Manufacturability-Driven Decomposition of Sheet Metal Products

Cheng-Hua Wang

September, 1997

CMU-RI-TR-35-97

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Designers often need to decompose a product into functioning parts during the product design stage. This decomposition is critical for product development, as it determines the geometric configuration of parts, and has direct impact on product cost. Most decomposition decisions are based primarily upon end-user requirements instead of product manufacturability. The resulting parts can be expensive to manufacture or are sometimes impossible to make. This thesis presents a manufacturability-driven approach which can help designers decompose bent sheet metal products into manufacturable parts.

The decomposition approach presented in this thesis takes the geometric description of an initial product design, analyzes its manufacturability, and decomposes the product into manufacturable parts. The decomposition continues until all decomposed parts are manufacturable. Near-optimal solutions are generated based on some primary concerns of *design for manufacture* (DFM) and *design for assembly* (DFA). Designers can then examine the decomposition results and decide whether they meet end-user requirements. Cutting, bending, and assembly processes are considered as the major manufacturing processes in bent sheet metal production. Key manufacturability analyses for these processes are based upon *part unfoldability*, *tool accessibility,* and *product assemblability.* The final results are sent to the corresponding process planning systems and a complete production plan is generated.

A prototype system, consisting of design and decomposition modules, as well as process planners for cutting, bending and assembly, has been implemented to validate the decomposition approach. A range of geometric and industrial products have been tested in the system and the results are given. We show that as the approximations of these process planners are improved, the results will also improve, without any changes to the decomposition approach.

Acknowledgments

I would like to thank my advisor, Professor David Bourne, for his guidance and advice through my study. He helped me to focus on key issues and ensured that I was always on the right track. It was a great pleasure to work with David.

I thank the other members of my committee--Professor Michael Erdmann, Professor Susan Finger, Professor Robert Sturges, and Professor Mark Cutkosky (Stanford University), for serving on the committee, for their encouragements, and for their many useful comments and discussions. I also want to thank Professor Kenji Shimada for teaching me in-depth knowledge of solid modeling and computational geometry to improve my thesis work.

Working with the people in the Rapid Manufacturing Laboratory has been pleasant and very helpful in my work. I am especially grateful to Dr. Satyandra Gupta and Duane Williams for their advice and discussions. I also want to thank Champ Knecht for proof-reading my thesis draft.

During my Master and Ph.D. study, many friends provided support and advice in my academic study and personal life. I want to thank Dr. Weileun Fang, Dr. Jung-Je Tseng, Dr. Ssukuei Wang, Dwight Tseng-Min Lin, Ming-Shen Hsu, Korntham Sathirakul, and Venkateswara Ayyadevara for making my study and life more enjoyable.

Special thanks must go to my parents who taught me the importance of education and supported me all the time. I am grateful to my sister Carol for taking good care of my parents when I was away. This dissertation would not be possible without the support of my wife Jia-Ying and my daughter Jenny. They were always there for me even when I was down.

Last but not least, I would like to thank my junior-high teacher Mr. Chi-Hsiang Tsai, who encouraged me on to higher education and made this thesis possible.

This thesis work is supported by the Robotics Institute and Amada Inc. In particular, I would like to thank Mr. Ken Hazama of AmadaSoft America for providing experience and expertise in sheet metal production.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Industry is always looking for ways to reduce the cost of manufacturing products, often by introducing the use of automation or implementing state-of-the-art production methods. Many analyses of product costs in the development cycle indicate that about 70% of the total product cost has been committed by the time the design is completed, while less than 10% of this cost is spent during the design stage (Andreason et al. 1983). As a result, bringing in manufacturing concerns during the design stage addresses the bulk of a product's cost. Since product/part manufacturability requires in-depth manufacturing knowledge and sometimes is too complex to evaluate, incorporating manufacturing considerations can be difficult.

During the product design stage, designers usually need to decompose a product into functioning parts. Decomposition of a product directly determines its geometric configuration and has significant impact on product cost. Because the decomposition task is driven by the end-user requirements of the product, the manufacturability issue is often not addressed. The resulting design may be expensive, or sometimes impossible to manufacture.

1.1 Product Decomposition

A product is a set of parts connected by assembly fasteners (Dixon and Poli 1995). An early stage of product design is to decompose a product into parts and subassemblies that meet product function and producibility concerns (Nevins et al. 1989). This task is shown in Figure 1.1. Traditionally, the decomposition is primarily function-driven (Figure 1.2). However, the design also needs to satisfy the constraints imposed by downstream manufacturing processes.

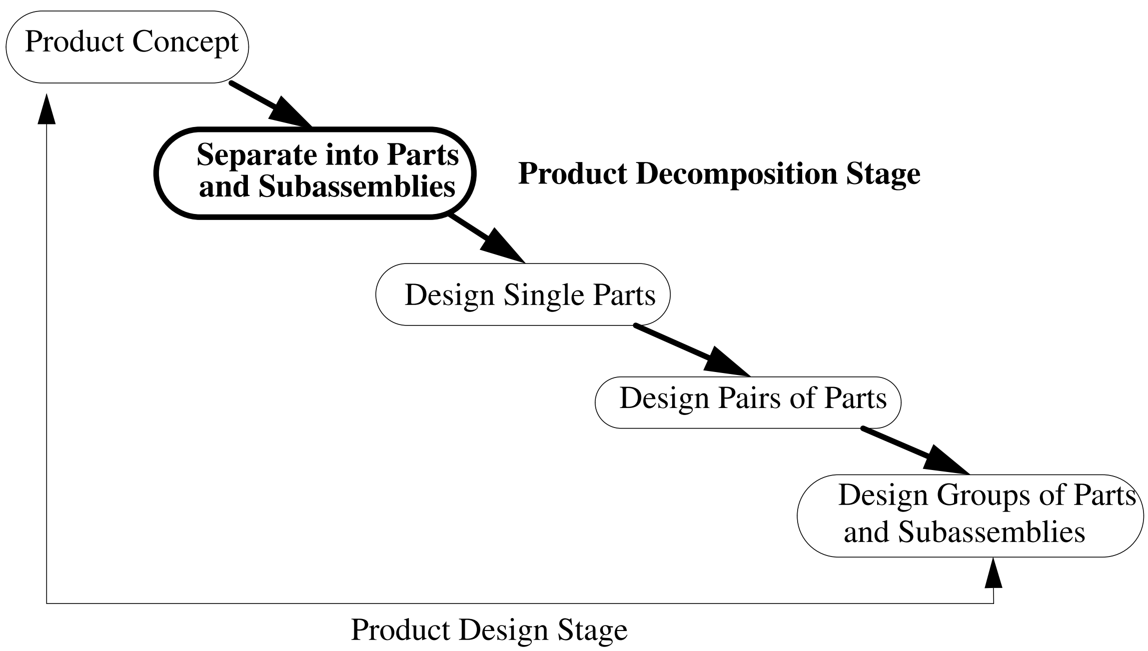


Figure 1.1: Product decomposition in product design (Nevins et al. 1989).

Product decomposition is highly complex due to its combinatorial nature. There may be many ways to decompose a product to meet functional needs, as the mapping of function-to-form is usually not one-to-one. However, there may be only a few decompositions that are manufacturable. Current decomposition is done mostly by using previous experience with similar designs. This approach provides limited decomposition alternatives which might not be optimal with respect to manufacturing concerns. It is also sensitive to the capabilities of the manufacturing facilities used to fabricate the products. When it comes to designing new products, this approach might not be applicable at all. Therefore, a trial-and-error approach usually has to be used and the product development cost inevitably increases.

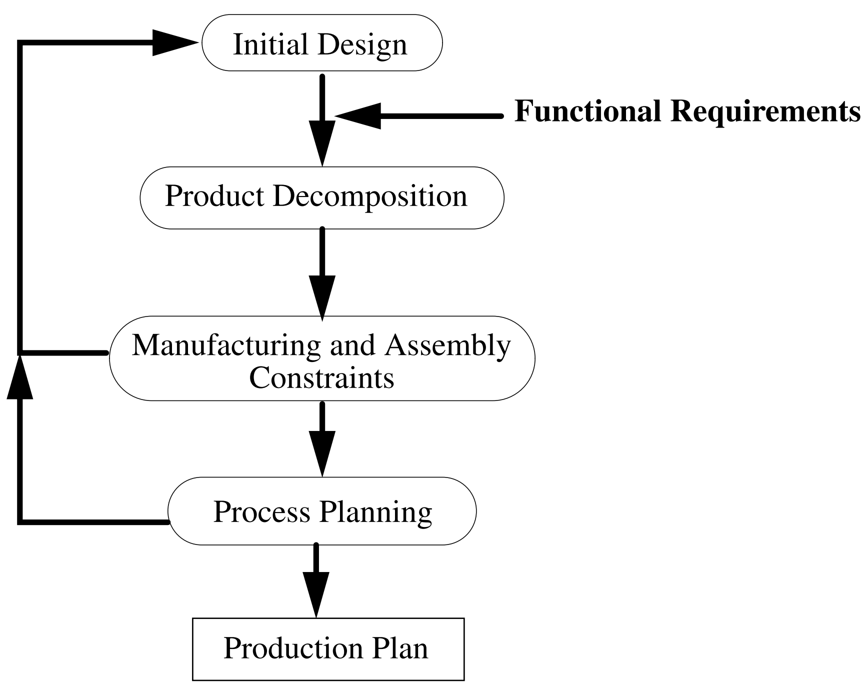


Figure 1.2: Traditional product decomposition approach.

1.2 Motivation

The motivations of this thesis are the following:

- **Products need to be decomposed in order to be fabricated.** For instance, Figure 1.3 shows two sheet metal products that cannot be unfolded into a single piece without decomposition. Another example is the initial design of a sheet metal electronic enclosure shown in Figure 1.4. The product cannot be made using a bending machine, because some of the bends cannot be reached by the bending tools. As a result, further decomposition is necessary. Four possible decompositions are also shown in Figure 1.4. Note that decompositions (a) and (b) are still not accessible by the tools. This indicates that even after a product is decomposed, it may still not be manufacturable.
- **Appropriate decomposition can reduce product cost significantly.** This is especially true when the batch sizes of products are large. For instance, a decomposition with fewer parts usually has a lower manufacturing cost than one with more parts.
- **Small changes in manufacturing facilities and product design may change the decompositions.** It is a common industrial practice to adjust the manufacturing facilities or to make changes to the product design. These changes may have a strong impact on the way the product should be decomposed. Figure 1.5 shows how different bending tools can affect the decompositions. If the gooseneck punch is available, no decomposition is needed. In Figure 1.6, two similar designs (with different heights) result in different decompositions with respect to the bending process. The decomposition of the second design is necessary due to inaccessibility of the bending tools.

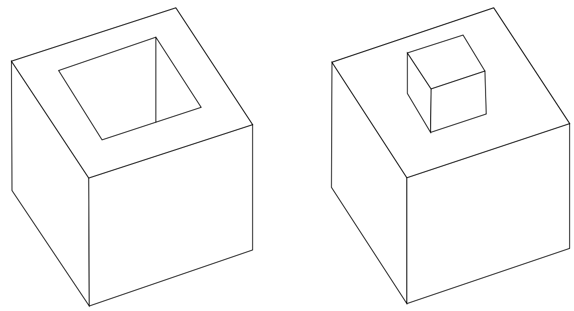


Figure 1.3: Products that cannot be unfolded without overlapping.

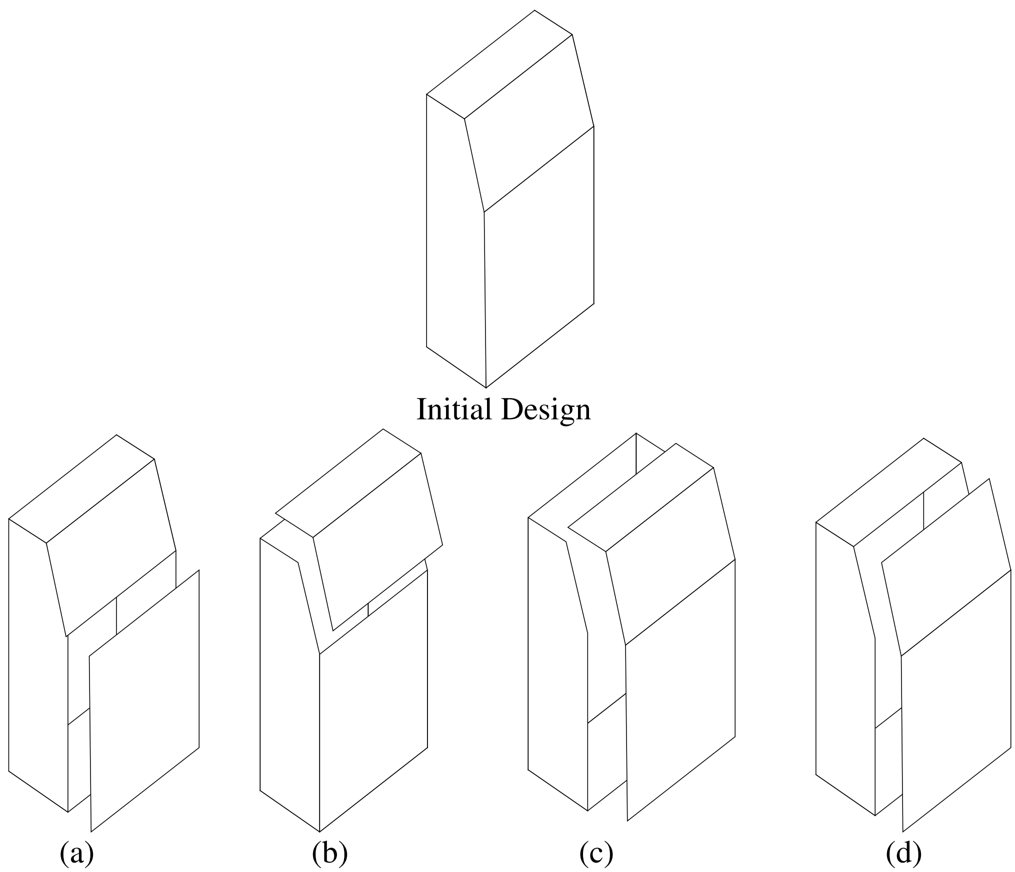


Figure 1.4: Different decompositions of an electronic enclosure.

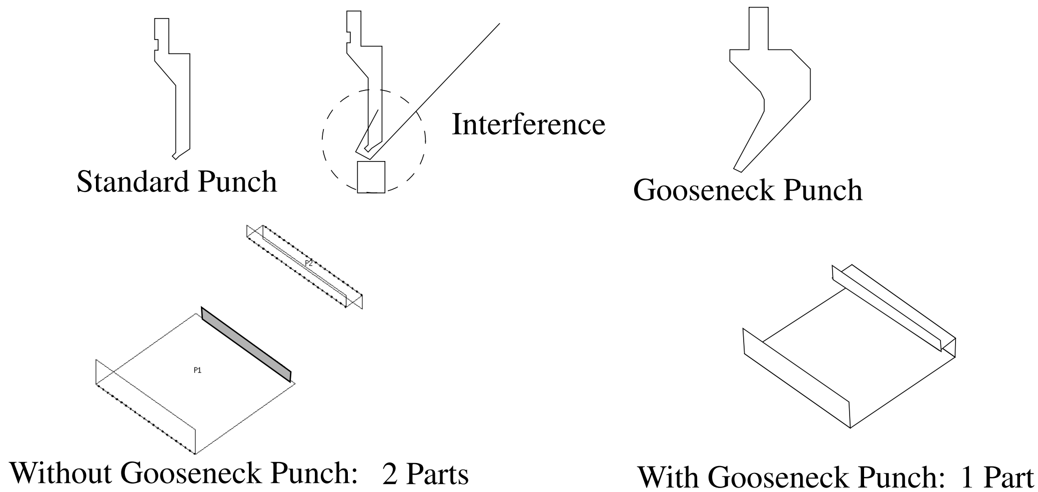


Figure 1.5: Manufacturing facility and decomposition.

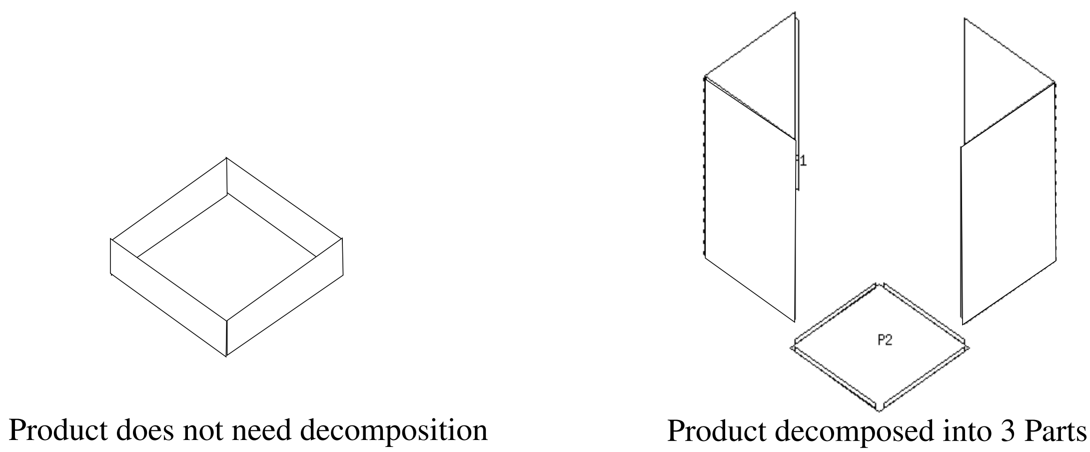


Figure 1.6: Similar product designs and decomposition.

1.3 Thesis Approach

This thesis describes a manufacturability-driven decomposition approach to help designers decompose bent sheet metal products into manufacturable parts. The thesis focuses on the cutting, bending, and assembly processes of the production of bent sheet metal products. This approach is primarily based on manufacturability related to product geometry. We develop a prototype system based on this approach. The input to the system is an initial geometric representation of a bent sheet metal product. The decomposition starts with the manufacturability analysis of the product and decomposes the product if it is not manufacturable. The process is recursive until all decomposed parts are manufacturable. To evaluate the complete manufacturability of each part would often be time-consuming and impractical; as a result, only primary manufacturing concerns are considered as feasibility checkers during the decomposition. The design of the parts may need to be modified to allow for assembly operations. For instance, additional assembly tabs are necessary to provide space for spot welding. The resulting decomposition has to be evaluated by process planners to ensure its manufacturability. The outputs are the 3D/2D geometric representation and the production plan of each resulting part. This decomposition process is fully automatic and the designers can provide additional information to guide the decomposition.

1.3.1 Requirements of Product Decomposition

Our decomposition approach was engineered to address the following issues:

- **Functionality.** Decompositions must satisfy the functional specifications of the product. When designing a mechanical part, a designer transforms an abstract functional description of an artifact into a physical description that satisfies the functional requirements. However, this function-form transformation is not well understood (Flemming et al. 1992).
- **Manufacturability**. This issue is usually not addressed during traditional product design. Our approach fits the current design paradigm in which manufacturing constraints are considered while designing the part to ensure success.
- **Complexity.** In order to arrive at optimal or near-optimal decomposition solutions, we need to be able to generate many, if not all, possible decompositions. However, the optimal decomposition problem is basically intractable due to its combinatorial nature. To address this complexity issue, certain simplifying assumptions must be made. For instance, to decompose a bent sheet metal product like the electrical enclosure shown in Figure 1.4, components are split along connecting edges between faces without splitting faces. This maintains the rigidity and original appearance of the product. It also makes the decomposed parts easier to bend and assemble. Most importantly, decompositions of bent sheet metal products can be easily modeled as a graph problem and efficient algorithms can be applied. Unfortunately, even though some decomposition simplifications are made, exponential explosion problems still cannot be avoided. For example, even if the product in Figure 1.4 is decomposed based upon the above simplifications, there are still 52 possible binary decompositions.
- **Cost**. Product cost is always a dominant factor in product development and thus we try

to achieve a decomposition which reduces product cost.

- **Flexibility**. The ability to easily adapt to different manufacturing facilities and machine capabilities is critical to survive in the highly competitive global market. Decompositions should be flexible enough so that currently-available manufacturing facilities can be fully utilized.
- **Autonomy**. The decomposition approach should be able to generate feasible decompositions and evaluate their manufacturability automatically. It requires minimal effort from the designer. In fact, only questions related to product functional requirements need to be asked.
- **Efficiency**. The approach needs to arrive at decomposition results in a reasonable amount of time. To achieve this goal, first the generation of feasible decompositions needs to be tractable. Then efficient algorithms have to be developed to evaluate product manufacturability and assemblability. This also includes looking for the geometric and manufacturing constraints of the manufacturing processes to speed up the evaluation process.
- **Design Feedback**. After the decomposition is performed, the decomposed product has to be assembled. To allow for assembly operations, the original design may need to be modified. The decomposition approach should provide feedback to the designer so that necessary design modifications can be made.

A satisfactory decomposition is a decomposition which meets all end-user requirements. The goal of my decomposition approach is to achieve a satisfactory decomposition with the least cost.

1.4 Thesis Contributions

This thesis describes the following progress toward the goal of efficient product decomposition of bent sheet metal products:

1. A systematic and tractable decomposition approach driven by the manufacturability constraints of sheet metal products
2. A decomposition approach which adapts to sheet metal design and facility changes
3. A formalization of the modeling of cutting, bending and assembly processes for sheet metal products/parts and the development of efficient algorithms for manufacturability analyses
4. An approach to flattening 3D parts into 2D flat patterns (pattern development) considering geometric and topological unfoldability and compactness
5. Automatic design modification (assignment of assembly tabs) of the product after decomposition for bent sheet metal products
6. A prototype decomposition system for bent sheet metal products

1.5 Previous Work

This section is an overview of some selected work related to product decomposition. It addresses the following topics: product design, product decomposition, manufacturability analysis, computer aided process planning, and concurrent engineering. Additional research is described in later chapters as appropriate.

1.5.1 Product Design

Engineering design is a highly iterative process which involves the following phases: product planning and clarifying of the task, conceptual design, embodiment (preliminary) design and detail design (Pahl and Beitz 1996). Flemming et al. (1992) discuss the function-form relationship in engineering design. They show that several strategies are usually used in engineering design: top-down (refinement), bottom-up (constructive) and middle-out design. Each of these strategies can be driven by either product function or product form. It is understood that functional specifications normally do not map uniquely into component configurations. Thus, the number of alternatives is usually very large and a standard search approach is difficult to employ.

A current trend in product design is to consider concerns other than product functionality early in the design stage. These considerations result in concepts like *design for manufacture* (DFM) and *design for assembly* (DFA). Boothroyd et al. (1994) show that DFMA (DFM and DFA) provides a systematic procedure for analyzing a proposed design from manufacture and assembly points of view. This procedure results in simpler and more reliable products, which are less expensive to fabricate. It also encourages teamwork between designers and manufacturing engineers, which can serve as a foundation for further concurrent design implementations.

1.5.2 Product Decomposition

The published literature on systematic product decomposition is not extensive. Kusiak et al. (1991) propose a knowledge-based approach to decomposing design specifications with an AND/OR tree. A number of alternative conceptual design solutions are found based on a set of production rules. In die design, it is often necessary to build dies with complicated shapes or sharp corners from a number of different pieces or sections (Stanley 1919). Such dies have several advantages; not only is the manufacturing process considerably simplified, but replacement is much easier in case of wear or breakage. Elber (1995) proposes an algorithm to decompose a free-form surface using a set of developable ruled surfaces and then assemble them into paper models. Chung and Huang (1996) present an approach to partitioning sheet metal products into unfoldable patterns based on depression and protrusion features. Unfortunately, unfoldability is the only concern of their work and the manufacturability and the assemblability of the products are not addressed.

The “decompose-and-assemble” (or divide-and-conquer) concept has been used in fields other than product development. For instance, the finite element method (FEM) divides complicated shapes into many basic elements to simplify the analysis of stress, temperature and pressure (Zienkiewicz and Morgan 1983), and the binary search algorithm recursively divides ordered problems into halves and reduces the computational complexity of the original problem (Aho et al. 1974).

1.5.3 Manufacturability Analysis

The ability to evaluate the manufacturability of a proposed design in the design stage is an essence of concurrent design. It can eliminate early conflicts between design and manufacturing processes so that manufacturing costs will not go up and re-design can be reduced. Most manufacturability analyses are highly constrained by geometry. A detailed survey of automated manufacturability analysis can be found in Gupta et al. (1997). Gupta and Nau

(1995) use machining features to analyze the manufacturability of machined parts. Their results can be used as design feedback to speed up product development. Tseng and Joshi (1994) describe an algorithm to decompose machined volumes of prismatic parts and then they reconstruct the features for multiple interpretations of interacting features. Bourne and Wang (1995) show that features related to tool selection can be used to resolve some manufacturing problems in an automatic bending system. Radhakrishnan et al. (1996) develop a design rule checker for sheet metal parts. An algorithm based on medial axis transformation is used to find the minimum distance information. Their system then uses the medial axis information and a stamping design rule database to ensure the design is valid.

The major problem of using features in process planning is that features can interact in negative ways. One feature may suggest process-A and another feature, process-B. Hayes (1987) uses production rules to resolve negative interactions and Nau et al. (1989) use feature algebras to derive alternative interpretations of features to avoid the interactions. Wang and Bourne (1995) justify the use of features and their corresponding constraints to avoid negative interaction for bent sheet metal parts.

The ability to evaluate manufacturability can further provide design feedback to simplify process planning and reduce product development cost. Gupta et al. (1994) propose an architecture to integrate Design for Manufacture and CAD through some design critiquing systems. These systems provide advice about ways to improve the design. Lee et al. (1993) develop an integrated system for assembly planning and redesign of mechanical products. This system is based on Design for Assembly (DFA) principles. The redesign plan will be generated if the new design could reduce assembly cost.

1.5.4 Computer-Aided Process Planning

Bedworth points out that the essence of concurrent engineering is the integration of product design and process planning into one common activity (Bedworth et al. 1991). Process planning is the key link between design and manufacture. It starts with the product description, the customer's requirements, and company manufacturing guidelines. It involves the preparation of a plan that outlines the routes, manufacturing operations, and sequence, as well as the fixtures and machine tools (Kusiak 1990).

Most process planning systems can be divided into either variant or generative systems. The variant approach is primarily based on part similarity. It assumes similar parts will have similar manufacturable features, and therefore similar plans. Parts are first classified based on a number of features and coded using a coding system. The code and its associated process plan are stored in a database. When a process plan for a new part is required, the process plan for a part similar to the new part is retrieved from the database. The retrieved plan is modified if necessary. This approach depends heavily upon similarity between parts. In addition, the process plan should not be sensitive to small changes of part feature or geometry. The generative approach, on the other hand, generates the process plan from basic manufacturing knowledge and requirements. Typically Artificial Intelligence (AI) techniques such as search algorithms or expert systems are used to derive the plan. The generative approach is more flexible than the variant approach which makes it a better candidate for automated process planning, especially when similar parts do not have similar plans.

Many automated process planning systems have been developed for various purposes. A detailed survey can be found in Alting and Zhang (1989).

1.5.5 Concurrent Engineering

Concurrent Engineering aims to integrate downstream concerns as early as possible in the design stage. Cutkosky and Tenenbaum (1992) develop NEXT-Cut: a concurrent engineering system for the design and manufacture of machined parts. The designers use a design-with-machined-features approach to create a design. This design approach is similar to the way a machined part is made. NEXT-Cut uses a knowledge-based approach to analyze the design's manufacturability; however, it requires that the designer have good machining knowledge. Finger et al. (1990) develop a computer-based design system called *Design Fusion*, which assists in the design of mechanical parts. Computer programs act like experts and advisors to provide feedback so that the designers can consider the interactions and trade-offs among different requirements simultaneously. Darr and Birmingham (1993) propose an automated-synthesis approach to automate computer design. This approach first generates many possible design solutions. The feasible designs are then obtained by simultaneously applying constraints and preferences. Toye et al. (1994) further extend a concurrent engineering environment called *SHARE* to the internet. This testbed focuses on how information technologies can help design teams to build a "shared understanding" of the design and design process.

1.6 Thesis Outline

The rest of the thesis is organized as follows:

In Chapter 2, an introduction to sheet metal product design and production is given. A new design approach which employs concurrent multiple representations of the part and the process design is used. A sheet metal product design system has been developed based on this representation scheme and used in the thesis.

Chapter 3 describes a manufacturability-driven product decomposition approach for bent sheet metal products. Products are decomposed if they are not manufacturable or too expensive to fabricate. Representations of sheet metal products/parts and their decompositions are presented. Cutting, bending, and assembly processes are chosen as the major manufacturing processes for sheet metal part production. Manufacturability evaluations of decomposed products are based upon part unfoldability, tool accessibility, and product assemblability. I show a develop-first, decompose-later strategy that makes the combinatorial decomposition problem tractable. This strategy generates its manufacturable decompositions based on the evaluation of manufacturability. Near-optimal decompositions are then produced according to DFM and DFA guidelines. The resulting decomposition is sent to the cutting, bending, and assembly planning system and the production plan of the product is generated.

A prototype software system has been implemented to illustrate this decomposition approach. Chapter 4 describes the components of this prototype system: a product designer, a decomposition module, a cutting planner, a bending planner, and an assembly planner. The system takes a geometric representation of a sheet metal product and decomposes it into manufacturable parts. The final result includes the 2D/3D designs of the resulting parts and the production plans for the decomposed product.

The next three chapters describe methods that are used in evaluating the manufacturability of the decomposed product in cutting, bending and assembly processes. As the evaluation of complete manufacturability is time-consuming and impractical, part unfoldability, tool accessibility, and product assemblability are employed as feasibility checkers for manufacturability. Chapter 5 discusses a pattern development approach to flattening 3D part designs into 2D flat patterns. Geometric and topological unfoldability are both taken into account during pattern development. A compactness measure is used to select the most compact pattern for parts with multiple possible patterns.

In Chapter 6, tool accessibility is used to determine part bendability. A backward (unfolding) planning approach is employed to determine the accessibility of the tools for a given product. Tools (punches and dies) are coded with geometric information in a tool database which is easily maintained. Inaccessible components are identified and necessary decompositions are inferred. A cache implementation to store computational results is used to speed up the analysis.

Chapter 7 explains an assembly planning approach which utilizes the local contact information derived from decomposition results. The assembly process considers translational motions only. The final assembly plan consists of the assembly sequence and directions. After a product is decomposed, additional assembly tabs are added to provide space for assembly operations. An approach to the automatic assignment of assembly tabs is also given.

Decomposition results and extensive discussion of the results are given in Chapter 8.

The decomposition results of both unfoldable and non-unfoldable products are shown. Cutting, bending, and assembly plans for each decomposition are also given. Critical issues related to decomposition are explained in detail.

Finally, in Chapter 9, I conclude my thesis with limitations, contributions and future work. The potential applications of the proposed decomposition approach are also discussed.

Chapter 2

Contemporary Design and Production of Sheet Metal Products

There are many products made with sheet metal parts. Typical products include communication equipment, kitchen appliances, and medical instruments. With recent progress in machine capability and process planning software, high precision sheet metal production has become possible. This chapter describes the current practice of sheet metal design and production.

2.1 Design of Sheet Metal Products

Currently, sheet metal part design typically proceeds as follows: in a CAD environment, a designer interprets the functional and geometric constraints of a part and creates a form representation with appropriate line types, group, and layer information; different line types are used to distinguish boundary lines and bend lines; and different layers are used to separate boundary and bend lines. Internal tabs and other lines e.g. holes are ignored during unfolding. This model is then passed to the manufacturing engineer, who uses additional software to specify material properties, invoke process rules for bend allowances, and apply unfolding algorithms that lead to a 2D flat pattern (Figure 2.1).

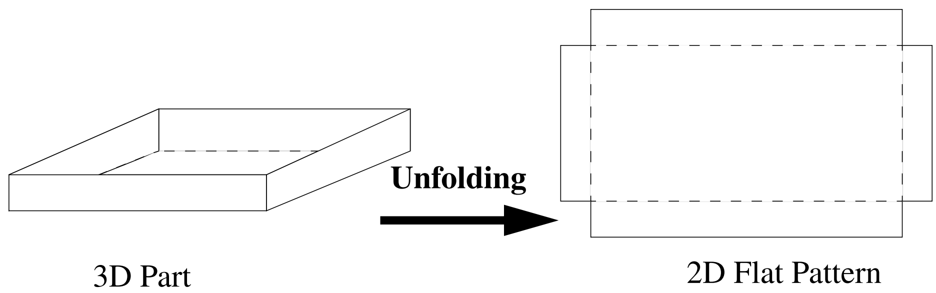


Figure 2.1: Unfolding process of a four-bend box.

There are three kinds of form representations that are commonly used for sheet metal parts: wireframe models, boundary representations (B-rep), and solid models. Wireframe models are easy to prepare, as they are very similar to traditional drafting tasks. The hardware and software needed to make wireframe models is inexpensive compared to the other two representations. However, the absence of manufacturing features and ambiguity in topological relationships make wireframe modeling less suitable for complex parts.

Unfolding algorithms are intended to convert a 3D wireframe model into a 2D flat pattern. Current methods employ a graph representation of part topology. Algorithmic search of this representation for faces, bend lines, and other group information produces a 2D pattern. However, ambiguities may arise from the use of a wireframe model; for example, some wireframe models may be unfolded in several different ways, and solid panels may not have been explicitly distinguished from openings. In addition, the manufacturing engineer may not have a full understanding of the part model, as he or she deals with low-level entities such as lines, arcs and circles. There are other obvious disadvantages to the use of wireframe models: unfolding is a separate, non-concurrent process; additional information is needed to develop a process plan; and inexperienced designers are prone to produce invalid designs, owing to the large number of design rules. One common problem in wireframe designs is the imprecise connectivity of some entities (Figure 2.2). For instance, if two lines in the design are supposed to intersect at their end points, the designer may inadvertently design the model in such a way that these two lines do not intersect at their end points or do not intersect at all. This kind of mistake usually causes the unfolding algorithm and the following process planning to fail. While these kinds of problems are expected in any design domain, errors become especially critical as batch sizes go down, as they must be amortized over a smaller number of parts. When producing in unit quantities, a single mistake in design can double the cycle time and cost of the part.

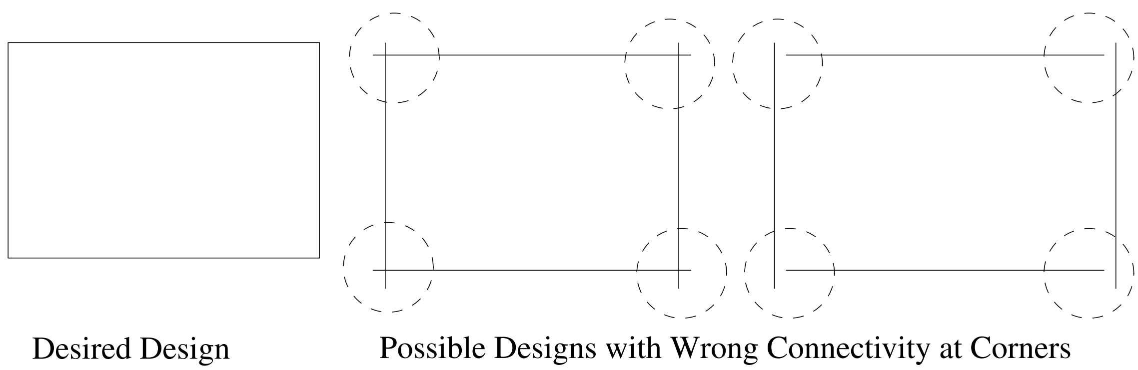


Figure 2.2: Common mistakes of connectivity in wireframe models.

Solid models, on the other hand, provide rich geometric and topological information. Solid modeling represents 3D parts truly, with material thickness, but the creation and the manipulation of parts is usually tedious and time-consuming. The complex data structure also complicates downstream planning tasks. In addition, solid-modeling CAD systems are currently expensive.

Boundary representation provides adequate information for both part design and subsequent production planning. Sheet metal parts are represented by neutral surfaces. The neutral surface is a surface the dimensions of which remain unchanged during bending. This representation resolves topological ambiguity and simplifies production planning. It can be easily converted to a solid model representation if necessary. In our Intelligent Bending Workstation Project at Carnegie Mellon University, we use a boundary representation with additional information (thickness, bend radius, and bend deduction) for various planning tasks of sheet metal part production.

In previous work (Wang and Sturges 1996), a new approach to concurrent part and process design with a series of representations, each of which embodies the part relative to a given process or series of processes, was proposed. A parallel design environment for sheet metal parts, BendCad (see Figure 2.3), has been developed in the Intelligent Bending Workstation Project. It employs concurrent multiple representations of bent sheet metal parts which avoid the difficulties of feature extraction and ambiguities in representation. BendCad offers designers a library of industry-standard primitive shapes with which to build a part. These shapes are defined in both two and three dimensions as functional forms, in contrast to the two-dimensional geometric outlines described by Kimpel and Richards (1991). Primitive shapes in BendCad are predefined generically and displayed in three dimensions, as rectangles, angles, channels, zees, boxes, holes and so on (Figure 2.4). Holes are defined as primitive shapes of negative material. Form features such as dimples and louvers are represented as bounding boxes and are included in the library. Users may also define their own primitives and save them in the library. Each of these primitives either takes parametric data from the designer to specify its dimensions, or inherits its dimensions from an adjacent primitive. The mapping between each individual 3D primitive and its flat pattern is parametrically defined and instantiated with material type and thickness.

Part construction in BendCad begins by connecting primitive shapes to each other at bend lines. The bend angle is defined by the designer. Holes are subtracted from the current model. Primitives may be added to either the 3D or the 2D view of the part. It is easier to add features in the 2D view, while it is much easier to monitor the 3D shape as the part is built.

Punching and shearing processes are represented with auxiliary views in BendCad. The outer and inner profiles of the 2D flat pattern are traced after the design is completed. The location and orientation of the part in a 2D flat pattern, and its order issue from the upstream shearing process, are variable and depend on the order horizon at the time of release to manufacture. This data is essential for integrating the bending process plan with the punch and shear processes (Figure 2.5).

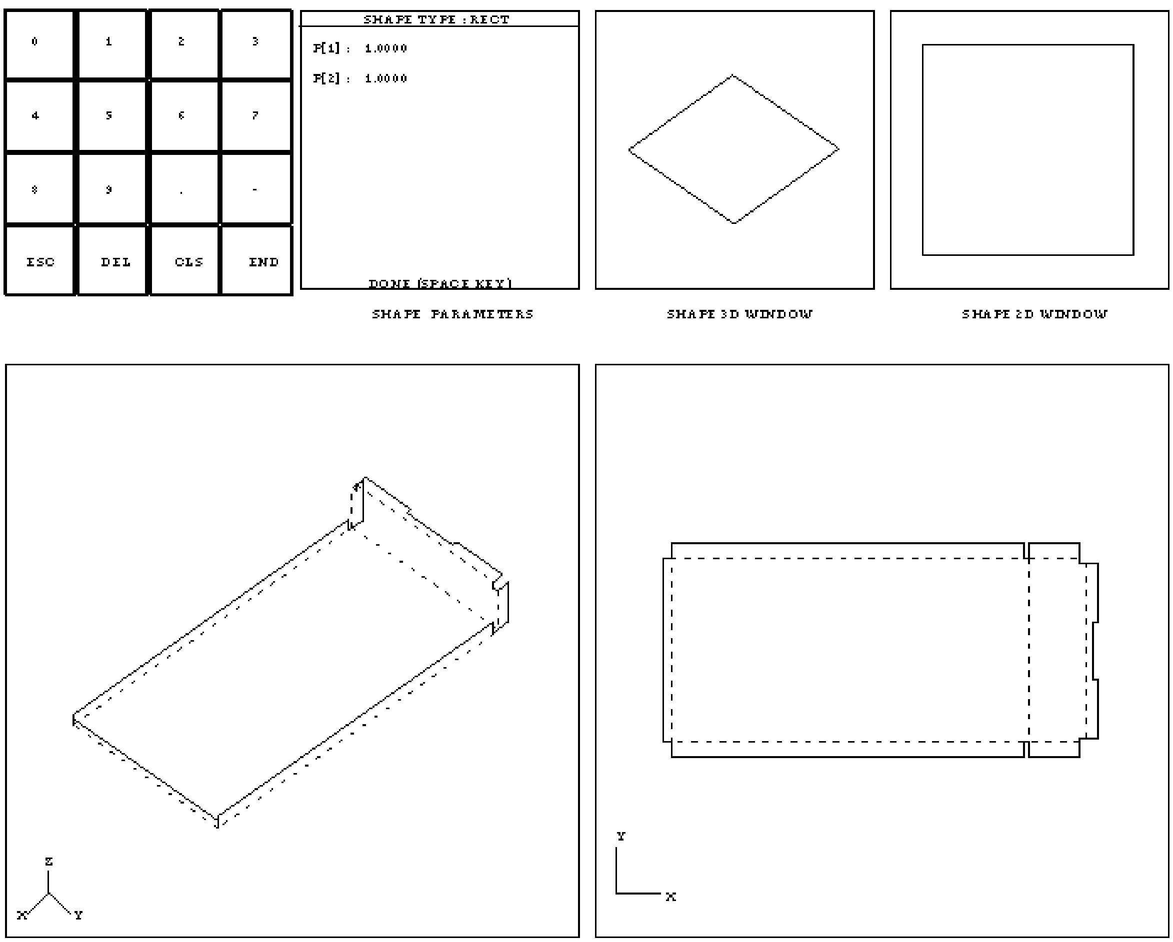
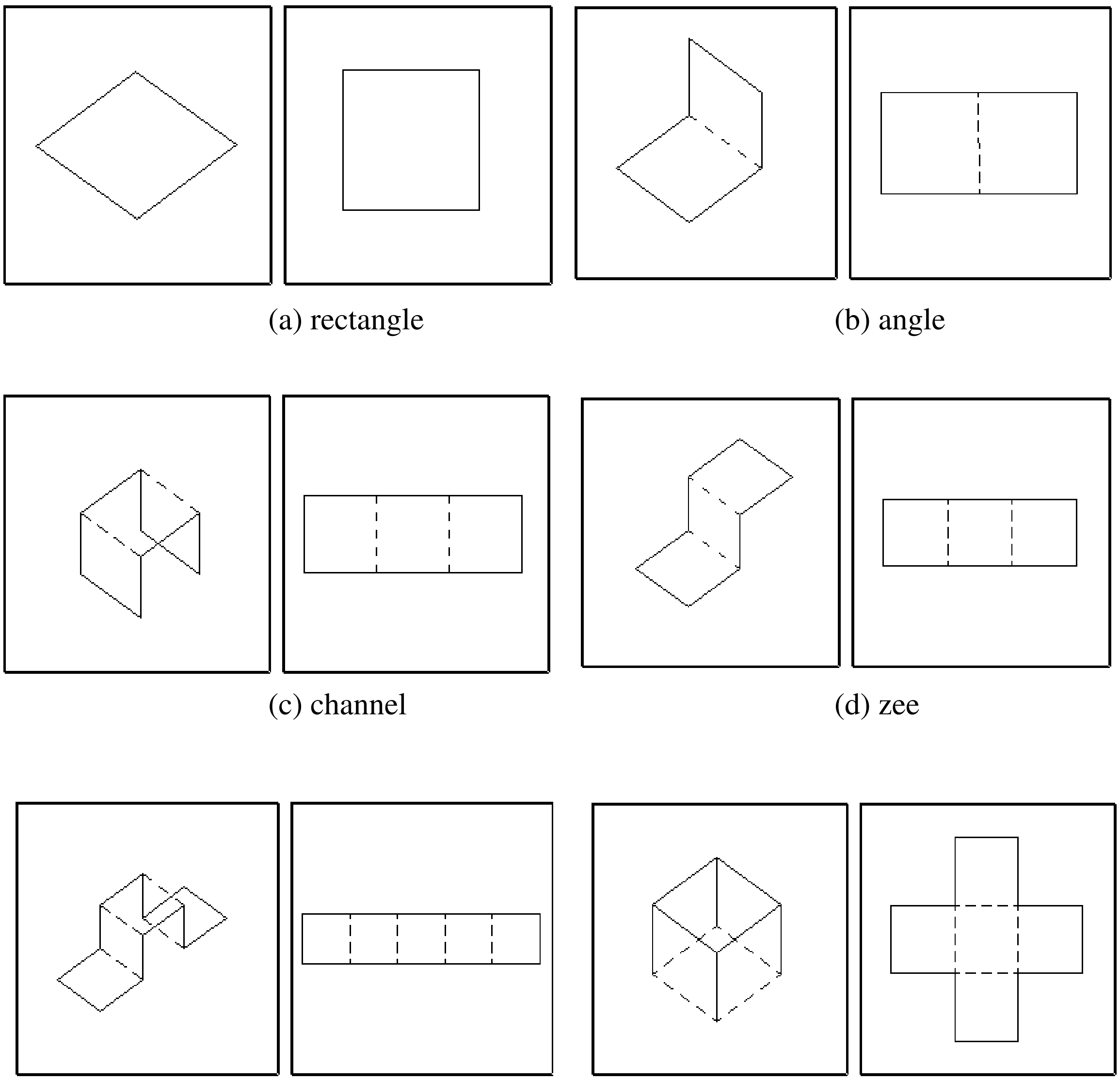


Figure 2.3: BendCad: A parallel design system for sheet metal parts.



