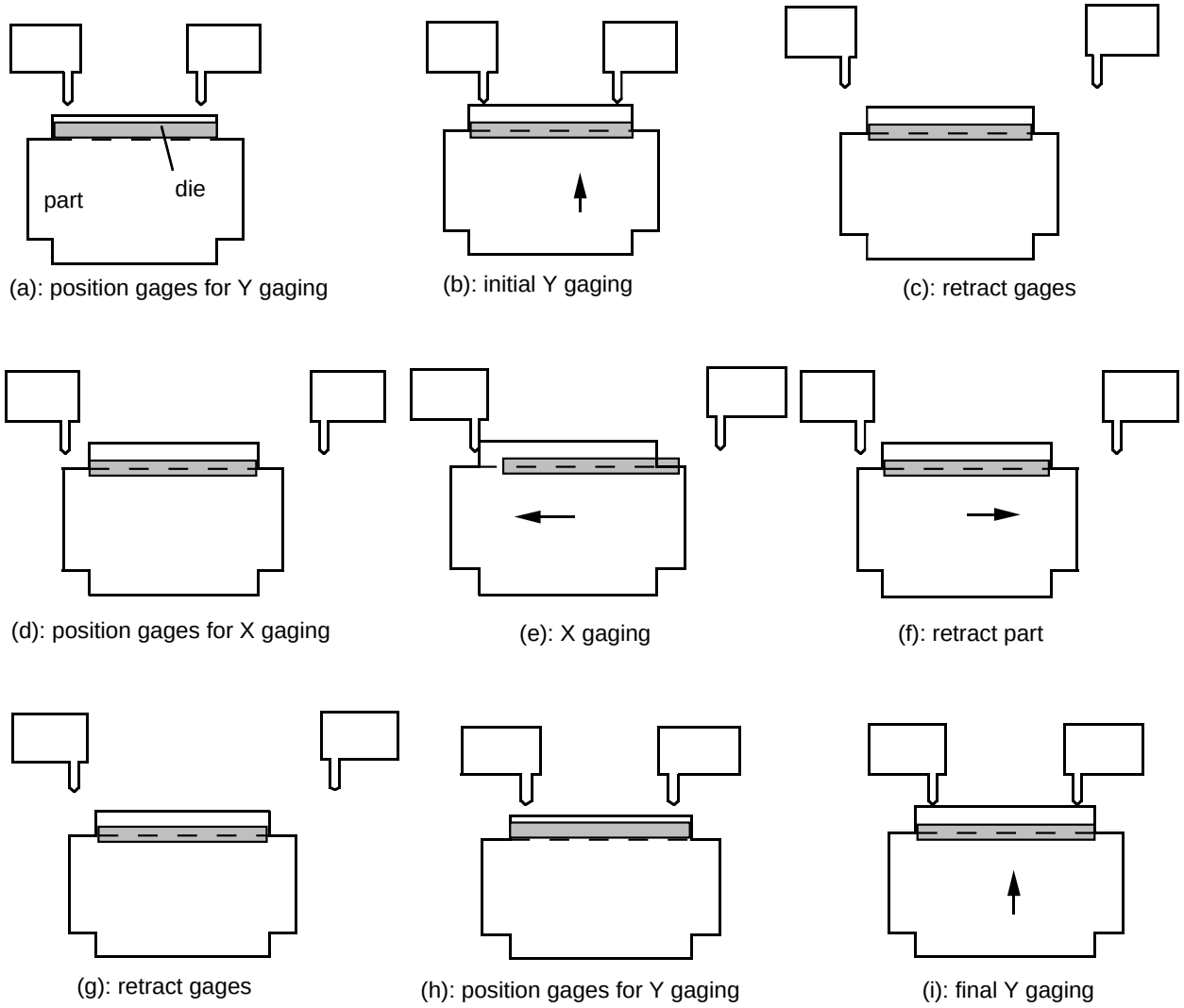


Figure 17: An Example of Gaging

In the automated sheet metal bending cell, the robot is used to handle the part and needs to accurately place each bend of the part on the die with the correct depth. Due to the uncertainty in robot kinematics, some errors in position are unavoidable while positioning the bend line on the die. In order to eliminate these errors, an initial Y gaging, a X gaging, and a final Y gaging are used for every bend.

