Methodology for automatic operation sequencing for progressive die design

Chu, Changyong

2007

Chu, C. Y. (2007). Methodology for automatic operation sequencing for progressive die design. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/36136>

<https://doi.org/10.32657/10356/36136>

9477841 36
2X
D/B/10

**A METHODOLOGY FOR AUTOMATIC OPERATION
SEQUENCING FOR PROGRESSIVE DIE DESIGN**

Chu Changyong

School of Mechanical and Aerospace Engineering

A thesis submitted to Nanyang Technological University in
fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

ACKNOWLEDGEMENTS

Firstly I am deeply grateful to my research supervisor A/Prof. Tor Shu Beng and co-supervisor Prof. Graeme A. Britton for their valuable guidance and encouragement of my thesis work. Without them this work might have not been completed.

I want to express my gratitude to School of Mechanical and Aerospace Engineering of the Nanyang Technological University, Singapore for its financial support and the necessary research facilities. I also appreciate all the staffs of the CAD/CAM Lab, for their kind help during my stay there; in particular, Dr Zhang Wenyu, who is also my senior, for his encouragement and many technical discussions during my studying in Singapore.

Finally I am deeply indebted to my parents, who have always been trying their best to provide me a good learning environment since my childhood.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	I
TABLE OF CONTENTS	II
LIST OF FIGURES	VII
LIST OF TABLES	XI
LIST OF PUBLICATIONS	XII
ABSTRACT	XIII
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Some basic concepts	3
1.2.1 Sheet metal	3
1.2.2 Die operations	3
1.2.3 Progressive die design process	6
1.3 Research aim	9

ii

1.4 Organization of thesis	10
 CHAPTER 2 LITERATURE REVIEW	 11
2.1 Introduction	11
2.2 Strip layout design	13
2.3 Graph theory	27
2.3.1 Data structures for graphs	30
2.3.2 Cycle detection algorithms	31
2.3.3 Graph colouring algorithms	32
2.3.4 Graph based operations sequencing methods for machining	34
2.4 Summary	35
 CHAPTER 3 FEATURE-BASED SHEET METAL PART AND STRIP MODELLING	 37
3.1 Introduction	37
3.2 Feature-based modeling of sheet metal parts	37

3.3	Generating of stamping features	41
3.4	Mapping stamping features to stamping operations	42
3.5	Representation of Stamping Operation Relations	43
3.5.1	Representation of operation precedence relations	45
3.5.2	Representation of operation cluster relations	47
3.6	Summary	48
CHAPTER 4	THE PROPOSED METHODOLOGY	48
4.1	Introduction	48
4.2	Problem statement	48
4.3	Proposed graph based operation clustering and sequencing approach	50
4.4	A case study	54
4.5	Proof of correctness and algorithm complexity	65
4.6	Summary	67
CHAPTER 5	SYSTEM IMPLEMENTATION	68

5.1 Introduction	68
5.2 System architecture	69
5.3 Objected-oriented classification of sheet metal features	73
5.4 Knowledge representation using CLIPS	77
5.5 User interface design	82
5.6 Summary	87
CHAPTER 6 CASE STUDIES	88
6.1 Introduction	88
6.2 Case 1: A sheet metal bracket	88
6.3 Case 2: A sheet metal electric iron part	96
6.4 Comparison between this research work and relevant previous research works	104
CHAPTER 7 CONCLUSIONS AND FUTURE WORK	106
7.1 Conclusions	106

7.2 Future work	108
REFERENCES	111

LIST OF FIGURES

Figure 1. 1 Progressive die design in sheet metal product life cycle (Wang <i>et al.</i> 1995)	7
Figure 1. 2 Three key processes of stamping process planning	8
Figure 2.1 A new graph theoretic approach for stamping operations sequencing	13
Figure 2.2 Two-stage hemming process (Duflou <i>et al.</i> 1999)	16
Figure 2.3 Knowledge base structure for consultation system (Ngoi and Chua 1994)	19
Figure 2.4 An example of DAG	29
Figure 2.5 A topological sort of example DAG	29
Figure 2.6 An acyclic graph on which the tree cycle detection algorithm would fail	31
Figure 2.7 An exact graph coloring algorithm (DSATUR) (http://mat.gsia.cmu.edu/COLOR/color.html)	33
Figure 3. 1 A sample part represented using standard sheet metal features	39
Figure 3. 2 Symbolic feature-tree representation of the sample part	40