(a) rectangle (b) angle

(c) channel (d) zee

(e) hat (f) box

Figure 2.4: Primitive shapes in BendCad.

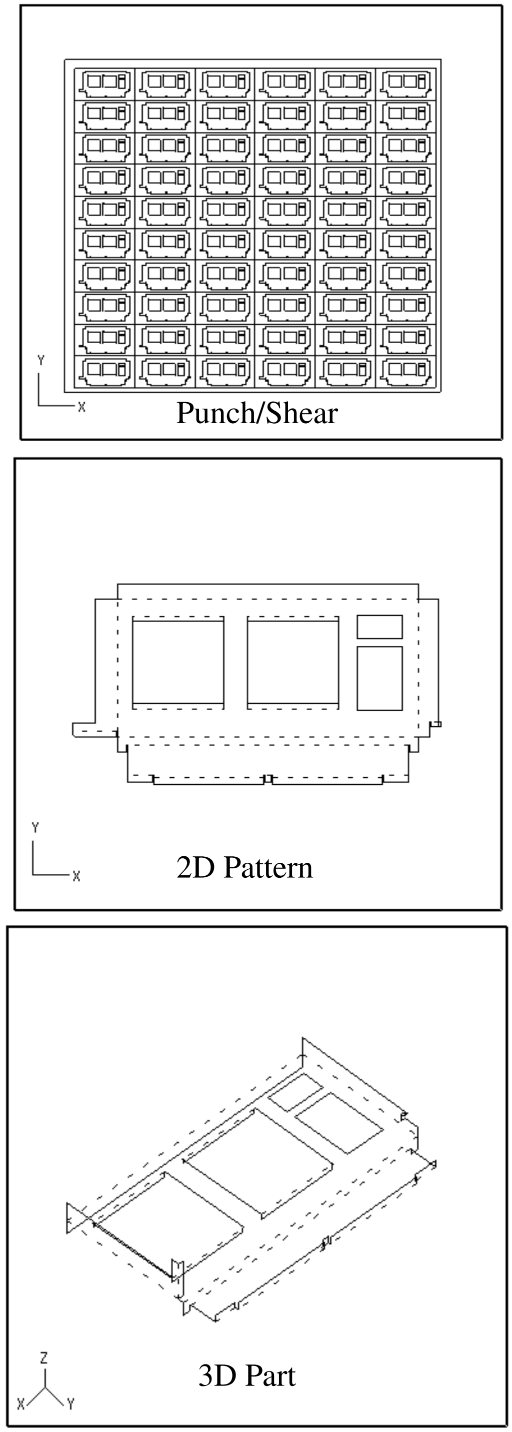


Figure 2.5: Concurrent multiple representations of a complex mounting tray.

The internal data structure in BendCad for implementing concurrent multiple representations is shown in Figure 2.6. A part is represented as a linked list of shapes. Each shape is represented as a linked list in three domains: punch, shear and bend. The punch and shear models are two-dimensional; the bend model is three-dimensional. For constant-width, variable-length, coil-fed applications, an additional view is included to model the initial cutoff process domain. Each shape is parameterized dimensionally, and can be readily visualized as a topology graph. With a set of functional design primitives and the connectivity analysis of the graph, BendCad automatically identifies process-specific features such as corners, internal and external tabs, corner notches, and external slots. Topological information is also stored. The initial bending process plan is derived from this data automatically, and it serves as the seed for a detailed process plan. The information in the preliminary plan comprises bend angles, bend radii, bend sequence, bend length, part orientation, and a standard tooling list. The final process design is carried out by a bending process planner described in Bourne (1992).

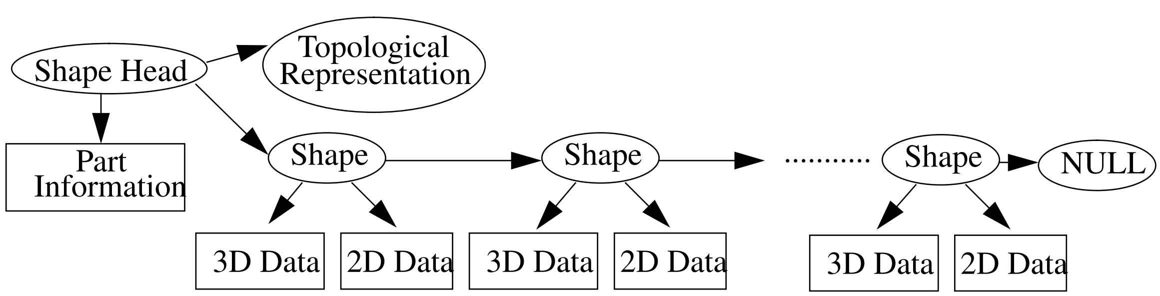


Figure 2.6: BendCad data structure.

The benefits of BendCad in a larger context will now be described in the sheet metal processing domains of shearing, punching and bending. Traditional sheet metal part design separates the two-dimensional flat pattern and three-dimensional wireframe model. Manufacturing process plans are manually developed for shearing and punching in 2D and for bending in 3D. The data embodied in these representations are incomplete and ambiguous with respect to the development of bending process plans. Additional information and skillful interpretation provided by the human process planner suffice for relatively large press runs, but this becomes increasingly uneconomical as batch sizes approach the idea of one-part-on-demand.

By contrast in BendCad, concurrent multiple representations of sheet metal parts prompt the designer to supply much missing information, while the several views resolve or eliminate sources of ambiguity. By having both 2D and 3D representations at the same time, the designer considers the shapes of both stock and finished part, while adding features to the most appropriate representation. For example, it is far more tedious to add a small tab to the 3D folded part model than to the 2D flat pattern. Moreover, a tab added to a 3D part at an inappropriate location may not be possible to produce from a flat pattern at all. The essential point of developing a master model that underlies all such views is that a preliminary process design is defined at the part design stage.

2.2 Production of Sheet Metal Products

Bent sheet metal part production typically involves punching (or blanking), bending, stacking, and assembly processes (Figure 2.7). Bent sheet metal parts usually are designed in 3D. Flat patterns are prepared by developing the 3D part. To reduce the amount of scrap, layout planning (nesting) of flat patterns is performed. Punching and blanking processes then prepare the 2D flat patterns. When the 2D patterns are ready, bending process planning can proceed and the final part shapes are bent. Before these parts are assembled into products, temporary stacking is necessary. Finally different parts of a product can be assembled. These processes are described in the following sections.

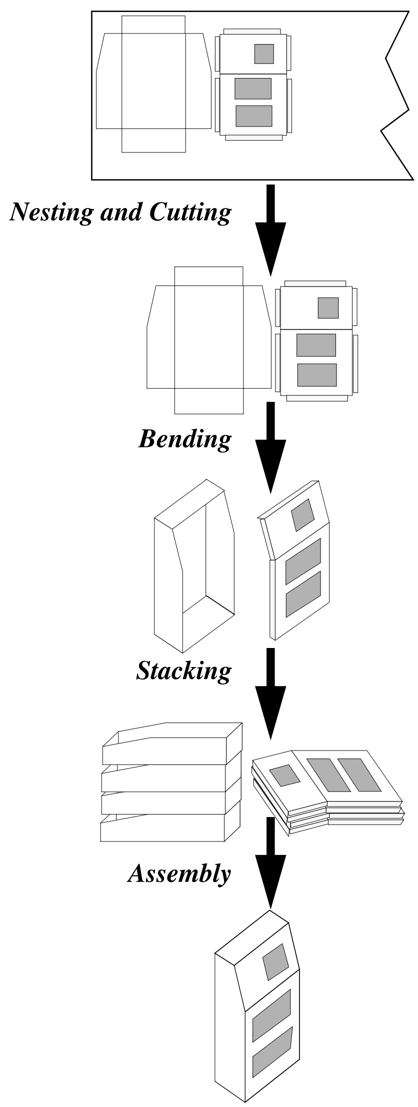


Figure 2.7: Sheet metal production flow.

2.2.1 Pattern Development

Pattern development involves unfolding 3D part designs into 2D flat patterns. This topic has been studied extensively in the garment industry and in shoe manufacture. In sheet metal pattern development, the bend deduction (the difference of the dimensions before and after bending) needs to be considered during the development process (Reich et al. 1991).

2.2.2 Layout Planning

Layout planning (nesting) aims to efficiently allocate regular and irregular patterns on a stock of sheets of finite dimension. The stock utilization ratio is usually the most critical factor to be considered. Both operational research and AI search approaches have been proposed and used in various layout systems. Albano and Sapuppo (1980) propose a heuristic search approach to handle both regular and irregular shapes. Pattern layout is known as an *NP-hard* problem (Li and Milenkovic 1995) and the optimal solution is usually impossible to find.

For traditional punch-turret pressing and flame cutting, gaps or lead-ins have to be provided among nested patterns to ensure cutting quality. In recent years, state-of-the-art laser cutting techniques have been developed and high cutting precision (up to 0.005 inch) can be achieved (Powel 1993). Safier and Grinberg (1996) propose an edge-sharing algorithm for layout planning. The goal is to maximize the total length of shared edges and minimize the convex hull area of the resulting layout. In their approach, a scheduling system with the Just-in-Time (JIT) (Morton and Pentico 1993) concept is incorporated into the layout module. Their results are used in our prototype system and illustrated later in the text. A detailed survey of various layout planning approaches can be found in Dowsland and Dowsland (1995).

2.2.3 Punching and Blanking

Previous work in the sheet metal industry has mainly focused on the automation of punching, shearing and nesting processes; Shibata et al. (1981) develop a CAD/CAM system using structural codes to represent a sheet metal product. Uzsoy et al. (1991) use a rule-based approach to prepare process plans for punched and bent parts. Hancock (1989) develops a prototype system to interpret the CAD design and to determine efficient punch and shear sequences. Recently, laser cutting has become more and more popular, because it is high precision and can handle complicated profiles.

2.2.4 Bending Process Planning

Bending is a material deformation process using punches and dies. A typical bending operation is shown in Figure 2.8. A bending plan usually includes: a bend sequence, tool (punch/die) selections, gage position planning, and stage layout planning. For robot-assisted systems (like the Amada BM100 in Figure 2.9), grasping position, gripper selection, and robot motion, also need to be determined. A good reference on sheet metal bending is Benson (1997).

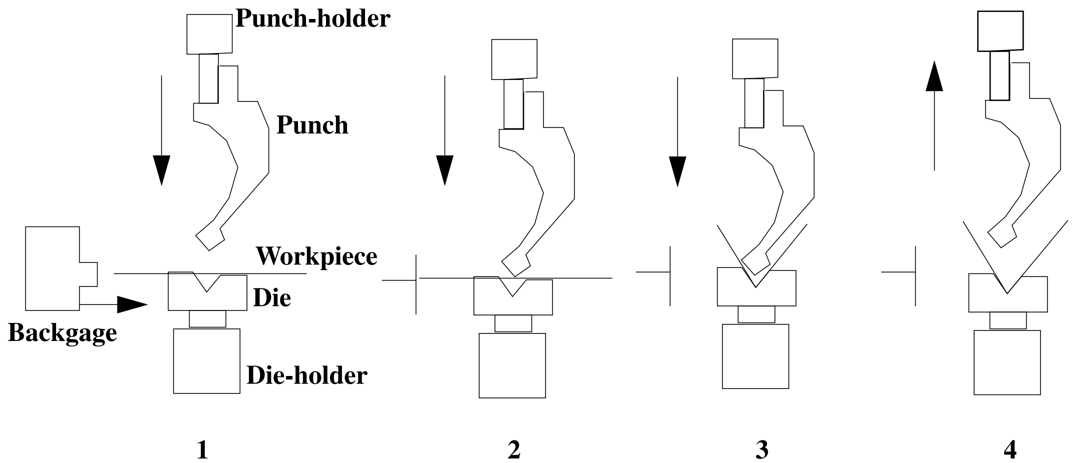


Figure 2.8: Bending operation.

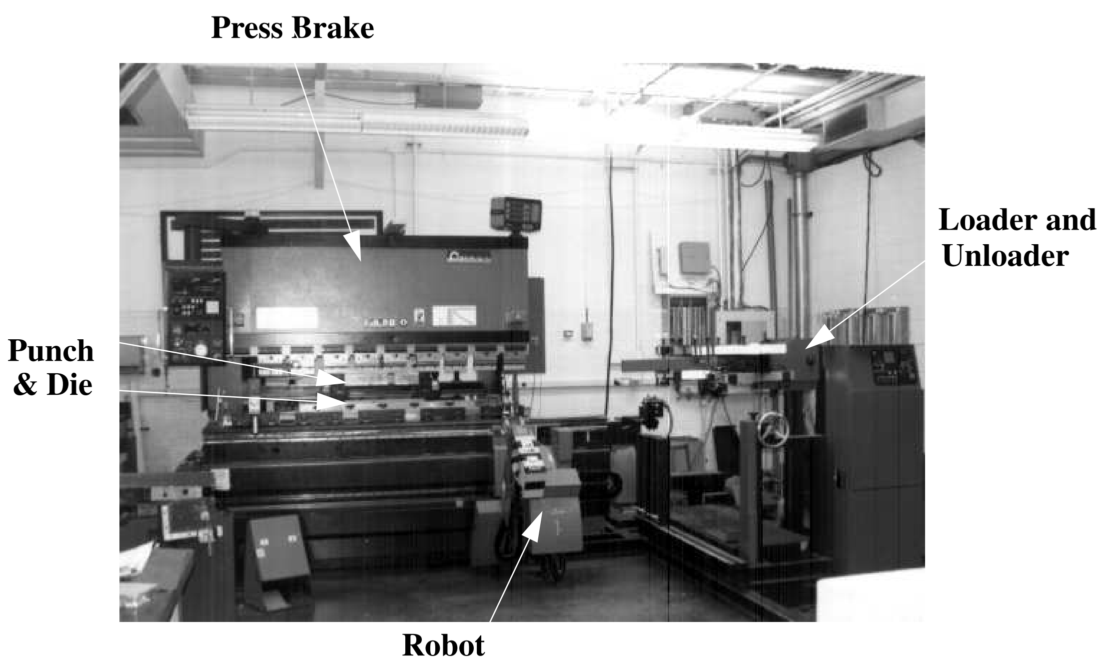


Figure 2.9: An automatic bending cell (Amada BM100).

2.2.5 Part Stacking

At the present time, stack planning for bent sheet metal parts is usually done manually. Stability and ease of stacking/retrieval are the major concerns. A matrix of pins is often used to help stack parts. Dedicated loading and unloading mechanisms and robots are used widely in sheet metal stacking. Ayyadevara and Sturges (1996) describe an algorithm to automatically generate stacking patterns for bent sheet-metal parts. They consider stacking one part at a time. First, a part is placed in the most stable position (considering the height and the position of the center of gravity of the stack), usually on the floor. A face or a set of edges of the new part is then chosen as the base and a set of faces and/or edges in the stack is chosen as the support for the base. The new part is positioned and rotated on the stack so that it is interference-free and stable. The process continues until all parts are processed.

2.2.6 Product Assembly

Typical assembly methods used for sheet metal products are welding, riveting and hinging. Ramsden et al. (1984) examine experimentally the fastening and joining problems of robot automated assembly for sheet metal products. The assembly of sheet metal products is still a labor-intensive operation due to the difficulty of part handling.

2.3 Conclusion

Bent sheet metal products are widely used in our everyday life. With the advanced technology and manufacturing facilities that have been developed, high precision sheet metal products and parts have become possible. Sheet metal components are being more widely used in the aircraft, automobile and shipbuilding industries.

In this chapter, background information on sheet metal design and production was presented. A design system with concurrent multiple representations has been developed and used in various sheet metal production projects. This chapter aims to provide a clearer connection between sheet metal product decomposition and various manufacturing processes. In later chapters, I will revisit these processes and explain them in more detail. In the next chapter, I focus on the product decomposition approach.

Chapter 3

Decomposition of Sheet Metal Products

In this chapter, a manufacturability-driven decomposition methodology for sheet metal products is presented. This chapter is organized as follows. First the representations of product and decomposition are discussed. Then the decomposition assumptions, strategies, goals, and algorithm are given. Several criteria which help to evaluate manufacturability in cutting, bending, and assembly during decomposition are presented. Design modifications due to decomposition are explained. The decomposition approach is also justified. A decomposition example is shown and used to explain the decomposition approach.

3.1 Product Representation

A *product* is a set of *parts* connected by assembly fasteners. A *sheet metal part* is a set of planar faces connected by bend lines. In previous work (Wang and Sturges 1996), we used concurrent multiple representations to represent sheet metal parts at different manufacturing stages. Figure 3.1 shows the representations of an electronic enclosure. This representation scheme has been used in several sheet metal planning systems, such as cutting, nesting, bending, and stacking in our Intelligent Bending Workstation Project.

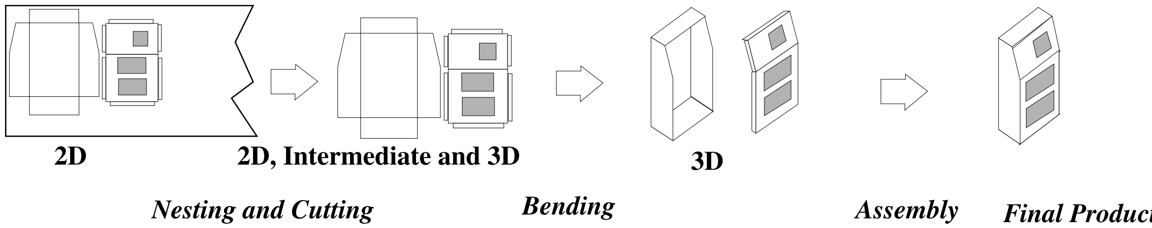


Figure 3.1: Multiple representations of a sheet metal enclosure.

A sheet metal product is usually represented in its 3D geometry. The corresponding topological graph (in 3D) with faces as nodes and connecting edges as links can then be derived (Figure 3.2). It is noted that the face connectivity graph of the 2D flat pattern corresponds to a spanning tree[1] of its 3D face connectivity graph. The links of the 2D connectivity graph represent the bend lines of the part. The major purpose of product decomposition is to determine what are the parts of the product and, for each part, what is the 2D flat pattern. Designers will assign the desired assembly methods/fasteners to the product after decomposition. When the decomposition is done, the complete representations (2D and 3D) of the parts can be obtained and further planning in cutting and bending can be performed. It should be noted that 3D part shapes are sufficient for the purpose of assembly planning.

1. A tree T is a spanning tree of a connected graph G if T is a subgraph of G and T contains all vertices of G (Deo 1974).

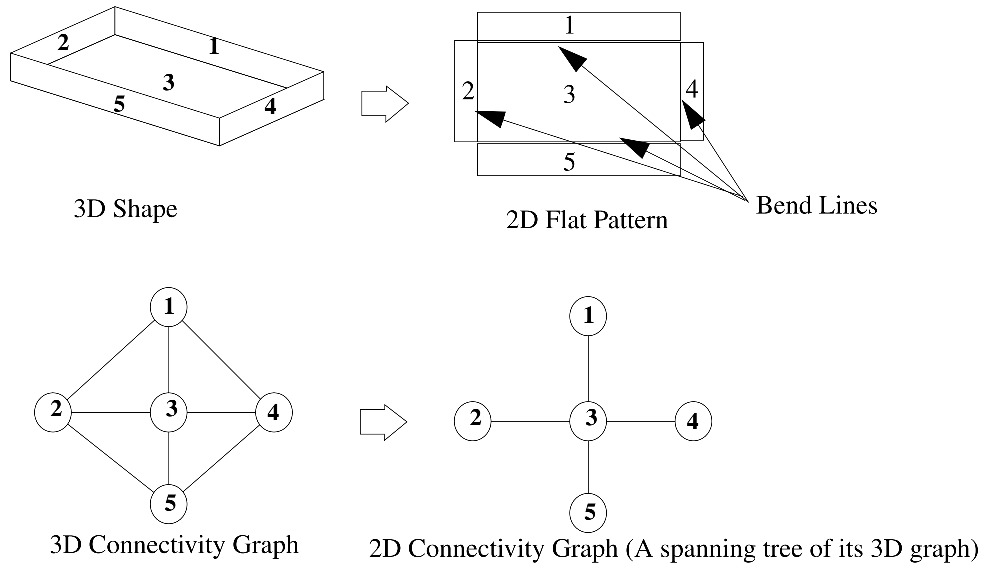


Figure 3.2: 3D shape of an open box and its connectivity graph.

3.2 Representation of Decomposition Space

Homem De Mello and Sanderson (1990) use the AND/OR Graph to represent assembly plans. They show that this representation is more compact than the state graph and other representations. In this thesis, the AND/OR graph is used to represent the decomposition space of a sheet metal product. Each node in the AND/OR graph represents a part (and a subassembly) and the AND-arc represents the assembly operation and fastener which bring the children parts together to make a parent. An OR-arc represents different ways of building the same parent part. In Figure 3.3, a 4-face part and its decomposition space are shown. Each decomposition is represented as a set of connected faces.

Unfortunately, even for a moderately complex product, the decomposition space is usually large. For instance, the closed box with 6 faces shown in Figure 3.4. has 29 possible binary decompositions. It should be noted that the decompositions correspond to the cut-sets[1] of the 3D connectivity graph of the product. Also, each node represents the 3D shape of the part which could have been made with more than one flat pattern. That is, the mapping from 3D shape to 2D flat pattern could be one-to-many. All such possible flat patterns correspond to the spanning trees of the part.

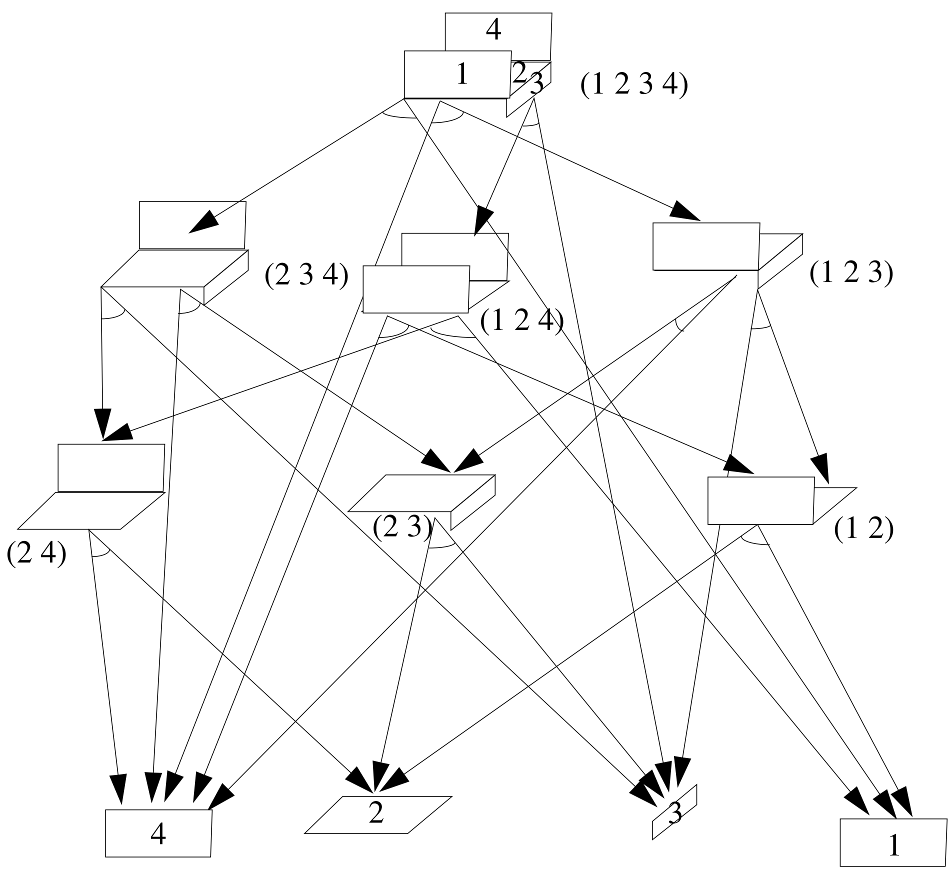


Figure 3.3: The decomposition space of a 3-bend part.

1. A cut-set is a minimal set of links in a connected graph whose removal cuts the graph into exactly two components (Deo 1974).

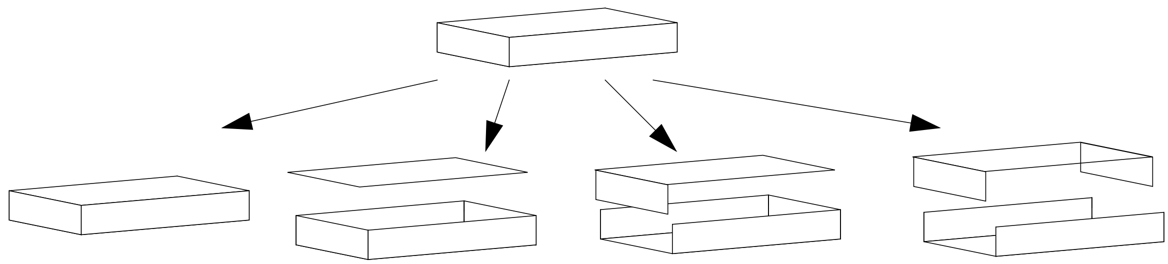


Figure 3.4: A closed box and its binary decompositions.

3.3 Decomposition Assumptions

As the decomposition space for a sheet metal product is usually large, to make the decomposition more tractable, several assumptions are made. First, decompositions can only occur along the connecting edges (links of the connectivity graph). This assumption is natural since most sheet metal designs are surface-based. Second, during the manufacturing and assembly processes, distortion or tearing of the faces is not allowed. The functional requirements of the product will not be addressed explicitly in this thesis. However, the designers will be able to guide or overrule the decomposition results during the decomposition process. In order to maintain the original functionality of the product, only the design modifications for mating requirements are allowed. In this thesis, welding is assumed to be the assembly method for decomposed products. Assembly tabs are usually needed to provide space for welding operations and these are added automatically but can be changed by the user.

Different manufacturing facilities usually have different machine capabilities. In this thesis, it is assumed that the following manufacturing facilities are used: a laser cutter, an automatic bending system with a 5-axis robot and a loading/unloading device, and an assembly cell equipped with an assembly robot.

Different manufacturing facilities also suggest different decomposition constraints and heuristics. For instance, replacing the automatic bending machine with a manual press-brake certainly relaxes the constraints imposed by the material handling component, the robot and the grippers. Making paper models by hand will also relax the manufacturing and assembly constraints quite a bit, since human beings are very dexterous compared to machines and robots. Each facility will have certain constraints and preferences associated with it. For instance, in this thesis, the part unfoldability, tool accessibility and product disassemblability are the constraints associated with the laser cutter, automatic bending machine and assembly robot. By using different manufacturing facilities, different decompositions can be generated even for the same product.

3.4 Decomposition Goal

The goal of product decomposition is to find the decompositions which optimize the manufacture and assembly of the resulting parts. The major considerations of decomposition are the following:

- **Minimize the number of parts.** The number of parts in an assembly is one of the major factors in determining the cost of an assembly (Dieter 1991).
- **Minimize the number of assembly tabs.** The assembly tabs are added to the decomposed parts after decomposition. Each assembly tab introduces additional material

cost and bending cost.

- **Maximize the compactness of the 2D flat pattern**. According to our study of sheet metal design and production, compactness is an important property for sheet metal products. A compact pattern can reduce the cutting cost, make it easier to handle in machine loading, and reduce the drooping problem during handling.

3.5 Decomposition Strategy

In this thesis, cutting, bending and assembly processes are considered during decomposition. For each process, an efficient and effective method is needed to evaluate the manufacturability and assemblability of the decomposed results. It is time-consuming and impractical if full function process planners are used to evaluate all possible decompositions. As a result, only the most critical issue in each process is taken into account and used as a quick feasibility test. Part unfoldability, tool accessibility and product disassemblability are used for the cutting, bending and assembly processes, respectively.

Two approaches can be used to decompose the product: the decompose-first-develop-later approach and the develop-first-decompose-later approach. Figure 3.5 shows these two decomposition approaches for a closed box.

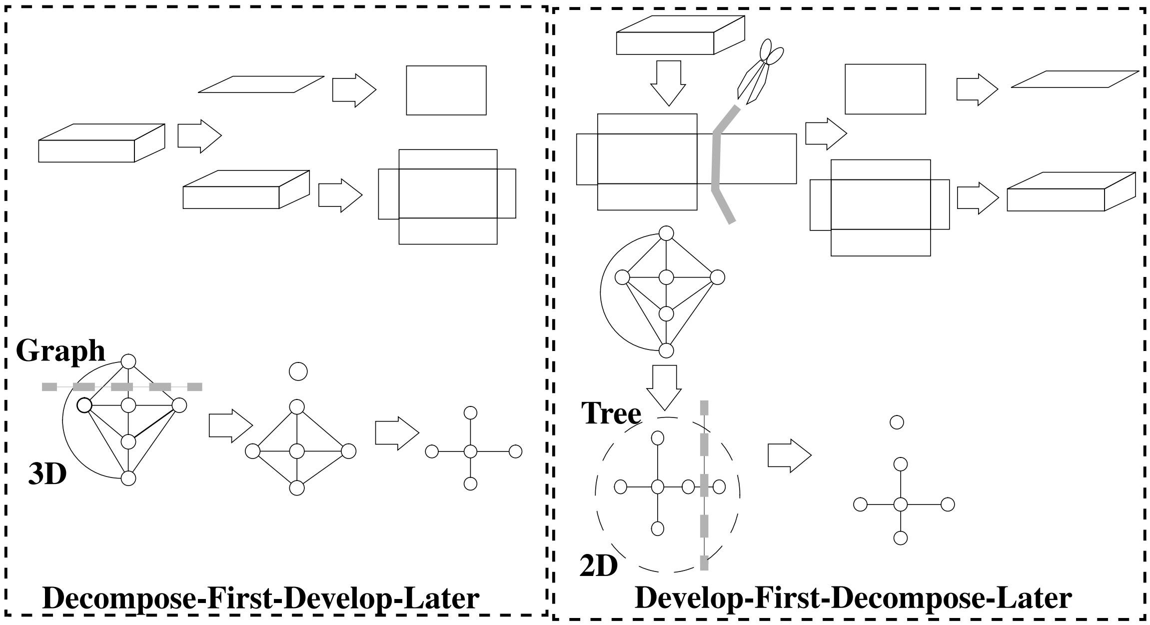


Figure 3.5: Decomposition strategies for a closed box.

3.5.1 Decompose-First-Develop-Later Strategy

The first approach, "decompose-first-develop-later," is to decompose the product into 3D parts first, and then to find the compact 2D patterns of the 3D parts. We refer the latter process as "developing" the part. This approach is general but it does not avoid the combinatorial explosion problem. All possible 3D decompositions can be represented as the cutsets in the topological space, which is exponential in complexity. As a result, this approach is intractable while decomposing the product, not to mention the later issue of unfolding and manufacturability evaluation.

3.5.2 Develop-First-Decompose-Later Strategy

The second approach, "develop-first-decompose-later," is to develop (flatten) the 3D product into a compact 2D flat pattern first, as compactness is a high priority for decomposed parts in cutting and bending, and then to decompose the developed product into 2D parts. The associated 3D part can be found when a 2D flat pattern is known. Decomposition optimality is traded for tractability. This approach cannot guarantee global optimality but the results are satisfactory. In this thesis, the develop-first-decompose-later strategy is employed in our decomposition approach to reduce decomposition complexity (since we convert the problem of decomposing a graph into the problem of decomposing a tree!).

3.5.3 Comparison of the Decomposition Strategies

A comparison of the two decomposition strategies mentioned above is given in Table 1. It is easier to consider further manufacturing concerns in the cutting and bending processes if the 2D flat pattern of a product is known.

	Decompose-First-Develop-Later	**Develop-First-Decompose-Later**
Decomposition Domain	3D (Face Connectivity Undecided)	2D (Face Connectivity Decided)
Product Representation	A Graph (3D Face Connectivity Graph)	A Tree (2D Connectivity Graph)
Decomposition Representation	Cut-Sets of the Graph	Edges of the Tree
Complexity	Exponential	Linear
Enumerating All Decompositions	Yes	No

Table 1. Comparison of decomposition strategies.

3.6 Decomposition Algorithm

The decomposition algorithm takes a geometric representation of a sheet metal product as input. The 3D face connectivity relationship is determined. The most compact 2D flat pattern is generated. Based on the 2D pattern, a set of manufacturing feasibility checkers is then used to evaluate the manufacturability of the product. The product is recursively decomposed if it fails to pass these tests. The product design may be modified by adding assembly tabs to enable the assembly operations. The resulting parts are evaluated by the cutting, bending, and assembly process planning systems and complete production plans for these parts are generated. The desired decomposition is the one with fewer parts and assembly tabs. An A* search algorithm is used to search for the desired decomposition. The flowchart of the decomposition algorithm is shown in Figure 3.6.

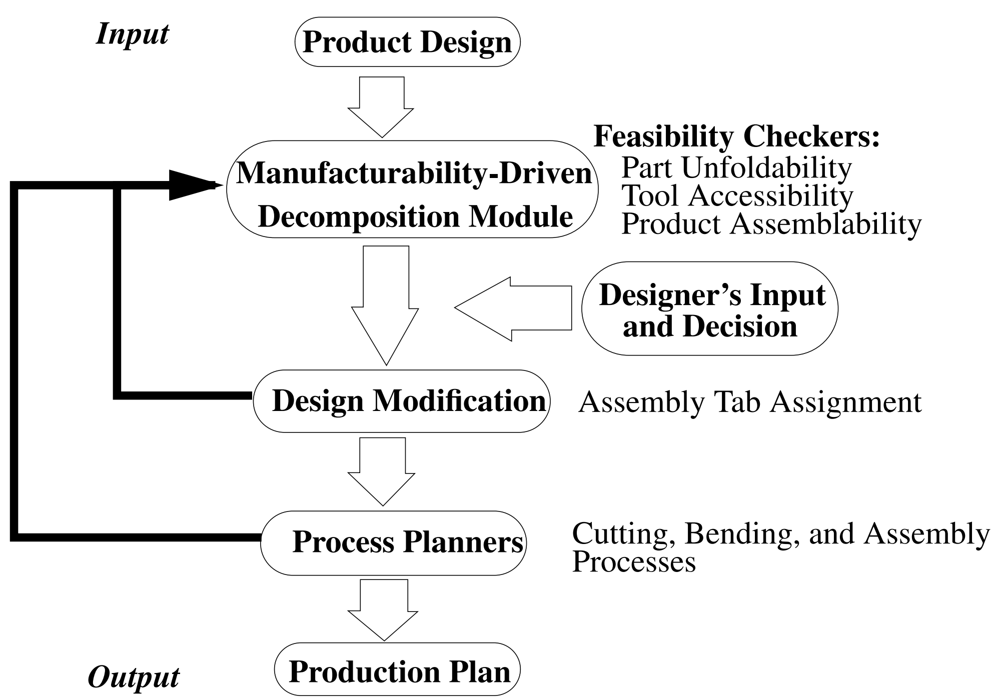


Figure 3.6: Manufacturability-driven decomposition approach.

3.7 Manufacturability Evaluation

To efficiently evaluate the feasibility of cutting, bending and assembly, several evaluation criteria are encoded as feasibility checkers. In this thesis, part unfoldability serves as the evaluation criterion for cutting, tool accessibility for bending, and product assemblability for assembly (Figure 3.7). These feasibility checkers are explained next.

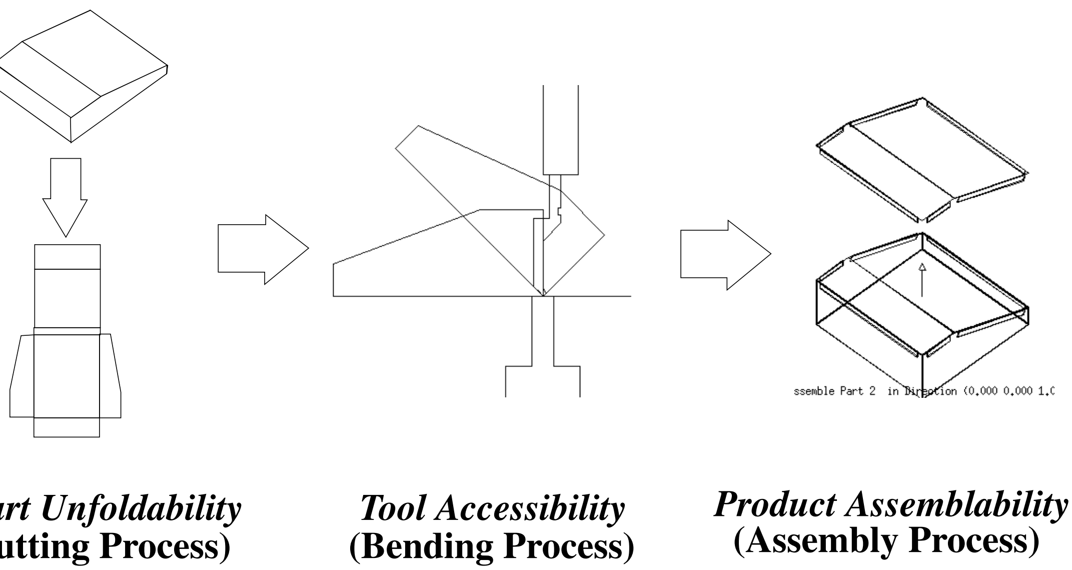


Figure 3.7: Feasibility checkers for manufacturing sheet metal products.

3.7.1 Part Unfoldability for Cutting

The most critical issue in the cutting process is part unfoldability. In our previous work (Wang and Bourne, 1995), a pattern development approach that generates compact patterns and avoids face overlapping is proposed. Compactness is defined as (area/perimeter^2) of the 2D flat pattern. To maximize the compactness, the perimeter of the 2D pattern has to be minimized since the area of the 2D pattern is a constant. The 2D pattern development is modeled as a minimum spanning tree problem. A local non-overlapping constraint regarding corners is also derived. The pattern development procedure needs to be done before the tool accessibility of the part can be evaluated.

3.7.2 Tool Accessibility for Bending

Different bending machines have different bending capabilities. Typical bendability considerations are: part thickness, part dimensions, tool availability and accessibility. Tool availability and accessibility are usually tested at the same time. A backward approach (de Vin et al. 1992) has been developed that determines whether an unfolding sequence exists. This approach starts with the final bent shape and tries to unfold the part into a flat pattern. The inaccessibility of any non-bendable part can usually be found in a short time.

Each bending machine will have a tool database which contains all the tools (punches and dies) available to that machine. Currently only standard tools are considered in this approach. Special tools such as closing, sash and multi-stage tools are not considered. As a result, this decomposition approach may generate more parts, but all the parts can be made using standard tools.

Although only the tool availability and accessibility of parts tested, these alone are quite expensive to compute. To further speed up the process, an *adjacent self-blocking component* is defined as a smallest topologically connected set of faces, all the bends of which are inaccessible for all tools in the tool database. That is, the component has no bend sequence without part-tool interference. Figure 3.8 shows an adjacent self-blocking component of a 4-bend part. A product with one or more self-blocking components is inaccessible and further decomposition of the links of the self-blocking components is needed. To obtain accessible decompositions, all adjacent self-blocking components have to be broken. An adjacent self-blocking component can be broken by decomposing any of its connecting links. The search for self-blocking components is done right after the part is unfolded. The number of bends in the adjacent self-blocking components can be limited to speed up the search. In this thesis, the search is limited to adjacent self-blocking components with at most five bends.

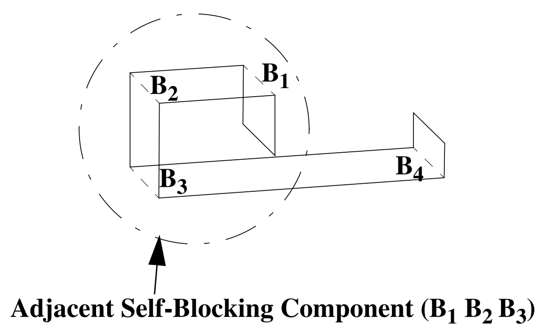


Figure 3.8: An adjacent self-blocking component.

3.7.3 Product Assemblability for Assembly

A quick disassemblability test can be performed between decomposed parts as the evaluation criterion for product assemblability. Designers can indicate the mating (inside/outside) relations between decomposed parts and the desired assembly fasteners. The specified mating relations determine the local contact information which can be used to derive the local feasible disassembly directions later in the assembly process planner. The result is used to generate the complete assembly sequence after the decomposition is finished. For instance, in Figure 3.9 a closed box is decomposed into two parts: top and base. Suppose additional assembly tabs are added to the top part. Possible mating relations between the tabs and their connecting faces are shown.

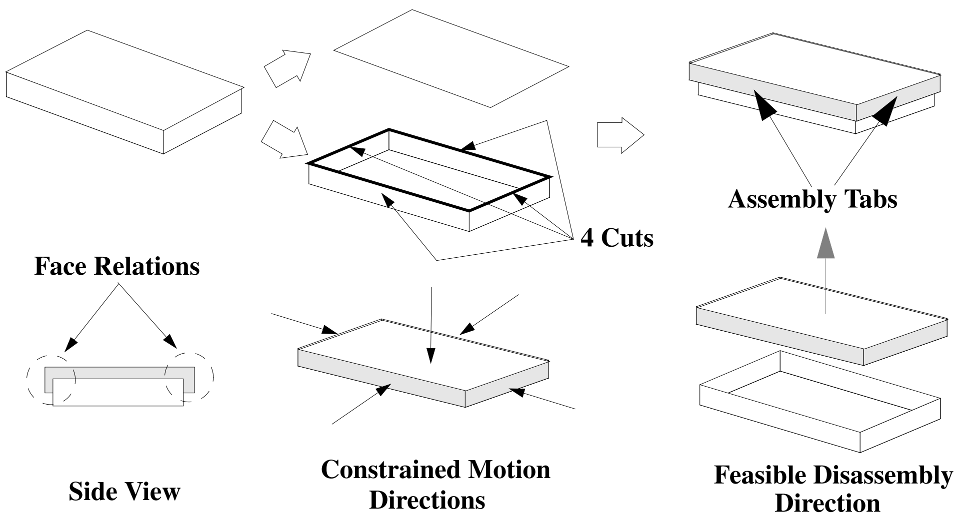


Figure 3.9: Possible design modifications of a closed box.

3.8 Design Modification

In order to compensate for assembly operations introduced by the decompositions, sometimes design modifications are necessary to meet the mating requirements between decomposed parts. For some assembly methods, such as welding, additional assembly tabs must be added to provide space for assembly operations. The tabs are added to cuts of the decomposed parts. A *cut* is a connecting edge which is separated by decomposition. The additional assembly tabs result in additional bending operations. The example in Figure 3.9 shows one possible decomposition of a closed box. The decomposed cover of the box has expanded by the amount of two thicknesses of the bottom part. There are four cuts in the decomposition and four additional assembly tabs are added to the top. As a result, four additional bends are also added to the top. The changes made are updated simultaneously in the geometric models and in the graph representation. The final results are tested again to ensure the part's manufacturability and assemblability.

Currently, assembly tabs are generated which are automatically added to the part with fewer bends. It is less likely this part will be inaccessible. However, there might still be interference between the assembly tabs and the part or other assembly tabs. The system will check for interferences and warn the designers of problems, and of whether further modifications of the tabs' positions and dimensions are needed.

3.9 Process Planning of the Decomposed Parts

During the decomposition process, only some manufacturing concerns are included in the manufacturability feasibility checkers. To ensure complete manufacturability, the cutting, bending, and assembly process planning systems are employed. The cutting and bending planning systems were developed under the Intelligent Bending Workstation project at Carnegie Mellon University. A primitive assembly planning system has been developed to evaluate the assemblability of the products.

3.10 Justification of the Develop-First-Decompose-Later Strategy

The decomposition approach presented in this thesis utilizes the "develop-first-decompose-later" strategy, which is based on the property of compactness of sheet metal parts. As the approach addresses the tractability of the decomposition problem, the quality of the approach requires further justification. The preservation of compactness after the 2D pattern decomposition and assembly tab assignment are discussed.

3.10.1 Preservation of Compactness after Decomposition

One important question about the decomposition approach is "what can be said about the compactness of the decomposed parts *if the original 2D pattern is the most compact pattern*?" That is, the develop-first-decompose-later approach tries to decompose the product in its most compact 2D flat pattern. Can compactness be preserved after it is decomposed? Or equivalently, are the subtrees, which result from cutting a link in a minimum spanning tree, are also minimum spanning trees with respect to their graphs? To answer this question, the following theorem must first be considered:

Theorem:

Let $U \subset V$ and e be of minimum length among the edges with one endpoint in U and the other endpoint in V-U. There exists a minimum spanning tree T such that e is in T (Even 1979).