While the Y gaging requires two gages, the X gaging is done with one gage by having a contact between the gage and a part edge perpendicular to the bend line. During the X gaging, the part may be rotated around the gripper pads due to the contact force between the part and the gage. Thus, a final Y gaging is used to correct any error caused during the X gaging. While the initial Y gaging requires a large approach value for the part, the final Y gaging can be done faster with a smaller approach value.

The gaging planner automatically determines gage positions for each gaging operation and all of the required gage movement. The gages are positioned at the theoretically correct locations before each bend according to the gaging sequence. Then the robot moves the part until gaging is complete and correctly positions the part on the die according to the gaging result.

Our gaging algorithm uses the following steps in identifying the gaging location for every bend:

- Identify all of the possible gaging locations on the part for the Y gaging.
- Select the best gaging location with the least error estimation.
- Identify all of the possible gaging locations on the part for the X gaging.
- Select the best X gaging location with the least travel from the initial Y gaging positions.
- Generate intermediate points for gage movements from the initial Y gaging, the X gaging, and the final Y gaging.

For Y gaging, each colinear line parallel to the bend line is tried to see if two gages can be placed against the part. When there is more than one gaging location possible, a positioning error for each candidate is estimated. For X gaging, each colinear line perpendicular to the bend line is tried against one gage. When there is more than one candidate gaging location possible, the one that needs the least travel from the initial Y gaging locations is selected. The gaging mechanism only allows both gages to simultaneously move in Y. Thus, while the selected gage is used for X gaging, the other gage's X position should be verified to be clear from the part so that it does not interfere with the part.