Figure 3. 3 Stamping features and its stamping feature tree of the example part	42
Figure 4. 1 Relationship between stamping features, die operations and workstations.	49
Figure 4. 2 Proposed graph based operation clustering and sequencing for progressive die design	53
Figure 4. 3 Precedence graph of the example part (automatically generated using WinGraphviz API)	55
Figure 4. 4 Operation adjacency graph for the example part (automatically generated using WinGraphviz API)	56
Figure 4. 5 S-OAGs of the example part	61
Figure 4. 6 3-D strip layout for the example part.	65
Figure 5. 1 System architecture of AutoSOS	70
Figure 5. 2 UML class diagram of AutoSOS	72
Figure 5. 3 Object-oriented classification of sheet metal features	74
Figure 5. 4 User interface of AutoSOS	83

Figure 5. 5 Create a blank strip window	84
Figure 5. 6 Feature operation mapping window	85
Figure 5. 7 Operation adjacency matrix window	86
Figure 5. 8 Operation precedence matrix window	86
Figure 6. 1 A sheet metal bracket	88
Figure 6. 2 Parameters setting dialog in nesting process	90
Figure 6. 3 Top view of scrap design for the sheet metal bracket	91
Figure 6. 4 Feature operation mapping process	92
Figure 6. 5 Operations precedence graph for the sheet metal bracket	93
Figure 6. 6 Operations adjacency graph for the sheet metal bracket	94
Figure 6. 7 Macro planning result for case 1	94
Figure 6. 8 Micro planning result for case 1	95
Figure 6. 9 Final strip layout of the sheet metal bracket	95

Figure 6. 10 A sheet metal electric iron part	96
Figure 6. 11 Scrap design for the sheet metal electric iron component	97
Figure 6. 12 Feature-operation mapping process for the sheet metal electric iron	98
Figure 6. 13 Operations adjacency graph for the sheet metal electric iron	99
Figure 6. 14 Operations adjacency graph for the sheet metal electric iron	99
Figure 6. 15 Operations precedence graph for the sheet metal electric iron	100
Figure 6. 16 Macro planning result for case 2	101
Figure 6. 17 Micro planning result for case 2	102
Figure 6. 18 Final 3D strip layout for the sheet metal electric iron part	103
Table 6. 1 Comparison between this research work and relevant previous research works	104

LIST OF TABLES

Table 1. 1 Various die operations	6
Table 2. 1 References in unfolding, nesting, strip layout and computer-aided progressive die design systems	25
Table 2. 2 Space complexity of AMR and ALR	30
Table 3. 1 Partial operation precedence matrix	46
Table 3. 2 A partial operation adjacency matrix	48
Table 4. 1 Operation precedence matrix of the example part	57
Table 4. 2 Operation adjacency matrix for the example part	58
Table 6. 1 Comparison between this research work and relevant previous research works	104

LIST OF PUBLICATIONS

1. Chu C.Y.; Tor S.B. & Britton, G.A. (2003). A Graph Theoretic Approach for Stamping Operations Sequencing. Journal of Engineering Manufacture, Part B, 4(1), pp. 467-471.
2. Chu, C.Y.; Tor, S.B. & Britton, G.A. (2004). A Graph Theoretic Operations Sequencing Approach for Progressive Die Design. Proceedings of the 34th International MATADOR Conference, 9-11 July, UMIST, Manchester, UK, pp. 69-74.
3. Chu, C.Y.; Tor, S.B. & Britton, G.A. (2006). An efficient algorithm for automatic operation sequencing for progressive die design. International Journal of Production Research (accepted for publication).