Claim:

Let's assume the uniqueness of minimum spanning trees for this claim. For a weighted connected graph G and its minimum spanning tree T, if we cut a link e in T, we get two trees, say T_1 and T_2. Let G_1 and G_2 be the corresponding graphs of T_1 and T_2 (i.e. T_1 and T_2 are spanning trees of G1 and G2). If T_1' and T_2' are minimum spanning trees of G_1 and G_2 respectively, then T_1=T_1' and T_2=T_2'.

Proof: By the theorem above let T' be the minimum spanning tree of G constructed from T_1', T_2', and the minimum cost edge e between G_1 and G_2. By uniqueness of minimum spanning trees, T=T', and thus we can conclude that T_1=T_1' and T_2= T_2'.

From this proof an important property of our decomposition approach is obtained: the subtrees which result from cutting a link in a minimum spanning tree are also minimum spanning trees, which implies that if the most compact 2D pattern is decomposed by cutting one of its links, the resulting patterns remain the most compact pattern with respect to their graphs (Erdmann, 1997). This proves that the compactness will be preserved after decomposition. This is also true if there is more than one minimum spanning tree as they all have the same compactness. Figure 3.10 shows a closed box and its most compact 2D flat pattern. Three possible decomposition on the most compact 2D pattern are also shown (Figure 3.10(a), (b), and (c)). The resulting parts have the most compact 2D flat patterns according to our compactness definition.

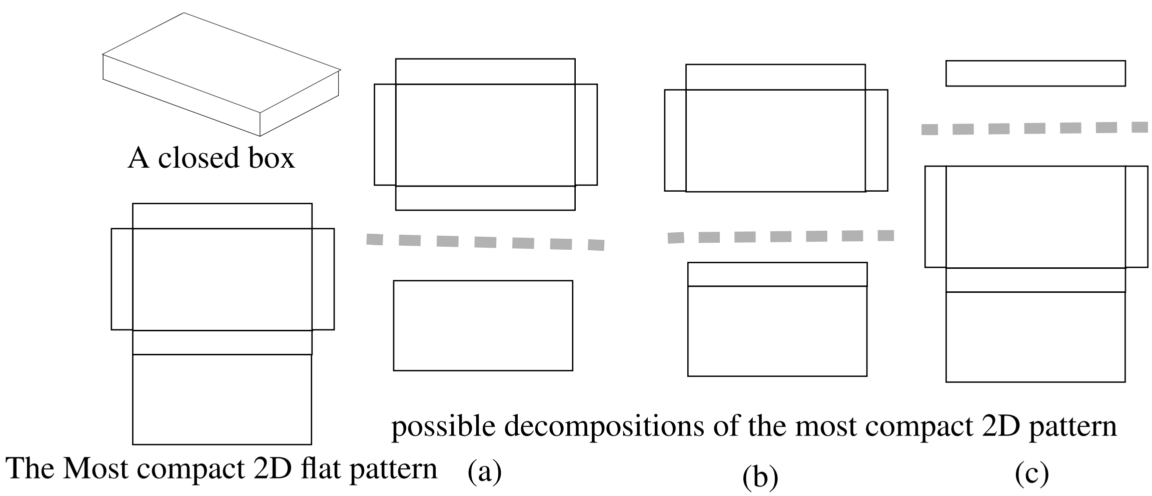


Figure 3.10: Preservation of compactness after decomposition

3.10.2 Preservation of Compactness after Assembly Tab Assignment

We have proven that compactness will be preserved after decomposition. The question which follows this is: will compactness be preserved after assembly tabs are assigned to the most compact pattern? If so, under what conditions will this be true? Consider the minimum spanning tree T in Figure 3.11. Suppose a new node N (the assembly tab) is added to T. Under what conditions will T be preserved in the new minimum spanning T' tree of $(G+N)$? To answer this question, assume that N connects to N_i in G with link e'_i, where i = 1 to n, and N_i is connected by e_i in T. By the theorem in Section 3.10.1 we know that the minimum edge e_{min} (in e_i) will be in T'. To preserve T in T', e_i has to be in T'. If $|e_i| < |e'_i|$, then $T' = T + e_{min}$. As the cost of a link is determined by its length, this condition is generally satisfied, since the assembly tab is usually connected to the minimum edge and the costs of the remaining edges connected to the original spanning tree are much larger than the ones in T. This property ensures that compactness will be preserved after assembly tabs are assigned to the parts.

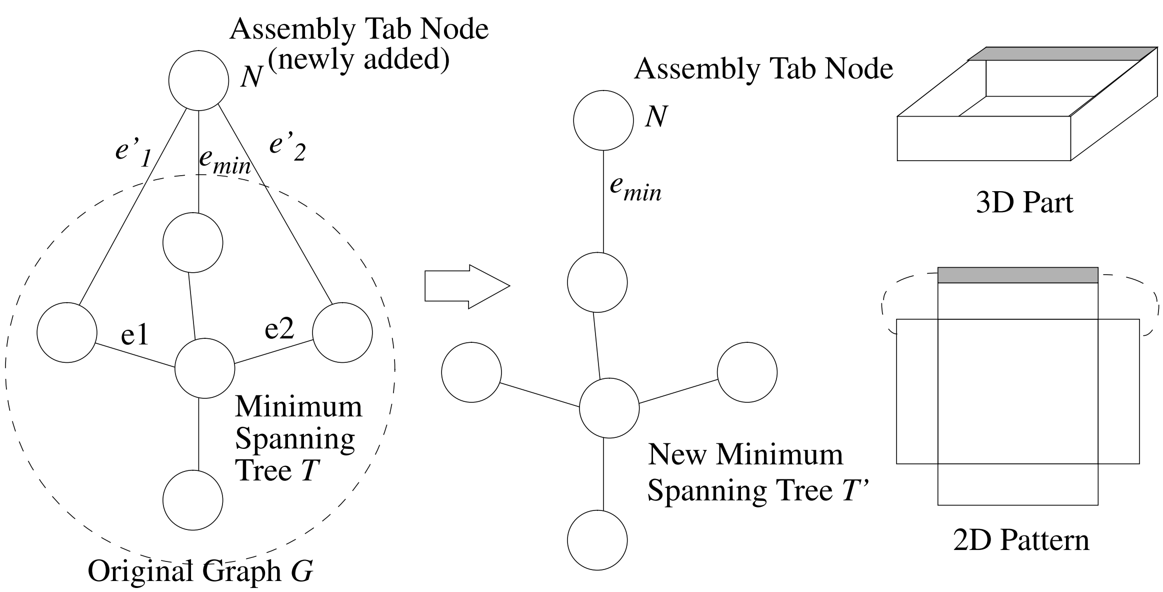


Figure 3.11: Preservation of compactness after assembly tab assignment.

3.11 Decomposition Example

Two different decompositions for a product are shown in Figure 3.12. The product is originally inaccessible for bending tools and further decompositions are necessary. The most compact 2D flat pattern of the product is first generated. The inaccessible components (we call these *adjacent self-blocking components*) are then identified with respect to the 2D pattern. Accessible decompositions are determined by breaking all inaccessible components. Additional assembly tabs are added to parts with fewer bends.

3.12 Conclusion

In this chapter, a manufacturability-driven decomposition approach for sheet metal products is presented. This approach adopts a "develop-first-decompose-later" decomposition strategy by first flattening the product into the most compact 2D flat pattern. The decomposition is then performed on the 2D pattern. Several manufacturing feasibility checkers for cutting, bending, and assembly have been applied to approximate complete manufacturability. Feasible decompositions are determined according to the results of these feasibility checkers. The product design is modified to account for assembly operations by adding assembly tabs. The decomposition with the minimum number of parts and cuts is selected as the desired decomposition. The complete manufacturability of the decomposition is ensured by applying the cutting, bending, and assembly process planners to the decomposed product. An example is used to illustrate the decomposition approach step by step.

In the next chapter, a prototype system based on this decomposition approach is developed to validate the proposed approach.

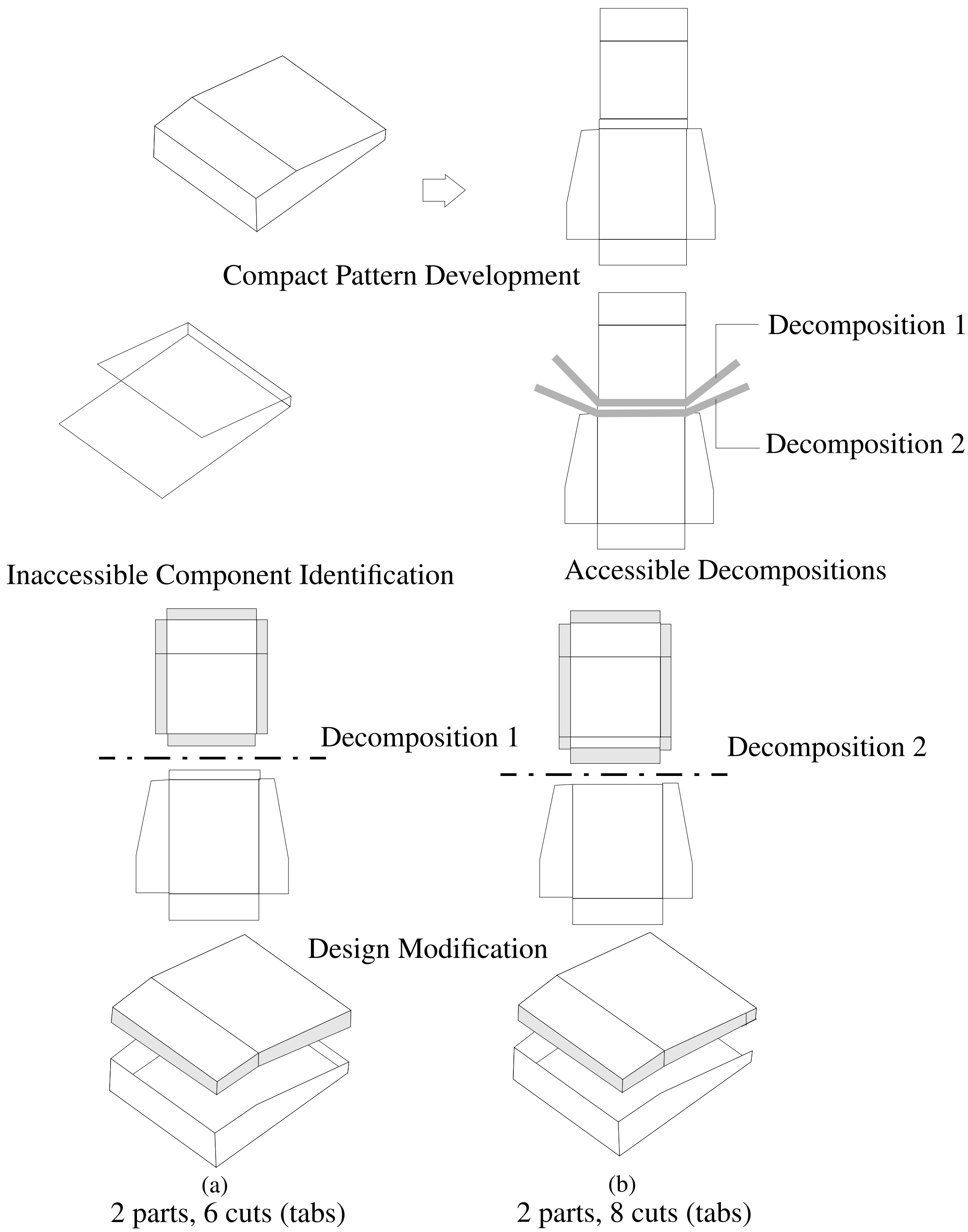


Figure 3.12: Decomposition results of a closed electronic enclosure.

Chapter 4

The Prototype System

A prototype system has been developed for sheet metal product decomposition based on the proposed decomposition approach. The system consists of the following modules: a product design system, a decomposition module, and process planners for cutting, bending and assembly. The architecture of the system is shown in Figure 4.1. The system first reads in the geometric representation designed by the product design system. The three-dimensional topological relationship of the faces is then determined. Once the geometric and topological representations are completed, the decomposition module adopts the develop-first-decompose-later strategy by first developing the product into unfoldable components. Feasibility is analyzed for each component and the necessary decomposition is performed. The designer can then examine and approve the decomposition results and additional decomposition information such as the preferred type of assembly method, the dimensions of the assembly tabs, etc. The result can be rejected if it violates end-user requirements such as the product's functionality. After the decomposition has been generated, the feasibility of the modified design is re-evaluated. The design is modified by adding assembly tabs to decomposed parts to provide space for assembly operations. The decomposition result is sent to the cutting, bending and assembly planners and a complete production plan is generated. The feasibility checkers and the process planners are implemented based on an A* search algorithm (Rich and Knight 1991). The A* algorithm is a heuristic search method that utilizes a heuristic estimate of the cost to get from the current state to the goal state. It is guaranteed to return a minimum-cost path if one exists, and to return failure otherwise, as long as the cost of the heuristic is never over-estimated.

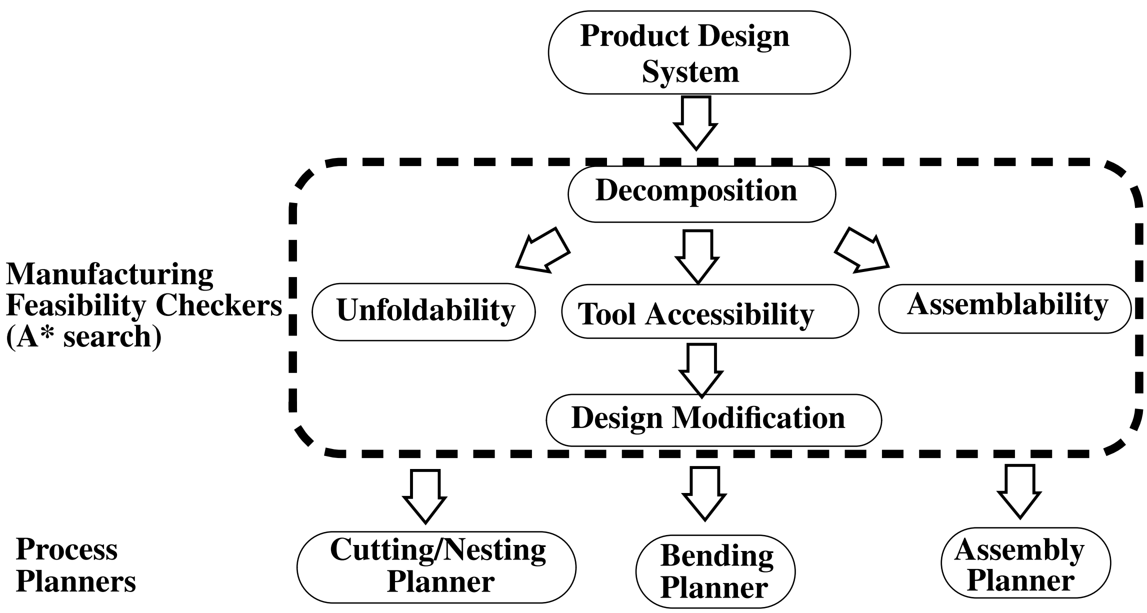


Figure 4.1: Architecture of the prototype system.

4.1 The Product Design System

The product design system is a face-based computer-aided design tool with multiple representations of the product for different manufacturing stages (Figure 4.2). The design system was implemented using the winged-edge data structure (Mantyla 1988) as shown in Figure 4.3. It generates the product models as input for the decomposition module. It also provides the input to the process planners after decomposition. The geometry and topology are updated as the decomposition proceeds. The user interface was implemented using Tk/Tcl (Ousterhout 1994).

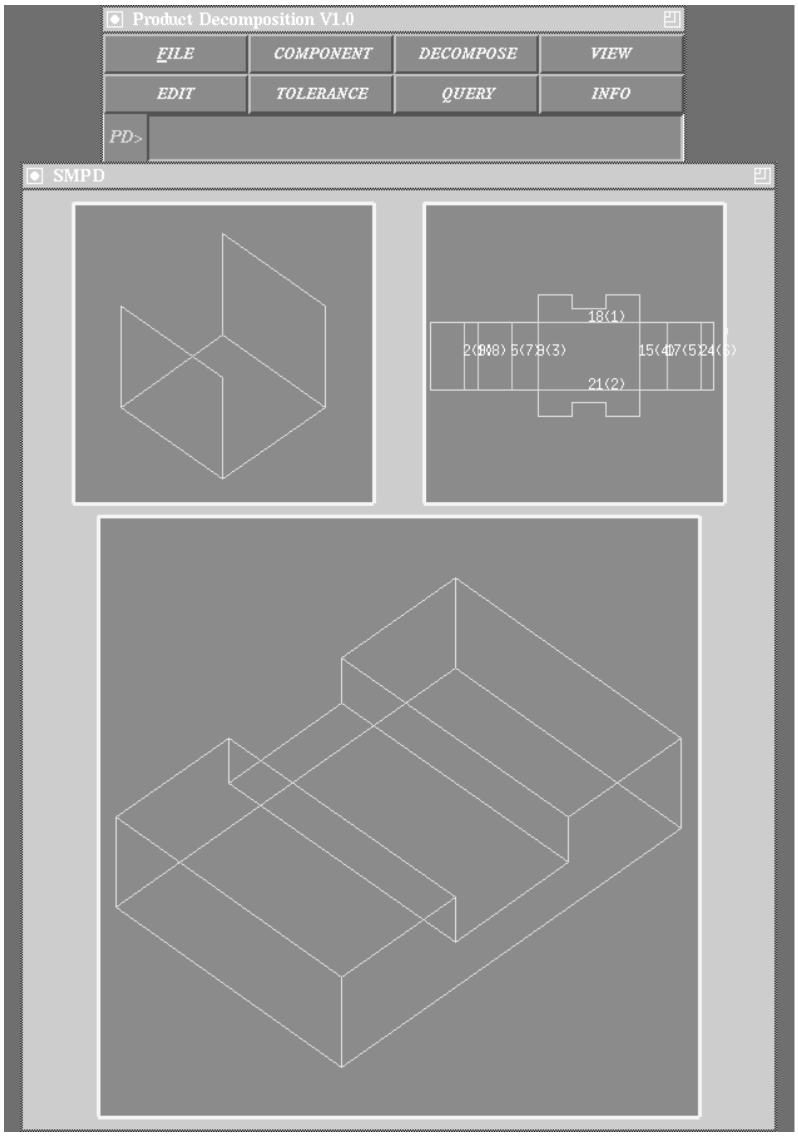


Figure 4.2: Product design system.

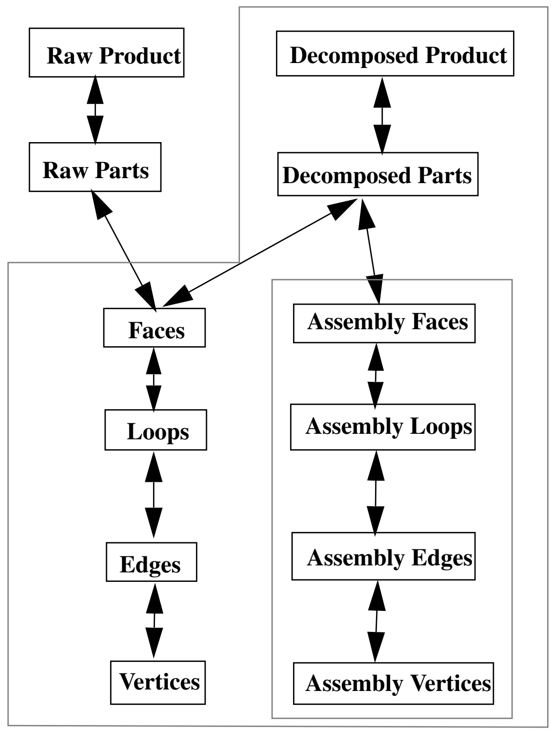


Figure 4.3: Data structure of a sheet metal product.

4.2 The Decomposition Module

The decomposition module takes a product designed in the product design system and unfolds the product into unfoldable components. For each component, the part unfoldability, tool accessibility and product disassemblability are used as feasibility checkers of manufacturability to ensure that the decompositions are feasible. Necessary decompositions are performed according to the feasibility analysis results. Currently, decompositions with the least number of parts and cuts are generated first. For each decomposed part, the most compact 2D flat pattern is preferred. A design modification module handles the assignment of assembly tabs to the decomposed parts after the decomposition. The decomposition results are then evaluated by the individual process planners and the complete production plan is generated.

The following feasibility checkers are employed to determine the manufacturability of the cutting, bending, and assembly processes. They serve as strong manufacturing constraints to filter out infeasible products/parts efficiently. However, the corresponding process planners (we assume they are complete and correct) have to be used to ensure the complete manufacturability in each process.

4.2.1 Part Unfoldability Checker

This module determines if a given product is unfoldable. If the product is unfoldable, the most compact 2D flat pattern is selected. If the product is not unfoldable, further decompositions for unfoldability are performed.

4.2.2 Tool Accessibility Checker

This module checks the accessibility of the given product with respect to the current available tools of the bending machine. If the product is accessible, a feasible unfolding sequence is generated. The tools for each bend in the sequence are selected. If the product is not accessible, all inaccessible components (self-blocking components) are identified and accessible decompositions are determined to break all self-blocking components.

4.2.3 Product Assemblability Checker

The system (or designer) assigns assembly tabs to the decomposed parts of a product after it is decomposed. The contact information between decomposed parts specified by the designer is used to derive the feasible assembly directions among the parts that are in contact. The decomposition is not assemblable if the decomposed product cannot be disassembled considering only the local contacts. However, passing this test does not guarantee global assemblability since further interference checking among non-contact parts needs to be performed.

4.2.4 Design Modification Module

The design modification module adds assembly tabs to the decomposed parts. The tabs are used to provide attachment points for assembly operations such as welding. Tab information can be assigned by the designers, or default settings are used. The part unfoldability and the tool accessibility are taken into account during the tab assignment.

4.3 The Cutting Process Planner

Currently, it is assumed that the cutting operation will be performed by a laser cutter. The cutting cost is assumed to be proportional to the perimeters of the 2D flat patterns and the amount of scrap. A cutting planner (see Fig. 4.4) is developed in the Intelligent Bending Workstation Project. The planner uses pattern profiles to generate a layout which maximizes the length of shared edges and minimizes the convex hull area of the layout.

The cutting planner consists of two modules: the cutting module and the placement (nesting) module (Safier and Grinberg, 1997). The cutting module generates potential layouts and the placement module evaluates the fitness of the layouts based on the area used and the amount of scrap of each layout. The cutting planner returns a near-optimal layout.

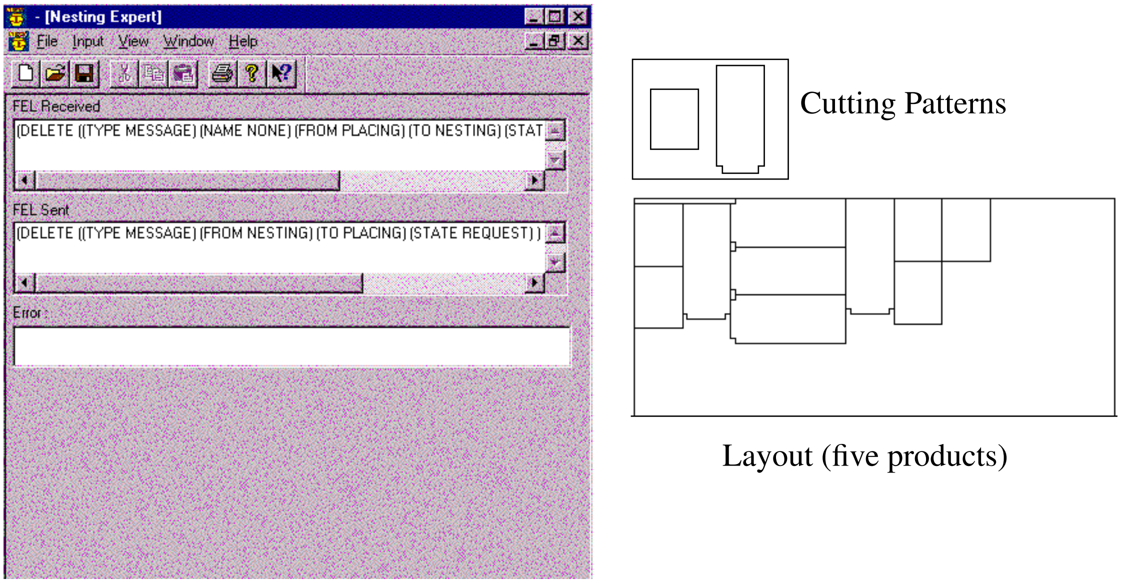


Figure 4.4: Cutting planner (running on Windows NT)

4.4 The Bending Process Planner

An automatic bending process planner has been developed for sheet metal parts (Bourne 1992). This bending process planner consists of five primary sub-systems --an operations planner, a tooling system, a grasping system, a robot motion planner, and an open architecture controller. The system architecture is shown in Figure 4.5.

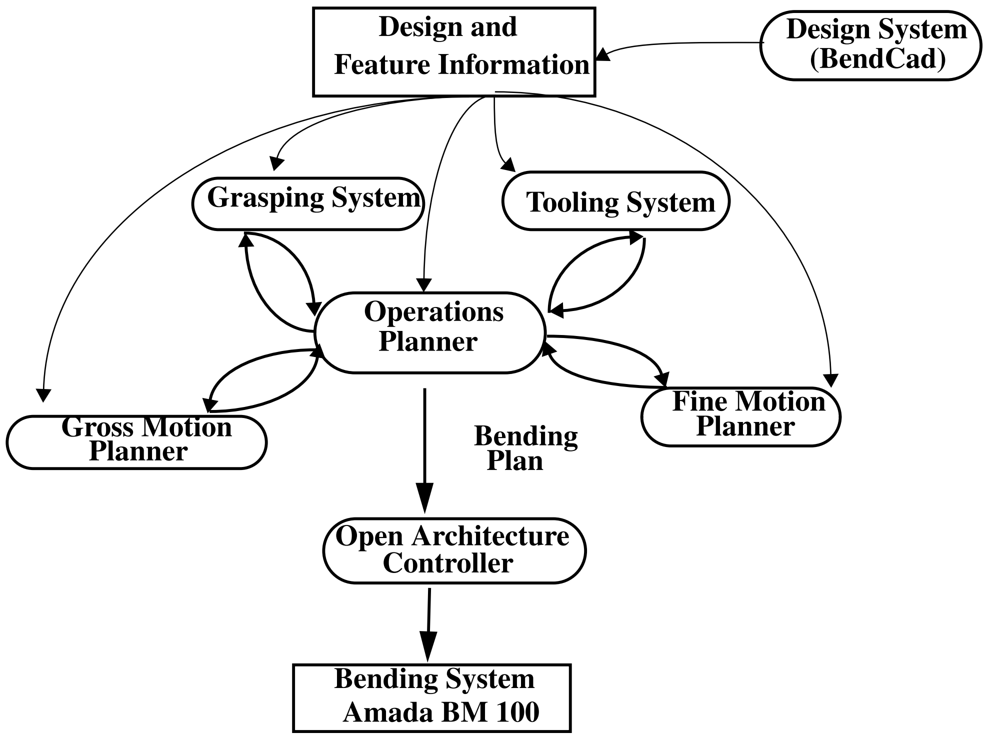


Figure 4.5: Architecture of the automatic bending process planner.

The various modules are described as follows:

***Operations Planner**:* The planner generates possible bending sequences and asks the sub-systems to evaluate the manufacturing costs of these sequences. These costs are measured in terms of manufacturing time. The goal is to generate a bending plan which minimizes the costs.

The search begins with the flat pattern (zero bends) and continually expands the most promising bending sequence until all bends are done. As the determination of a bending sequence is a permutation problem, to make it more tractable, precedence constraints and heuristics suggested by the features are used to prune the search space and guide the search. The precedence heuristics are used to adjust the costs of the nodes, and the precedence constraints are used to limit the breadth of the search tree.

***Tooling System*:** The tooling system, which selects punches and dies, determines the number of tool stages (segments of punch-die pairs), and calculates stage lengths, and performs interference checking between parts and tools. Once the bend sequence has been determined, the tooling system assigns each bend to a stage of tooling and lays out the complete set of tools on the machine.

***Grasping System*:** The grasping system selects the grippers, determines the best grasping position for each bend, and predicts the number of gripper repositions. Repositions are necessary whenever the gripper grasps across unbent bend lines or whenever there is no adequate clearance between the gripper and the bending machine. A reposition gripper holds the part while the robot re-grasps the part at a different location to continue bending. For example, three bends of a large four-bend box can often be made from the same grasping location, but the gripper must be repositioned before the last bend can be made. Repositions are more frequent for small parts; sometimes two or three repositions are necessary, one for each bend.

***Motion Planner*:** The motion planner consists of two sub-systems: the "*Gross Motion Planner*" that determines the transfer motion of the robot, and the "*Fine Motion Planner*" that determines the motion of the robot when the part is inside the punch-die space, particularly the retraction of the part after it is bent.

Open Architecture Controller: The bending machine's controller has been replaced with an off-the-shelf engineering workstation (currently a 100 MHZ PC) that controls all of the low level machine operations. This workstation makes it convenient to integrate all control activities with higher level planning activities, which have been traditionally kept separate. The controller manages all the real-time machine functions, interprets the plan (thus producing the part), and provides feedback so that the plan can be improved on an operation-by-operation basis. In some cases, this feedback process can nearly double the performance of the bending system.

Each sub-system is designed to cooperate with the operations planner in order to develop a near-optimal plan. The controller improves the plan further as parts are produced. A screen dump of how these sub-modules communicated is shown in Figure 4.6.

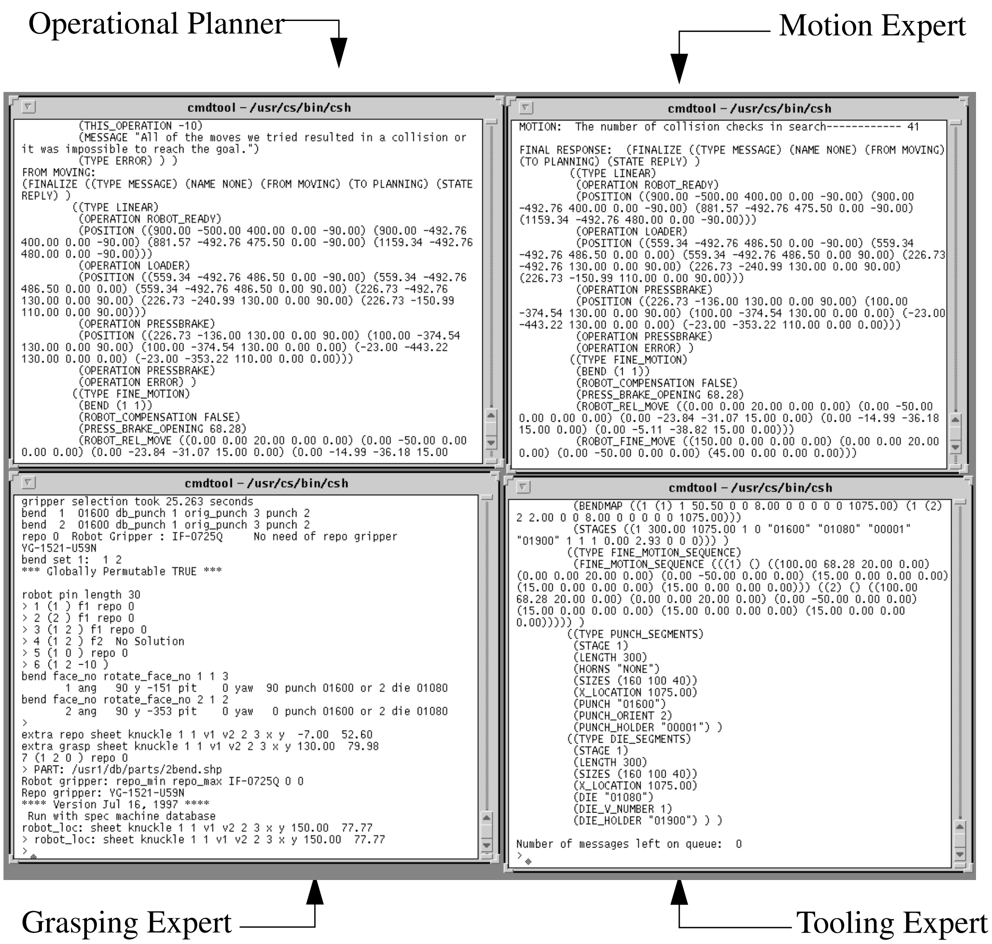


Figure 4.6: Bending process planner.

4.5 The Assembly Process Planner

A primitive assembly planner which considers only translational motions has been developed. Decomposed parts are first converted into Noodles (Gursoz 1991) solid models, considering part thickness. The assembly planner then searches for valid assembly sequences via disassembly. The heuristic of selecting a preferred disassembly direction is to consider the assembly operations of major directions first. The preference is: Z direction beofore (X and Y direction) before other directions. It is assumed that the assembly is performed on an assembly table, and the original configuration of the product is used as the configuration of the assembly. The assembly planner generates the assembly sequence and directions according to the specified mating relations between the decomposed parts. Interference checking among non-contact parts is performed during planning.

Chapter 5

Part Unfoldability

Pattern development involves mapping 3D surfaces into 2D flat patterns. To reduce the complexity of patterns, approximation of the surfaces into polygonal surfaces such as quadrilateral or triangular facets is usually performed. Sheet metal parts are often designed as polygonal surfaces in current practice. The development process considers the unfoldability of the 2D patterns to determine if they can be cut from a flat sheet.

Since unfoldability is an important issue for pattern development, two kinds of unfoldability are studied in this chapter. Geometric unfoldability requires that the developed 2D pattern does not overlap itself; topological unfoldability requires that every face of the part remains connected and planar during bending except for local bending deformation. Geometric and topological unfoldability are treated as constraints during the pattern development process.

In this chapter, an algorithm is developed to flatten polygonal surfaces into 2D patterns. First, a connectivity graph is constructed with faces as nodes and bends as links. By considering the unfoldability and the compactness preference (minimization of the perimeter of the 2D pattern) as link costs, the compact 2D pattern corresponds to the minimum cost spanning tree of the connectivity graph. An A* algorithm is used to search for the compact 2D pattern. This algorithm ensures that the developed 2D patterns will satisfy the geometric and topological unfoldability constraints.

5.1 Previous Work

Pattern development has been studied extensively. It is known that a surface is unfoldable (developable) if the Gaussian curvature of every point in the surface is zero (Faux and Pratt, 1981). For instance, plane and single-curved surfaces are known to be always unfoldable, while double-curved surfaces are not. Manning (1980) proposes an approach based upon an isometric tree to map curved surfaces onto a plane for shoe manufacture. Hindes et al. (1991) consider the Gaussian curvature to develop patterns for garment pieces.

Elber (1995) proposes an algorithm using a set of developable ruled surfaces to approximate a free-form surface. Paper models are assembled using these ruled surface patterns. This approach takes geometric unfoldability into account. However, several pieces are usually needed to make a model. Yeh et al. (1995) propose an unfolding approach that considers topological unfoldability. This approach first determines the topological relationship from a wire-frame representation of the part. The unfoldability of a part can then be determined by examining the corresponding connectivity graph. However, the geometric overlapping problem of 2D patterns is not addressed.

5.2 Pattern Development

During the development of 2D patterns, unfoldability is an important factor to certify the validity of the designs (3D shapes). There are two types of unfoldability to be considered: topological unfoldability and geometric unfoldability. Topological unfoldability requires that faces of the part remain connected with no ripping, while geometric unfoldability ensures that the 2D pattern does not overlap itself and will not be distorted during bending operations.

Essential collinear bends are bends that must be bent simultaneously or distortion or ripping will occur. The bend graphs of parts with essential collinear bends correspond to graphs with cycles instead of trees. To identify essential collinear bends, the positions of bends and the bending requirements such as bend radii and bend angles, must be examined first. A cut along a set of essential collinear bends results in exactly two components. The 2D connectivity graph of an unfoldable part with essential collinear bends should be transformed into a tree by grouping the corresponding essential collinear bends and faces. An example is shown in Figure 5.1. After all the essential collinear bends have been identified and grouped, the connectivity graph of an unfoldable part corresponds to a tree (see Figure 5.1).

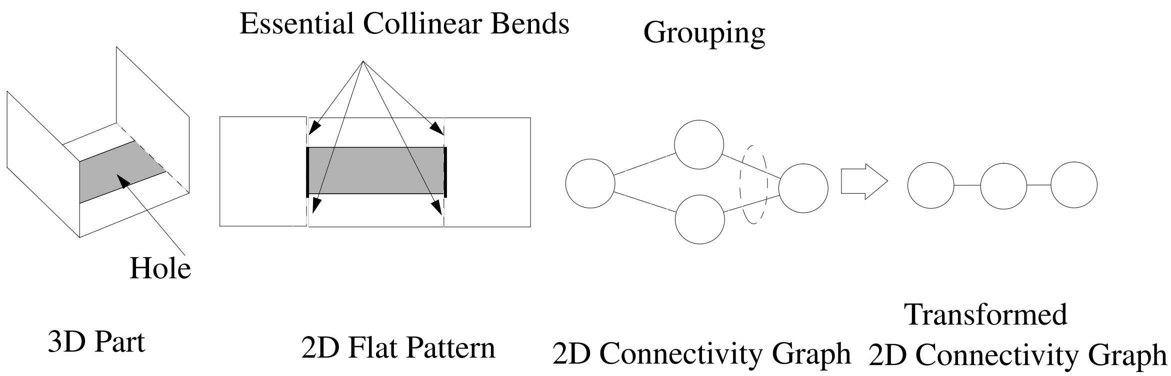


Figure 5.1: A part with essential collinear bends.

Topological unfoldability is one necessary condition for the development of a valid 2D pattern. However, even if a part is topologically unfoldable, geometric unfoldability needs to be checked. Global geometric unfoldability demands that all faces of the 2D pattern do not overlap, while local geometric unfoldability only requires that the faces of a locality do not overlap. Figure 5.2 shows two parts that satisfy the topological unfoldability constraint (i.e. the bend graphs all correspond to trees). However, the part in Figure 5.2(b) fails to satisfy the geometrical unfoldability constraint because the flange does not fit in the hole. Global geometric unfoldability cannot be checked before the 2D connectivity graph is resolved. However, some properties that are related to local unfoldability can be used to validate the designs and to develop the 2D patterns more efficiently.

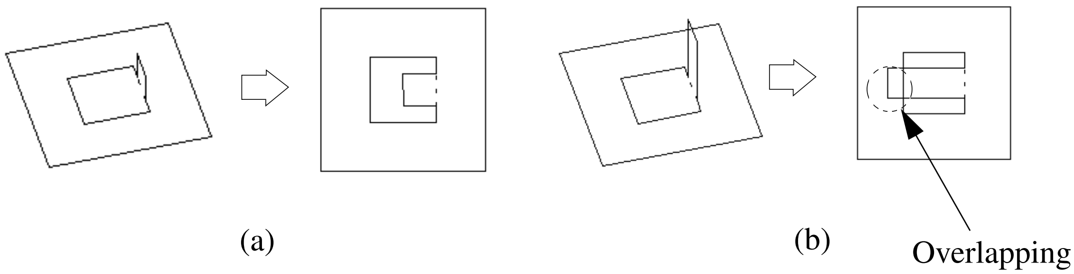


Figure 5.2: Geometric unfoldability analysis: (a) is unfoldable and (b) is not.

5.3 Unfoldability Evaluation

5.3.1 Geometric Unfoldability

Geometric unfoldability is further classified into global geometric unfoldability and local geometric unfoldability.

- ***Global Geometric Unfoldability.*** Global geometrical unfoldability demands that all faces do not overlap during development. The complexity is O(N^2), where N is the number of faces, because every face has to be checked against every other face.
- ***Local Geometric Unfoldability.*** Before local geometric unfoldability is studied, some terms need to defined. Corners are common features for sheet metal parts/products. A corner is defined as the intersection of three or more connecting faces. In Figure 5.3, the angle $\Delta\theta$ is the angle between two non-connected faces around the developed locality of the corner C_p. If $\Delta\theta > 0$, the corner is locally unfoldable (no local overlapping). Otherwise, overlapping will occur ($\Delta\theta < 0$). Overlapping corners are corners the sum of whose face angles is greater than 2π. During the development of 2D patterns, any formation of overlapping corners will violate geometric unfoldability around the locality of the corners and should be avoided. For instance, in Figure

5.4(b), face 1 needs to be separated completely from the overlapping corner (2 cuts needed in the graph). For the part in Figure 5.4(a), only one cut in the graph is needed.

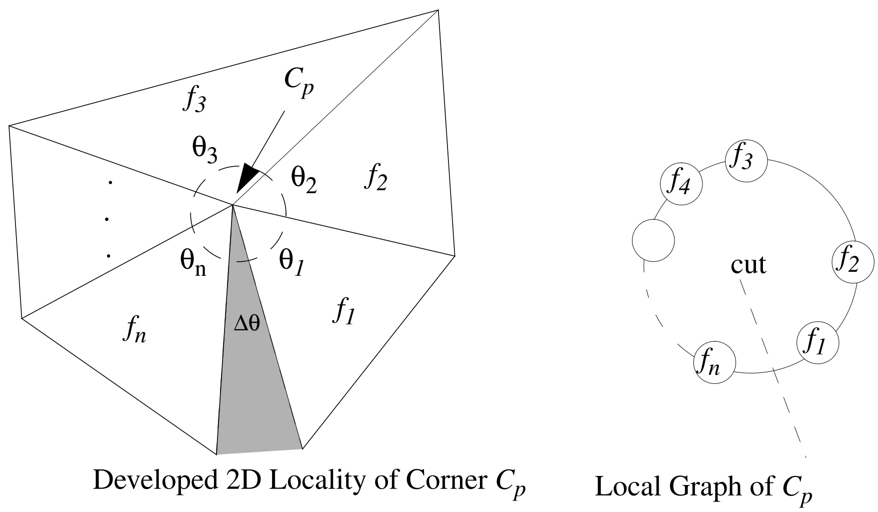


Figure 5.3: Local unfoldability of corner *Cp*.

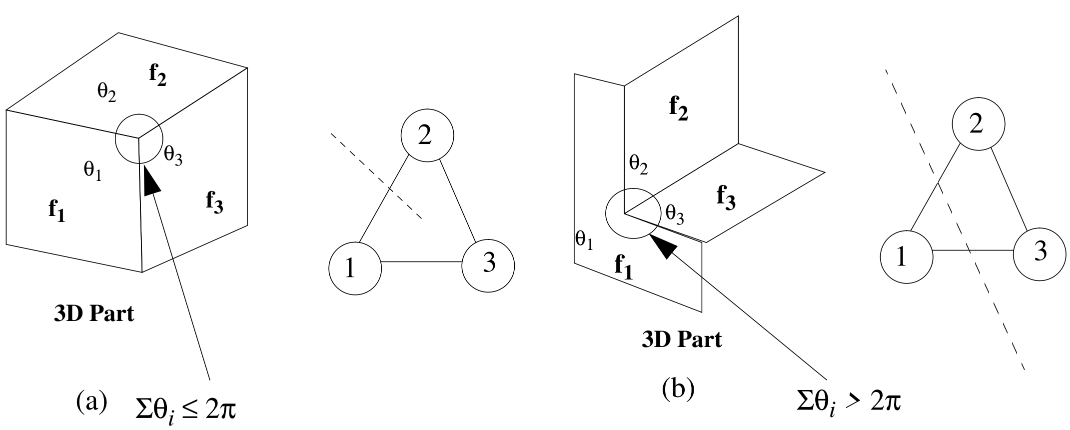


Figure 5.4: Corners. (b) is an overlapping corner and (a) is not.

Checking global geometrical unfoldability is computationally expensive. Therefore, local geometrical unfoldability (considering the faces around the corners only) of overlapping corners is considered during development. The resulting 2D pattern will be tested for global geometric unfoldability (considering all faces) after the development procedure is complete.

5.3.2 Topological Unfoldability

Topological unfoldability requires that the pattern remain connected; no distortion or ripping is to occur during and after the development process. This constraint implies that the connectivity graph of the resulting pattern corresponds to a spanning tree of the connectivity graph. If the connectivity graph corresponds to a graph with cycles, some of the bends must be cut in order to make the bend graph into a tree. Every link in that tree represents a bend of the part. Each bend will divide the faces into exactly two groups. As a result, the faces in these two divided groups are free to move during bending. Thus, all faces remain planar and connected during bending operations.

5.4 Pattern Selection

Some designs suggest ambiguous (one-to-many) mappings from 3D shapes to 2D patterns. Figure 5.5 shows six different 2D patterns that can all be bent into the same 3D shape and connectivity (based on the 3D edge-face relation). For parts with a multiple choice of 2D patterns, each pattern corresponds to a spanning tree of the unresolved bend graph. The complexity of finding all spanning trees of a graph is exponential in n, where n is the number of nodes in the graph. It is not practical to enumerate all the spanning trees and evaluate their fitness. In selecting the desirable 2D pattern, some preferences should be considered to make the selection more tractable. It is assumed that the faces of the parts can only be separated along connecting edges. No splitting of faces is allowed during development. It has been found that compactness of 2D patterns is usually desirable. This property directly relates to lower costs in cutting and less drooping during bending. Two types of compactness are considered: geometrical compactness and topological compactness.