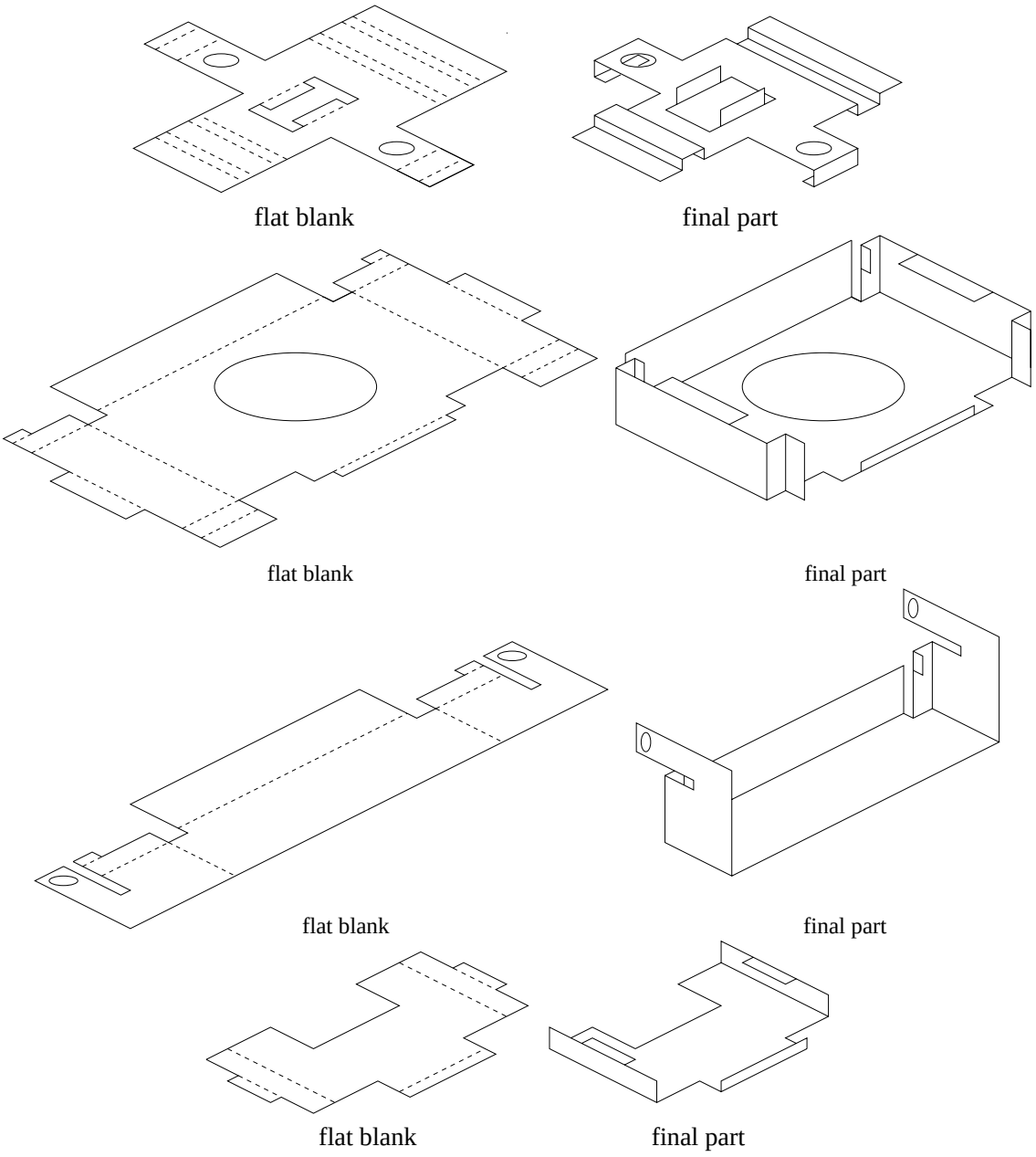


Figure 18: Representative Parts

7 Discussion and Conclusions

7.1 Implementation

Our system has been implemented using the C++ programming language [26]. For geometric modeling and reasoning we have used NOODLES geometric kernel [14]. For graphical interface we have used HOOPS graphics library [18]. All message passing among planners is accomplished by Feature Exchange Language [5].

On average parts (8 to 12 bends), our system produces process plans in less than 5 minutes on a 200 MHz Pentium workstation. Our system has been tested with over two hundred of low to high complexity industrial parts. Figure 18 shows a few representative parts.

7.2 Summary

This paper describes a completely automated process planning system for a robotic sheet metal bending press-brake. Given a CAD file for the desired part, our system generates the instructions for driving all components of the robotic press-brake, which includes a material handling robot, a gaging system, and an NC press-brake. Our system is based on a generative approach and employs a distributed planning architecture. Our system architecture offers the following advantages:

- The central planner performs macro planning. Specialized planners perform micro planning. Communications between the central planner and specialized planners allow us to tightly integrate macro and micro planning functions.
- Specialized knowledge is encapsulated into modular planners. Each specialized planner uses the representation most efficient for its planning activity and employs the most efficient problem solving technique.
- The modular nature of the architecture also makes this system easy to update. For example, each specialized planner can be be upgraded without changing the rest of the system.

Planning complex tasks is often difficult, because there are several objective functions to jointly satisfy. There are inherently conflicting requirements between material handling and material processing components. In order to ensure a good grasp, the part needs to be held such that a large portion of it is covered by the grasping (or holding) device. On the other hand, in order to ensure proper accessibility conditions for material processing a large portion of it should be exposed. We have resolved such problems by sharing constraints between planners. While it is in general impossible to find a complete and correct set of constraints without exhausting the search space (a priori), it is possible to develop a partial list of correct constraints, and based on practice this list can be expanded on an as needed basis. The most useful application of these constraints can be used in the predictive stage of

planning: before the search begins. For example, if grasping has a list of tooling constraints then it can make its repo predictions to be consistent with tooling and the reverse is true for tooling. With this approach, each system can work with the constraints of all of the other systems.