ABSTRACT

Sheet metal components produced by progressive dies play an important role in modern day life. Progressive die design in most industries remains an art rather than a science. The job is mainly carried out manually, based on individual designer's skills and knowledge, which are accumulated primarily through work experience. Strip layout is the most critical task in progressive die design. The optimal die operation sequence enables both the precise part production and the manufacturing cost reduction.

The aim of this research is to develop a novel, rigorous approach for strip layout design in progressive die design. The proposed approach incorporates rules and heuristics used by human process planning experts and synthesizes the reasoning process using graph theoretic algorithms. Two types of constraints between stamping operations are identified in this work: cluster and precedence constraints. Operations have cluster constraints should be performed simultaneously at the same station. While a precedence constraint implies that an operation or operation cluster must be performed before another operation or cluster. The proposed approach first generates the stamping features from the sheet metal features consisting of the CAD model of stamping part to-be-manufactured. In this process, the blank orientation is determined using nesting algorithm to obtain the optimal material utilization. The sheet metal features in the CAD part model are mapped to stamping features in a strip model. Auxiliary stamping features (pilot features and scrap features) are devised by the process planner. Then, stamping features are mapped to stamping operations, followed by generating the operation precedence and adjacency graphs. Next the operation precedence graph is verified to be acyclic using a colored DFS (depth-first search). Based on the operation

precedence graph, a modified topological sort algorithm is applied to cluster the operations into partially ordered sets. Finally, a graph colouring algorithm is applied to the operation adjacency graph on the partially ordered operation sets.

The approach has been implemented in a prototype system (AutoSOS - Automatic Stamping Operation Sequencing), which is written in C++ and is fully integrated with the SolidWorks CAD system. This system can be used in either of two ways: as an automated process planning system, *i.e.*, a tool by someone who is not an expert but requires the use of expert knowledge; or as a checking system, *i.e.*, an assistant to an expert in the field.

If it is used as an automated process planning system, the result is a sequenced cluster of operations that meet the constraints of both graphs. It can output 2 numbers, the minimum number of stations based on topological sort (the ideal minimum) and the feasible number after colouring (the feasible minimum). The 3D strip layout is automatically generated.

On the other hand, if it is used as a checking system, it can output 3 numbers: the actual number of stations set by planner, the feasible number and the ideal number. The numbers are useful to help process planner see the impact of constraints and improve the process plan according to them.

CHAPTER 1 INTRODUCTION

1.1 Introduction

Sheet metal components produced by progressive dies play an important role in modern day life. According to a survey in the US, some 100,000 metal stampings could be found in the average American home in the 1980s (Cheok and Nee1998). Progressive dies for producing sheet metal parts in mass production have been widely applied in various industries such as aerospace, electronics, machine tools, automobiles, refrigeration, etc., resulting in production automation at higher productivity, higher product quality and lower tooling cost. A metal stamping is produced from a progressive die by advancing a metal strip from one station to another one, where it undergoes one or a combination of operations. These operations can be piercing, notching, cut-off, blanking, lancing, shaving, drawing, embossing, coining, trimming, and other miscellaneous forming operations (Smith, 1990). A progressive die can range from a simple two-station die to a complicated multi-station die involving cams and sliding mechanisms. The degree of precision and complexity of die components may vary depending upon the specifications of the sheet metal components.

Progressive die design in most industries remains an art rather than a science. The job is mainly carried out manually, based on individual designer's skills and knowledge, which are accumulated primarily through work experience (Wang and Chang, 1995). Even if commercial solid modeling software such as Pro/Engineer, SolidWorks and Unigraphics are used to aid the design process in some stamping companies, the design process always remains time and cost exhaustive (Li *et al.* 2001). There are strong

demands for the special CAD systems to speed up the process of progressive die design.

The process of progressive die design can be defined as follows: firstly, unfolding the 3D metal stamping into a flat pattern; secondly, the flat patterns are nested to achieve optimum utilization of stock material, which is called nesting or blank layout; thirdly, planning for the sequence of die operations that need to be performed to manufacture a part, which is called strip layout. On the basis of the strip layout, the die is constructed by assembling together die components such as punches, die block, punch plates, strippers, pilots, strippers, stock guides, guide bushings, guide posts and die sets located by dowel pins and held together by fastening screws. This process is called die structure design.