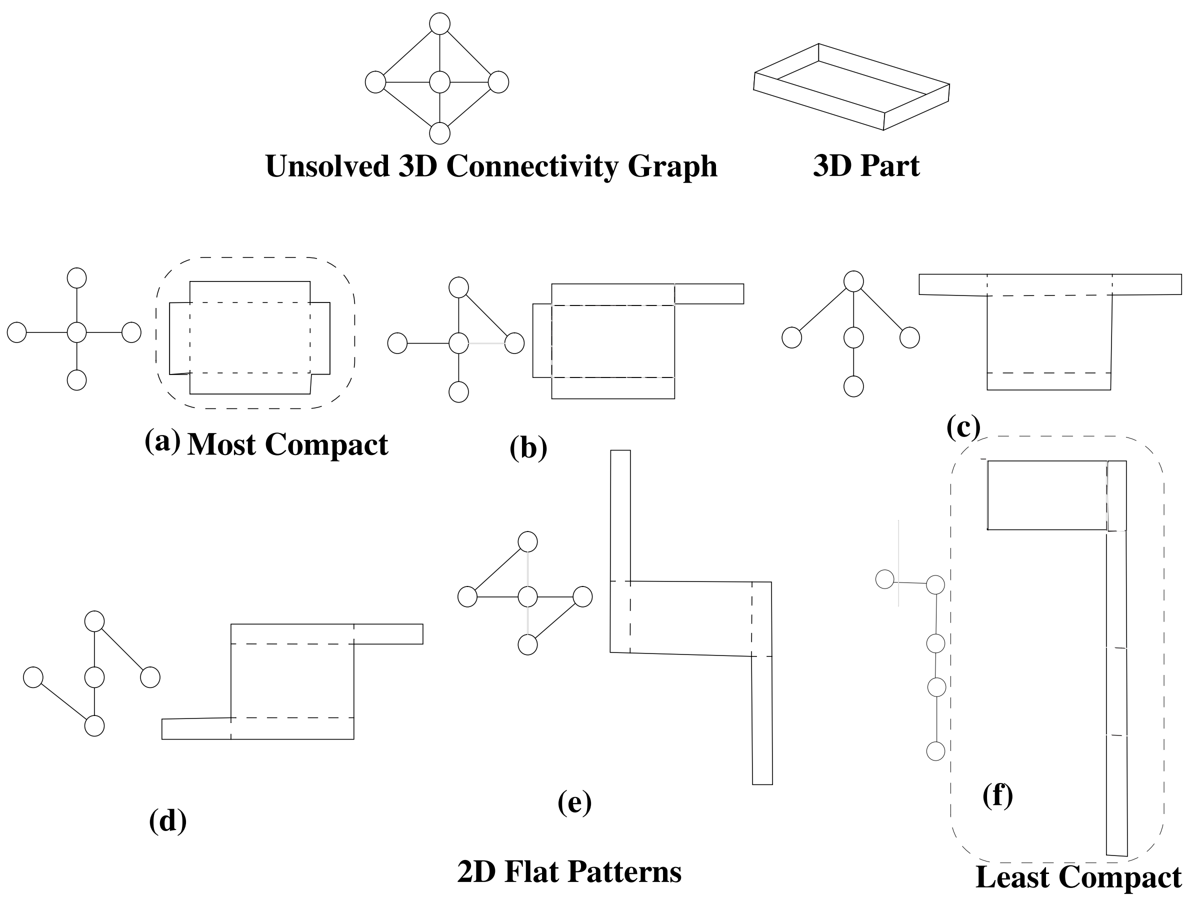


Figure 5.5: Five different 2D patterns that can be bent into the same 3D shape.

5.4.1 Geometric Compactness

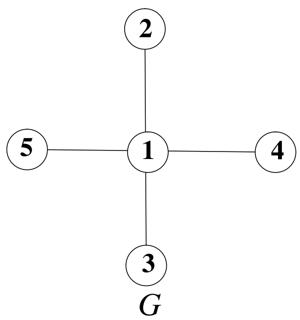
Geometric compactness is defined as (area/perimeter^2) of the 2D pattern, which is similar to Ballard and Brown's (1982) definition. According to this definition, minimizing the perimeter of the 2D pattern also maximizes the compactness of the pattern. This means that during development, longer connecting edges are always considered for connection first. This preference reduces the cutting cost. The compactness property is used in the pattern development approach as the measure of fitness. In the case of links that are the same length, further topological compactness has to be considered.

5.4.2 Topological Compactness

Although geometrical compactness usually dominates the selection of patterns, for links of the same weight (i.e., connecting edges are the same length), topological compactness needs also to be considered.

Before topological compactness is outlined, some terms need to be defined. The *distance* between two vertices v_i and v_j, $d(v_i,v_j)$ in a graph G is defined as the length of the shortest path between these two vertices of the graph (Deo 1974). The *eccentricity* of a vertex v, $E(v)$ in a graph G is defined as the maximum distance between v and the farthest vertex in G (Deo 1974). It gives a measure of how close the given vertex is to the farthest vertex in the graph. Figure 5.6 gives the distances and eccentricity for a tree.

A measure of topological compactness is defined as the reciprocal of the sum of the eccentricity of all vertices in the current developing tree. As it is desirable to have a tree with maximum topological compactness, this definition aims to minimize the sum of eccentricity of the developing tree. Three possible spanning trees and their topological compactness are shown in Figure 5.7. The one with the largest topological compactness is usually the most compact in topological aspects. That is, the link which gives the current developing tree the maximum topological compactness should usually be selected during pattern development.



$$d\,(V_1,V_2) = d(V_1,V_3) = d(V_1,V_4) = d(V_1,V_5) = 1$$
$$d(V_2,V_3) = d(V_2,V_4) = d(V_2,V_5) = 2$$
$$d(V_3,V_4) = d(V_3,V_5) = 2$$
$$d(V_4,V_5) = 2$$
$$E(V_1) = 1$$
$$E(V_2) = E(V_3) = E(V_4) = E(V_5) = 2$$
$$\sum E(V_i) = 9$$

Figure 5.6: Distance and eccentricity of a tree.

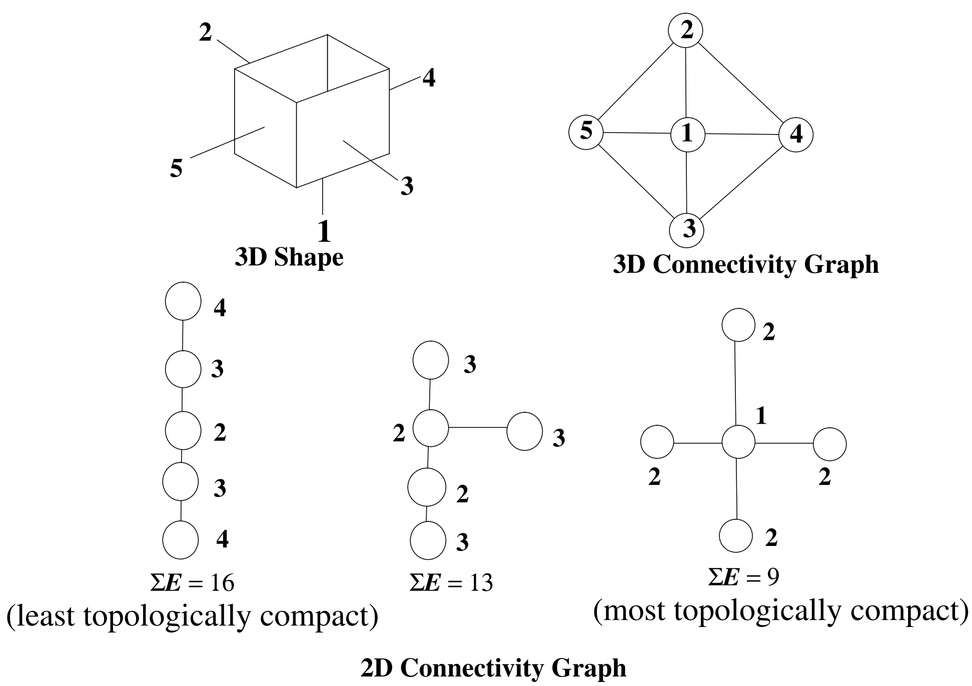


Figure 5.7: Different 2D patterns and their topological compactness.

5.5 Modeling of Pattern Development

To model the development of 2D patterns, constraints and preferences in geometric and topological aspects are taken into account. Weights are assigned to the links of the unresolved connectivity graph of the 3D part. The objective is to minimize the perimeter of the resulting 2D pattern while maintaining the topological and geometric unfoldability. In other words, the goal is to maximize the total length of connecting edges. If the weights of links are proportional to the negative length of the connecting edge, the pattern development can be transformed into a minimum cost spanning problem. Topological unfoldability is preserved during the generation of the spanning tree. Global geometrical unfoldability can be checked during development in $O(N^2)$, where N is the number of faces in the current spanning tree. Since this is computationally expensive, local geometric unfoldability (considering the faces around the corners only) of overlapping corners is considered during development. The resulting 2D pattern is evaluated for its global geometric unfoldability (considering all faces) after the development procedure is complete.

This approach finds the most compact spanning tree by adding new nodes (links) to the current tree, which is similar to Kruskal's algorithm (1956). Standard minimum cost spanning tree algorithms such as the ones proposed by Prim (1957) or Kruskal (1956) generally do not handle constraints in the search. Therefore, an A* algorithm has been implemented to search for the most compact 2D pattern. In cases where there are links of the same weight (i.e., the connecting edges are the same length), topological compactness is considered.

The A* algorithm uses the difference between the total number of nodes and the number of nodes in the current tree as its heuristic function (*h* cost). It is designed to encourage promising paths to go deep. As a result, this algorithm does not guarantee global optimality. However, it generates satisfactory results in a reasonable amount of time, especially for parts with some connecting edges that are the same length. To further speed up the development process, the algorithm caches previous overlapping nodes. This knowledge is used to eliminate global overlapping and potential overlapping during further node expansion.

5.6 Pattern Development Algorithm

The development algorithm has been implemented with the following steps:

1. Generate the connectivity graph corresponding to the 3D part.
2. Group essential collinear bends and transform the original connectivity graph.
3. Identify overlapping corners.
4. Decompose the links that fail on-bend test.
5. Assign weights to the links of the connectivity graph.
6. Search for the minimum cost spanning tree and unfold the pattern based upon the resulting tree of the compact 2D pattern.
7. Check global geometric overlapping of the resulting 2D pattern. Backtrack if necessary.

5.7 Examples

Several developed 2D patterns are shown in Figure 5.8 and Figure 5.9. Figure 5.8 gives some closed polygonal shapes with no overlapping corners. In Figure 5.9, three sheet metal components of the bending robot are used as examples with overlapping corners. Table 2 gives the computation time of various 2D patterns. The results were tested on a Sun Sparc 10 workstation. Several paper models were made from these developed 2D patterns (Figure 5.10). All models were made from a single sheet and assembled by hand. Bend deduction (change in dimension during bending) is neglected in any example.

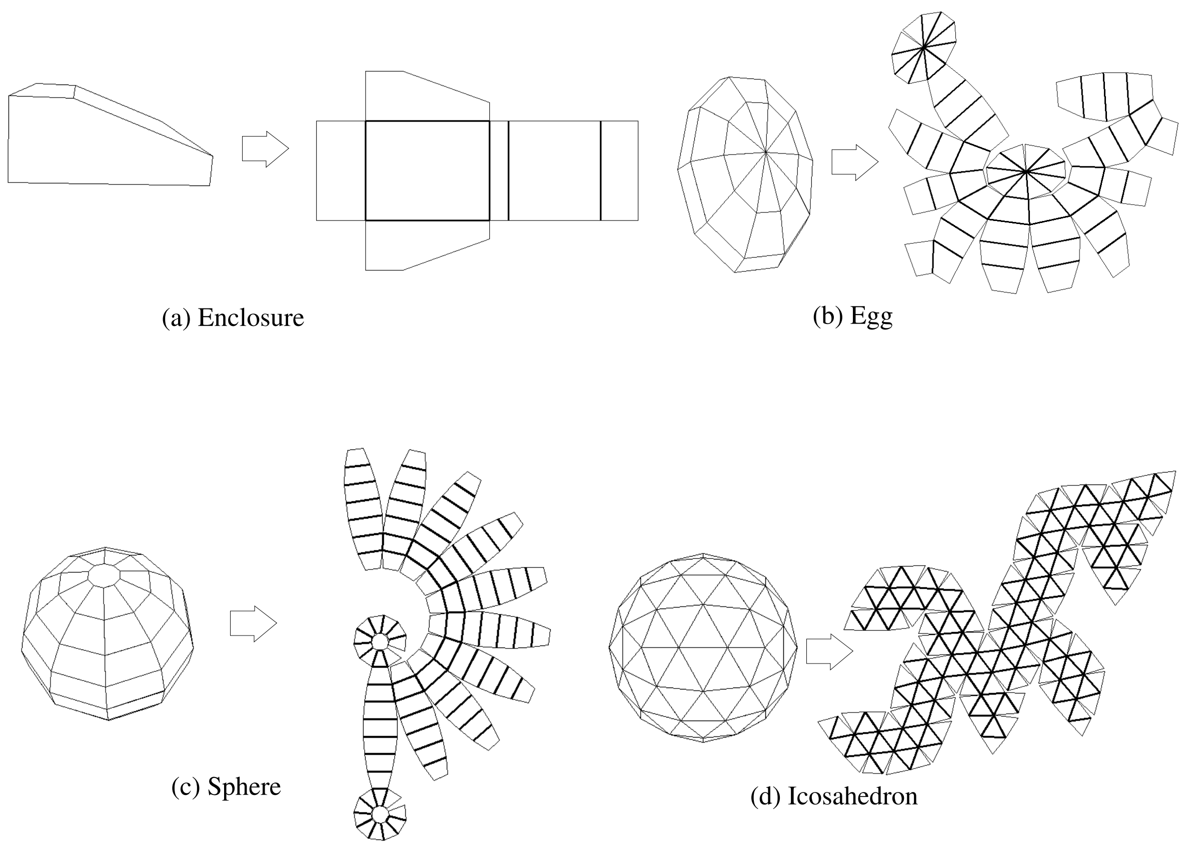


Figure 5.8: Developed 2D patterns with no overlapping corners.

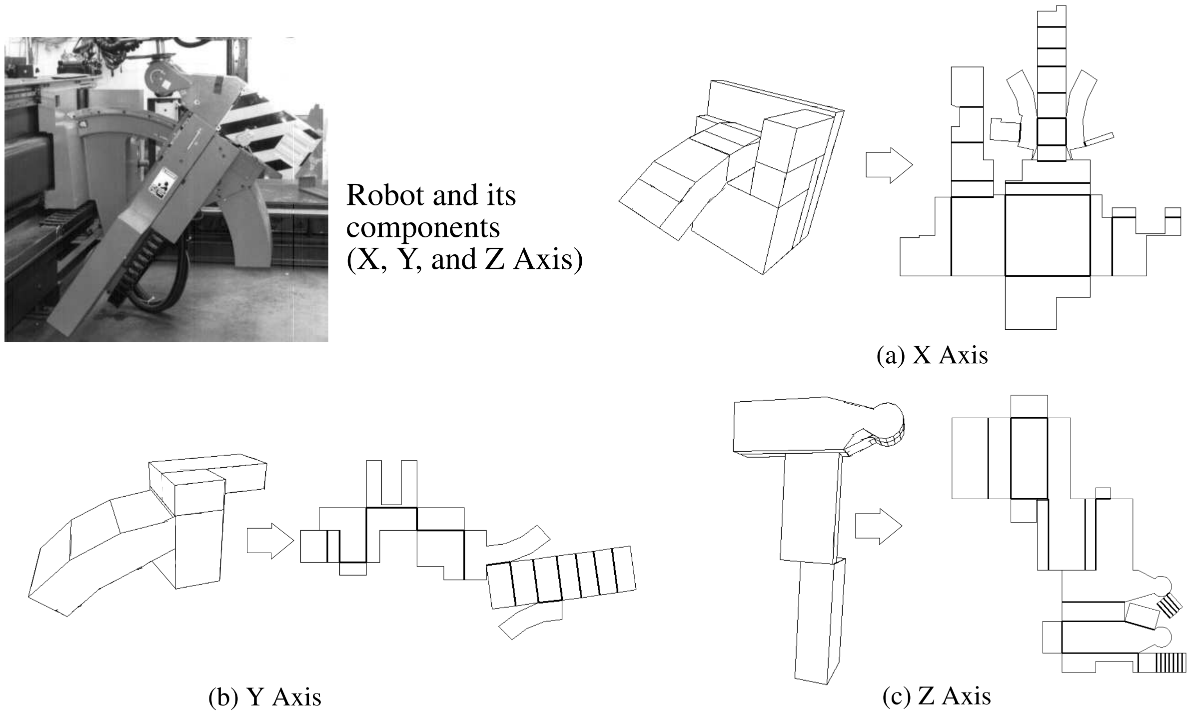


Figure 5.9: Developed 2D patterns with overlapping corners.

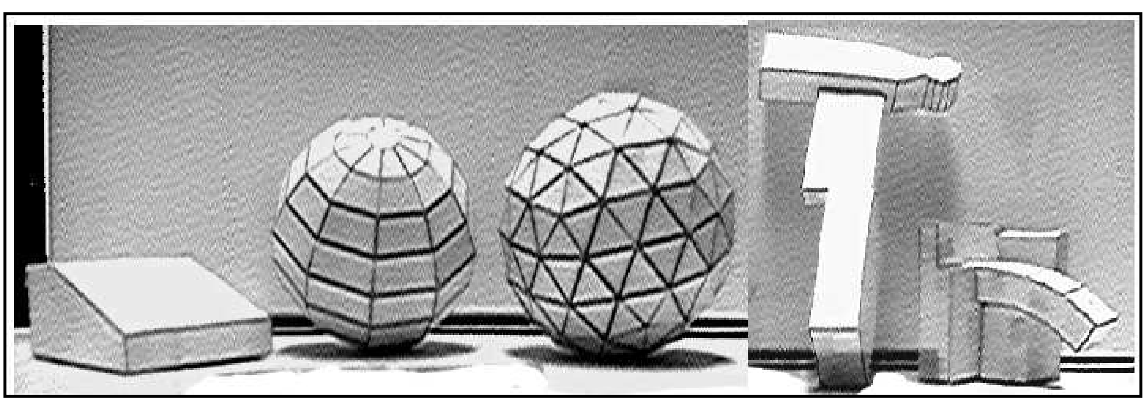


Figure 5.10: Paper models made from the developed 2D patterns.

Models	No. of Faces	No. of Links	No. of Overlapping Corners	Computation Time (Sec.)
Electric Enclosure	7	15	0	1
Egg	60	110	0	18
Sphere	92	190	0	62
Icosahedron	180	270	0	257
X Axis	25	75	23	9
Y Axis	17	44	11	7
Z Axis	29	79	12	18

Table 2. Pattern development results

5.8 Non-Unfoldable Parts

There are parts which are not unfoldable considering geometric and topological unfoldability. Figure 5.11 shows some examples of parts which cannot be unfolded. Unfortunately, there is no theorem or efficient algorithm that can tell if a given 3D shape is unfoldable or not. However, there are some ways to detect if a shape is not unfoldable. For example, considering one bend (and two faces connected by the bend) at a time, if this one-bend part cannot be unfolded, the part cannot be unfolded into a single piece without overlapping. However, this one-bend test only covers some non-unfoldable cases.

Chuang and Huang (1996) propose an approach to decomposing a 3D shape into unfoldable components. They define protrusion and depression features, and based upon these features, they cut along what they call “concave edges” before development. Their approach is conservative and sometimes can result in more components (parts).

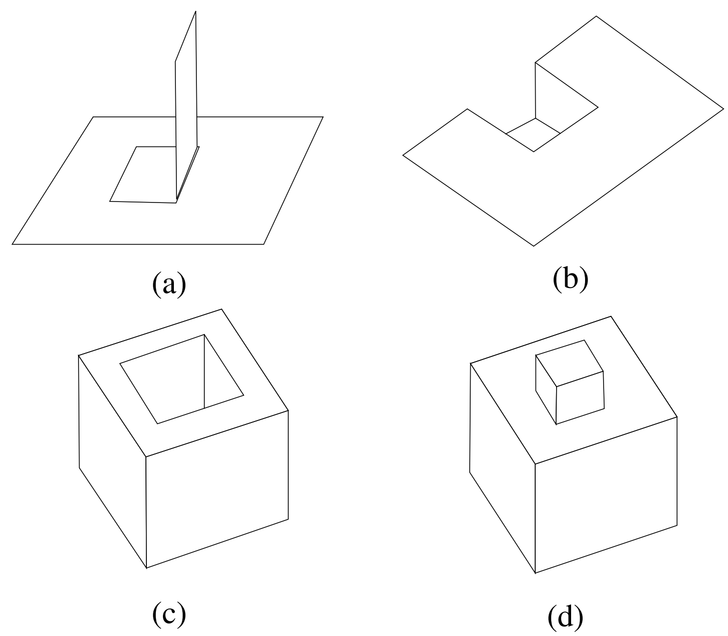


Figure 5.11: Parts that cannot be unfolded into single pieces.

5.8.1 Decomposition for Unfoldability

The pattern development approach proposed here works for parts that can be unfolded into a single piece. For non-unfoldable parts, further decomposition is needed. In order to minimize the number of decomposed parts, we define *overlapping edges* as the edges (links) connected to an overlapping corner. The pattern development algorithm can choose to decompose these overlapping edges before the search. This algorithm usually results in fewer parts than Chuang and Huang's approach. Figure 5.12 shows how the approach presented here decomposed the non-unfoldable parts into unfoldable components. The link in Figure 5.12(a) is decomposed because it fails the one-bend test. The rest of the parts in Figure 5.12 are obtained by decomposing the overlapping edges. The overlapping edges are marked with thick lines and the overlapping corners are circled in the figure.

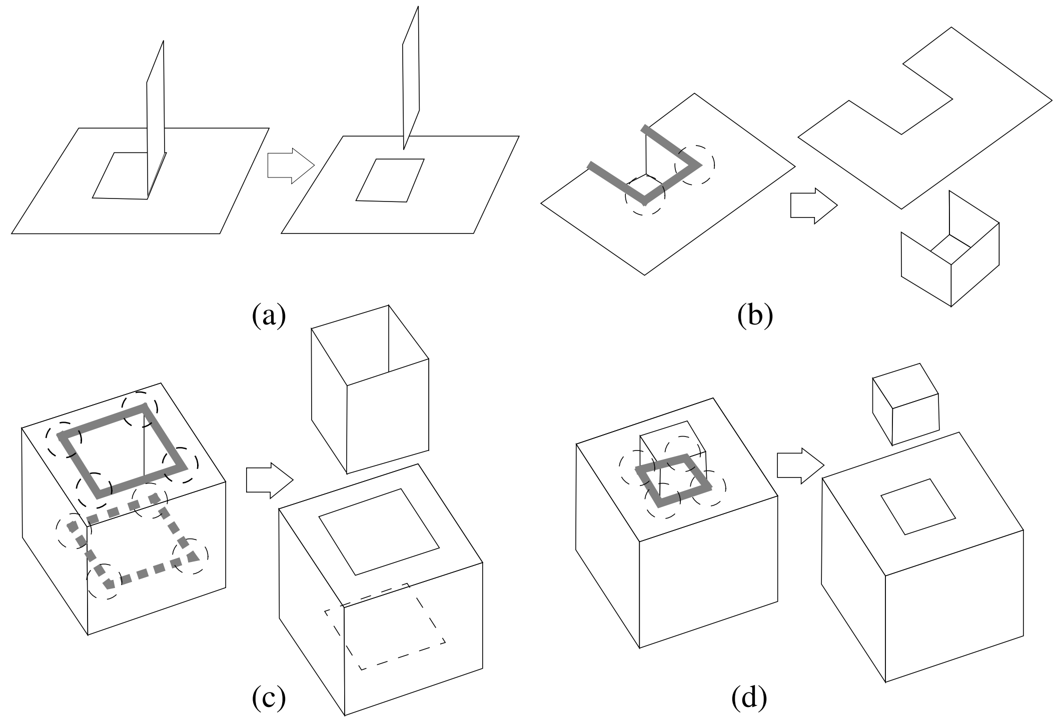


Figure 5.12: Decomposition for unfoldability.

5.9 Conclusion

2D pattern development is the first step in the production of bent sheet metal parts. This chapter proposes an algorithm for pattern development that considers topological and geometric unfoldability. This approach generates 2D patterns subject to a compactness preference. Some local unfoldability properties are used to make the development process more tractable.

Various parts have been tested, and it has been found that if the parts are convex (i.e., no overlapping corners, see the parts in Figure 5.8), they tend not to have global geometric overlapping after development. We state this as a conjecture for proof. For parts with overlapping corners (see Figure 5.9), global overlapping is mostly likely to occur. Since this approach only considers local unfoldability during development, it is possible that the resulting 2D patterns do not satisfy the global geometric unfoldability constraint. As a result, backtracking is sometimes needed. The existence of overlapping corners in the parts usually increases the incidence of backtracking, especially when the lengths of the connecting edges around the corners are smaller than other edges (see the Z axis in Figure 5.9(c)). For parts with some edges that are the same length, such as the sphere and icosahedron in Figure 5.8, topological compactness proves to be a good heuristic. Although the algorithm presented here does not guarantee global optimality, it finds satisfactory results in a reasonable amount of time. For parts which cannot be unfolded into a single piece without overlapping, an approach based on the overlapping corners and overlapping edges of the part is given. This approach decomposes the part into unfoldable components. The most compact 2D flat pattern is then obtained for each component.

In this chapter, the geometric and topological constraints of 2D pattern development are studied. Bent sheet metal parts are used as examples to illustrate the approach. Constraints are imposed by the manufacturing process that are not considered in our approach. For instance, the bendability of parts requires the availability and accessibility of tools for all bends. This constraint may reject the 2D patterns generated by our approach. Parts that form closures cannot be bent since the punch and die cannot access all bends. As a result, the parts may need to be decomposed into bendable components before they can be unfolded. This issue will be discussed in the next chapter.

Chapter 6

Tool Accessibility

The bending process is a material forming procedure that deforms sheet metal into a desired shape. A bending operation involves making one or several bends at a time. The tooling used during the bending operation includes a set of punches and dies. A major issue in planning bend operations is to determine the feasible bend sequence and select the appropriate tooling. Tool accessibility is a key step in forming the bend sequence. In this thesis, to simplify the evaluation of the bendability of a given sheet metal part, tool accessibility is used to determine the manufacturability of a bending process, although it is not the only concern in manufacturability.

6.1 Bending Process Planning

Typically, bending is performed using a press brake. A press brake is usually operated by a CNC controller. A common automatic bending cell consists of a CNC press brake, a device that handles material during bending (i.e., a robot), a gaging system to ensure bend orientation, and a loading/unloading device for 2D blanks and finished parts. A complete bending plan includes the following components:

- A bending sequence
- Tooling (punches and dies) selection and set-up
- Robot gripper selection and grasping positions*
- A robot motion plan*
- A backage motion plan*
- A loading/unloading plan*
- *: necessary if an automatic bending cell is used.

One of the goals of product decomposition is to guarantee that the workpiece that does not interfere with the tooling before, during, or after bending. For a manually operated and automatic bending machine, the bend sequence determination and the tooling selection are always necessary.

6.1.1 Previous Work

De Vin et al. (1992) develop a computer aided process planning system for sheet metal components called PART-S. The system uses a backward (unfolding) approach to generate the bend sequence. That is, the system tries to find a feasible unfolding sequence and then reverse that sequence. Bourne (1992) develops a planning system for an automatic robot-assisted bending workstation. The system consists of an operational planner and four sub planners: a tooling planner, a grasp planner, a backgage planner and a motion planner. These planners work cooperatively to produce a near-minimum cost manufacturing plan. However, their approach toward determining the bend sequence is forward, i.e., the system starts with the 2D blank and searches for the best bend sequence by folding the bends.

Radin et al. (1996) propose a two-stage backward planning algorithm for bend sequence determination and tool assignment. The algorithm first finds a feasible solution based on collision avoidance. Then, in the second stage, the algorithm seeks an alternative bend sequence with a lower cost. It uses an A* search algorithm and a set of heuristics that are based on bend position, bend line length, bend angle, bend flatness, and reduction of the sum of parallel bend angles. However, these heuristics might mislead the search for some particular parts and the interactions (conflicts) between heuristics remain to be resolved. Cser et al. (1991) use a case-based learning approach to classify sheet metal parts for feasible bending sequences. Inui and Kimura (1991) relate product models to features and processes for bending operations. Expert knowledge on blanking and bending are represented as production rules to make the generation of the plan more efficient. However, these two approaches might be limited by the lack of similar cases and knowledge.

6.2 Bend Sequence Planning

The bend sequence determination is the major component of a bending plan. All other components of the bending plan are based on the bend sequence. Either of two approaches to bend sequence planning are usually used: a forward planning approach or a backward planning approach. The forward planning approach starts with a 2D flat pattern and searches for a *folding* sequence that avoids tooling interference during the folding process. The backward planning approach, in contrast, works from the final 3D shape and searches for an *unfolding* sequence that avoids tooling interference during the unfolding process.

The reverse of an unfolding sequence will yield a feasible folding sequence.

Both approaches expand the search graph during the search, but the search direction is different in each case. The search expands the search graph if the current node is interference-free. The determination of a bend sequence is a combinatorial problem and has the complexity of $O(N!2^N)$, where each bend can have two possible holding sides and N is the number of bends in the part. The number of gripper reposition[1] operations is not included.

Figure 6.1 shows a sample sheet metal part which will be used to illustrate these two planning approaches.

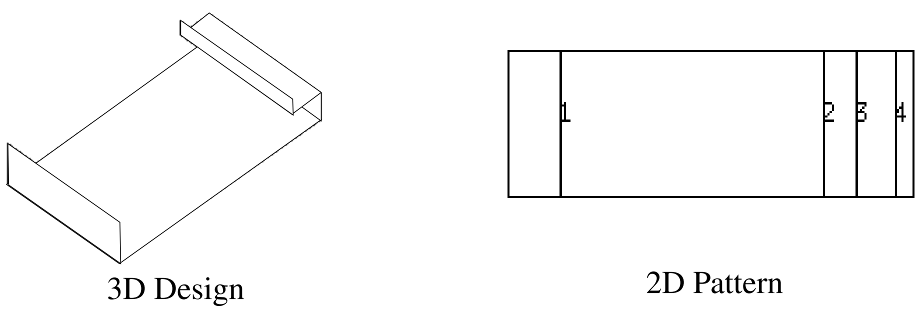


Figure 6.1: Example part.

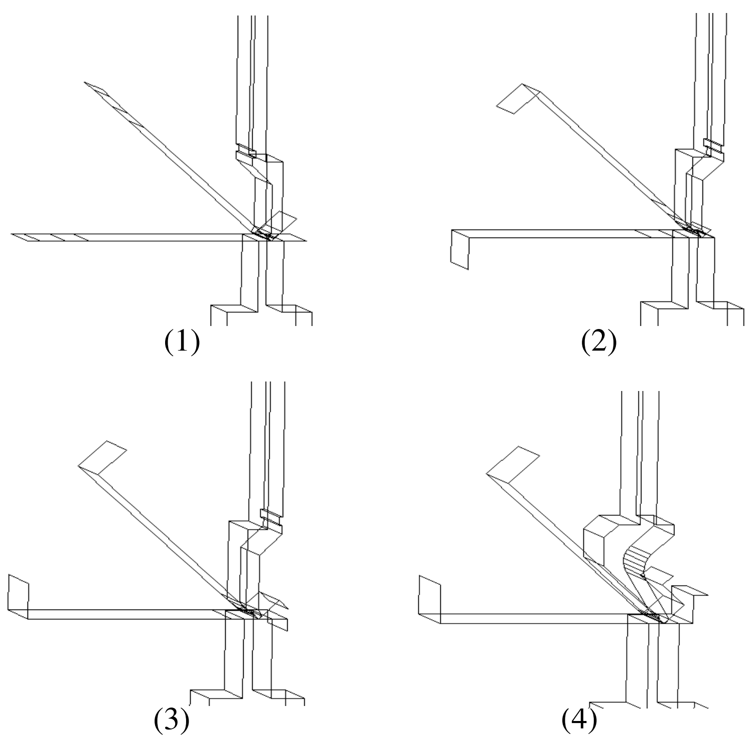
6.2.1 The Forward Planning Approach

The forward planning approach starts with the 2D flat pattern (no bend has been bent) and adds one bend at a time, considering all possibilities until all bends are done. This approach is similar to the actual bending process. Representative research for this approach is by Bourne (1992). The main reason to adopt the forward planning approach, in Bourne's opinion, is that the number of repositions for grasping can be accurately pre-

1. A gripper reposition for a bending robot is necessary if no further bending operations can be performed.

dicted. Gripper repositions are necessary if the rest of the bends cannot be made without changing the current grasping position. Reposition is considered expensive during bending operation. As a result, the number of repositions has significant influence on the optimality of the bending plan. However, Bourne recognizes the geometric constraints that available during backward planning as so constraints are generated to prune the forward search tree.

Figure 6.2 shows the folding sequence of the example part using the forward planning approach. Bend index (**B_i**) and tool index (**T_i**) are included in the sequence. The holding face and tool orientation are also determined.



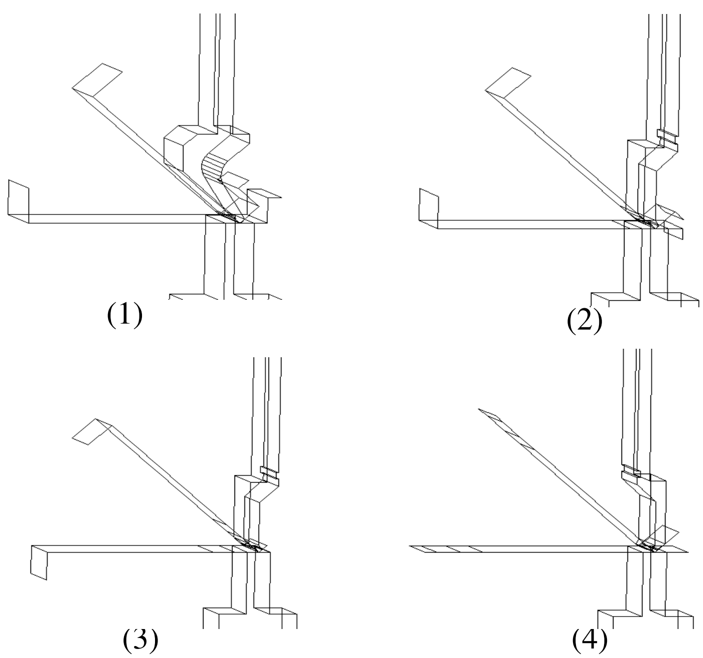
Folding sequence:((B_1:T_1)->(B_4:T_1)->(B_3:T_1)->(B_2:T_2))

Figure 6.2: Forward planning approach for the part in Figure 6.1.

6.2.2 The Backward Planning Approach

The backward planning approach uses the opposite search direction from the forward planning approach. It starts with the final shape (all bends are bent) and subtracts one bend at a time, considering all possibilities until all bends are unfolded. De Vin et al. (1992) use the backward approach in their bending planning system--PART-S. The backward approach tends to have fewer nodes in its search, since the interference between the work-piece and the tooling is most likely to happen at later bends. Bends which come later in the bend sequence are the first bends which a backward planning approach will unfold.

An unfolding sequence of the example part using the backward planning approach is shown in Figure 6.3.



Unfolding sequence: $((B_2{:}T_2)\text{->}(B_3{:}T_1))\text{->}(B_4{:}T_1)\text{->}(B_1{:}T_1))$

Figure 6.3: Backward planning approach for the part in Figure 6.1.

6.2.3 Preference for Bend Sequence Planning

In order to arrive at a better and faster bending plan, it is desirable to consider the following preferences during the planning stage:

- **Grasp the largest side.** This preference makes the grasping, or workpiece handling much easier, and reduces the possibility of the workpiece colliding with the press brake or the tools.
- **Bend adjacent bends first.** A desired bending plan always tries to minimize the manufacturing time. Bending adjacent bends first will reduce the material handling time between consecutive bends.
- **Use a standard tool.** Using standard tooling reduces the tooling cost, and makes the standardization of product components possible. Also, by using the same standard tooling, it is also possible to reduce the number of tooling stages[1] setup in the machine.

6.3 Tool Accessibility Analysis

To evaluate the bendability of the decomposed product, a tool accessibility analysis should be performed on all parts of the product. The tool accessibility analysis serves as a feasibility checker for bendability. In this analysis, the backward planning approach is used to search for an interference-free unfolding sequence. This approach selects the tooling and generates a simplified plan that takes into account the unfolding sequence preferences. These preferences are described in the next section.

---

1. A tooling stage is a contiguous collection of tooling (punch and die) that is assigned to one or more bends in a part.

6.3.1 Precedence Preferences for Tool Accessibility Analysis

To speed up the search for a feasible unfolding sequence, an additional set of precedence preferences are used. These preferences in backward planning are: outside bends later, tall bends first, and internal tab bends first. These are combined with the bending precedences described in Section 6.2.3.

Outside bends are classified as outside with respect to the grasping position. For instance, in Figure 6.4(a), bend 1 is marked as an outside bend, and bend 2 is marked as an inside bend. If outside bends are unfolded later, the workpiece most likely will not interfere with the tools with later bends.

Tall bends are defined as tall with respect to the flange heights. Unfolding taller bends earlier in the unfolding sequence tends to result in less interference in the later bends.

Internal tab bends are bends connected to holes (Figure 6.4(b)). It is desirable to unfold internal tab bends first to avoid interference.

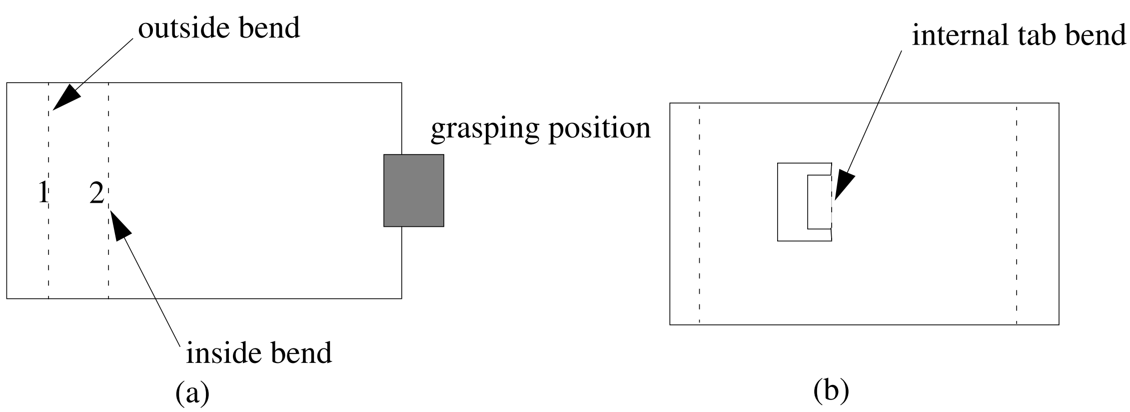


Figure 6.4: Inside bend, outside bend, and internal tab bend.

These preferences guide the search effectively to a feasible unfolding sequence. However, they can also mislead the search because of interactions among them. In our previous work (Wang and Bourne, 1995), we have discussed feature interactions during bend sequence planning and how to resolve these interactions. A channel part which violates the "outside-bend-later" precedence is shown in Figure 6.5.

These feature interaction problems frequently cause search to backtrack. One way to resolve these problems is to identify these interactions and assign different heuristic costs during the search.

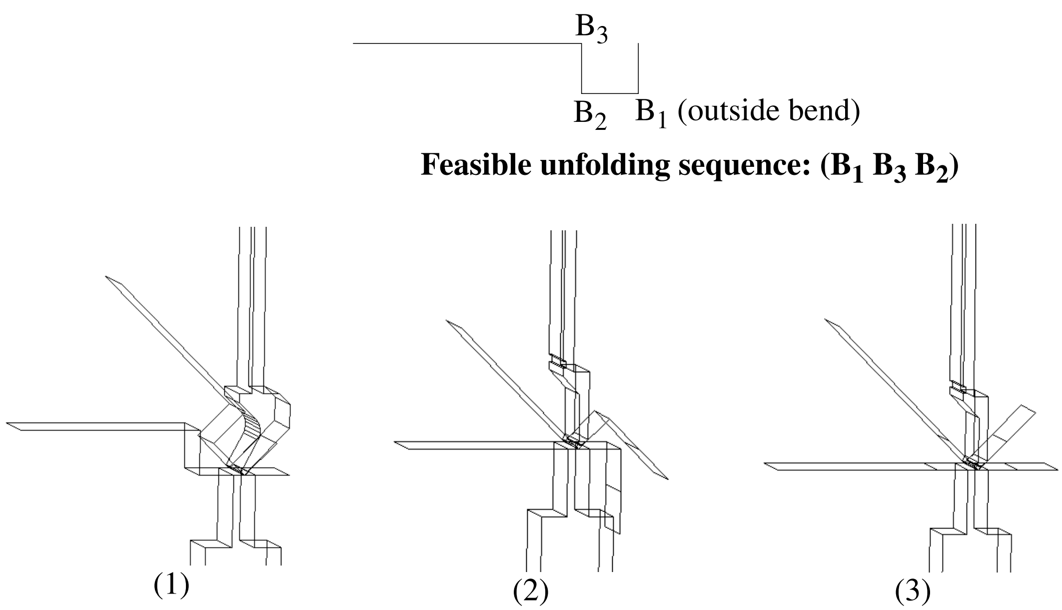


Figure 6.5: A channel part that violates the outside-bend-later precedence.

6.3.2 Tool Selection

Tool selection involves choosing a set of punches and dies that meets the bending requirements. The major considerations for selecting tools are: material type, part thickness, bend angles, bend radii and flange heights.

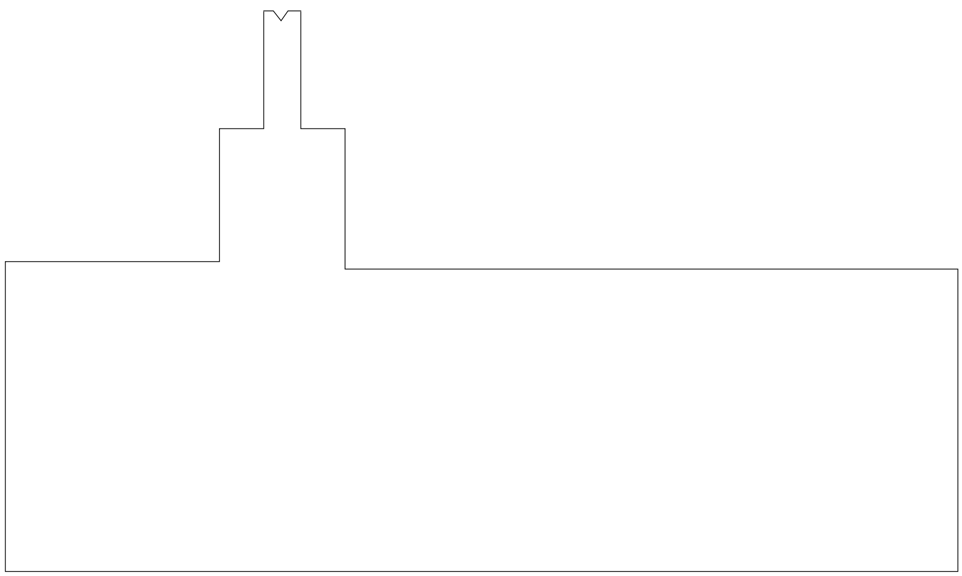
6.3.3 Tool Database and Machine Database

During tool selection, appropriate tooling (punch and die) is chosen. The tools are represented in a tool database with a geometric description and information about what kind of bends each tool can make. Each tool is represented in 2D and extruded for interference checking. This 2D representation makes the maintenance of tools much easier and more flexible. In the current implementation, only two standard punches are in the tool database. The die and the bed of the press brake are modeled as a single piece. Figure 6.6 shows these tools. The tool database file format is given in Figure 6.7. New tools can be added following this format.

The capacities of the bending facility are encoded into a machine database. These machine capacities include the maximum/minimum bend length and flange height. Any parts that violate the capacities in the machine database cannot be bent and need to be decomposed further. The machine database can eliminate expensive computation and provide a flexible way to control decomposition.

The tool and machine databases separate the accessibility analysis module into tool and facility dependent information. Different sets of tools and bending facilities can have their own databases, which can be easily maintained and exchanged.