7.3 Limitations

Currently, our system has the following three main limitations:

- We have used our system successfully for parts upto 23 bends. In order to handle a wide range of complex parts (e.g., part with with more than 20 bends), we will need to develop better prediction heuristics and search algorithms to solve the process planning problem efficiently.
- Currently, we do not explicitly account for part tolerances in the evaluation of operation sequence. We plan to make use of the specified part tolerances in the selection of the operation sequence.
- Our system currently does not make recommendations for tool redesign or rework. In practice, sometimes none of the existing tools can be used for a given part. In such situations, human process planners quite often recommend grinding an existing tool to eliminate part-tool interference. We are planning to add capabilities for suggesting tool redesign.

Acknowledgements Dr. Gupta's participation in this project was supported in part by Amada America Inc. and National Science Foundation Grant DMI-9713781. Drs. Bourne's, Kim's, and Krishnan's participation in this project was supported by 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] L. Alting and H. Zhang. Computer aided process planning: The state of the art survey. *Int. J. of Prod. Res*, 27(4):553–585, 1989.

[2] Steven Ashley. Intelligent sheet metal bending. *Mechanical Engineering*, 119(1), January 1997.

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

[4] Steve D. Benson. *Press Brake Technology: A Guide to precision sheet metal bending*. Society of Manufacturing Engineers, 1997.

[5] D.A. Bourne, J. Baird, P. Erion, and D.T Williams. The operational feature exchange language. Technical Report CMU-RI-TR-90-06, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1990.

[6] David Bourne. Intelligent manufacturing workstations. In *Knowledge-Based Automation of Processes, ASME Winter Annual Meeting*, Anaheim, CA, 1992.

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

[8] 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.

[9] M. R. Cutkosky and J. M. Tenenbaum. Toward a framework for concurrent design. *International Journal of Systems Automation: Research and Applications*, 1(3):239–261, 1992.

[10] M.R. Cutkosky, R.S. Engelmore, R.E. Fikes, M.R. Genesereth, T.R. Gruber, W.S. Mark, J.M. Tenenbaum, and J.C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, 26(1):28–37, January 1993.

[11] 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.

[12] 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.

[13] S.K. Gupta, D.S. Nau, W.C. Regli, and G. Zhang. A methodology for systematic generation and evaluation of alternative operation plans. In Jami J. Shah, Martti Mantyla, and Dana S. Nau, editors, *Advances in Feature Based Manufacturing*, pages 161–184. Elsevier Science Publishers, 1994.

[14] Levent Gursoz. *Reference Manual for the NOODLES Library.* Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA, 1991.

[15] I. Ham and S. Lu. Computer aided process planning, the present and the future. *Annals of CIRP*, 37(2), 1988.

[16] C. C. Hayes and P. Wright. Automatic process planning: using feature interaction to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[17] Caroline Hayes. A manufacturing process planner for a concurrent engineering environment. In *IEEE International Symposium on Assembly and Task Planning*, 1995.

[18] Ithaca Software, Alameda, CA. *HOOPS© Graphics System Reference Manual*, 1992. Release 3.20.

[19] D. S. Nau. Automated process planning using hierarchical abstraction. *TI Technical Journal*, pages 39–46, Winter 1987.

[20] Nils Nilsson. *Principles of Artificial Intelligence*. Morgan Kaufmann Publishers Incorporated, CA, 1980.

[21] 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.

[22] 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.

[23] S.E. Sarma and P.K. Wright. Algorithms for the minimization of setups and tool changes in simply fixturable components in milling. *Journal of Manufacturing Systems*, 15(2), 1996.

[24] 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.

[25] D.N. Sormaz and B. Khoshnevis. Process sequencing and process clustering in process planning using state space search. *Journal of Intelligent Manufacturing*, 7:189–200, 1996.

[26] Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, second edition, 1991.

[27] H.P. Wang and R.A. Wysk. A knowledge-based approach for automated process planning. *International Journal of Production Research*, 26(6):999–1014, 1988.

[28] 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.

[29] 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.