Strip layout is the most critical task in progressive die design. The optimal die operation sequence enables both precise part production and manufacturing cost reduction. Many CAD systems provide powerful functions for sheet metal product modeling, unfolding, and nesting. On the other hand, strip layout design is still manually performed by manufacturing engineers. Only a few studies have been reported for automating the strip layout process.

In this chapter, some basic progressive die concepts are introduced such as sheet metal, die operations, and the whole progressive die process. The chapter concludes with the research goals and organization of this thesis.

1.2 Some basic concepts

1.2.1 Sheet metal

A metal workpiece that has uniform thickness much less than its length and width and not exceeding 6.35mm is called sheet metal. The sheet can be of either ferrous or non-ferrous metals. Currently low carbon steel (0.05 to 0.20% C) is the most widely used material though there is an increasing trend of using high strength low alloy steels. About 56% of all steel produced is in sheet form (Early 1974). A wide variety of non-ferrous metals including aluminum, copper, magnesium, zinc, titanium, nickel and their alloys are widely used in sheet form.

1.2.2 Die operations

The following operations are commonly performed in progressive dies.

Piercing: Piercing generally refers to cutting openings such as holes and slots in sheet stock, strip material, or a part. Where possible, all holes and openings in a part are pierced in one stroke. This results in a much more consistent part than punching features in separate strokes of the press. Since all punches are permanently mounted in the same master die and enter the workpiece at the same time, once they are checked and located correctly, dimensional repeatability of all features is high.

Blanking: Blanking is the cutting of flat metal sheet or strip stock into the required size and shape. The piece of material removed from the sheet or strip is the part that is wanted and is referred to as the blank. Usually, blanking is the last operation performed in a progressive die in that it cuts the sheet metal component from the strip.

Bending: This is one of the most common processes in sheet forming. The part

may be bent not only along a straight line, but also along a curved path (stretching, flanging). The minimum bend radius, measured to the inside surface of the bend, is important and determines the limit at which the material cracks either on the outer surface of the bend or at the edges of the part. This radius, which is usually expressed in terms of multiples of the sheet thickness, depends on the ductility of the material, width of the part, and its edge conditions. Springback in bending and other sheet-forming operations is due to the elastic recovery of the metal after it is deformed. Determination of springback is usually done in actual tests. Compensation for springback in practice is generally accomplished by overbending the part; adjustable tools are sometimes used for this purpose.

Drawing: Drawing is broadly defined as stretching sheet metal stock, commonly a blank that is restrained around its periphery, into a cuplike, cylindrical or box shape. In practice, there's virtually no limit to the shapes that can be drawn. Drawing usually takes place in either a mechanical or a hydraulic press, where a tool (punch) forces the metal to flow into a cavity (die) creating the basic shape.

Forming: Forming, in practically all instances, can be considered as a bending operation. Unlike drawing, which can significantly stretch and deform the metal, forming generates shapes with virtually no change in thickness. Part configurations include those with simple flanges, V-bends, U-shaped channels, and more complex contours. Typically, basic tools like V-punches and wipe dies are built into standard holders. As with blanking and piercing operations, repeatability and high precision are readily achieved.

<i>Operation</i>	<i>Description</i>	<i>Diagram</i>

Piercing	Piercing is the operation of cutting internal features (holes or slots) in stock.	
Blanking	Blanking is similar to piercing except that the slug produced is the part and the strip or blank is the scrap piece.	A diagram showing a circular punch cutting a circular slug from a rectangular workpiece. The slug is labeled 'workpiece' and the remaining material is labeled 'scrap'.
Bending	Bending is a process by which metal can be deformed by plastically deforming the material and changing its shape.	A diagram showing a workpiece being bent into a 90-degree angle. The workpiece is labeled 'workpiece'.
Forming	Forming is a process that is used to both bend and shape the part into a specific shape.	A diagram showing a punch pressing a workpiece into a die. The punch is labeled 'punch', the workpiece is labeled 'workpiece', and the die is labeled 'die'.
Drawing	In drawing, a blank of sheet metal is restrained at the edges, and the middle section is forced by a punch into a die to stretch the metal into a cup shaped drawn part.	A diagram showing a punch pressing a workpiece into a die. The punch is labeled 'punch', the workpiece is labeled 'workpiece', the die is labeled 'die', and a pressure pad is labeled 'pressure pad'.