Robot Punch (Punch-01600) Gooseneck Punch (Punch-15508)



Profile model for die, die-holder and machine lower part

Figure 6.6: Tool profiles in the tool database.

```
2 . . . . . . . . . . . . . . . . . . . . /* number of punches in the database */

Punch-00301 . . . . . . . . . . /* punch number (standard robot punch)*/
0.600 88.000 90.000 . . . . /* punch radius, punch angle and punch height */

16 . . . . . . . . . . . . . . . . . . ./* number of vertices for the 2D profile */
-20.000 120.000 0.000 . . /* vertex coordinates for punch profile*/
-7.000 120.000 0.000
-7.000 90.000 0.000
11.000 90.000 0.000
11.000 11.000 0.000
3.000 4.000 0.000
3.000 3.000 0.000
0.000 0.000 0.000
-3.000 3.000 0.000
0.000 9.000 0.000
0.000 64.000 0.000
-20.000 85.000 0.000
-20.000 94.000 0.000
-17.000 94.000 0.000
-17.000 102.000 0.000
-20.000 102.000 0.000

Punch-15508 . . . . . . . . . . . /* punch number (gooseneck punch)*/
0.769 88.000 90.458 . . . . . /* punch radius, punch angle and punch height */

22 . . . . . . . . . . . . . . . . . . . ./* number of vertices for the 2D profile */
7.077 90.458 0.000 . . . . . ../* vertex coordinates for punch profile */
-7.051 90.458 0.000
-7.051 80.869 0.000
11.051 62.767 0.000
13.743 59.434 0.000
15.666 55.613 0.000
16.743 51.459 0.000
16.922 47.203 0.000
16.204 42.973 0.000
14.589 38.998 0.000
-4.102 3.846 0.000
-0.641 0.282 0.000
0.000 0.000 0.000
0.667 0.282 0.000
4.231 3.974 0.000
3.487 4.718 0.000
49.485 52.357 0.000
49.485 78.356 0.000
37.357 90.458 0.000
20.204 90.458 0.000
20.204 120.739 0.000
7.077 120.739 0.000
```

Figure 6.7: Punch description in the tool database.

6.3.4 Interference Checking

During the search for an unfolding sequence, the workpiece cannot interfere with the tools and the machine. It is necessary to perform some interference checking to make sure that this does not happen. Interference checking is performed as follows. First, the intermediate shapes of the workpiece before and after the current bend are generated. The sweeping volume between these two shapes represents the work space of the workpiece for that particular bend. If the sweeping volume does not intersect with the chosen tooling and the machine, the current bend is feasible and can be further expanded. Otherwise, the sequence is infeasible and it will be discarded. In Figure 6.8, I show interference checking for a part. The model of the bending machine is simplified to speed up the analysis.

In tool accessibility analysis, it can be determined if a given part is accessible or not. For inaccessible parts, it is necessary to determine why the parts are inaccessible and to make further decomposition decisions to ensure the accessibility of the decomposition. Currently, only one bend is considered at a time for the tool accessibility analysis.

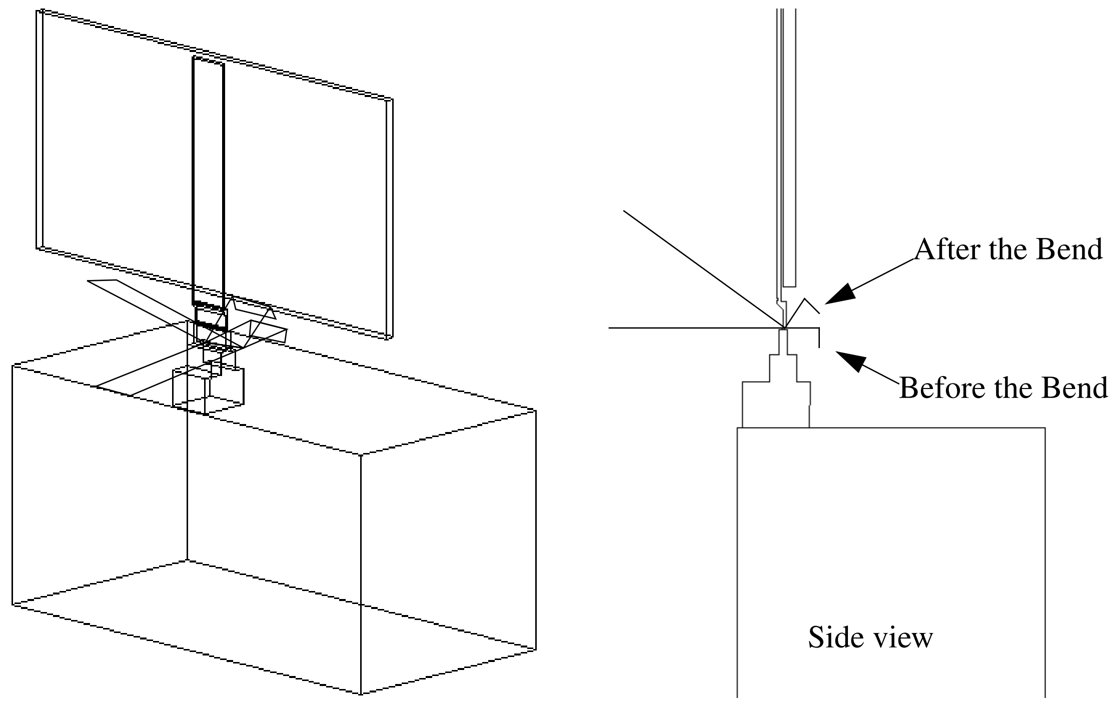


Figure 6.8: Interference checking (no collision found in this example).

6.3.5 Adjacent Self-Blocking Components

For an inaccessible part, an adjacent self-blocking component is defined as the smallest connected subset of the original part for which all bends are inaccessible to tools. A given part is inaccessible if any self-blocking components exist within it. The self-blocking components of two example parts are shown in Figure 6.9 and Figure 6.10. In these two examples, cuts are defined as the connecting links which are connected in 3D shape, but are separated due to the decomposition.

To find the adjacent self-blocking components of an inaccessible part, the faces with which the tool may interfere for all connecting links (bends in 2D flat pattern) of the final shape must first be computed. The sets of the smallest number of connecting links (bends) which interfere with the faces connected by these links represent the self-blocking components of the given part. Consider the example in Figure 6.9. A 3D part and its 2D flat pattern are shown. There are five connecting links (bends 1,4,5,6, and 7) after the compact pattern development is done. The interfered faces are also shown in the figure. The search looks for sets of inaccessible connecting links and checks if they completely interfere with the faces they connect. In this example, links 1 and 4 form an adjacent self-blocking component.

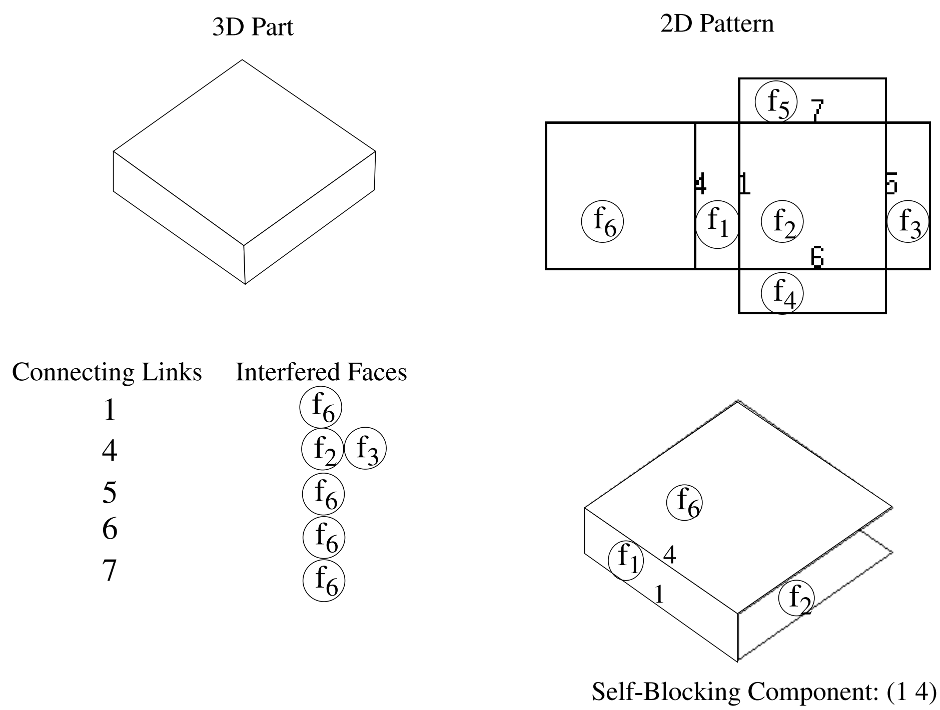
The maximum number of links allowed in an adjacent self-blocking component can be defined by the user in order to speed up the search. It should be noted that in these self-blocking components all links are inaccessible and adjacent to each other. However, there are cases in which parts that do not have any adjacent self-blocking components, are still inaccessible. These components are called non-adjacent self-blocking components. A detailed discussion of non-adjacent self-blocking components and how to deal with them will be given in Chapter 8.

The algorithm for identifying all adjacent self-blocking components is the following:

1. For every connecting link in the 2D flat pattern, treat the link as the last bend to be unfolded in the 3D shape, and determine the faces which interfere with the tools during an unfolding operation. Build the "connecting link (bends)" to "interfered face" relationship.

2. Find all connected components based on the 2D connectivity. Each connected component is represented by a set of connecting links (bends).

3. Compute the smallest connected components which are inaccessible (from the interference information in Step 1).



Accessible Decompositions:
(1) : 2 Parts, 6 Cuts
(4) : 2 Parts, 4 Cuts

Figure 6.9: Self-blocking component example 1.

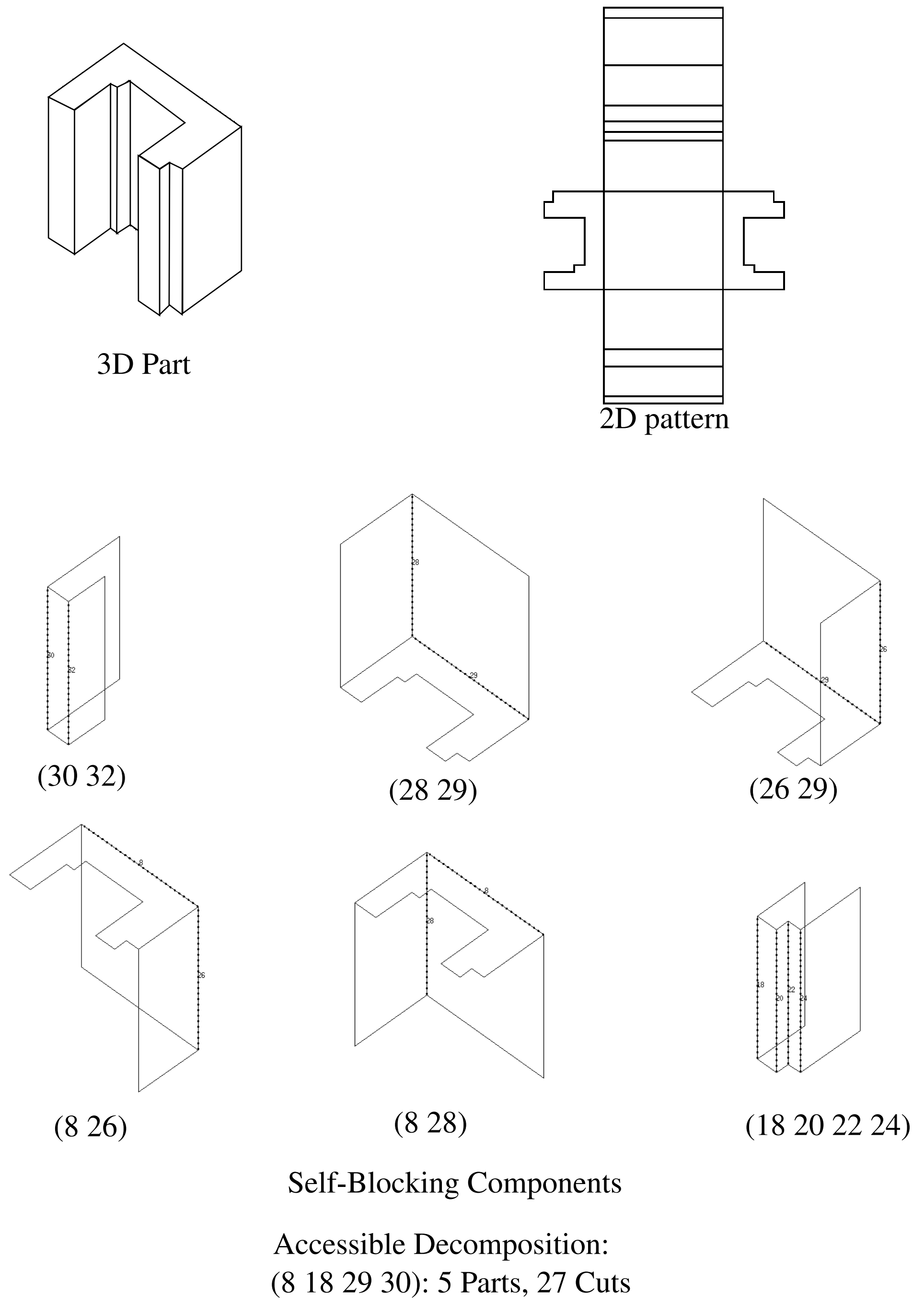


Figure 6.10: Self-blocking component example 2.

6.3.6 Accessible Decompositions

The resulting parts of an accessible decomposition are accessible with respect to the current tools. After all self-blocking components of an inaccessible part have been identified, the accessible decompositions need to break all self-blocking components of the original part. Breaking one link from every self-blocking component will result in an accessible decomposition. The product sets of all self-blocking components represent all accessible decompositions which break all self-blocking components. The links in the accessible decomposition are the links to be decomposed. A decomposition with *N* decomposing links will result in *(N+1)* parts. To minimize the number of parts, the accessible decomposition with the smallest number of decomposing links is chosen.

The accessible decompositions of the product in Figure 6.9 are link 1 and link 4. A decomposition along link 1 will result in 2 parts and 6 cuts, while decomposition along link 4 will result in 2 parts and 4 cuts. The other example in Figure 6.10 has 6 self-blocking components. Note that only 4 links need to be decomposed in order to break all self-blocking components, since some self-blocking components share common links.

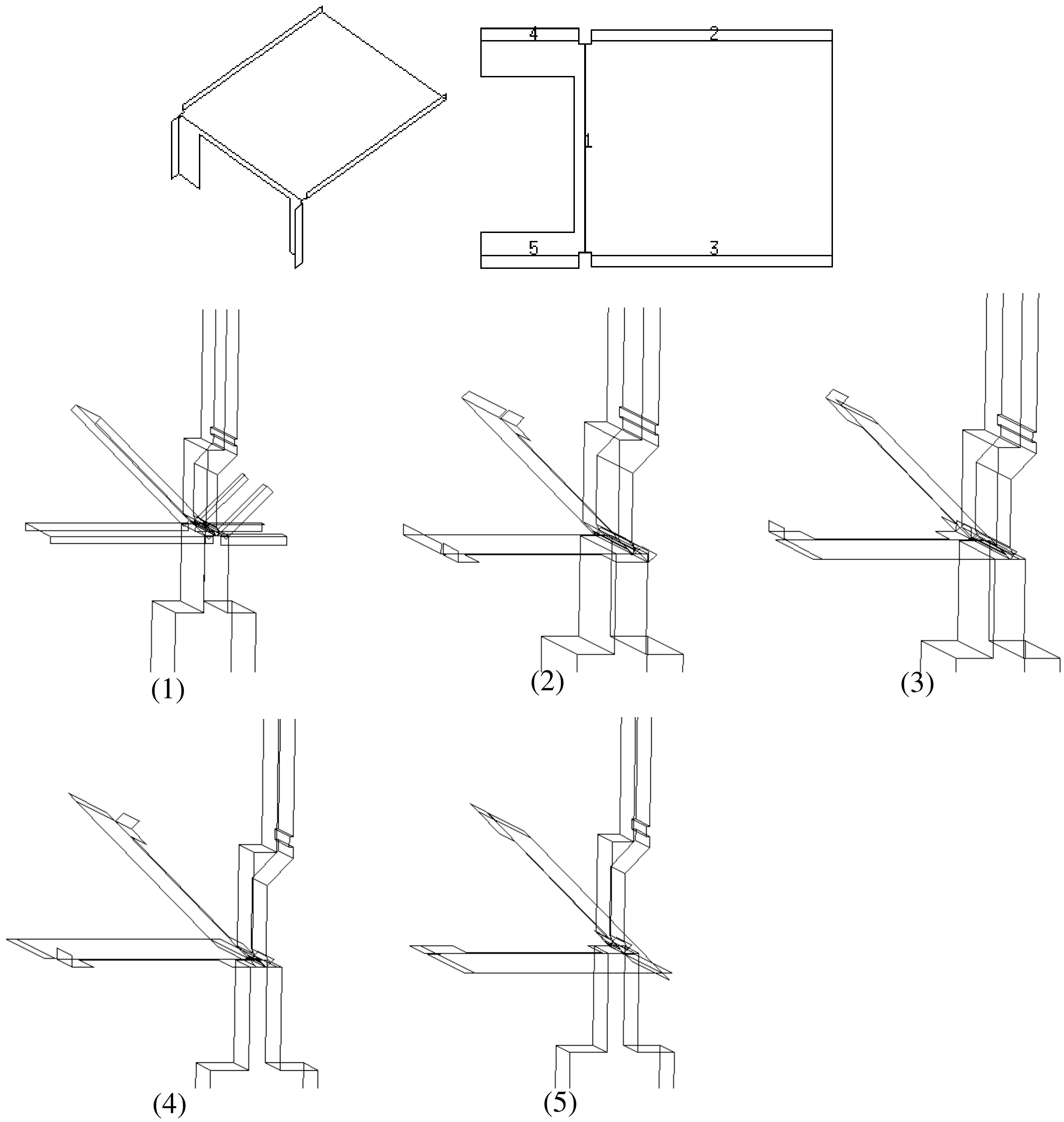
It is desirable to have a decomposition which results in the least number of parts and cuts. Currently, all accessible decompositions are generated first, and the decomposition with the smallest number of parts and cuts is selected. However, if the number of self-blocking components is large, to enumerate all accessible decompositions is impractical. A heuristic search (like A*) can be used instead to find the decomposition with the smallest number of parts and cuts.

6.3.7 Cache Implementation

Tool accessibility analysis is computationally expensive and has to be conducted on all resultant parts during the product decomposition stage. However, some of the computation is redundant or can be determined from previous results. A cache implementation has been developed to store previously computed results. This mechanism saves a significant amount of time when evaluating tool accessibility during decomposition. Two rules are used to take advantage of the cache implementation. First, any part which is a connected super-set of an inaccessible part is also inaccessible. Second, any part which is a connected sub-set of an accessible part is also accessible.

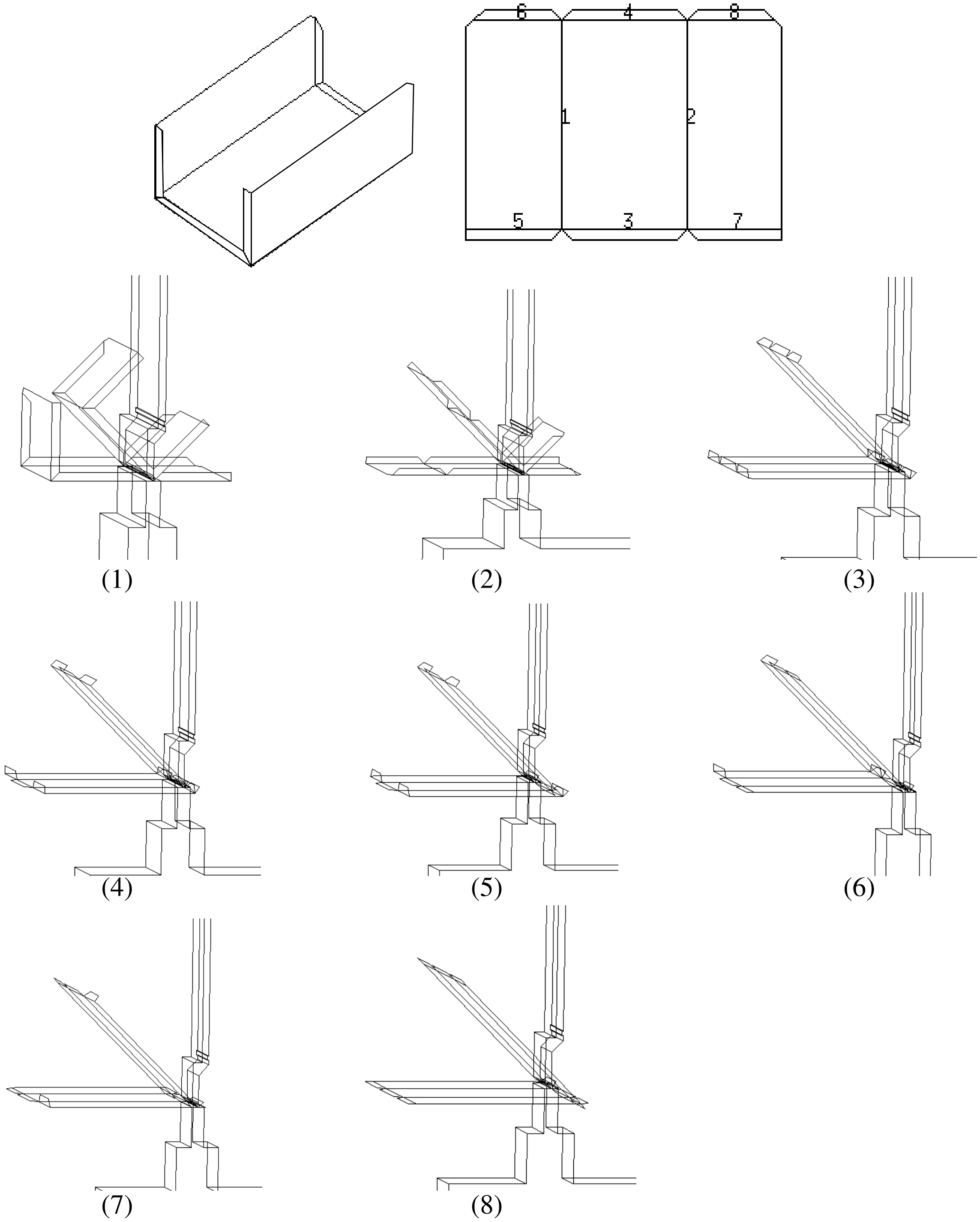
6.4 Examples

We have applied the tool accessibility analysis to two example products (see Figure 6.11 and Figure 6.12). These examples are both accessible with respect to the standard tools in the tool database. Their unfolding sequences and the selected tools for each bend in the sequence are shown. The computational time on a Sun Sparc 10 is also shown in the figures.



Unfolding sequence: ((B_1:T_1)->(B_2:T_1)->(B_3:T_1)->(B_4:T_1)->(B_5:T_1)) (7.3 sec.)

Figure 6.11: Unfolding sequence of example 1.



**Unfolding sequence: ((B_1:T_1)->(B_2:T_1)->(B_3:T_1)->(B_4:T_1)->
(B_5:T_1)->(B_6:T_1)->(B_7:T_1)->(B_8:T_1)) (32.4 sec.)**

Figure 6.12: Unfolding sequence of example 2.

6.5 Conclusion

In this chapter, an approach to evaluating the tool accessibility of bent sheet metal parts is presented. This approach adopts a backward planning approach to determine an interference-free bend sequence and select tooling. Tool accessibility imposes manufacturing constraints during the bending process planning. The tools and machine capacity are encoded into the tool and machine databases, which are used during the analysis and can be easily updated and replaced.

This approach aims to find a feasible unfolding sequence quickly. However, passing the tool accessibility test does not guarantee bendability of the given part. For instance, a part passing the tool accessibility test may not have enough room for the robot to grasp, or the stage length for the tooling may be too long for the bed of the press brake. As a result, a thorough bending process planning has to be performed after the decomposition to ensure complete bendability.

Chapter 7

Product Assemblability

Traditional assembly sequence planning starts with the known geometric representations and final configurations of parts in a product. An assembly plan consisting of a feasible (stable and interference-free) assembly sequence and directions is derived from the known geometric representations. Common geometric representations for assembly planning are constructive solid geometry (CSG) and boundary representation (B-Rep) (Hoffmann 1989). With a given part geometry and configuration, the contact information of parts can be computed. Based on the contact information, the local feasible assembly motion (translation or rotation) for the chosen part can be determined. The motion is verified by sweeping the chosen part from the designated initial configuration to the final configuration against all parts in the current assembly to guarantee that it is interference-free.

When a product assembly is considered during the decomposition stage, assembly planning differs from the standard assembly problem. First, the number of parts in the product is not known before the product is decomposed. Second, even after the product has been decomposed, the resulting parts may need to be modified to account for the specified mating conditions and assembly methods. As the decomposition proceeds, the contact relations between parts need to be specified by the designers since they are related to product functionality and accessibility conditions.

7.1 Assembly Sequence Planning

The assembly planning problem is characterized as a manipulation planning problem (Wilson, 1992). An assembly is given as a set of parts with their geometric representations and initial and final configurations. The rest of the objects to be manipulated are the tools used to assemble the parts, such as robots or fixtures. One important constraint for assembly planning is that all parts must be in their final relative positions in the goal configuration and there should not be any interference between any parts and tools during the assembly operation.

Assembly sequence problems are known to be highly geometry-dependent. The idea of assembly planning by disassembly is used in most assembly planners. That is, an assembly sequence can be obtained as the reverse of a valid disassembly sequence. This is because the disassembly approach usually provides more constraints early in the search. As a result, the search space for disassembly is much smaller than for an assembly approach. In reality, some reverse disassembly operations may not be feasible due to stability constraints, and the non-rigid characteristics of parts. However, in this thesis I assume all assembly sequences can be obtained by reversing their disassembly sequences; the terms assembly and disassembly are used interchangeably.

7.1.1 Previous Work

Homem de Mello and Sanderson (1990) use an AND/OR graph to represent assembly plans. They subsequently develop an algorithm to generate all feasible assembly plans based on this AND/OR graph (1991).

Bourjaut (1984) represents the connection relationship of an assembly as a graph. The nodes in the graph represent parts in the assembly and the links represent the contact relations between parts. In his approach, the geometric constraints which are used to derive the liaisons of the graph are obtained by querying the users whether certain liaisons can be established between parts. The assembly plan is then inferred.

De Fazio and Whitney (1987) extend Bourjaut's work to capture the user's intuition about precedence constraints by asking users a small set of questions. However, once the number of parts in an assembly becomes large, the number of questions grows and becomes tedious to answer. Also, for a complex part geometry, it is difficult to determine the precedence constraints among parts.

Baldwin et al. (1991) develop a computer-aided tool to generate all feasible assembly sequences and help users to evaluate the assembly sequences; however, human interactions and judgements are required.

Wilson (1992) uses an approach to represent the blocking relationship for the parts in an assembly. The concept of non-directional blocking graphs (NDBGs) is used to capture the blocking relationship for translational motion. Both local and global blocking relationships are considered in NDBG.

Mattikalli (1994) focuses on the kinematic and dynamic issues of assembly planning. He uses a linear programming approach, which is based on potential energy minimization to find the most stable configuration for an assembly. He further develops an approach to synthesize the grasping and fixturing of an assembly which considers the stability of the assembly.

7.1.2 Types of Assembly Sequence

In order to simplify assembly planning, some terms must be defined:

- **Number of hands**. The minimum number of hands needed to perform a given assembly operation. A robot, an assembly table, and fixture are usually considered hands for assembly operations. For an assembly with n parts, a minimum number of n hands may be required (see Figure 7.1). Assembly plans with more than two hands are usually difficult to execute and plan. In fact, most assembly planners assume the assembly plans will be executed with two hands. This is called a binary assembly sequence.
- **Monotonicity**. For some assemblies, a part may have to be placed in some intermediate configurations during assembly operations, and later moved to its final configuration. For example, part B in Figure 7.2 needs to be moved to the intermediate configuration to avoid collision with part A before it can be moved to its final configuration. An assembly sequence is monotonic if no intermediate configuration of the part is needed. Non-monotonic assembly sequences are more difficult to deal with, since additional part configurations have to be considered.
- **Linearity**. A binary assembly sequence is linear if only one part has to be moved for each step of the assembly. The assumption of linearity allows one part to be assembled at a time. This assumption also greatly reduces the number of possible subassemblies generated during the planning stage. Figure 7.3 shows an assembly which

cannot be done with a linear sequence.

- **Connectedness**. The connectedness constraint demands that all subassemblies be connected during assembly operations. In Figure 7.4, a disassembly sequence is shown which violates the connectedness assumption. Connected subassemblies are easier to fixture and manipulate during assembly operations. This constraint also reduces the number of combinations in non-linear planning.

A detailed discussion on the types of assembly sequences can be found in Natarajan (1988),Wolter (1988) and Wilson (1992).

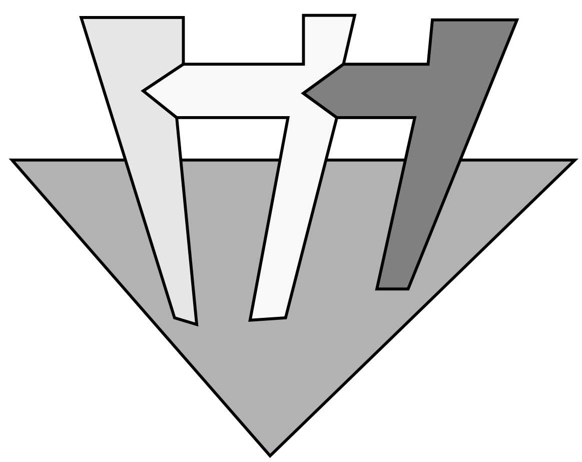


Figure 7.1: N-handed assembly operation (Natarajan, 1988).

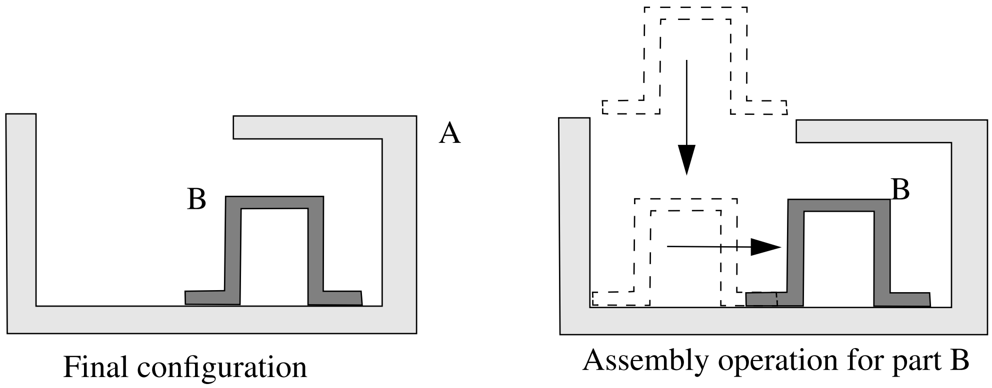


Figure 7.2: Non-monotone assembly operation for part B.

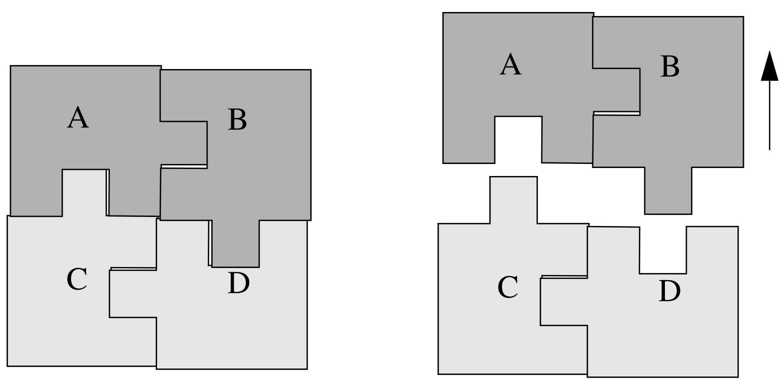


Figure 7.3: Non-linear assembly operation.

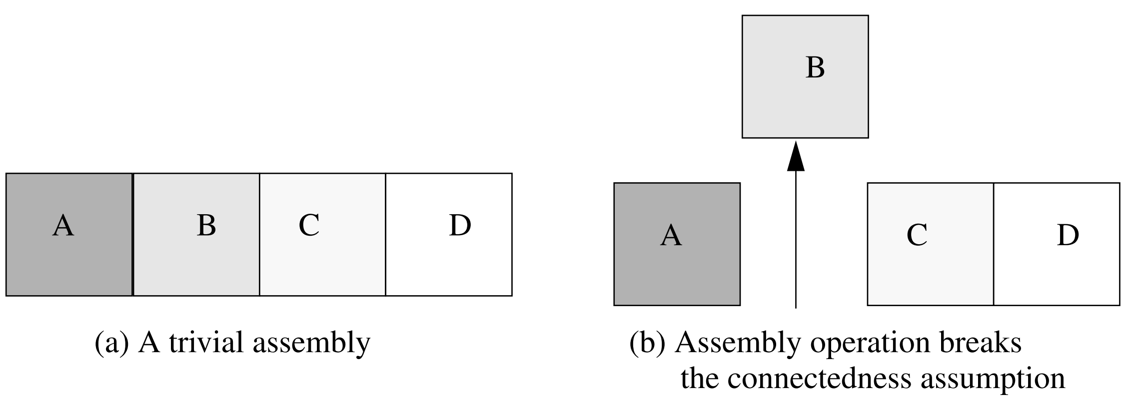


Figure 7.4: Non-connected assembly operation.

7.2 Our Assembly Planning Approach

An assembly planner has been developed based on an assembly-via-disassembly approach. To simplify planning, the planner assumes that all assembly operations are binary, monotonic, linear and connected. I further restrict the possible assembly motion to translational motion only (Wilson and Schweikard 1991).

To generate an assembly plan, a subassembly must first be selected and its feasible disassembly directions found. A disassembly sequence is feasible if there is no interference between parts during the disassembly operations. The feasibility of the disassembly directions is checked locally and globally. Usually the local feasible directions are derived from mating conditions between parts. These directions are further verified by sweeping operations to ensure there is no global interference among parts. This process continues until all parts have been disassembled.

7.2.1 Local Feasibility Checking

To derive the local feasible disassembly directions, the mating conditions between parts have to be determined. This is usually done by a small amount of perturbation of the part to be disassembled along the disassembly direction to check if there is any interference in the current assembly. If so, it indicates that these two parts which interfere with each other are in contact. Another way to compute the mating information is to perform a boolean intersection operation between parts. If the intersection is a surface, it indicates that these two parts are in contact. The normal of the intersection surface indicates the contact direction.

During the product decomposition stage, the contact information or the inside/outside relations between decomposed parts are usually determined by designers. Additional assembly tabs are added to parts to provide attachment points for assembly operations. This kind of information usually relates to product functionality. With local contact information between parts, the local disassembly directions can be derived.

A general approach to representing feasible disassembly directions is a Gaussian map (Gan et al. 1994). A Gaussian map is a unit sphere with the origin at its center. All vectors originating from the sphere center to a point on the spherical surface represent possible disassembly directions in Cartesian Space. A single local contact between two parts represents a hemisphere with disallowed disassembly directions, and the other hemisphere, feasible disassembly directions (see Figure 7.5(a)). The feasible disassembly directions for a subassembly can be found by intersecting all feasible directions with its neighboring parts (see Figure 7.5(b)).

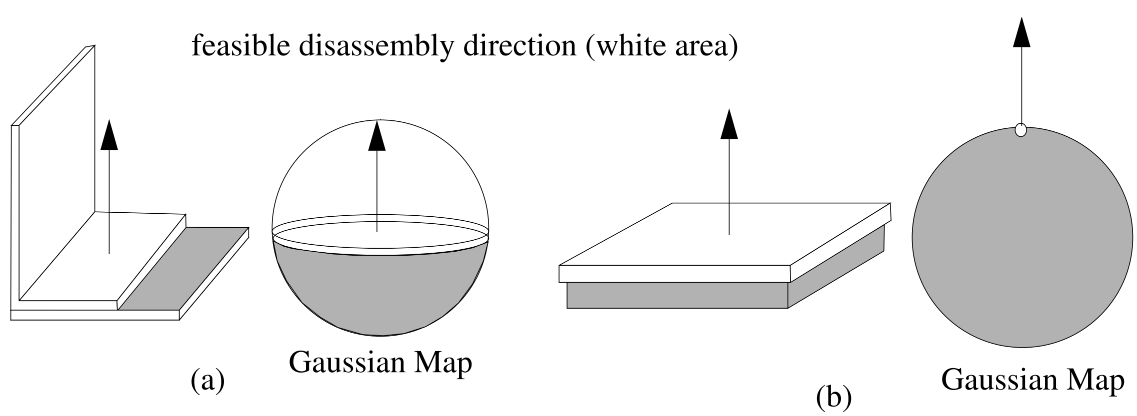


Figure 7.5: Gaussian map and feasible disassembly directions.

7.2.2 Global Feasibility Checking

The local disassembly directions derived from a Gaussian map only consider the mating conditions between parts in contact. To ensure that there is no interference with any other parts, further interference checking has to be performed. Global interference checking is done by sweeping the part to be disassembled along the disassembly direction to determine if there is any interference with parts in the current assembly. For instance, Figure 7.6 shows a product with two parts and two sweeping tests (Figure 7.6(b) and (c)).

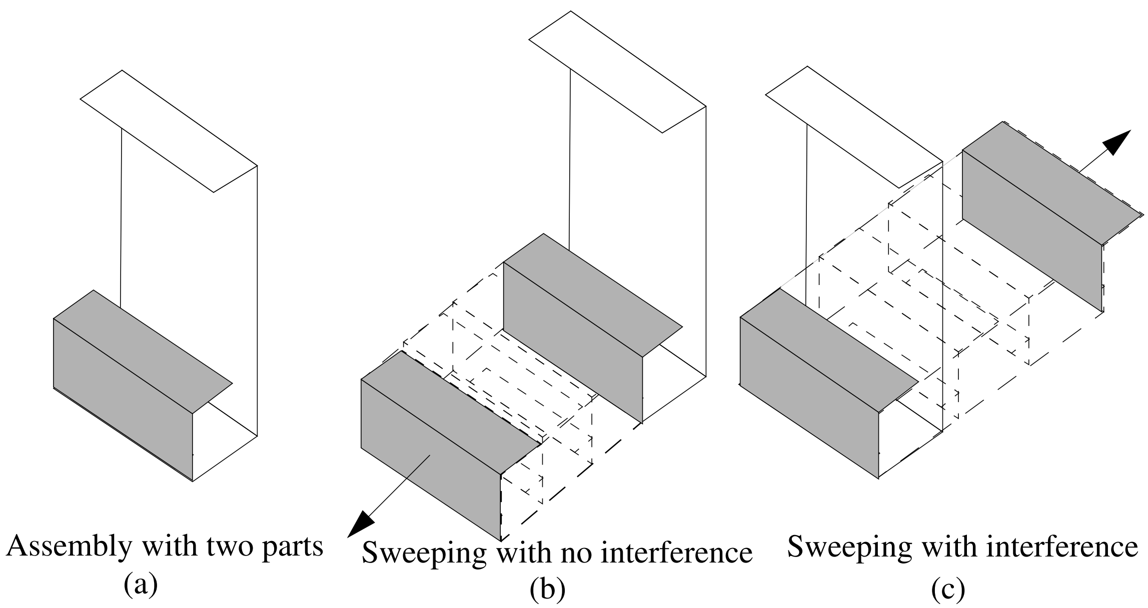


Figure 7.6: Global interference checking: the sweeping test.

7.3 Design Modification

In order to compensate for the assembly operations introduced by product decompositions, design modifications are necessary to meet the mating requirements between decomposed parts. For some assembly methods, such as welding and riveting, additional assembly tabs must be added to provide space for assembly operations. Two welding methods are usually used in assembly: spot welding and seam welding. Spot welding is easier to perform, but needs additional assembly tabs. Seam welding does not need additional assembly tabs but is much more expensive and difficult to perform. In our system, additional assembly tabs are added between decomposed parts to facilitate spot welding. The additional assembly tabs result in additional material and bending operations. The changes made are updated simultaneously in the geometric models and the graph representation. The final results are tested again to ensure the part's manufacturability and assemblability.

The assembly tab is assumed to be rectangular and the full length of the cut. Currently, assembly tab information is generated automatically with default values. Designers can override the default values if necessary. An interface is used which provides the following information (Figure 7.7):

- Assembly method
- Tab dimension and position
- Part to which the tab will be assigned
- Inside/outside relation of the tab with respect to the mating part

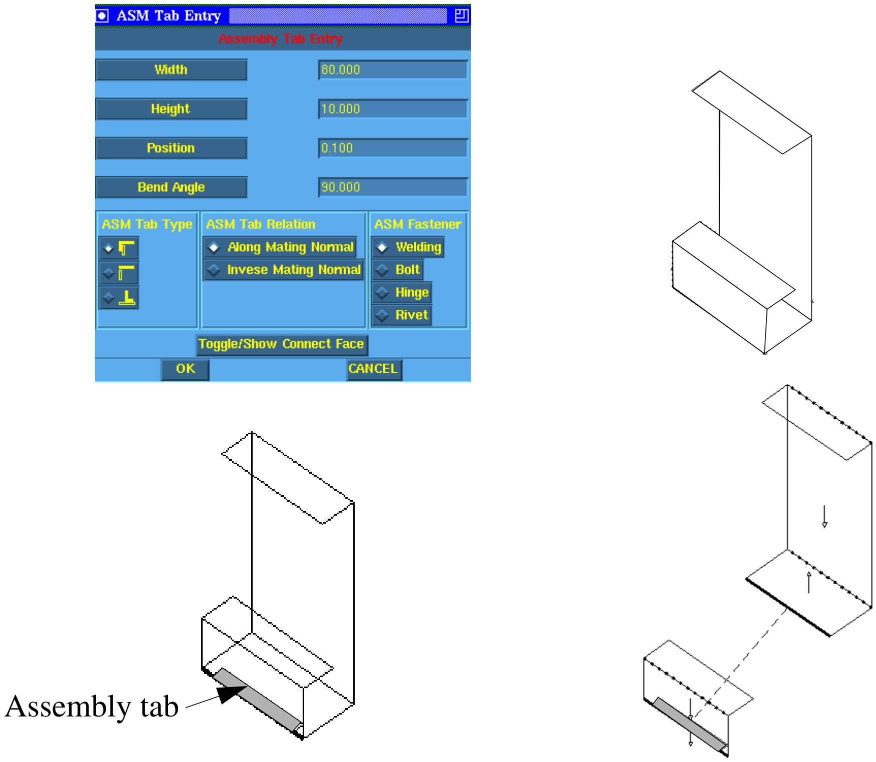


Figure 7.7: A user interface for assembly tab information.

Assembly tabs are assumed to be rectangular. Currently, assembly tabs are automatically assigned to the part with fewer bends. By doing so, it becomes less likely that the resulting part will be inaccessible during bending. Tool accessibility will be tested on the final results. Also, interferences between the assembly tabs and the parts need to be avoided. The system checks for these interferences and warns the designers as appropriate. However, designers can acknowledge or overrule the automatic assignment if necessary. Designers can also choose not to assign any assembly tabs between decomposed parts.

7.3.1 Automatic Assembly Tab Assignment

In our assembly planning approach, assembly tabs are automatically assigned to avoid overlapping between the assembly tabs and the part. Before assembly tabs can be assigned, several terms have to be defined. A *cut* is a decomposition between two connecting edges of a product. The connecting edges of a cut are defined as the *cut edges* of the cut. The data structure of a cut is shown in Figure 7.8. To assemble a decomposed product, each cut needs to be assigned an assembly tab to guide or provide welding space for the assembly operation. The tab can be assigned to either part. There are two kinds of cuts for a decomposed product: *cuts between decomposed parts* and *cuts in the same part* (Figure 7.9 and Figure 7.10). The first kind of cut is due to product decomposition, and the second one to pattern development. By examining the 2D flat patterns of decomposed parts, local interference between assembly tabs and parts can be avoided. A cut edge is said to be *fit* if none of the following conditions is true:

- Previous_Angle < 90 *OR* Next_Angle < 90
- Previous_Angle < 180 *AND* Previous_Cut is already assigned
- Next_Angle < 180 *AND* Next_Cut is already assigned

All the conditions described above indicate that the cut edge to which a rectangular assembly tab will be assigned will cause 2D pattern overlapping. In Fig. 7.11(a), the assembly tabs can be assigned to the cuts without interference. However, in Fig. 7.11(b), the cuts are on concave edges and interference occurs when assembly tabs are assigned to the concave cut edges.

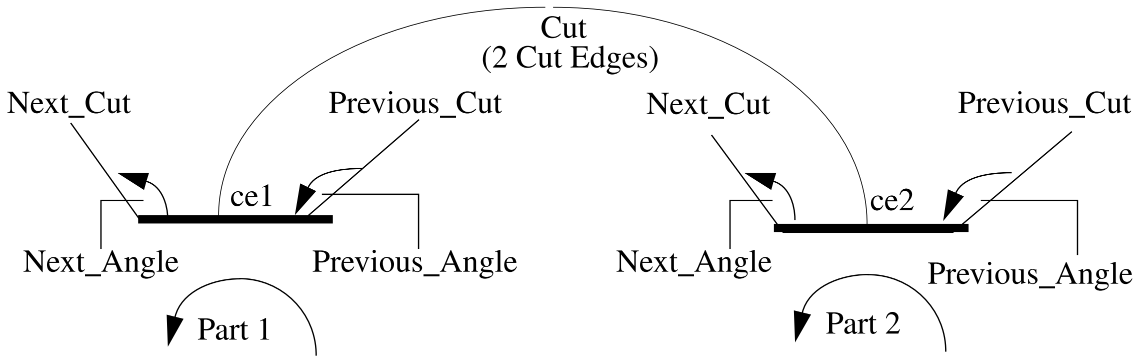


Figure 7.8: Data structure of a cut

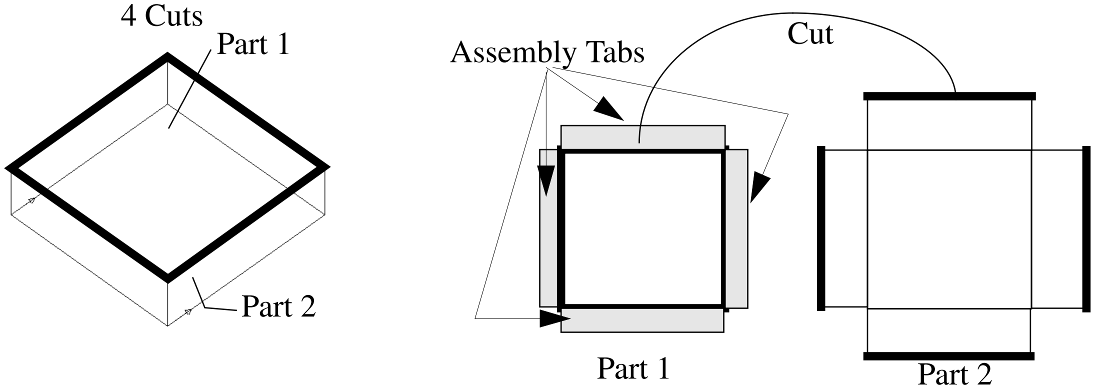


Figure 7.9: Cuts between decomposed parts.

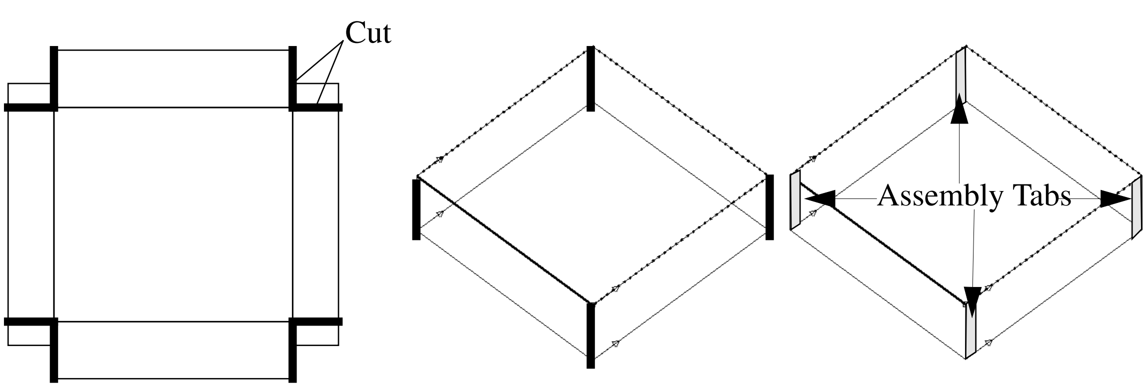


Figure 7.10: Cuts in the same part.

Similarly, a cut edge is said to be *unfit* if any of the above conditions holds. A cut is defined as *fit* if any of its cut edges is *fit*. A fit cut has at least one cut edge that can be added an assembly tab. A cut is said to be *unfit* if both its cut edges are *unfit*. An *unfit cut* cannot be assigned an assembly tab to any of its cut edges.

An example of assembly tab assignment between decomposed parts is shown in Figure 7.9. The product has 4 cuts that are between decomposed parts (part 1 and part 2). They are all fit cuts with two fit cut edges. That is, the assembly tabs can be assigned to either part. Since part 1 has no bend and part 2 has four bends, all tabs are assigned to part 1. No assembly tab will be assigned to unfit cuts.

Currently only cuts between decomposed parts are automatically assigned assembly tabs. Cuts in the same part are not automatically assigned assembly tabs because it is assumed that the rigidity of parts can be maintained after they are bent. However, following the same analysis for cuts between decomposed parts, the assembly tabs can be also assigned to the same part. Figure 7.10 shows one possible assignment for cuts in the same part. This design is common for industrial products.

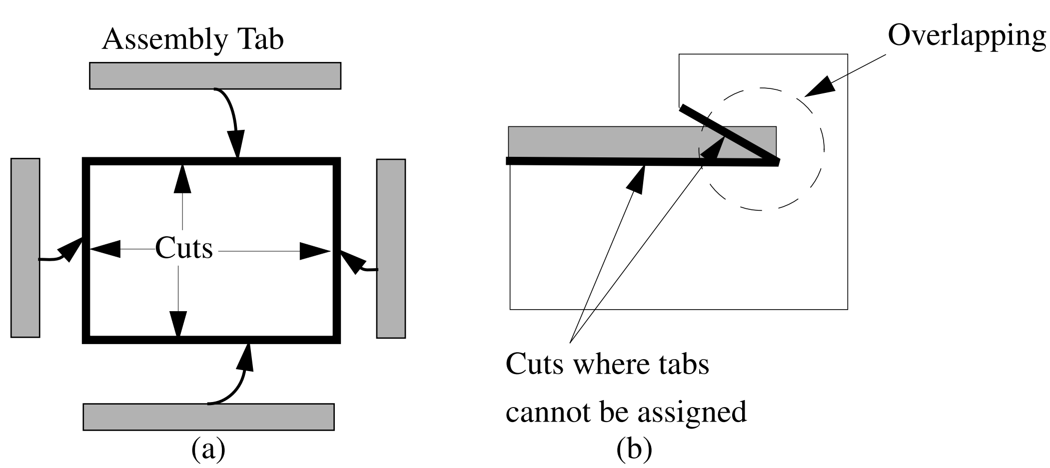


Figure 7.11: Assembly tab assignment and 2D pattern interference.

After the assembly tabs are assigned, the face relations between the assembly tabs and their connecting faces have to be determined. Fig. 7.12 shows three possible face relations. These relations are usually related to product functionality. In this thesis, the designers have to provide such information. These face relations imply local contact information between the assembly tabs and their connecting faces. The local contact information is used for assemblability checking and assembly planning.

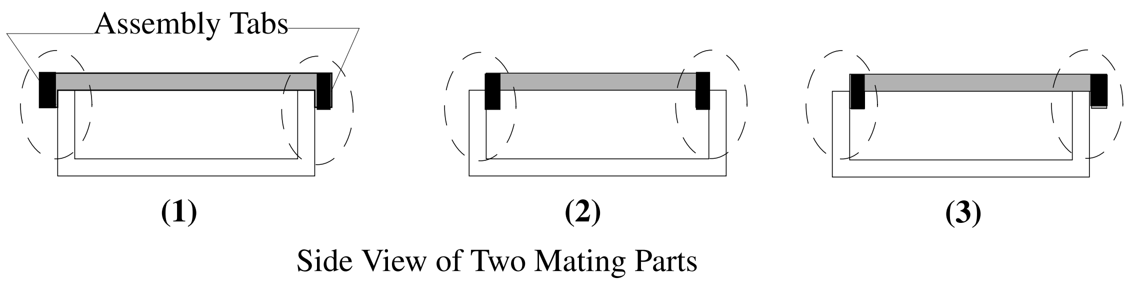


Figure 7.12: Face relations between assembly tabs and their connecting faces.

7.4 A Primitive Assembly Planning System

Based on the planning assumptions mentioned in this chapter, A primitive assembly planning system has been developed to derive the assembly plan for decomposed products. An A* search algorithm is used to search for the disassembly plan. The preference for disassembly direction selection is: +Z direction before (X direction and Y direction) before any other direction. Two examples are given in the next section.

The planner takes an input file that contains the geometric models and local contact information of an assembly, and generates the assembly plan. The geometric models are represented using *Noodles* (Gursoz, 1991). The input file format of the assembly is shown in Figure 7.13. An input example (for the assembly in Figure 7.15) is given in Figure 7.14. Note that part number 0 refers to the assembly table. Finally, a simulation file is generated to visually verify the plan.

To speed up the global sweeping interference checking, A cache mechanism has been implemented to store and re-use the computational results.

```
Product Name
Product No.
Number of Parts in the Product
File Name of the Product Solid Model (Noodles V7 file)

    Part Name 1:
    Part No.
    File Name of the Part Solid Model (Noodles V7 file)
    Initial Configuration of the Part
        Number of Contact Parts:
            Contact Part 1:
                Number of Contacts
                Contact Directions
            ...
        ...
    ...
```

Figure 7.13: Input file format for disassembly planning.

```
Box_1
1
2
/usr0/cwwang/PD/ASM/V7/Box_1_p2.v7

        part-1
        1
        /usr0/cwwang/PD/ASM/V7/Box_1_p1.v7
        0.0 0.0 0.0

        2

                2
                4
                -1.000 -0.000 -0.000
                1.000 -0.000 -0.000
                -0.000 -1.000 -0.000
                -0.000 1.000 -0.000

                0
                1
                0.000 0.000 -1.000

        part-2
        2
        /usr0/cwwang/PD/ASM/V7/Box_1_p2.v7
        0.0 0.0 0.0


                1
                1
                4
                1.000 0.000 0.000
                -1.000 0.000 0.000
                0.000 1.000 0.000
                0.000 -1.000 0.000
```

Figure 7.14: Input file for the assembly in Figure 7.15.

7.5 Examples

This section shows the assembly plans of two example products. Each assembly plan consists of an assembly sequence (an ordered list of part indices) and the assembly directions (unit vectors). The assembly tabs are added and the face relations between the tabs and their connecting faces are given. In each example, the final assembly configuration and the local contact relationship among parts are shown. The simulations of the assembly plans are also shown.

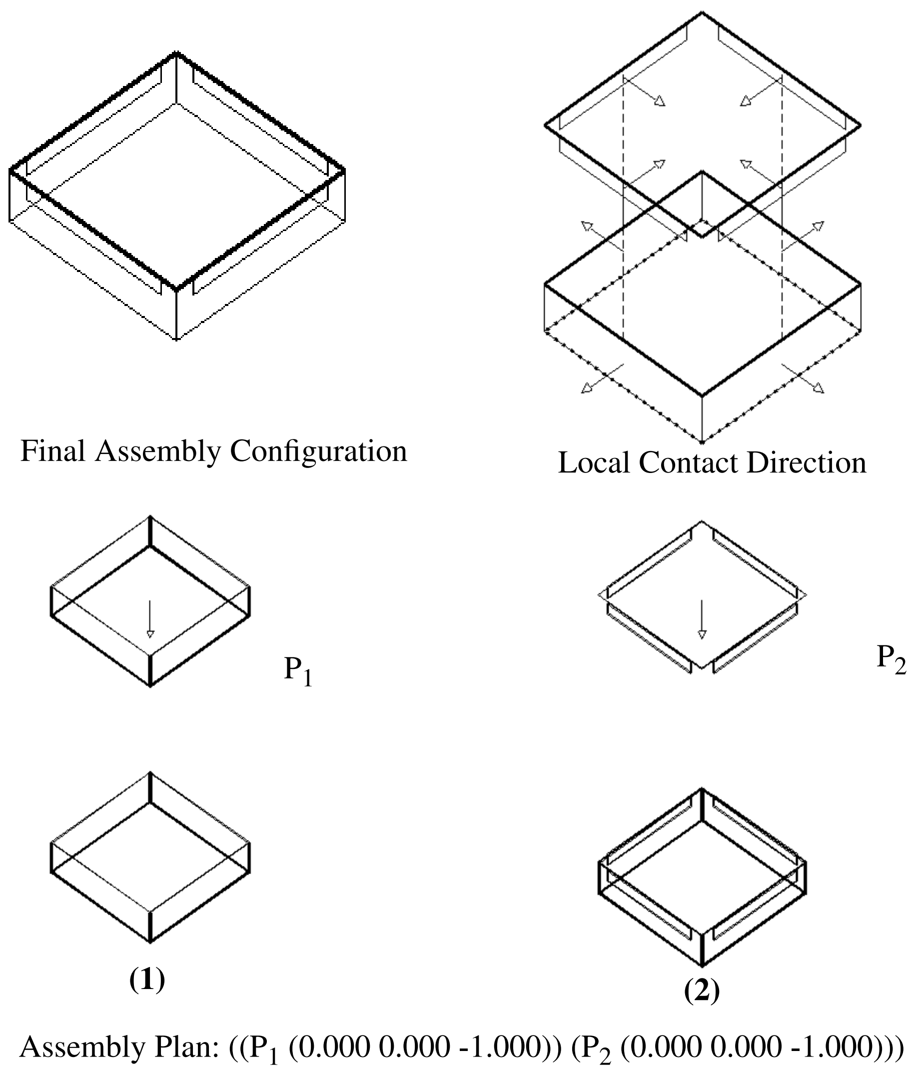
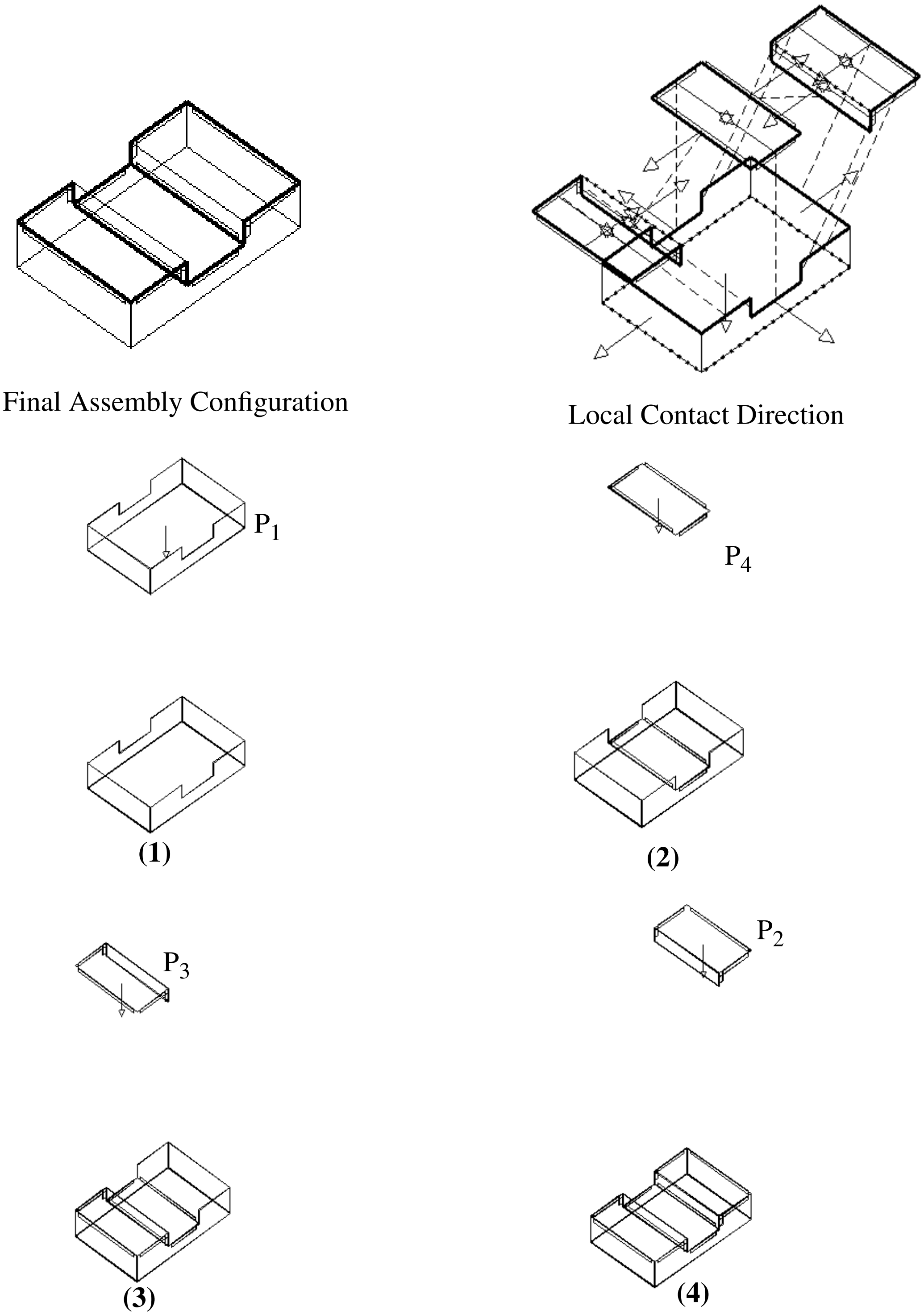


Figure 7.15: Assembly simulation example 1.



Assembly Plan: ((P_1 (0.000 0.000 -1.000)) (P_4 (0.000 0.000 -1.000))
(P_3 (0.0000.000 -1.000)) (P_2 (0.000 0.000 -1.000)))

Figure 7.16: Assembly simulation example 2.

7.6 Limitations

The assembly planning approach proposed in this thesis does not aim to address all aspects of the assembly process. The approach is limited by the following assumptions:

1. Assembly planning is based on part geometry only.
2. Stability and accessibility of assembly operations are not considered.
3. Fixture planning is not considered.
4. Only translational motion is considered.

My approach can be extended to include the assembly sequence of components other than sheet metal parts inside the product as long as the components are modeled as parts of the assembly. This is especially true when the sheet metal products are used as covers or enclosures.

7.7 Conclusion

This chapter presents an approach to planning the assembly sequence of decomposed products. In this approach, either the decomposition module can automatically assign assembly tab information or designers can make their own decisions about assembly tabs. The local contact information can be derived from the inside/outside relations between parts in the assembly tab information. This step contrasts with the traditional assembly planning approaches which derive assembly tab information from product geometry. Human intervention is necessary because the inside/outside relation between parts can directly relate to the product functionality.

The planning system searches for the interference-free assembly sequence of a product, which has been decomposed. This planning system assumes the assembly operation is binary, monotonic, and uses translational motion only. The final assembly contains the assembly sequence and assembly directions.

The product geometry needs to be modified according to the mating conditions specified by the designers. For most bent sheet metal assembly methods, additional assembly tabs are usually needed to provide attachment points for assembly methods. An approach to automatically assigning assembly tabs has been developed. This approach is based on 2D part geometry to avoid further overlapping between tabs and part.

An assembly feasibility checker utilizes the local contact information among parts in an assembly during the decomposition stage. An assembly planner based on the approach described in this chapter has been developed and is used in the prototype system.

Chapter 8

Decomposition Results and Discussion

In this chapter, several decomposition examples are presented using the proposed decomposition approach. Examples of both unfoldable and non-unfoldable products are shown. I will explain how these products are decomposed using the develop-first-decompose-later approach. The assembly tabs are automatically assigned to the decomposed products. The final production plans for cutting, bending and assembly are also produced using the prototype system. In the examples, only the results with the minimum number of parts and cuts are given unless otherwise specified.

Several key issues related to decomposition are addressed. I show how different 2D flat patterns of the same compactness, tool availability, and design variation affect the decomposition results. The differences between the tool accessibility analysis and the bending plan produced by the bending planner are also discussed.

8.1 Decomposition of Unfoldable Products

A bent sheet metal part needs to be developable (unfoldable) before its bendability can be analyzed. The develop-first-and-decompose-later approach accounts for this concern. In any event, further decomposition of any unfoldable part will also be unfoldable.

Three unfoldable products are shown in this section. In each example, the compact 2D flat pattern, adjacent self-blocking components, and decomposition results are illustrated. The corresponding cutting, bending and assembly plans are also given. Inside/outside relations between assembly tabs and their connecting faces are assumed to be along the face normal directions of the connecting face.

8.1.1 Example 1: An electronic enclosure

An electronic enclosure is shown in Figure 8.1. We start with the 3D shape of the enclosure. The most compact 2D pattern is determined according to the pattern development algorithm described in Chapter 5.

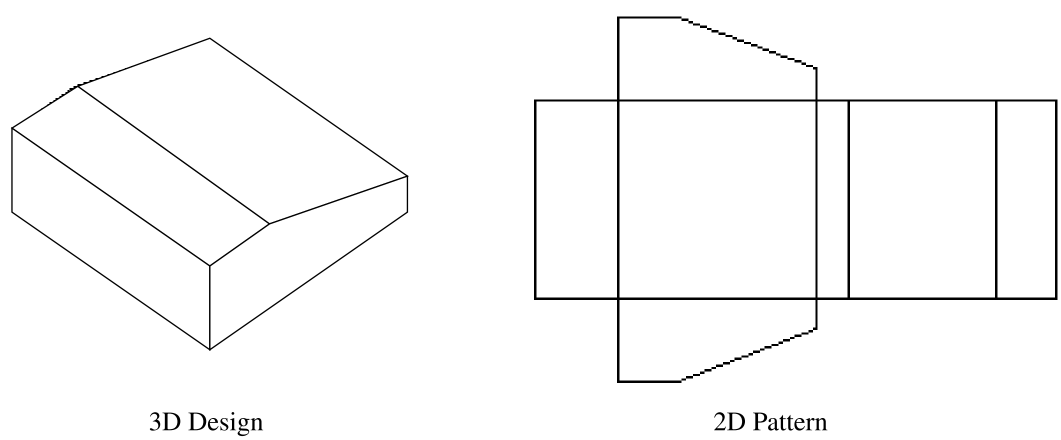


Figure 8.1: Example product 1.

Once the 3D shape and the 2D pattern of the product are determined, the tool accessibility analysis is performed. In this example, one adjacent self-blocking component is found. There are two possible decompositions to break this self-blocking components. The decomposition which yields two resulting parts and six cuts is shown in Figure 8.2. Assembly tabs are assigned to P_2 because it has less bends than P_1.

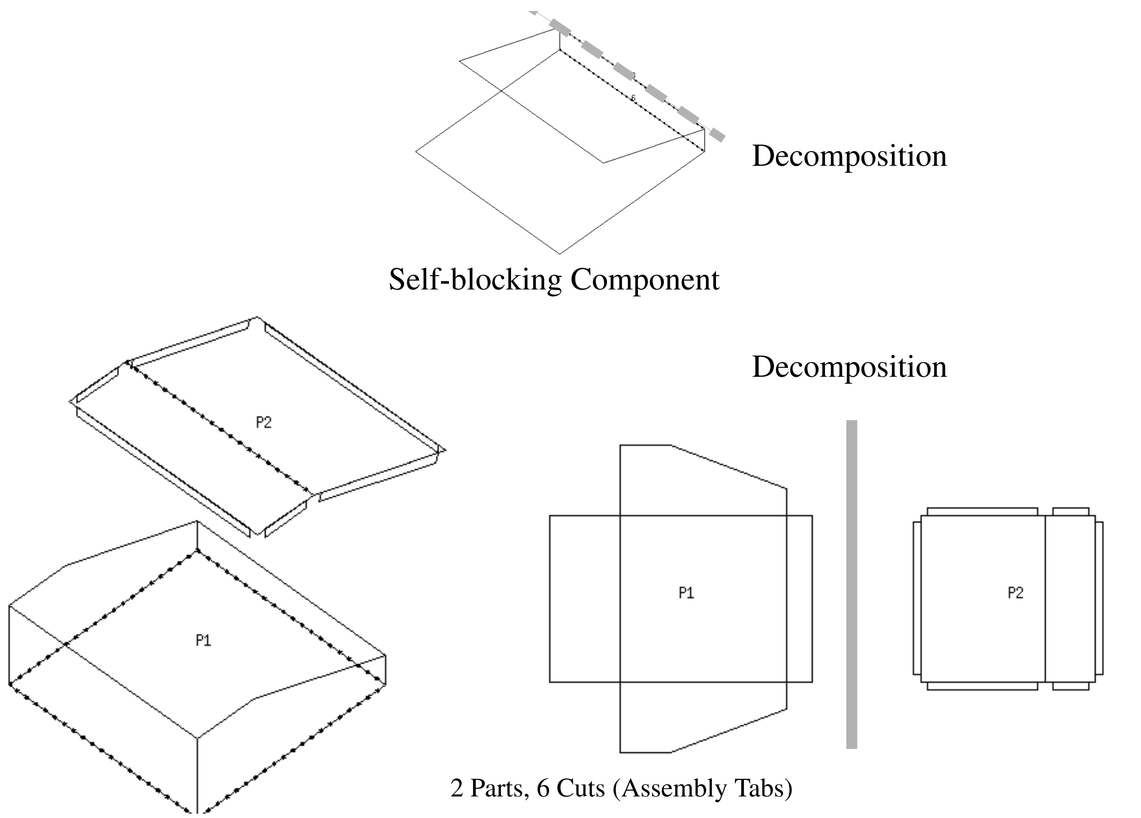


Figure 8.2: Decomposition of example 1.

Now the 3D shape and the 2D flat pattern of each resulting part are determined, the planning of the cutting, bending, and assembly processes proceeds. The cutting planner generates a nesting layout (Figure 8.3). In this layout, ten products are considered in the layout planning.

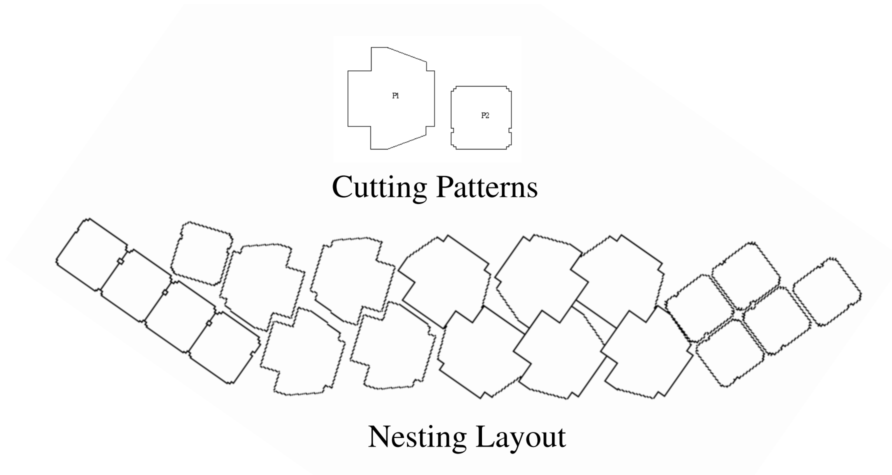


Figure 8.3: Cutting/nesting plan of example 1

The bending planner produces the bending plans for both parts (Table 3). The tools used in the bending plans are shown in Figure 8.4. Note that bend 0 in a bend sequence indicates the reposition operation for robot re-grasping. Bends that appear in the same parenthesis are to be bent simultaneously (For example, bends 7 and 4, and bends 2 and 5, in the bend sequence of P2 in Table 3). The decomposition results for P1 and P2 agree with their bending plans.

.

Bending Tools	**P1**	**P2**
Bend Sequence	(4 3 0 2 1)	((7 4) 1 0 (2 5) 6 3)
Robot Gripper	TG-1521-U55Q	IG-0725-U332Q
No. of Stages	1	1
Punch(es)	01600	01600
Punch Holders(s)	00001	00001
Die(s)	01080	01080
Die Holder(s)	01900	01900
Reason for Failure		

Table 3. Bending plans of example product 1.

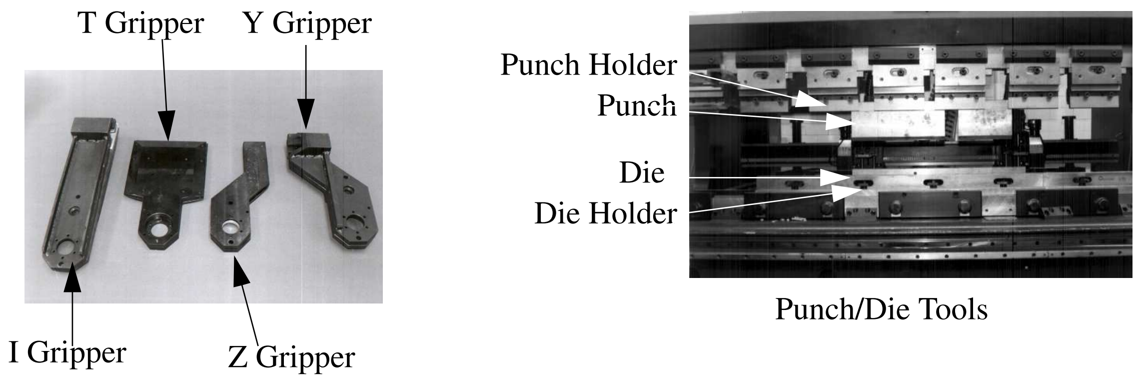


Figure 8.4: Tools in a bending plan.

The assembly planner generates an assembly plan which contains the assembly sequence and assembly direction of each part. Figure 8.5 shows the graphic simulation of the assembly plan. In the simulation, P_1 is moved to the assembly table and P_2 is assembled on top of P_1. All assembly directions are along thee negative Z direction.

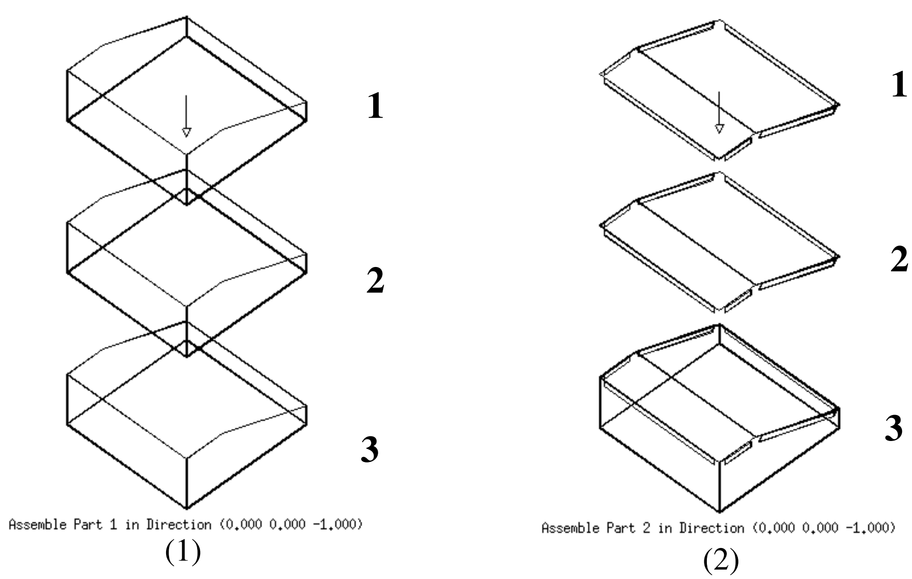


Figure 8.5: Assembly plan of example 1.

8.1.2 Example 2: An equipment cover

Figure 8.6 shows the 3D design of the cover for a profile measuring machine. The most compact 2D flat pattern is determined and shown in the same figure.

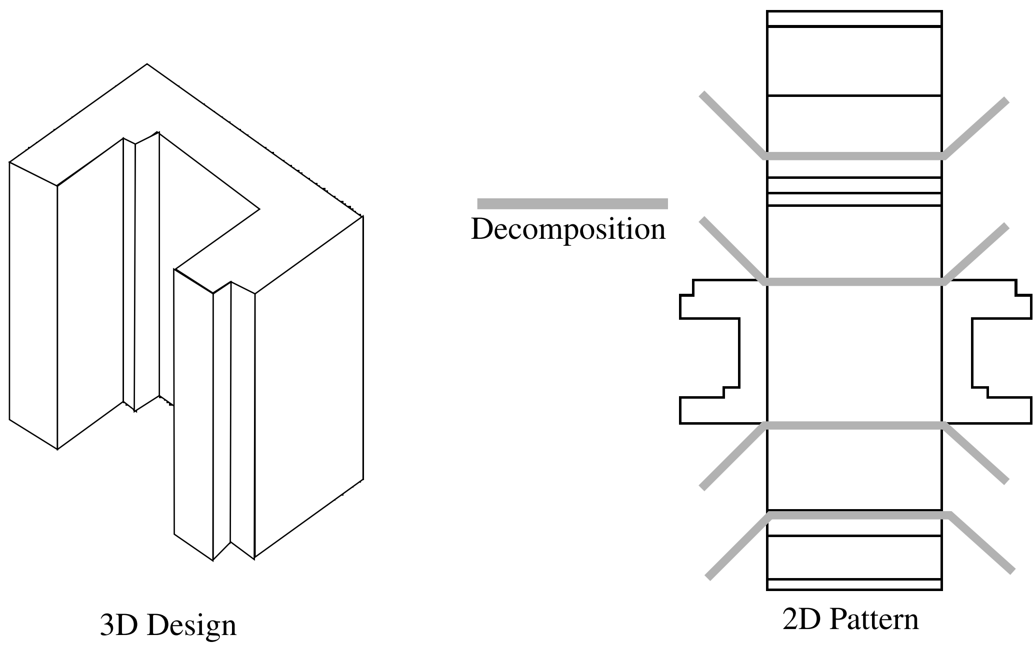
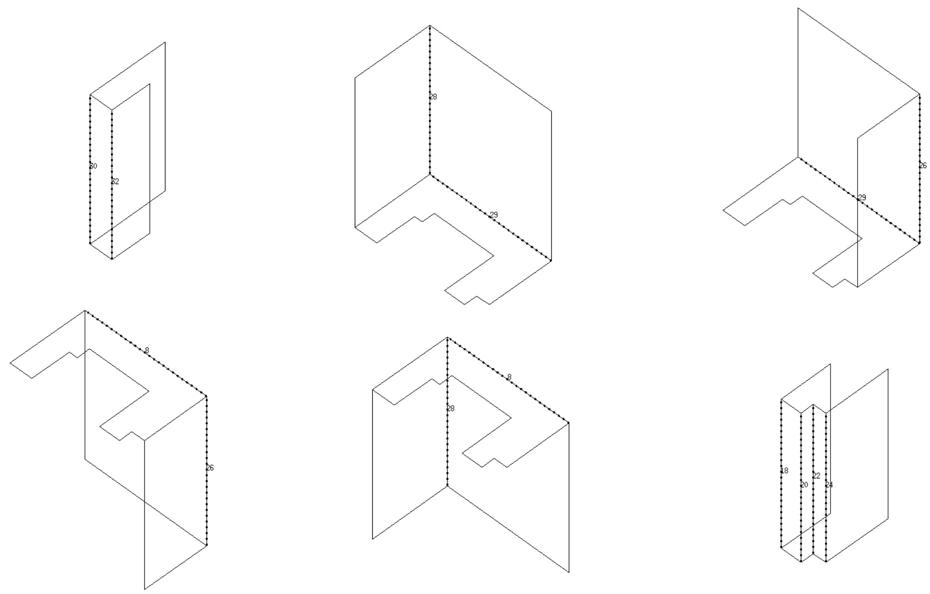
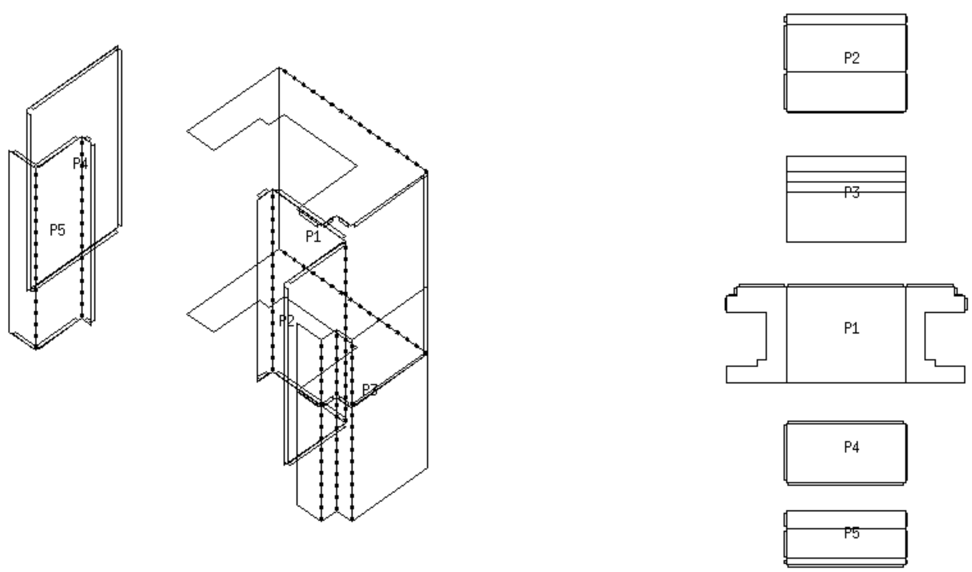


Figure 8.6: Example product 2.

In this example, six adjacent self-blocking components are found for the 2D pattern shown in Figure 8.6. The decomposition which results in five resulting parts and twenty-seven cuts is shown in Figure 8.7. Assembly tabs are assigned to parts with fewer number of bends.



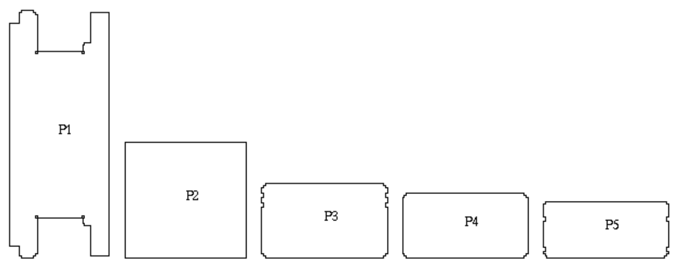
Self-blocking Components



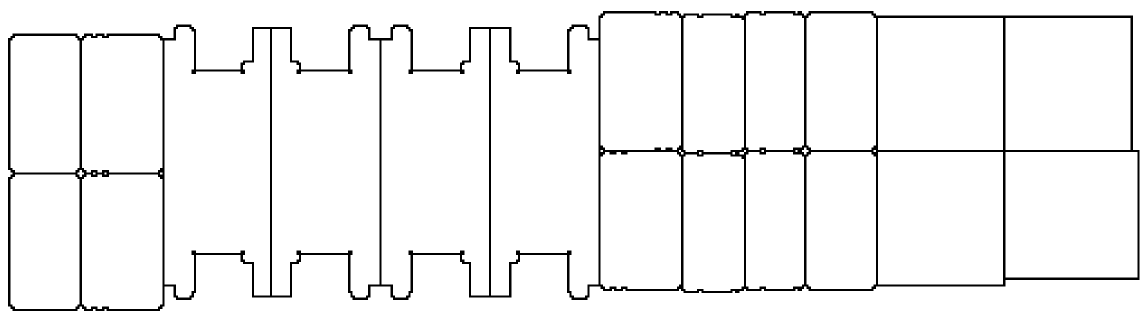
5 Parts, 27 Cuts (Assembly Tabs)

Figure 8.7: Decomposition of example 2.

The patterns and the nesting layout of five products are shown in Figure 8.8.



Cutting Patterns



Nesting Layout

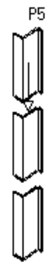
Figure 8.8: Cutting/nesting plan of example 2.

The bending plans are shown in Table 4. For Part P_1, the bending process planner cannot find a bending plan. For Part P_2, the bending planner reported a interference among part, tools, and the robot gripper. This is because in our tool accessibility analysis, the robot gripper is not modeled. The bending planner has successfully generated the bending plans for Part P_3, P_4, and P_5.

Bending Tools	P1	P2	P3	P4	P5
Bend Sequence			((10 7 4) (2 5 8) 9 0 1 3 6)	(4 3 2 0 1)	((9 6 3) (1 4 7) 8 5 2)
Robot Gripper			TG-1521-U55Q	TG-2025-U222Q	TF-3025Q
No. of Stages			1	1	1
Punch(es)			01600	01600	01600
Punch Holders(s)			00001	00001	00001
Die(s)			01080	01080	01080
Die Holder(s)			01900	01900	01900
Reason for Failure	No solution found	Part, tool and gripper interference			

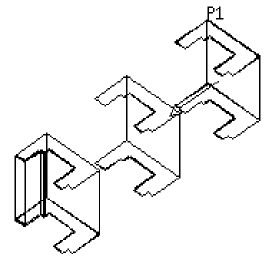
Table 4. Bending plans of example product 2.

The assembly sequence and assembly direction of each part for example 2 is shown in Figure 8.9.



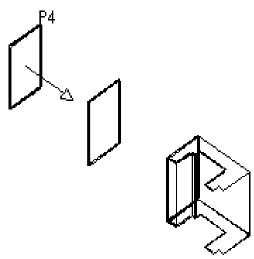
(1)

Assemble Part 5 in Direction (0.000 0.000 -1.000)



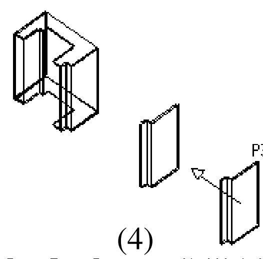
(2)

Assemble Part 1 in Direction (-1.000 0.000 0.000)



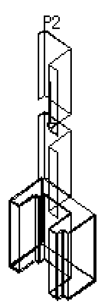
(3)

Assemble Part 4 in Direction (0.000 -1.000 0.000)



(4)

Assemble Part 3 in Direction (0.000 1.000 0.000)

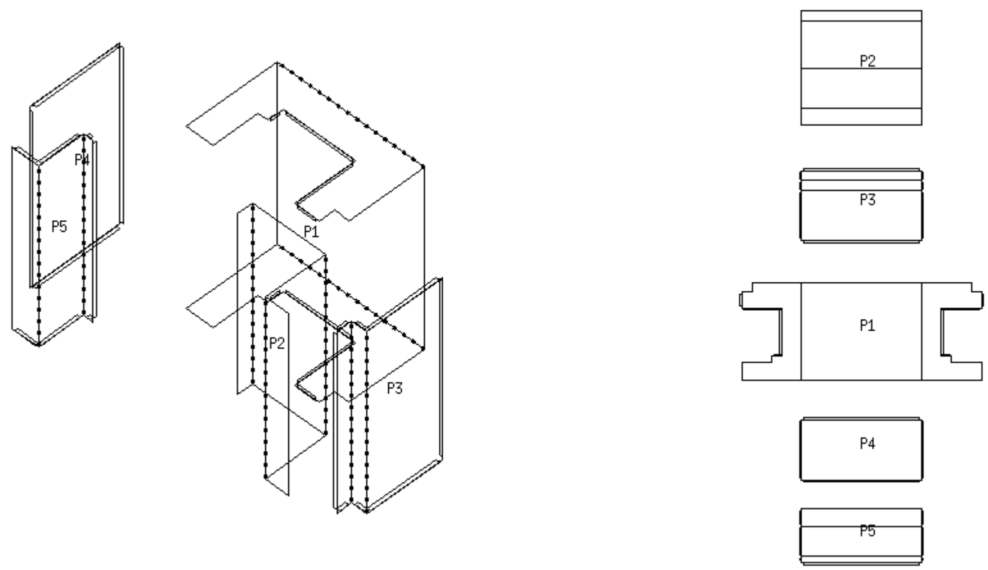


(5)

Assemble Part 2 in Direction (0.000 0.000 -1.000)

Figure 8.9: Assembly plan of example 2.

Another decomposition alternative with the same number of resulting parts and cuts is shown in Figure 8.10. Part P_2 and P_3 are different than the ones in Figure 8.7. The assembly tab assignment is also different.



5 Parts, 27 Cuts (Assembly Tabs)

Figure 8.10: Decomposition alternative of example 2.

8.1.3 Example 3: The cover of a robot component

The third example is the cover of the Y axis of our bending robot (Figure 8.11). For the most compact 2D pattern, there are twelve adjacent self-blocking components. The decomposition results in the minimum number of parts and cuts (seven parts and thirty-three cuts) is shown in Figure 8.12. The cutting patterns and nesting layout of three products are shown in Figure 8.13.