Table 1. 1 lists the above operations that can be performed by progressive die including descriptions and diagrams of these operations.

Operation	Description	Diagram
Piercing	Piercing is the operation of cutting internal features (holes or slots) in stock.
Blanking	Blanking is similar to piercing except that the slug produced is the part and the strip or blank is the scrap piece.
Bending	Bending is a process by which metal can be deformed by plastically deforming the material and changing its shape.
Forming	Forming is a process that is used to both bend and shape the part into a specific shape.
Drawing	In drawing, a blank of sheet metal is restrained at the edges, and the middle section is forced by a punch into a die to stretch the metal into a cup shaped drawn part.

Table 1. 1 Various die operations

1.2.3 Progressive die design process

The design cycle of progressive dies consists of three primary stages: product (sheet metal

component) modeling, process planning, and die structure design. A typical CAD/CAM system for progressive dies (Wang *et al.* 1995), ignoring the management and business aspect, is shown in Figure 1.1. 17 functions are identified in Figure 1.1, but those enclosed by dashed lines are relevant from the point of view of CAD/CAM system development.

Figure 1. 1 Progressive die design in sheet metal product life cycle (Wang *et al.* 1995)

There are two polygons represented by dashed lines in the figure. The first one encloses those functions for a CAD system for product design, *i.e.* product design, functionality analysis and design optimization. The CAD system may also include the functions for the selection of processes, types of dies, and equipment, and that for manufacturability analysis. According to concurrent engineering, the product life cycle may be shortened if these functions are performed at the design stage. The second

encloses all the functions related to the design and manufacturing of a progressive die. A dotted line in the figure separates the second polygon into two regions. Functions in the first region are related to process planning in progressive die design, while those in the second take care of the design of die structure. Stamping process planning includes the development of the blank, nesting, and strip layout (Figure 1.2).

Figure 1. 2 Three key processes of stamping process planning

Strip layout is actually a general name of several processes, which includes scrap

design, design of the pilot system, feature operation mapping, determination of the overall operation sequence, and the assignment of operations to die stations. The result is represented by a strip layout.

1.3 Research aim

Recent advances in the field of feature modeling and Artificial Intelligence (AI) have given rise to the possibility to construct feature- and AI-based systems that incorporate built-in intelligence and apply it to solve progressive die design problems including strip layout design automation. However, most of them lack practicability and scalability due to loss of design and manufacturing information and computational inefficiency in representing the stamping metal parts and stamping processes associated with stamping process planning. No scientific and mathematic approach exists for operation sequencing in progressive die design.

As the productivity, accuracy, cost, and quality of a progressive die mainly depend on the strip layout, strip layout design becomes a backbone in progressive die design. However, it has been well recognized that strip layout design, even after many years of practice, still remains more of an art rather than a science. Historically, the work is mainly carried out manually, based on designers' experience, skill and knowledge. To improve productivity and to build a computer integrated manufacturing environment, the automation of strip layout is essential. The aim of this research is, given a sheet metal component, to develop a novel, scientific strip layout approach that is capable of generating a stamping sequence automatically. The proposed system is intended to decrease process planning time, resulting in increased productivity.

1.4 Organization of thesis

This thesis is organized into seven chapters. Chapter 1 (the current chapter) presents background information about progressive die design. The research aim is defined. The literature relevant to this research is reviewed in chapter 2. In chapter 3, feature-based sheet metal part design and creation of stamping features are discussed. In addition, mapping stamping features to stamping operations and representation of stamping operation relation are also illustrated. In chapter 4, the problem statement is defined first, followed by a detailed discussion of the proposed automated strip layout approach. At the end, a proof of the approach and discussion of the computational complexity are also provided. In chapter 5, the work for software implementation of the proposed approach is expounded. Chapter 6 shows several industrial cases to illustrate the capabilities of the approach. Chapter 7 wraps up the work with conclusion and potential areas for future work.