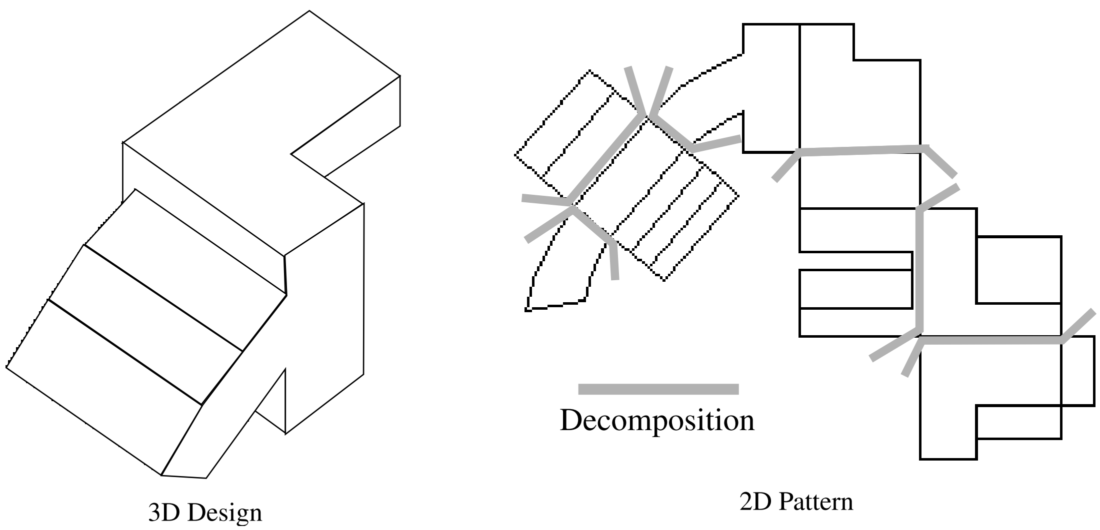
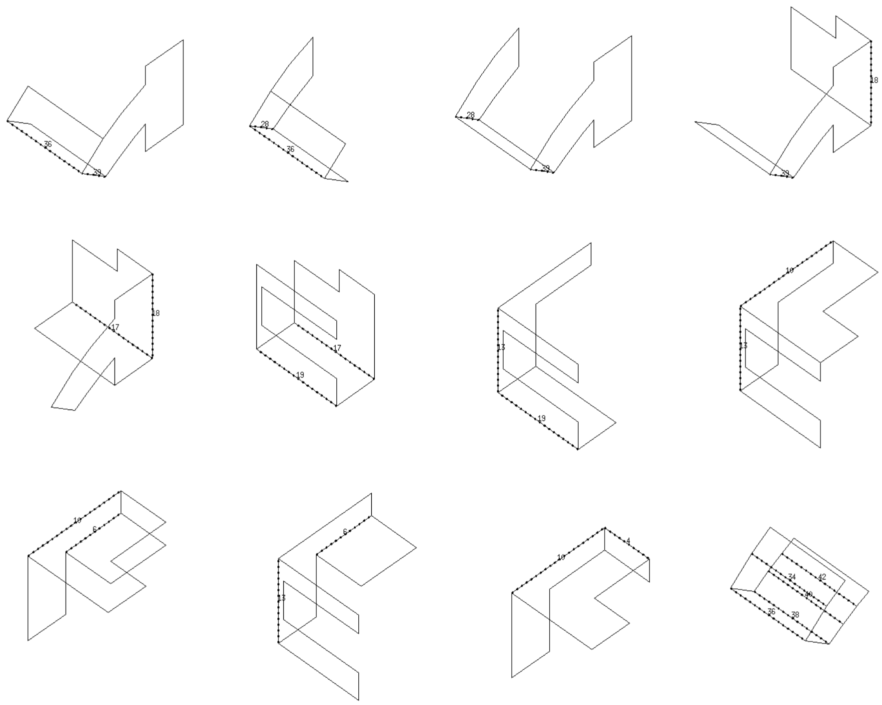
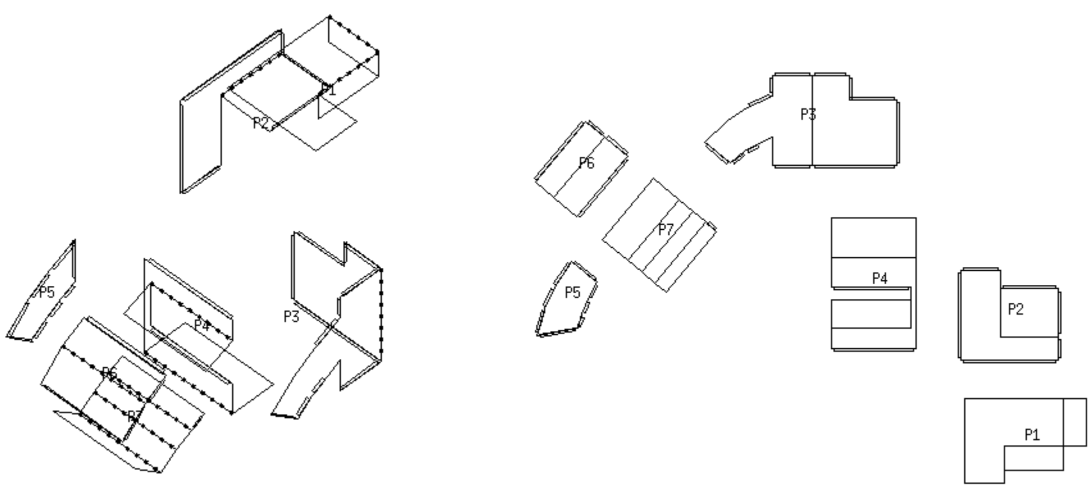


Figure 8.11: Example product 3

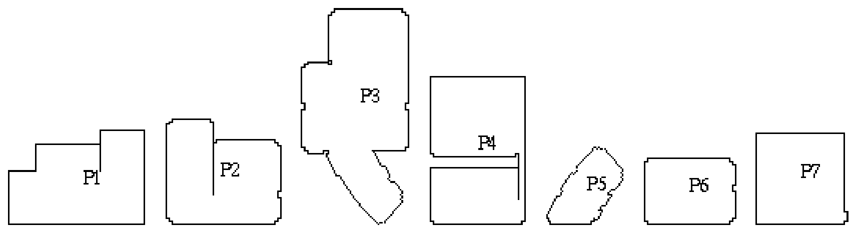


Self-blocking Components

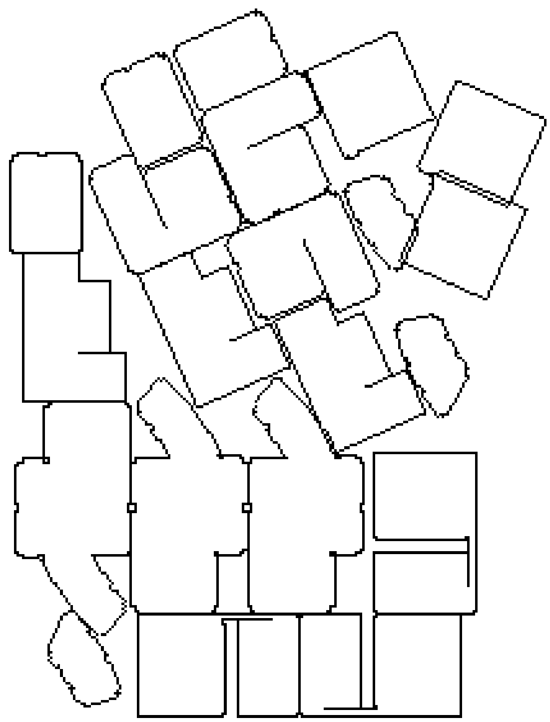


7 Parts, 33 Cuts (Assembly Tabs)

Figure 8.12: Decomposition of example 3.



Cutting Patterns



Nesting Layout

Figure 8.13: Cutting/nesting plan of example 3.

Although all seven parts of example 3 pass the tool accessibility test, the bending planner only generates the bending plan for Part P_1. In this example, most resulting parts have complex geometry. Actually, most of the resulting parts are considered difficult even for manual bending. The reasons of failure are listed in Table 5. We discuss the discrepancy between the decomposition and planning results in Section 8.4.5.

Table 5. Bending plans of example product 3.

Bending Tools	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**
Bend Sequence	(2 1)						
Robot Gripper	TF-2025Q						
No. of Stages	1						
Punch(es)	01600						
Punch Holders(s)	00001						
Die(s)	01220						
Die Holder(s)	02900						
Reason of Failure		Fine motion failed	Gaging failed	No feasible punch/ die	Gaging failed	No feasible punch/ die	Gaging failed

There is no feasible assembly sequence of example 3 with respect to the automatic tab assignment. However, with manual assignment, feasible assembly sequences are possible for this decomposition.

8.2 Decomposition of Non-Unfoldable Products

Unfoldability is the first factor to be considered when evaluating the manufacturability of bent sheet metal parts. However, some designs may not be able to be unfolded as a single piece, and further decomposition is necessary. Non-unfoldable parts must first be decomposed into unfoldable components. Unfortunately, there are no theorems or proofs that can determine if a given part is unfoldable or not without exhaustive tests. My approach to developing 2D flat patterns is to go through the search until the most compact pattern is found or the part is not unfoldable. Chung and Huang (1996) propose an approach to decompose non-unfoldable parts into unfoldable components. Their approach considers the unfoldability of each component after decomposition. However, their approach may be too conservative, and it sometimes generates more parts than necessary. For example, the part in Figure 8.14 can be unfolded into a single piece using my pattern development approach. Using Chung and Huang's approach, the product needs to be decomposed into two components.

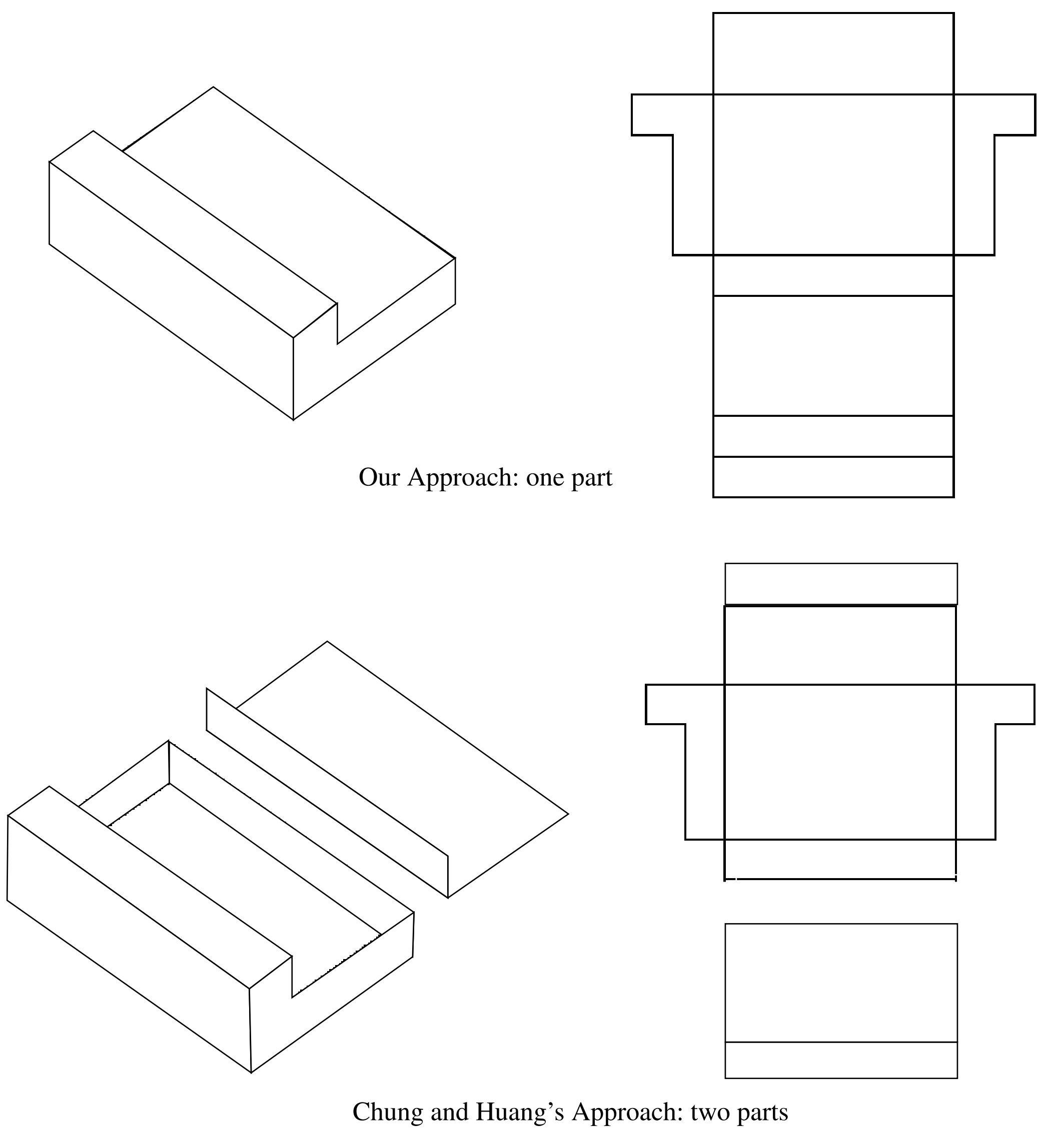


Figure 8.14: Pattern development considering unfoldability.

The decomposition of a non-unfoldable product is shown in Figure 8.15. Our pattern development approach first decomposes the product into two components (decomposition for unfoldability). Each component is then unfolded and the tool accessibility is analyzed (decomposition for accessibility). The decomposition is produced considering both components (see Figure 8.16).

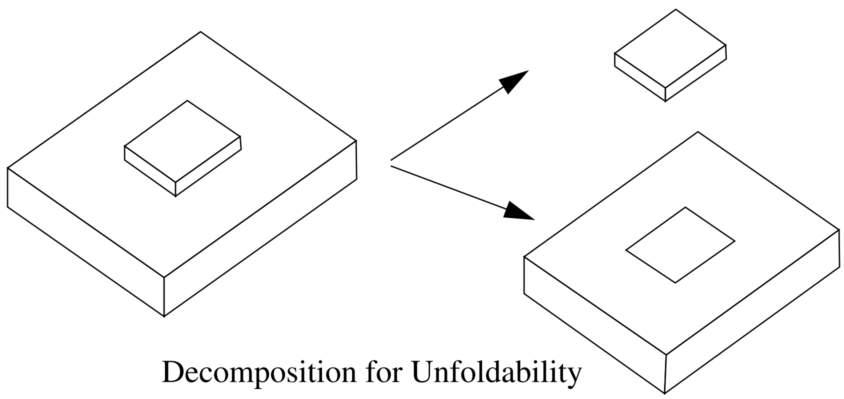


Figure 8.15: Decomposition of a non-unfoldable product.

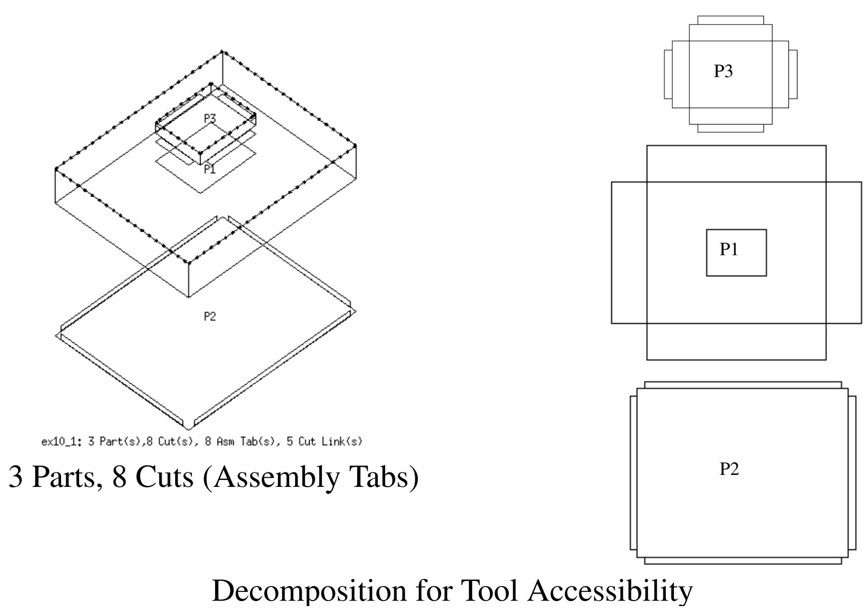


Figure 8.16: Decomposition of a non-unfoldable product (contd.)

The cutting patterns and the nesting layout of five products are shown in Figure 8.17.

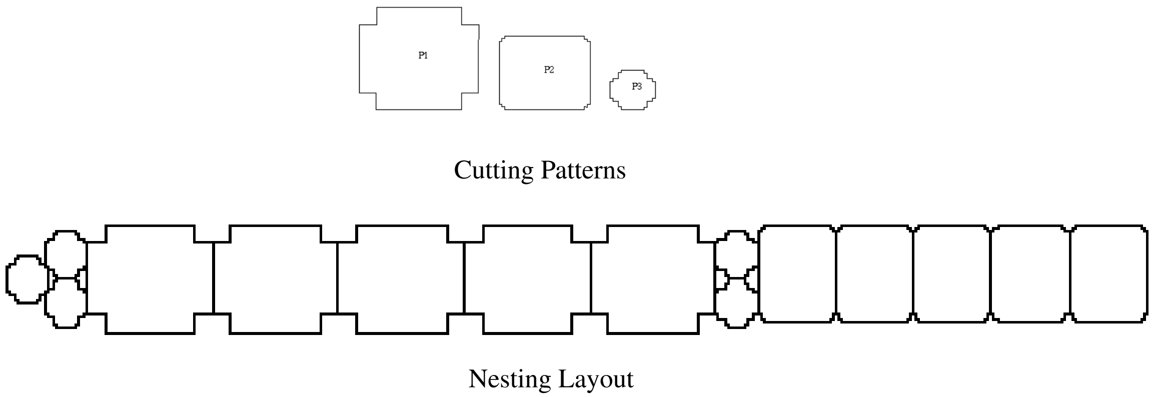


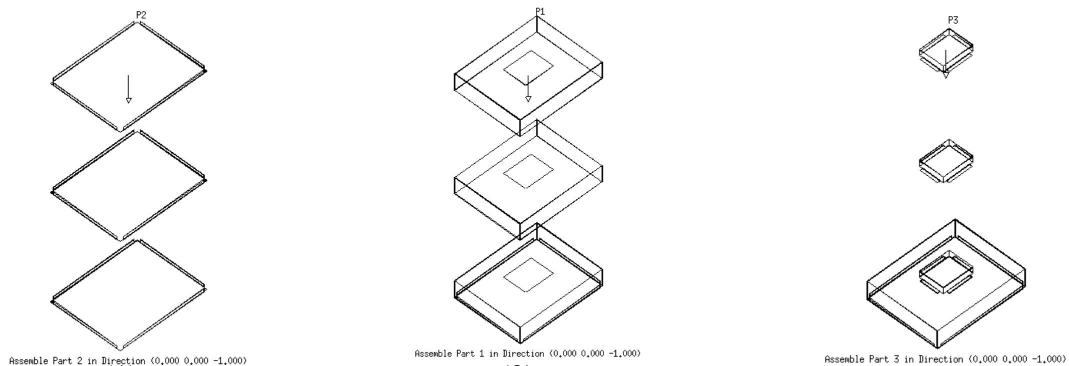
Figure 8.17: Cutting/nesting plan of the non-unfoldable product.

The bending plans of Part P_1, P_2, and P_3 are shown in Table 6.

Bending Tools	**P1**	**P2**	**P3**
Bend Sequence	(4 1 0 3 2)	(4 3 0 2 1)	(8 4 0 6 5 1 0 2 0 7 3)
Robot Gripper	TG-1521-U956N	IG-0725-U332Q	ZG-0725-U332Q
No. of Stages	2	1	1
Punch(es)	01600	01600	01600
Punch holders(s)	00001	00001	00001
Die(s)	01120	01120	01120
Die Holder(s)	02900	02900	02900
Reason for Failure			

Table 6. Bending plans of the non-unfoldable product.

An assembly plan has been generated and the assembly simulation is shown in Figure 8.18.



(1) (2) (3)

Figure 8.18: Assembly plan of the non-unfoldable product.

8.3 Decomposition Efficiency

The efficiency of the decomposition approach is shown in Table 7. All results were tested on a SUN Sparc 10 Workstation. The time includes the 2D pattern development, tool accessibility analysis, the assembly sequence planning, and the assignment of assembly tabs. Layout planning and bending process planning are not included, as they usually take longer than the decomposition process.

	Unfoldable Example 1	**Unfoldable Example 2**	**Unfoldable Example 3**	**Non-unfoldable Example**
No. of Faces	7	14	17	11
No. of Links	15	36	44	25
No. of Self-Blocking Components	1	6	12	1
No. of Parts	2	5	7	3
No. of Cuts	6	27	33	8
Time (in seconds)	55.5	310.7	368.8	92.3

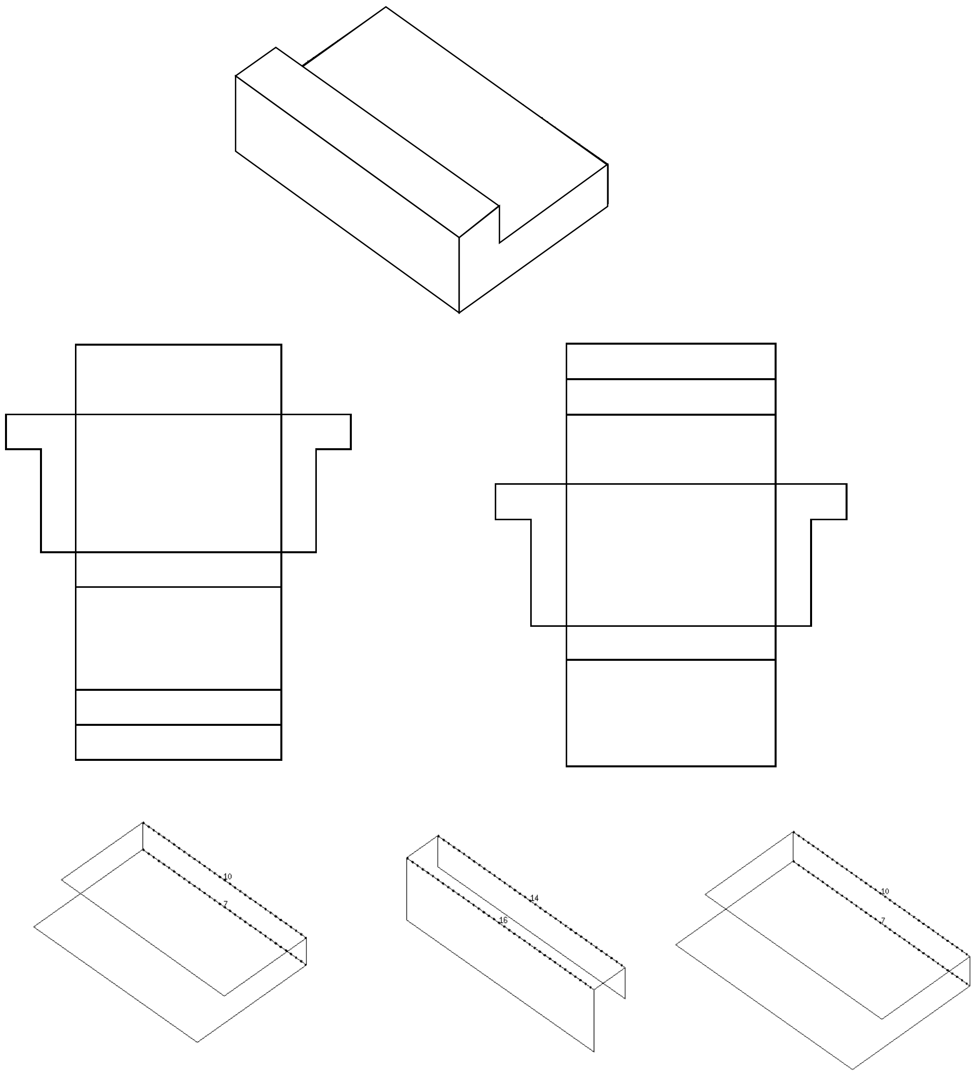
Table 7. Decomposition efficiency analysis

8.4 Discussion

Studying the decomposition examples described above, it becomes apparent that several issues significantly affect on the decomposition results. These issues include: the selection of compact patterns, tool availability, design variation, inaccessible parts with non-adjacent self-blocking components, and inconsistency between tool accessibility analysis and bending process planning. These are critical for further development of my current decomposition approach. They also suggest the direction of future work.

8.4.1 Selection of Compact Patterns

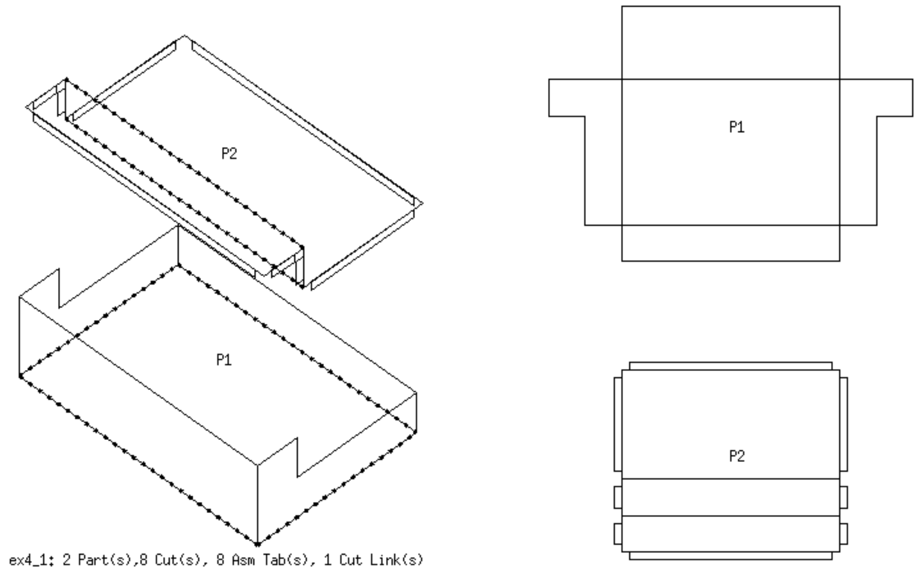
For a given product, the most compact 2D flat pattern might not be unique, even when both geometric and topological compactness are considered. For example, two of the 2D patterns of the L-shaped product in Figure 8.19 have the same compactness. Since the connectivity of the 2D patterns determines the adjacent self-blocking components of the product, it also determines how many parts result from decomposition. Figure 8.19(a) shows a pattern that has only one self-blocking component and results in two parts. On the other hand, the pattern in Figure 8.19(b) has two self-blocking components and three parts are decomposed. The corresponding decompositions are shown in Figure 8.20. This shows the correlation between the selection of the compact flat pattern and the number of self-blocking components. To generate less number of parts, the compact 2D flat pattern with less self-blocking components should be selected.



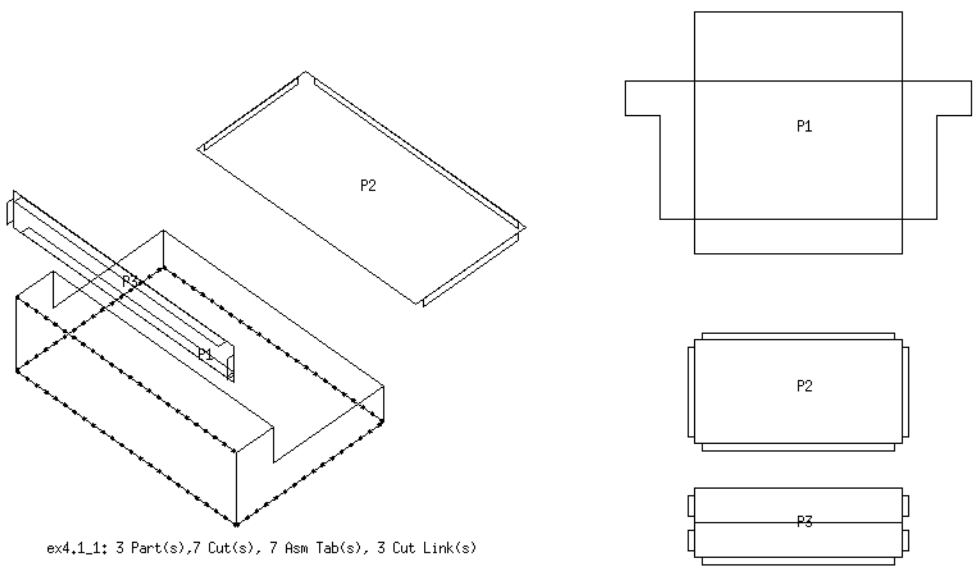
1 self-blocking component
(a)

2 self-blocking components
(b)

Figure 8.19: Patterns that have the same compactness value.



(a)



(b)

Figure 8.20: Decompositions due to different 2D patterns.

8.4.2 Tool Availability and Decomposition

The robot punch (Figure 8.21(a)) is the standard punch for most parts. However, there are special tools made to produce products with special features. For example, a goose-neck punch (Figure 8.21(b)) is used to access bends with stacked flanges (Figure 8.22). For an 8-bend box part (Figure 8.24), a hinged tool (Figure 8.25) is usually needed to be able to access some of the inner bends (Amada Sheet Metal Working Research Association, 1981).

Special tools are designed to handle particular features in bent sheet metal parts. They can simplify the bending operation or make parts bendable that are unbendable using standard tools. Unfortunately, they are expensive and geometry dependent.

Robot Punch (Punch-01600)
(a)

Gooseneck Punch
(b)

Figure 8.21: Standard punches used in the tool database.

My tool accessibility analysis uses a tool database to specify tools that are currently available. As the tool database changes, the decomposition results can be different. For instance, the part in Figure 8.22 can be made with a goose-neck punch. If a goose-neck punch is not available, further decomposition is necessary (Figure 8.23).

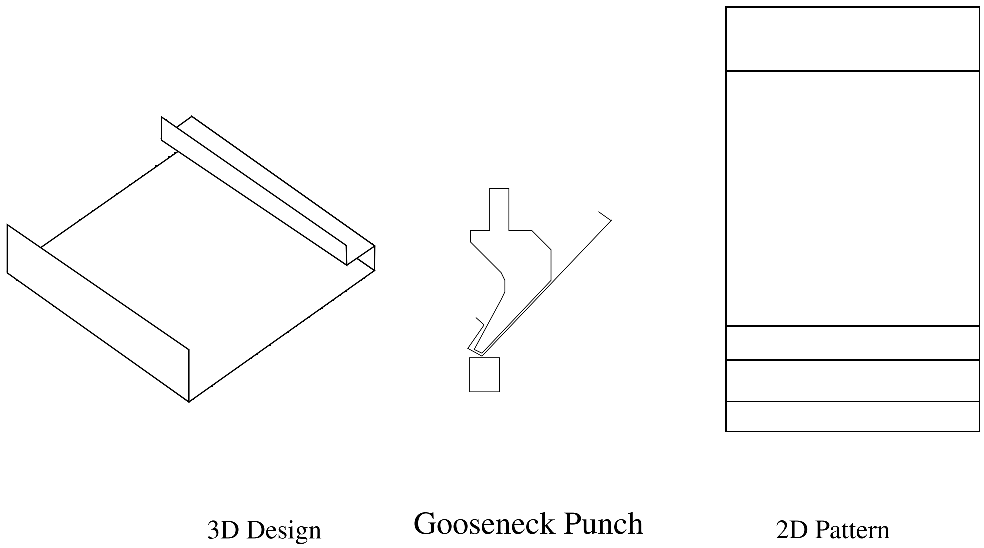


Figure 8.22: A part which can be made using a goose-neck punch.

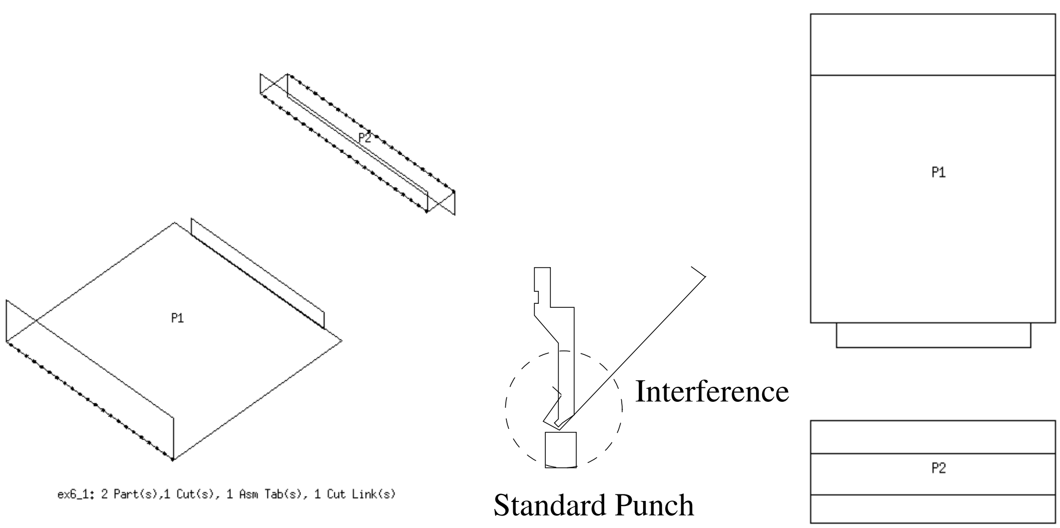
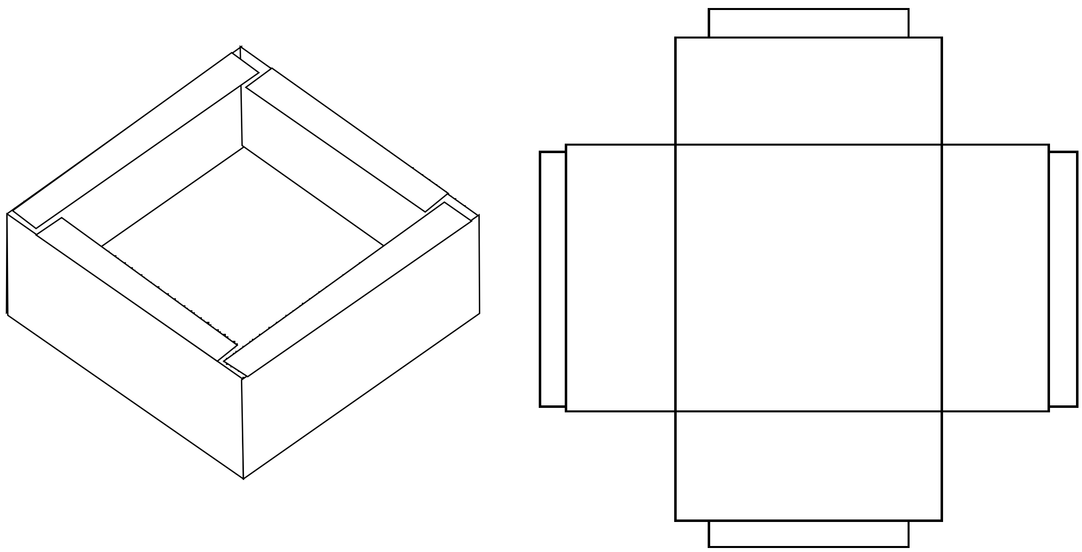


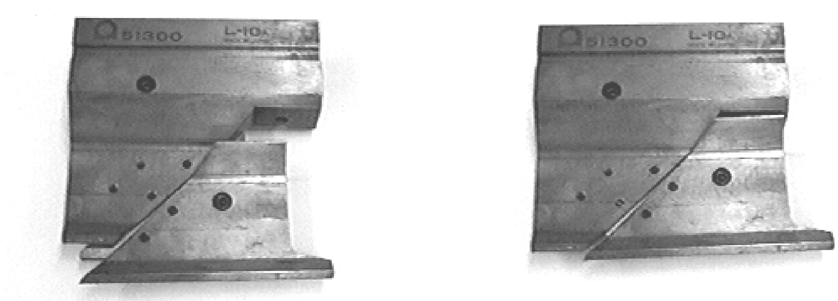
Figure 8.23: Further decomposition is needed without a goose-neck punch.

In Figure 8.24, we show an 8-bend box that requires a hinged tool (see Figure 8.25) in order to be made. A hinged tool will expand once it comes in contact with the die, and retract when it is not in contact. This property makes the bottom bends of the 8-bend box accessible. Without the hinged tool, additional decompositions are needed (Figure 8.26).



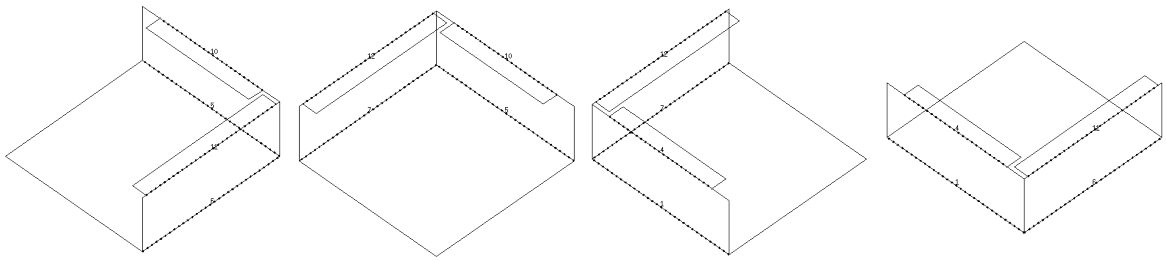
3D Design 2D Pattern

Figure 8.24: An 8-bend box that can be made using a hinged tool.



Retracted Tool Expanded Tool

Figure 8.25: Hinged tool (punch).



Self-blocking Components

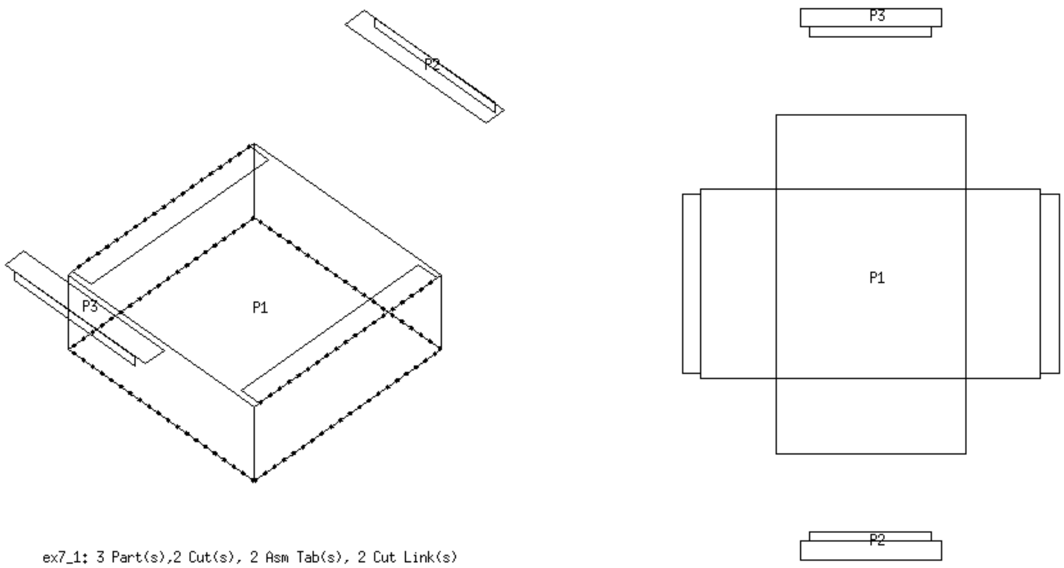


Figure 8.26: Further decomposition is needed without the hinged tool.

8.4.3 Decomposition of Similar Parts

Product design is an iterative process. A design needs to go through minor design variations in order to satisfy functional and manufacturability requirements. The ability to adjust the decomposition due to design variation, especially in the shape or geometry, is time-critical during product development. In this section, I show two examples to illustrate how parametric design changes directly affect the decomposition results. The first example is an open 4-bend box (Figure 8.27). It is a common sheet metal component. If the designers would like to increase the volume of the box, the height of the box (Figure 8.28) might be modified. The new design might no longer be bendable and may need further decomposition. The second example shows two designs with different heights, which could end up with the same number of parts but totally different decompositions in geometry (Figure 8.29).

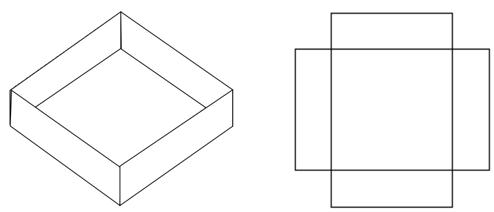
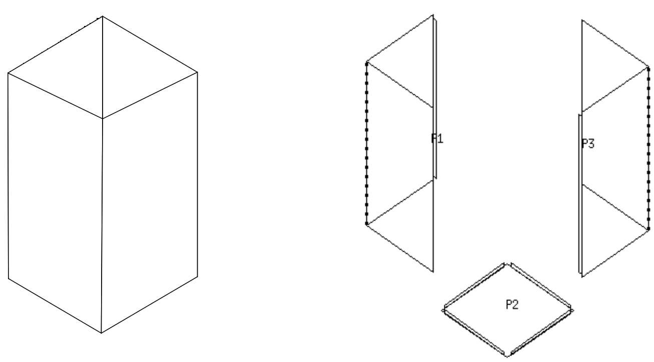


Figure 8.27: A bendable 4-bend box.



3 Parts, 6 Cuts

Figure 8.28: A tall 4-bend box and its decomposition..

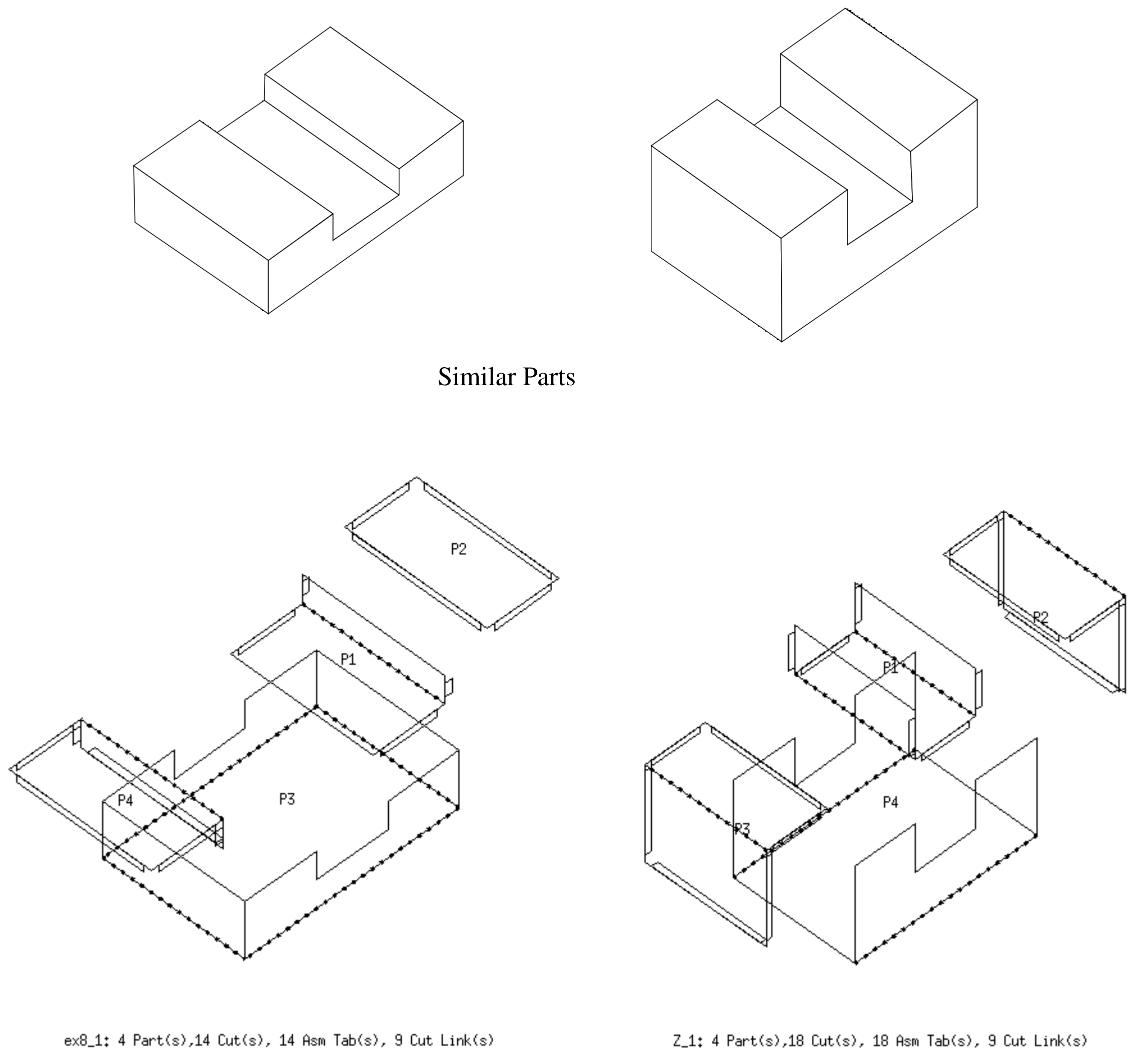


Figure 8.29: Different decompositions for similar parts (contd.).

8.4.4 Non-Adjacent Self-Blocking Components

The proposed decomposition approach is primarily based on the adjacent self-blocking components defined in Chapter 6. The goal of decomposition for tool accessibility is to break the connectivity of all self-blocking components. To make the search for self-blocking components tractable, only components that are connected by inaccessible links are considered. Self-blocking components are determined by testing the accessibility of every bend in the final shape. Only the smallest connected sets are generated. This kind of self-blocking component might only be a subset of the inaccessible components of the parts. There are some parts that do not have any adjacent self-blocking components, but are inaccessible. That is, not all bends in the part are inaccessible. This kind of part is known as a non-adjacent self-blocking component. Figure 8.30 shows an example. This part has three bends and no self-blocking component. But it is inaccessible for any unfolding (or folding) sequence. This is because some of the links are accessible before unfolding a certain bend but become inaccessible after that bend. It is possible that the resulting parts remain inaccessible even after all self-blocking components have been decomposed. To detect non-adjacent self-blocking components is usually difficult and intractable since all combinations must be considered. Currently we deal with this kind of part by further decomposing one link at a time until all resulting parts are accessible. One good heuristic, to decide which link to decompose first in a non-adjacent self-blocking component, is to decompose the link with the tallest flange height (this link usually causes accessibility problems). The decomposition in Figure 8.31 utilizes this heuristic; the link with the tallest flange height (B_1) is decomposed first. Actually, for this part, decomposing any link will make the resulting part accessible.