CHAPTER 2 LITERATURE REVIEW

2.1 Introduction

Strip layout design involves laying out the material strip ribbon of metal or other material that is passed through the press in order to produce stamping exactly as it will appear after all operations have been performed on its parts (Ngoi and Chua 1994). Strip layout design determines the number of workstations, the features punched or formed at each workstation, and the sequence of workstations. Once strip layout design has been finished, the productivity, accuracy, cost, and quality of a progressive die are dictated to a large extent. From this point of view, strip layout is the backbone in progressive die design. However, it has been well recognized that strip layout design, even after many years of practice, still remains more of an art rather than a science. Historically, the work is mainly carried out manually, based on designers' experience, skill and knowledge.

Recent advances in the field of feature modeling and Artificial Intelligence (AI) have given rise to the possibility to construct feature- and AI-based systems that incorporate built-in intelligence and apply it to solving progressive die design problems including strip layout design automation. However, most of these lack practicability and scalability due to loss of design and manufacturing information and computational inefficiency in representing the stamping metal parts and stamping processes associated with stamping process planning. Currently, there are no commercial CAD or intelligent CAD systems that support strip layout design automation successfully. Strip layout design is still accomplished by process planners manually or interactively.

The aim of this research is to develop a novel and analytical approach for strip

layout design in progressive die design. The approach is based on graph theory. Graph theory as a branch of mathematics has wide practical applications in engineering. Compared with other mathematical modeling methods, graph theory has the following advantages:

- i) Nice mathematical results can be obtained at a low level of abstraction.
- ii) Graphs can be easily visualized, which enhance model understanding.
- iii) Modelling a practical problem as a graph theoretical problem is often simpler and less dependent on abstract thinking than with other kinds of mathematical modelling.
- iv) Graphs have special properties that allow the solution of much larger problems than can be handled by any other optimization technique and are much more computationally efficient than other types of solution procedures (Evans and Minieka, 1992).
- v) There are a lot of graph based algorithms dedicated to address different kinds of problems, which can be easily borrowed and tailored to specific problems.

Bearing these advantages in mind, a graph theoretic approach for stamping operations sequencing is proposed. The focus of this research and the graph based algorithms that are used are shown in Figure 2.1.

Figure 2.1 A new graph theoretic approach for stamping operations sequencing

The main purpose of this chapter is to review research literature in stamping process planning, computer-aided die design, and relevant graph based algorithms. The literature review discusses three key processes of stamping process planning - unfolding, blank layout and strip layout. This is followed by a survey on computer aided die design systems. Then, a general introduction to key graph theoretic terms needed in the proposed approach is given. In subsequent sections, three graph based algorithms are reviewed: cycle detection algorithms, graph colouring algorithms and graph based operations sequencing methods. Finally, some comments on the current state-of-the-art will be made in the summary.

2.2 Strip layout design

As mentioned in the previous section, the unfolding process is the first major step in process planning for generating a stamping operation sequence from a sheet metal blank. In unfolding, bend allowances resulting from metal stretching at bends must be accommodated. In addition to bend allowances, flat patterning for a 3D design

involves unfolding walls and mapping features such as slots, cutouts, etc., onto unfolded walls. Manual determination of flat pattern is time-consuming and error-prone, especially for intricate parts. To shorten unfolding design cycle and eliminating errors, computer-aided flat pattern development systems have been developed either as prototypes or for commercial use (Yeh *et al.* 1994, See Toh *et al.* 1995, Shunmugam *et al.* 2002, Prasa *et al.* 2004).