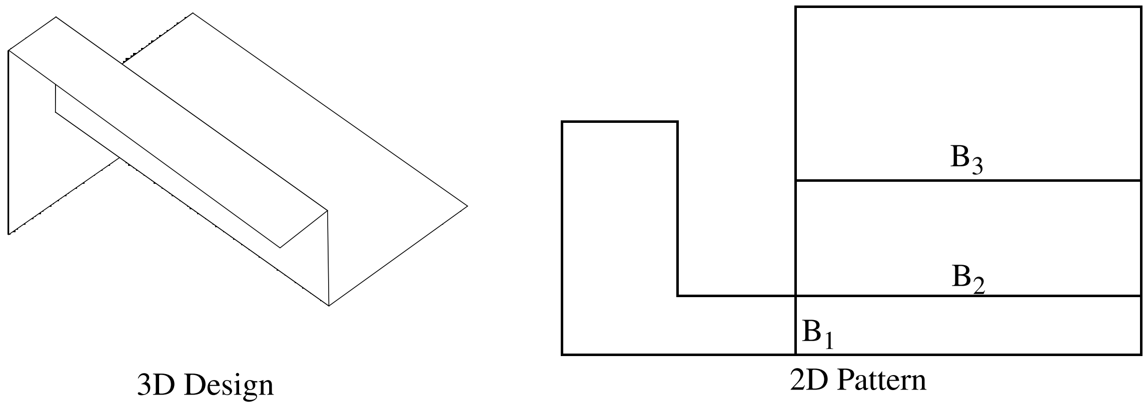


Figure 8.30: An inaccessible part without adjacent self-blocking components.

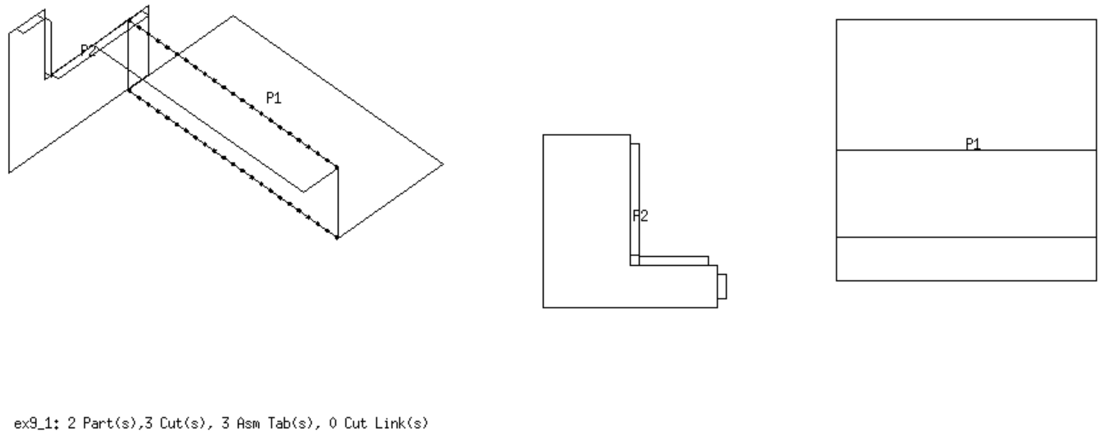


Figure 8.31: Decomposition of a non-adjacent self-blocking component.

8.4.5 Tool Accessibility Analysis and Bending Process Planning

In my decomposition approach, tool accessibility is used to simplify the evaluation of part bendability, which eventually must be determined by the bending process planner. Unfortunately, tool accessibility only represents one major concern in bending process planning.

For a manually operated bending machine or a robot-assisted bending system, there are other issues which need to be addressed. For instance, the automatic bending system operates with a robot that handles the part during bending. The selection of a robot gripper, the determination of a grasping position (Kim 1996), the gross motion and fine motion planning for the robot, and the motion of a backgage (see Figure 8.32)to ensure bend quality are other issues to be considered. The robot also has limitations on where it can reach (workspace limitations). On the other hand, manually operated machines have fewer problems in part handling, since human operators are dexterous. However, some basic problems still remain. There are parts that cannot be made manually. In some cases, the shape of a part prohibits the backage from aligning on any contact edge so that gaging is impossible, or upside-down tools are needed in order to access certain bends (which is not considered in my accessibility analysis). As a result, parts that pass the tool accessibility test may still not be manufacturable either on a manual or automatic bending machine, but this can be determined with more complete process planning.

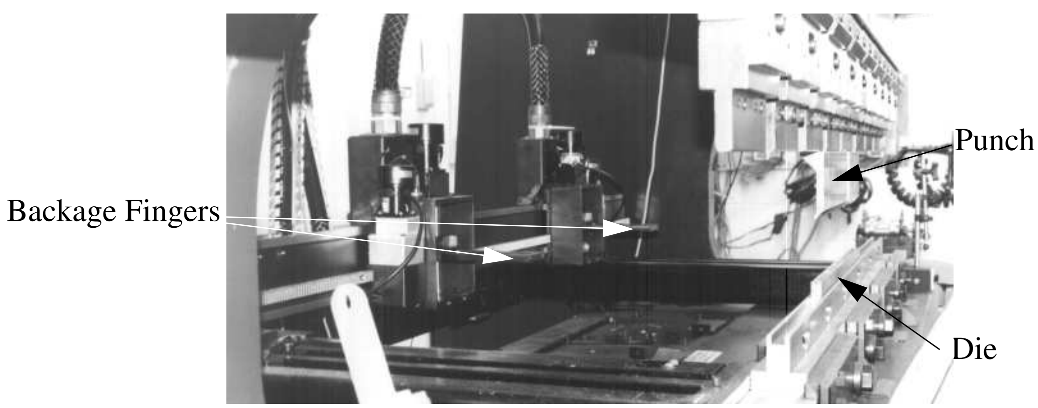


Figure 8.32: Backage.

Bending plans for example products are shown in Tables 3, 4, 5, and 6. For Although all the resulting parts are unfoldable and accessible, there are 8 out of 17 parts that fail to produce bending plans from the bending process planner. The resulting parts become more difficult to plan after the assembly tabs are assigned (more bends added to the parts). Typical reasons of failure are due to gaging problems, motion planning problems, and tool selection problems. This is because we only use tool accessibility to approximate the bendability during product decomposition.

Tool accessibility provides a strong constraint when determining if a given part can be accessed using available tools. Tool accessibility analysis aims to find an accessible unfolding (bending) sequence quickly. No attempt is made to try to achieve optimality of the sequence. On the other hand, the bend sequence derived by the bending process planner in the prototype system is optimal considering the tooling, grasping, robot motion, etc. Therefore, the sequences from the accessibility analysis and the bending process planner might not agree. However, the tools selected and the tool orientations are fairly consistent between these two approaches.

The failure of the decomposed parts provides different guidelines for part bendability. Most failure parts in our examples are considered difficult parts even for manual bending. We have learned that the gaging of bends plays a critical role in determining whether a part is manufacturable on either a manual or a robot-assisted bending machine. For example, the backgage fingers should be able to access all bends and provide good contact to ensure bend precision. Figure 8.33(a) shows one bend that is impossible to access due to part geometry and Figure 8.33(b) shows another gaging problem -- non-parallel contact edges. These kinds of problems may result directly from the assignment of assembly tabs, and can be avoided during tab assignment. Other gaging problems are shown in Figure 8.34. For example, Y gaging fails if the only place to gage is shorter than the minimum gaging distance (Figure 8.34(a)). Also, the backgage fingers are physically constrained so that they have to move to the same Y position, although they can move separately in X and Z directions. In order to obtain good alignment for gaging, parallel contact edges (with respect to the fingers) are necessary (Figure 8.34(b)).

Another problem causing planning failure is related to tool selection for acute angle bends. Acute angle tools have more constraints on tonnage and material thickness than tools for 90 degree bends. Two parts have this kind of problem and no feasible tools can be found. This problem can be resolved by adding acute angle tools to the tool database.

There are failures caused by the fine motion planning. The part may not be retracted from the punch/die space even it does not interfere with the tools. The part or tools may interfere with the robot gripper as the gripper is not modeled in our tool accessibility analysis. These problems are usually not encountered in manual bending.

It is noted that most of the problems are related to the addition of assembly tabs. These results can be used as feedback information in assembly tab assignment.

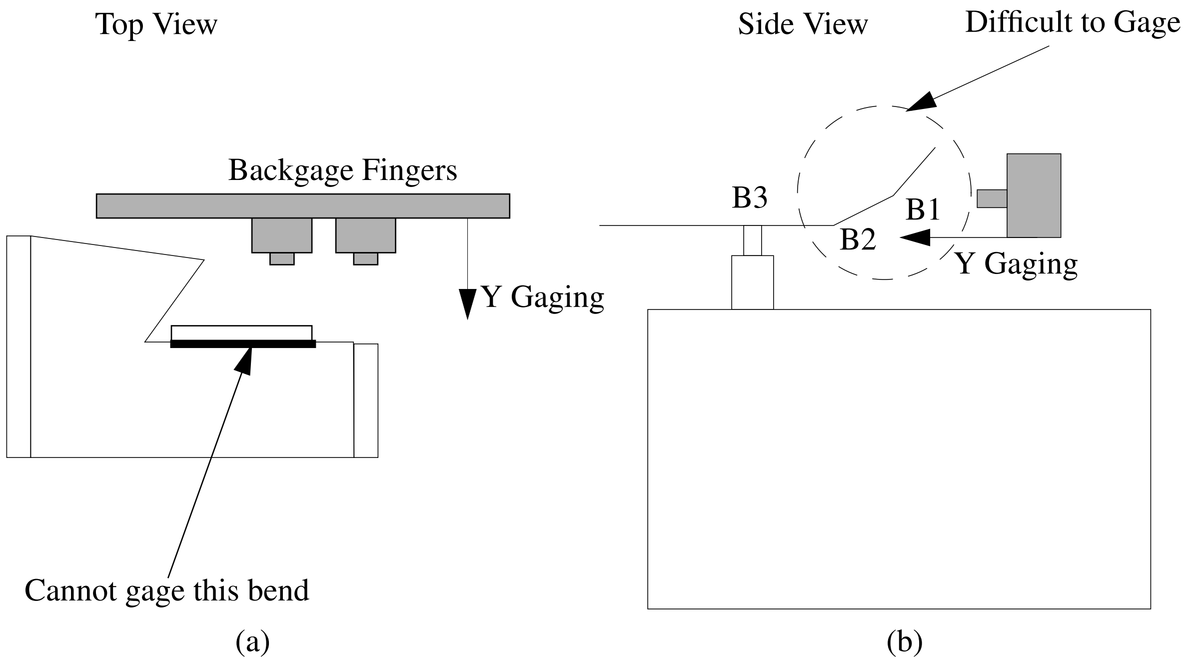


Figure 8.33: Possible gaging problems.

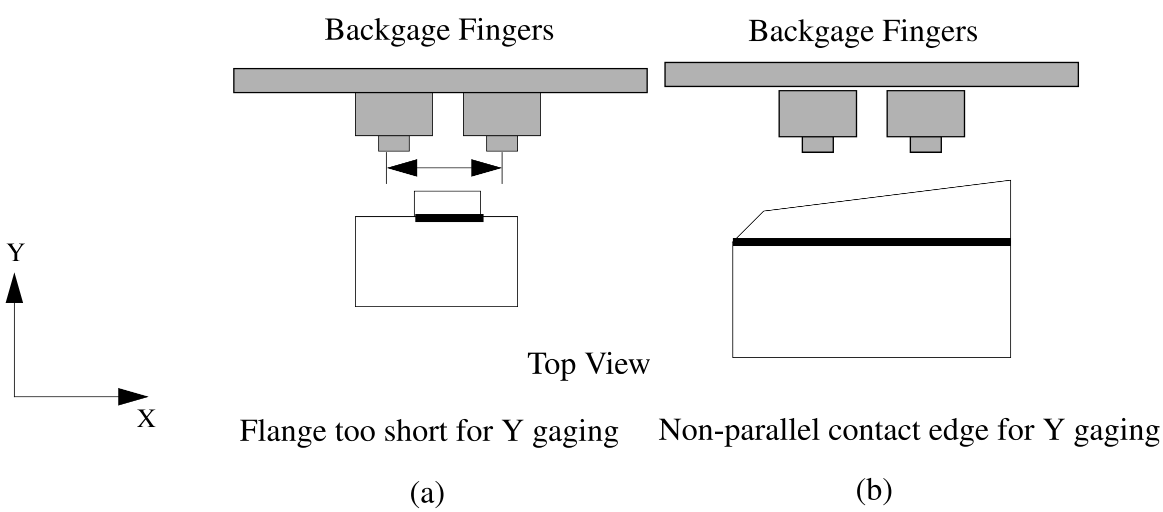


Figure 8.34: Possible gaging problems (contd.)

Assembly tab assignment plays an important role in deciding whether the resulting parts are bendable or not. The addition of assembly tabs can greatly simplify welding. Although we have taken into account the unfoldability issue in 2D, there could be further gaging or tool selection problems if the assignment is inappropriate. These issues are to be included in future methods of assembly tab assignment.

During the decomposition stage, manufacturability evaluation sometimes can help designers determine why a given part is not manufacturable. This information can provide useful feedback and improve designs. For instance, gaging problems and acute angle bends are also critical issues in sheet metal design. We also plan to conduct a detailed study on how to use our decomposition results to provide design feedback.

8.5 Conclusion

In this chapter, we have applied our decomposition system to several products. We have shown that the decomposition approach is practical and can adapt to facility and design changes. The discrepancy of the results between tool accessibility and the bending process planner is discussed. This discrepancy is due to the incomplete manufacturability evaluation during decomposition. We believe that by taking more manufacturability considerations into account during the decomposition process (adding more feasibility checkers), the discrepancy will be reduced and the performance of the decomposition system will improve.

Chapter 9

Conclusion and Future Work

9.1 Conclusion

This thesis presents a manufacturability-driven decomposition approach to product design. This approach contrasts with the traditional decomposition approach, which is primarily based on product functionality. I have shown that by taking account of product manufacturability during the decomposition stage, product design can be sped up and product cost reduced. Different decomposition alternatives are generated in order to reduce product cost. We anticipate that if products are primarily represented in their form or geometry, our decomposition approach will have many advantages over ones based on functionality.

We have applied the decomposition approach to bent sheet metal products and the results are promising. A prototype system has been developed to verify this decomposition approach.

I have listed the contributions and limitations of my thesis work. Further discussions of the future work and potential applications are also given.

9.2 Contributions

The major contributions of this thesis are the following:

- A systematic and tractable decomposition approach for bent sheet metal products driven by manufacturability
- A decomposition system which adapts to sheet metal design and facility changes
- The formalization of the modeling of cutting, bending, and assembly processes for sheet metal products/parts and the development of efficient algorithms for manufacturability analysis
- An approach to pattern development considering geometrical and topological unfoldability and compactness
- Automatic design modification (assigning of assembly tabs) of the product after decomposition for bent sheet metal products
- A product decomposition system for bent sheet metal product development

9.3 Limitations

Several simplifying assumptions are made for the proposed decomposition approach. These assumptions are listed as follows:

1. Functionality is not directly addressed in the decomposition. The decomposition approach still requires the designer's judgement and interaction during the decomposition stage.

2. Only key manufacturing processes are considered. In each process, only some critical issues are addressed. These issues serves as feasibility checkers to determine the corresponding manufacturability. As a result, the decomposition results might not reflect the complete manufacturability of the product as evaluated by these process planners.

3. Manufacturability evaluation is primarily based upon geometry. Non-geometric issues are not addressed in this thesis.

4. Global optimality is not guaranteed. I have shown that by adopting some manufacturing preferences based on DFM and DFA, the proposed approach will overcome the combinatorial nature of the decomposition problem and sub-optimal solutions can be obtained.

The limitations mentioned above suggest the direction of my future work.

9.4 Future Work

This thesis shows the potential of conducting product decomposition based on product manufacturability. The prototype system presented here demonstrates the feasibility of my approach. However, in order to develop a comprehensive decomposition system for sheet metal products we need to:

1. Take into account functional requirements in the product decomposition process. Investigate the possibility of defining limited function-form relationships and using them to automate the decomposition process. For example, we could try to improve the inside/outside assignment of the assembly tabs using more detailed design constraints.

2. Consider other manufacturing processes during the decomposition process, such as nesting, stacking, painting, welding and so on. Also, take into account more manufacturing constraints during manufacturability evaluation (adding more feasibility checkers).

3. Focus on non-geometric issues in manufacturability evaluation such as product strength and ease of assembly (Diaz et al. 1995).

4. Currently, no efficient algorithms can determine if a given sheet metal product is unfoldable or not. A convex product seems to be always unfoldable. However, a concave product might or might not be unfoldable. Further studies of product unfoldability, such as proofs and tractable algorithms, are necessary. This work is currently underway in our project.

5. Replace the preliminary process planning systems with more sophisticated planning systems. In my prototype system, two primitive process planning systems, cutting (nesting) and assembly planning, have been developed. In the future, we would like to replace these primitive systems with more sophisticated planning software. We would like to integrate the design, decomposition, and process planning modules so that they can work in a closed-loop manner.

6. Build better manufacturing cost models to reflect the realistic drives of the decomposition. Currently, the cost of an individual manufacturing process is estimated using some DFM and DFA guidelines. In reality, better cost models which reflect the true costs of manufacturing processes, are needed.

7. Take advantage of feedback from decomposition results to further improve designs.

Finally, we would like to apply the decomposition approach to different products/processes in other applications. Some of the possible applications of manufacturability-driven decomposition are discussed next.

9.5 Potential Applications

The success of applying the proposed decomposition approach in this thesis depends on the following considerations:

1. Product design is primarily form-driven and manufacturability can be evaluated mainly based on product geometry.
2. Manufacture of products becomes easier if further decompositions are applied.
3. There exist tractable decomposition assumptions and algorithms that meet product requirements.

It is true that for different products and manufacturing processes, different decomposition assumptions, decomposition preferences, modeling of manufacturing processes, and different manufacturability evaluation to decomposition, need to be applied. However, the manufacturability-driven concept remains the same for different products.

The following products/processes are considered as potential applications for my decomposition approach: paper packaging design, pressing dies, injection and casting model design, and laminated manufacturing.

9.5.1 Paper Packaging Design

Packaging boxes made of paper are used in everyday life. They vary from simple designs such as the tea box (with 12 bends) shown in Figure 9.1 to very complex designs like the one for computer accessories (with 37 bends) shown in Figure 9.2. However, packaging boxes have a lot in common; they are usually made of a single sheet by folding the sheet;

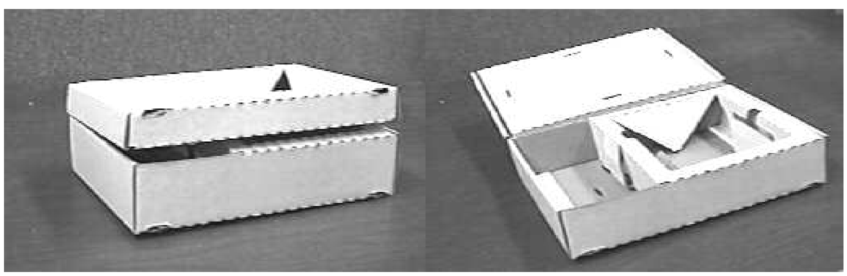
additional tabs are usually needed in order to glue the product; and inter-locking designs are sometimes used to maintain the shape of the product or to serve as one way to access the interior of the design. In current practice, highly automatic machines cut, fold and glue the packaging design all at once.

The manufacturing processes of paper packaging design are conceptually similar to those of bent sheet metal products. Both need cutting, bending (or folding), and assembly processes. In terms of economic considerations, the reduction of the number of parts and the compactness of the 2D flat pattern are common preferences. The unfoldability issue remains valid for both products.

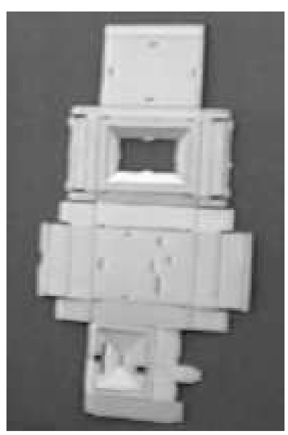
However, there are some major differences between these two products/processes. For instance, the bend deduction for paper is small and can be ignored. The folding process for packaging design is different than the bending process for sheet metal parts. The paper is flexible and the folding angles can be adjusted to fit the designed shape. As a result, some intermediate interferences are allowed during the folding operations. Tool accessibility is relaxed for packaging design. Unfortunately, the folding process is usually much more difficult than sheet metal bending in terms of final geometry. Currently, highly complex packaging designs still cannot be fully automated and need human assistance.

Paper Tea Box and Its Flat Pattern

Figure 9.1:A simple paper packaging design.



3D Views



2D Blank with 37 Bends (folded in halves)

Figure 9.2:A complex packaging design for computer component.

9.5.2 Pressing Dies, Injection and Casting Mold Design

For dies and molds that have complex shapes, it is desirable to break the dies or molds into pieces that are easy to manufacture (Stanley, 1919). In addition, maintenance and replacement due to wear or breakage become much easier in dies or molds which have been broken down. Figure 9.3 shows a sectional die for blanking a piece that is bent into a cover for a machine mechanism. There are a number of sharp corners in the die which usually cause machining problems. I show that by dividing the die into parts with much simpler outlines, the machining problem can be avoided. The machining and finishing of the sectional die must be done with the knowledge of the changes in dimension that are likely to occur when the parts are hardened, and a corresponding allowance can be made, if necessary, for correction before assembly.

The manufacture of sectional dies primarily involves machining and assembly processes. The number of resulting parts might not be the major concern, but instead the ease of machining and assembly should be taken into account first.

Injection or casting mold design is another candidate for the decomposition approach as the parting line/surface and draw direction determination are directly related to part geometry and are critical in model design.

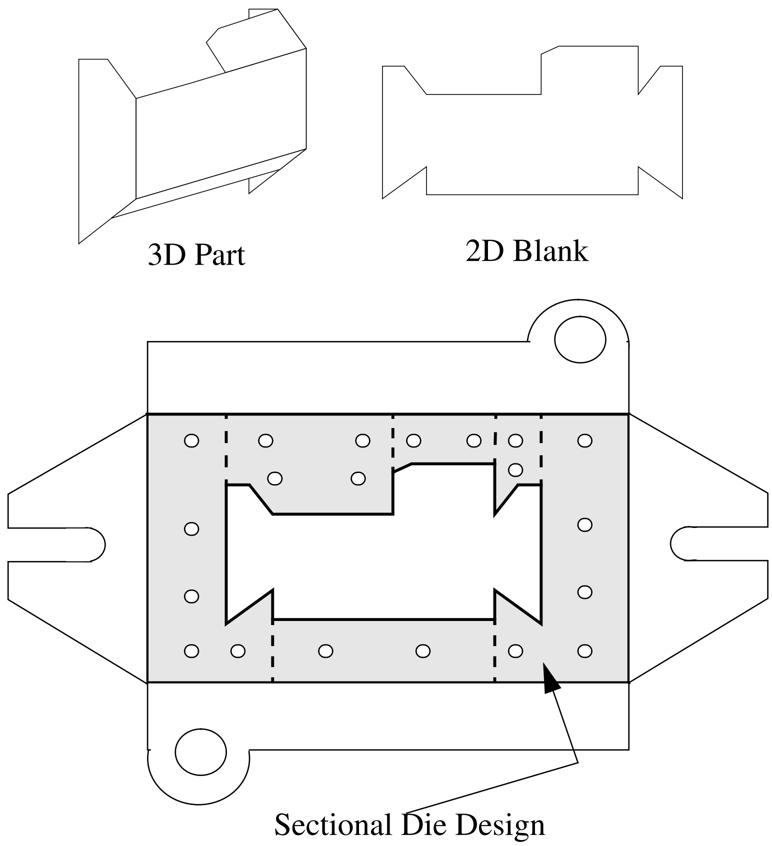


Figure 9.3: An example of a sectional die design (a mechanism cover).

9.5.3 Laminated Manufacturing (Rapid Prototyping)

Another possible application of the decomposition approach is in laminated (layered) manufacturing. In this process, the 3D solid model of a part is first decomposed into 21/2 D cross-sectional layers. A process planner then generates the guiding trajectories for the material additive process. The decomposed layers are ultimately built up layer by layer in an automated fabrication machine to form the part. To provide the necessary support for building a new layer, sacrificial supporting layers are generated simultaneously as the part fixtures.

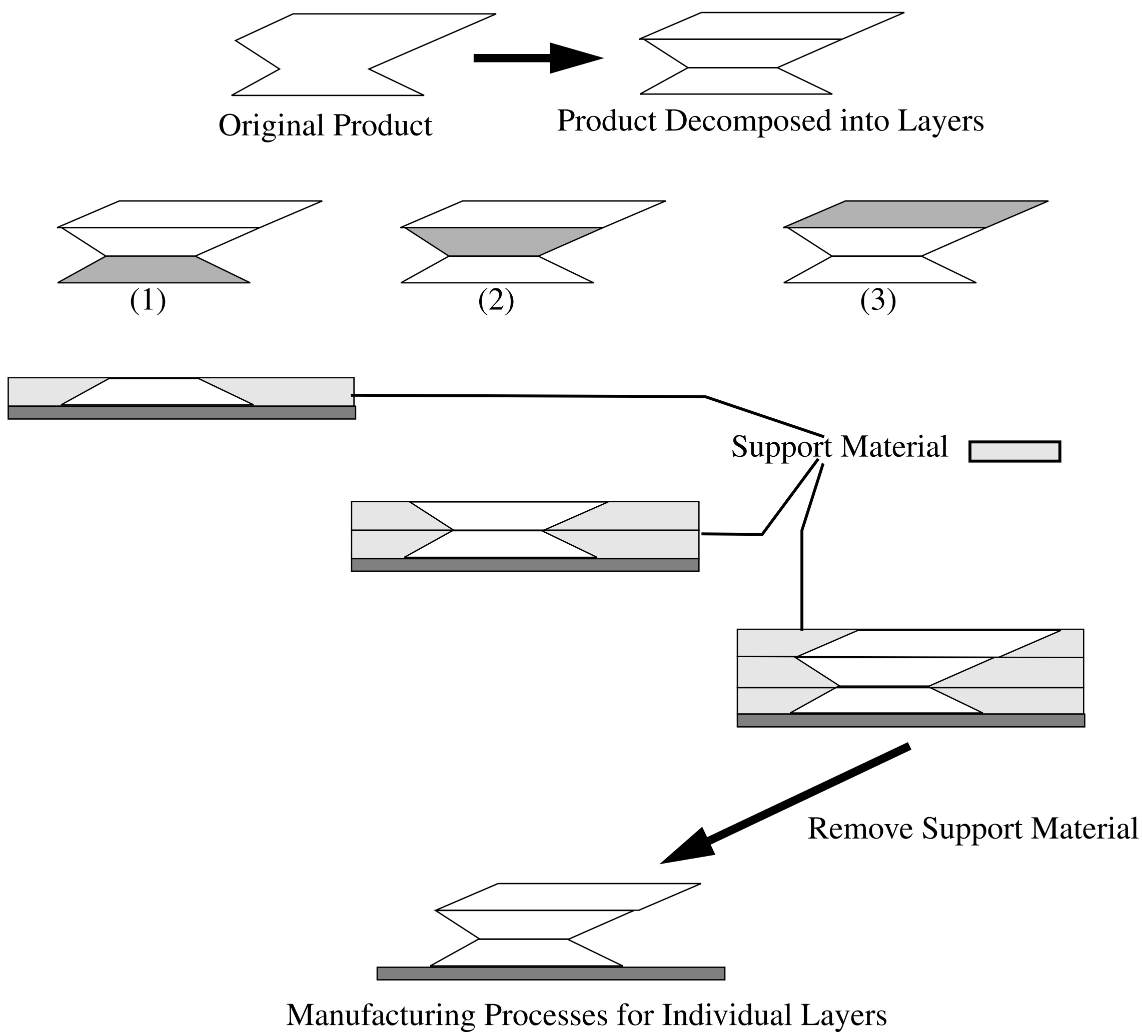


Figure 9.4: Shape decomposition in layered manufacturing (Weiss, 1995).

In Figure 9.4, a part fabricated by Weiss (1995) is shown. A fabrication approach called Shape Deposition Manufacturing (SDM) combines the material addition and removal processes. In SDM, the part is decomposed in such a way that the undercut features in the part need not be machined. For instance, the part in Figure 9.4 is decomposed into 3 layers first. Each layer is then built up by first depositing primary material into a near-net shape. A machining process is then followed to form the net shape of the layer.

SDM can use different deposit sources. For instance, stainless steel can be used as the primary material and copper as the sacrificial material.

One critical issue in laminated manufacturing is to decompose the part into appropriate layers that are easy to manufacture. Weiss also points out that small changes in the design might have significant effects on the decompositions. This characteristic is similar to the sheet metal bending process.

9.5.4 Extension of the Decomposition Approach

To apply our decomposition approach to the applications mentioned above, the geometric features which relate to manufacturability in various manufacturing processes have to be identified. For instance, the folding lines of a paper packaging design, the sharp corners of a sectional die, and the undercuts of a part in layered manufacturing. These features can serve as basic units of decomposition. The manufacturing processes that are of interest are selected also. For example, the cutting, folding, and gluing processes in paper packaging design, machining and assembly processes in sectional die design, and material depositing and removal (machining) processes for layered manufacturing. The next step is to determine how these features are related to the manufacturability of these manufacturing processes and how decomposition can improve the manufacturability. For instance, unfoldability in paper packaging design, and machinability in making a sectional die and layered manufacturing. In modeling the manufacturing processes, some manufacturing constraints are used to check manufacturing feasibility during the decomposition. The complete manufacturability has to be ensured using thorough process planning after the decomposition. To select a desired decomposition, some manufacturing preferences can be considered. For instance, reduction of the number of parts and the reduction of manufacturing costs, such as material costs, are common preferences for decomposition. It is possible that after the product is decomposed, additional design modifications are needed to account for assembly operations. For example, additional tabs are necessary for gluing purposes in paper packaging design.

In this section, some general guidelines on how to apply our decomposition approach to the potential applications are presented. Although the detailed implementation may be different, the principles of decomposition remain the same. We hope this decomposition approach can be applied to these applications in the future.

Bibliography

[1] Aho, A.V., Hopcroft, J.E. and Ullman, J.D., 1974, *The Design and Analysis of Computer Algorithms,* pp. 113-118, Addison Wesley.

[2] Albano, A. and Sapuppo, G., 1980, "Optimal Allocation of Two-Dimensional Irregular Shapes Using Heuristic Search Methods," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 5, pp. 242-248.

[3] Alting, L. and Zhang, H., 1989, "Computer Aided Process Planning: the state-of-the art survey," *International Journal of Production Research*, Vol. 27, No.4, pp. 553-585.

[4] Amada Sheet Metal Working Research Association, 1981, *Bending Knowledge*, Machining Co., Ltd.

[5] Andreasen, M.M., Kahler, S. and Lund, T., 1983, *Design for Assembly*, IFS and Springer-Verlag.

[6] Ayyadevara, V.R. and Sturges, R.H., 1997, "Automated Planning for Stacking of Bent Sheet Metal Parts," *Proceedings* of *ASME'97-DETC/DFM Conference*, Sacramento, CA.

[7] Baldwin, D.F., Abell, T.E., Lui, M.-C.M., De Fazio, T.L. and Whitney, D.E., 1991, "An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Products," *IEEE Transactions on Robotics and Automations*, Vol. 7, No. 1, pp. 78-94.

[8] Ballard, D.H. and Brown, C.M., 1982, *Computer Vision*, Prentice-Hall.

[9] Bedworth, D., Henderson, M.R., and Wolfe, P.M., 1991, *Computer-Integrated Design and Manufacturing*, McGraw-Hill.

[10] Benson, S.D., 1997, *Press Brake Technology: A Guide to Precision Sheet Metal Bending*, Society of Manufacturing Engineers.

[11] Boothroyd, G., Dewhurst, P. and Knight, W., 1994, *Product Design for Manufacture and Assembly*, Marce Dekker.

[12] Bourjault, Alain, 1984, Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires. Ph.D. thesis, Faculté des Sciences et des Techniques de l'Université de Franche-Comté.

[13] Bourne, D.A. and Wang, C.-H., 1995, "Design and Manufacturing of Sheet Metal Parts: Using Features to Resolve the Manufacturability Problems," *ASME International Computers in Engineering Conference*, Boston, MA.

[14] Bourne, D.A., 1992, "Intelligent Manufacturing Workstations," *Knowledge-Based Automation of Processes, ASME Winter Annual Meeting*, Anaheim, CA.

[15] Chuang, S.H. and Huang, S.F., 1996, "Feature Decomposition from Solid Models for Automatic Flattening," *Computer-Aided Design*, Vol. 28, No. 6/7, pp. 473-481.

[16] Cser, L., Geiger, M., Greska, W. and Hoffmann, M., 1991, "Three kinds of case-based learning in sheet metal manufacturing," *Computers in Industry*, Vol. 17, pp. 195-206.

[17] Cutkosky, M.R. and Tenenbaum, J.M., 1992, "Toward a Framework for Concurrent Design," *International Journal of Systems Automation: Research and Applications,* Vol. 1, No. 3, pp. 239-261.

[18] Darr, T.P. and Birmingham, W.P., 1993, "Automated Design for Concurrent Engineering," Technical Report CSE-TR-174-93, Dept. of Electrical Engineering and Computer Science, University of Michigan.

[19] De Fazio, T.L and Whitney, D.E., December, 1987, "Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, RA-3(6):640-658.

[20] De Vin, L.J., De Vries, J., Streppel, A.H. and Kals, H.J.J., 1992, "PART-S, a CAPP System for Small Batch Manufacturing of Sheet Metal Components," *Proceedings of the 24th CIRP International Seminar on Manufacturing Systems*, Copenhagen, pp. 171-182.

[21] Deo, N., 1974, *Graph Theory with Their Applications to Engineering and Computer Science*, Prentice-Hall.

[22] Diaz, A., Chandra, D.N. and P.K. Khosla, 1995, "Measuring the Difficulty of Assembly Tasks from Toll Access Information," *Proceedings of IEEE International Symposium on Assembly and Task Planning*, pp. 87-93.

[23] Dieter, G.E., 1991, *Engineering Design: A Materials and Processing Approach*, McGraw-Hill.

[24] Dixon, J.R. and Poli, C., 1995, *Engineering Design and Design for Manufacturing*, Field Stone.

[25] Dowsland, K.A. and Dowsland, W.B., 1995, "Solution approaches to irregular nesting problems," *European Journal of Operational Research 84*, pp. 506-521.

[26] Elber, G., 1995, "Model Fabrication Using Surface Layout Projection," *Computer-Aided Design*, Vol. 27, No. 4, pp. 283-291.

[27] Erdmann, A.M., 1997, Personal Discussion.

[28] Even, S., 1979, *Graph Algorithms*, Computer Science Press.

[29] Faux, I.D. and Pratt, M.J., 1981, *Computational Geometry for Design and Manufacture,* Ellis Horwood.

[30] Finger, S., Fox, M., Navinchandra, D., Prinz, F.B. and Rinderle, J.R., 1990, "Design Fusion: A Product Life-Cycle View for Engineering Designs," Technical Report EDRC 24-28-90, Engineering Design Research Center, Carnegie Mellon University.

[31] Flemming, U., Adams, J., Carlson, C., Coyne, R., Fenves, S., Finger, S., Ganeshan, R., Garrett, J., Gupta, A., Reich, Y., Siewiorek, D., Sturges, R., Thomas, D. and Woodbury, R., 1992, "Computational Models for Form-Function Synthesis in Engineering Design," Technical Report EDRC-48-25-92, Engineering Design Research Center, Carnegie Mellon University.

[32] Gan, J.G., Woo, T.C. and Tang, T.,1994, "Spherical Maps: Their Construction, Properties and Approximations," *Journal of Mechanical Design*, Vol. 116, pp. 357-363.

[33] Gupta, S.K. and Nau, D.S., 1995, "Systematic Approach to Analyze the Manufacturability of Machined Parts," *Computer-Aided Design*, Vol. 27, No. 5, pp. 323-342.

[34] Gupta, S.K., Regli, W.C. and Nau, D.S., 1994, "Integrating DFM with CAD through Design Critiquing," *Concurrent Engineering: Research and Applications*, Vol. 2, pp. 85-95.

[35] Gupta, S.K., Regli, W.C., Das, D. and Nau, D.S., 1997, "Automated Manufacturability Analysis: A Survey," to appear in *Research in Engineering Design*.

[36] Gursoz, E.L., 1991, *Reference Manual for the Noodles Library*, Engineering Design Research Center, Carnegie Mellon University.

[37] Hancock, T.M., 1989, "A Systems Approach to Sheet Metal Processing," *Computers in Industry*, Vol. 13, pp. 245-251.

[38] Hayes, C., 1987, "Using Goal Interactions to Guide Planning," *Proceedings of AAAI-87; the Sixth National Conference on Artificial Intelligence*, pp. 224-228.

[39] Hindes, B.K., McCartney, J. and Woods, G., 1991, "Pattern Development for 3D Surfaces," *Computer-Aided Design*, vol. 23, No. 8, pp. 583-592.

[40] Hoffmann, C.M., 1989, *Geometric & Solid Modeling*, Morgan Kaufmann.

[41] Homem De Mello, L.S. and Sanderson, A.C., 1990, "And/Or Graph Representation of Assembly Plans," *IEEE Transactions on Robotics and Automations*, Vol. 6, No. 2, pp. 188-199.

[42] Homem De Mello, L.S. and Sanderson, A.C., 1991, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences," *IEEE Transactions on Robotics and Automations*, Vol. 7, No. 2, pp. 228-240.

[43] Inui, M. and Kimura, F., 1991, "Design of Machining Processes with Dynamics Manipulation of Product Models," *Artificial Intelligence in Design*, ed. D. T. Pham, Springer-Verlag, pp. 195-227.

[44] Kim, K. 1996, "Automated Grasp Planning for Sheet Metal Bending," *International Sessions on Dynamics, Systems, and Design, KSME Fall Annual Meeting*, Seoul, Korea.

[45] Kimple, J.F. and Richards, L.G., 1991, "A Feature-Based Group Technology Coding System for Sheet Metal Parts," *Journal of Engineering Design 2(3)*, pp. 263-279.

[46] Kruskal, J.B. Jr., 1956, “On the Shortest Spanning Subtree of Graph and the Traveling Salesman Problem,” *Proceedings of the American Mathematical Society,* Vol. 7, pp. 48-50.

[47] Kusiak, A., 1990, *Intelligent Manufacturing Systems*, pp. 159-167, Prentice-Hall

[48] Kusiak, A., Szczerbicki, E. and Park, Kwangho, 1991, “A Novel Approach to Decomposition of Design Specifications and Search for Solutions,” *International Journal of Production Research*, Vol. 29, No. 7, pp. 1391-1406.

[49] Lee, S., Kim, G.J. and Bekey, G.A., 1993, “Combining Assembly Planning with Redesign: An Approach for More Effective DFA,” *Proceeding of IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 319-325

[50] Li, Z. and Milenkovic, V., 1995, “Compaction and Separation Algorithms for Non-Convex Polygons and Their Applications,” *European Journal of Operational Research* 84, pp. 539-561.

[51] Manning, J.R., 1980, “Computerized Pattern Cutting,” *Computer-Aided Design*, vol. 12, No. 1, pp. 43- 47.

[52] Mantyla, M., 1988, *An Introduction to Solid Modeling*, Computer Science Press.

[53] Mattikalli, R., 1994, Ph.D. Thesis, Mechanical Engineering Department, Carnegie Mellon University.

[54] Morton, T.E. and Pentico, D.W., 1993, *Heuristic Scheduling System*, John Wiley and Sons.

[55] Natarajan, B.K., 1988, “On Assembly Assemblies,” *ACM Symposium on Computational Geometry*, pp. 299-308, ACM.

[56] Nau, D.S. and Karinthi, R.R., 1989, “An Algebraic Approach to Feature Interactions,” Technical Report ERC-UMC 89-101, University of Maryland.

[57] Nevins, J.L., Whitney, D.E. and De Fazio, T.L., 1989, *Concurrent Design of Products & Processes*, McGraw Hill.

[58] Ousterhout, J.K., 1994, *Tcl and the Tk Toolkit*, Addison Wesley.

[59] Pahl, G. and Beitz, W., 1996, *Engineering Design,* Spring-Verlag.

[60] Powell, J., 1993, *CO_2 Laser Cutting*, Springer-Verlag.

[61] Prim, R.C., Nov. 1957, “Shortest Connection Networks and Some Generations,” *Bell System Tech. Journal*, Vol. 36, pp. 1389-1401.

[62] Radhakrishnan, R., Amsalu, A., Kamran, M. and Nnaji, B.O., 1996, "Design Rule Checker for Sheet Metal Components Using Medial Axis Transformation and Geometric Reasoning," *Journal of Manufacturing Systems*, Vol. 15, No. 3, pp. 179-189.

[63] Radin, B., Shpitalni, M. and Hartman, I., 1996, "Two-Stage Algorithm for Rapid Determination of the Bending Sequence in Sheet Metal Products," *ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Irvine, CA

[64] Ramsden, N.C., Shanks, I. and Rathmill, K., 1984, "The Development of Techniques for the Robot Automated Assembly of Sheet Metal," *5th International Conference on Assembly Automation*, Paris.

[65] Reich, R., Ochs, I.B. and Ozsoy, T.M., 1991, "Automated Flat Pattern Layout from Three-Dimensional Wire-Frame Data," *Journal of Engineering Design*, Vol. 2, No. 3, pp. 247- 261.

[66] Rich, E. and Knight, K., 1991, *Artificial Intelligence*, McGraw-Hill.

[67] Safier, S. and Grinberg, V., 1996, Personal Discussion.

[68] Shibata, Y. and Kunitomo, Y., Dec. 1981, "Sheet Metal CAD/CAM System," *Bulletin of the JAPAN Society of Precision Engineering*, Vol. 15, No. 4.

[69] Stanley, F.A., 1919, *Punches and Dies: Layout, Construction and Use*, pp. 54-56, McGraw Hill.

[70] Toye, G., Cutkosky, M.R., Leifer, L.J., Tenenbaum, J.M. and Glicksman, J., 1994, "SHARE: A Methodology and Environment for Collaborative Product Development," *International Journal of Intelligent & Cooperative Information Systems,* Vol.3, No. 2, pp. 129-153.

[71] Tseng, Y.-J. and Joshi, S.B., 1994, "Recognizing Multiple Interpretations of Interacting Machining Features," *Computer-Aided Design*, Vol. 26, No. 9, pp. 667-688.

[72] Uzsoy, R., Ramcharn, D.J. and Martin-Vega, L.A., 1991, "An Experimental Expert System for Process Planning of Sheet Metal Parts," *Computers ind. Eng.* Vol. 20, No.1 pp. 59-69.

[73] Wang, C.-H. and Bourne, D.A., 1995, "Using Features and their Constraints to Aid Process Planning of Sheet Metal Parts," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 1020-1026, Nagoya, Japan

[74] Wang, C.-H. and Bourne, D.A., 1996, "Compact 2D Pattern Development from 3D Polygonal Surface," *Proceeding of ASME'96-DETC/DFM Conference*, Irvine, CA.

[75] Wang, C.-H. and Sturges, R.H., 1996, "BendCad: a design system for concurrent multiple representations of parts," *Journal of Intelligent Manufacturing*, Vol. 7, pp. 133-144.

[76] Weiss, L. June, 1995, "Solid Freeform Fabrication Process," *NSF Workshop on Design Methodologies for Solid Freeform Fabrication*, Engineering Design Research Center, Carnegie Mellon University.

[77] Wilson, R.H. and Schweikard, A., 1991, "Assembly Polyhedra with Single Translation," Technical Report STAN-CS-91-1387, Computer Science Department, Stanford University.

[78] Wilson, R.H., 1992, *On Geometric Assembly Planning*, Ph.D. Thesis, Computer Science Department, Stanford University,

[79] Wolter, J, 1988, *On the automatic generation of plans for mechanical assembly*, Ph.D. Thesis, University of Michigan.

[80] Yeh, S., Kamran, M. and Nnaji, B.O., 1995, "Unfolding Sheet Metal Parts: A Graph-Based Approach," *International Journal of Production Research*. Vol. 33, No. 3, pp. 729-740.

[81] Zienkiewicz, O.C. and Morgan, K., 1983, *Finite Elements And Approximation*, John Wiley and Sons.