The design of an efficient blank layout is important as the optimum layout can reduce wastage of the strip material and reduce the overall cost of production. Great efforts from researchers worldwide have been taken to develop special CAD/CAM systems to automate blank layout process (Chow 1979, Adamowicz and Albano 1976, Dori and Ben-Bassat 1984, Qu and Sanders 1987, Dagli and Tatoglu 1987, Nee 1984, Nee 1989, Nee and Foong 1992, Ismail and Hon 1992, Lin and Hsu 1996, Tang and Rajesham 1994, Singh and Sekhon 1996, Cheok and Foong 1996, Choi *et al.* 1998, Zhao and Peng 2002).

Strip layout is the core sector in progressive die design, in that it determines the operation sequence performed on each die station. In this sense, the problem of strip layout design is in essence an operation sequencing problem. However, automated stamping operation sequencing for sheet metal progressive dies has not received as much attention as other aspects of process planning, as researchers have mainly focused on strip layout and die design using knowledge-based systems. The research that has been carried out mainly addresses the issue of sequencing of bending operations, although some efforts have been put on global stamping operation sequencing. The review on stamping operation sequencing will be in three sections: bending operation sequencing,

global stamping operation sequencing, and computer aided die design systems.

If a part has a few bends then manual planning of the bending sequence is straightforward. If, however, there are a large number of bends, the determination of the bending sequence will become quite complicated and be prone to error if the work is done manually. Computer aided process planning (CAPP) would then be very helpful. Existing successful applications of bending process planning systems primarily focus on press-brakes (Hoffmann *et al.* 1992, Lin and Peing 1994, Shpitalni and Saddan 1994, De Vin *et al.* 1994, Wang and Bourne 1995, Duflou 1997, Ong *et al.* 1997, Radin *et al.* 1997, Gupta *et al.* 1998, Inui and Terakado 1999, Duflou *et al.* 1999, Thanapandi *et al.* 2001, Thanapandi *et al.* 2002, Aomura and Koguchi 2002, Markus *et al.* 2002, Rico *et al.* 2003), most of which use heuristic search or global optimizing algorithms to find a practical or optimal operations sequence, such as A*, branch and bound and genetic algorithms.

It should be noted that Duflou *et al.* (1999) used two methods to generate feasible sequences. The first method uses precedence constraint solving. A precedence constraint graph is constructed taking into account hard and “soft” (heuristic) constraints between bends. Precedence constraints are identified by analysis of part details. Hard constraints are those constraints that cannot be violated. For example, a two-stage hemmed edge (Figure 2.2) is formed by a first bending operation with angle $\geq 90^\circ$, followed by a finishing step with a hammer tool that completes the bend over an angle of 180° . Soft constraints are from empirical rules based on observations of the way experienced process planners analyze workpieces. The precedence constraint graph is represented by an incidence matrix, precedence constraint matrix (PCM), in their system. The row and column numbers represent the bend ID numbers, while the matrix cells contain the information related to the priority level of the constraints (0 = no constraint,

1 = hard constraint, >1 = soft constraint).

Figure 2.2 Two-stage hemming process (Duflou *et al.* 1999)

To determine the sequence, an incidence vector C_i is calculated for each bend i , which is defined as the sum of all the cells in column i of PCM. The bend to be sequenced j is identified by $C_j = 0$. Then, the corresponding column and row will be removed from PCM, followed by a recalculation of the incidence vector C_i for the remaining bends. This process is repeated until no bends remain un-sequenced.

The authors state this method is efficient for reducing the search space, but is often not a sufficiently refined tool when conflicting heuristic rules are applied. Although the precedence graph provides an explicit and concise mathematical way to represent the bending operations sequencing problem, it does not take other constraints into account, e.g. minimum number of setups, which is an important shortcoming of the so called precedence constraint solving method.

Despite the fact that bending operations sequencing problem for press-brakes and progressive dies is quite similar, the constraints, die structures and bending types are very different. Usually, there are more constraints for the latter and it makes the problem more complicated. For example, in a progressive die, a child bend feature has to be formed before its parent bend feature, but there is no such constraint in press-brakes. It is because, in progressive dies, the feeding and bending directions are fixed; while in

press-brake the part can be aligned in any direction as long as it does not collide with the tool. The design of a progressive die is a complex and challenging task.

Wang *et al.* (1995) presented an algorithm for planning the bending sequence in progressive die design, with special attention paid to satisfying constraints imposed on the bending sequence by assembly features of sheet-metal parts. The constraint they mainly dealt with is that a wall with an insert on it should be bent before the wall with the corresponding slot. After examining the walls in the sheet metal part one by one, a list of bending sequence constraints can be established. In their algorithm, the constraints on a bending sequence are expressed by a linked list. For example (A B) means wall A should be bent before B. The procedure of their algorithm is as follows. At first, an empty bending sequence list is constructed. Then, the established constraints are examined one by one. If none of the two elements of the current constraint has been stored in the bending sequence list, they are both inserted at the head sequentially. If one of them is already inserted in the bending sequence list, the other is to be inserted either immediately before or after according to the constraint list. The above step is repeated until all the constraints have examined. The bending operations sequence will be the sequence of walls in the resulting bending sequence list. The algorithm can not deal with constraint conflicts in the bending sequence list. Furthermore, there is no check to ensure the list sequence is unique.

Li *et al.* (2002) described an integrated modeling and process planning system developed for planning bending operations of progressive dies. Case-based reasoning algorithms are explored in their system for automatic operations sequencing. However the representation, indexing and similarity metrics of the bending operations sequence are not discussed in detail. In addition, how to modify the operations sequence received

from the case base to cater for new cases is not mentioned. A major disadvantage is the case-based reasoning algorithm needs a huge library of past cases to support it. It is difficult and time-consuming to build such a case library.

Although Wang *et al.* (1995) and Li *et al.* (2002) have made great efforts in bending operations sequencing of progressive dies, the problem of strip layout design automation is still far from being solved. The stamping operations that can be performed by a progressive die are not only bending. Actually, a progressive die could perform all the fundamental sheet metal operations, including but not limited to the following: piercing, blanking, notching, embossing, lancing, and drawing. The diversity of the progressive die operations and the geometrical or technological constraints that may exist between them requires simultaneous planning of all types of operations. The author refers to this as global stamping operations sequencing.

Nee and Foong (1992) attempted to develop an automatic staging and punch shape design algorithm based largely on heuristic rules, shape decomposition and pattern-matching techniques. However, all the work focused on the general strip layout design process for blanking and the expert rules involved do not include other stamping process such as piercing, bending and trimming.

Ngoi and Chua (1994) presented an experimental knowledge-based consultation system developed for the purpose of capturing strip layout design knowledge. In their system, the knowledge base is categorized into two main modules, part orientation module and sequence of operations module. The sequence of operations module is further divided into four smaller sub-modules which encompass four types of operations of a progressive die — piercing, blanking, bending and trimming.

Figure 2.3 Knowledge base structure for consultation system (Ngoi and Chua 1994)

The design rules were contained in their knowledge base as IF-THEN production rules. When a specific part is loaded, a series of screen menu will appear to prompt the user to choose the type of orientation, the end product removal method and so on. It is more like a consultation system rather than an automation system. The operations sequence is determined by the user interactively.

Other knowledge based systems for stamping operations sequencing include Duffey and Sun (1991), Cheok *et al.* (1994), Ismail *et al.* (1995), Lin and Hsu (1996), Park *et al.* (1998), Choi *et al.* (1999), Choi *et al.* (2000), Choi and Kim (2001), Kim, C., Kim *et al.*

(2001), Kim *et al.* (2002), Kim and Park *et al.* (2004), Kim, Choi *et al.* (2004).

Lin and Deng (2001) describe a genetic algorithm that was applied as a solution tool to obtain the optimal strip working sequence for a shearing-cut and bending progressive die. Their work only considers the problem of torque equilibrium. The stamping operations sequence is obtained according to a particular optimization objective, e.g. minimum number of die stations. The constraints that are imposed by function specifications, technical requirements as well as spatial restrictions must be satisfied before optimization. Otherwise, the resulting sequence may be infeasible even though it is optimal from the view point of torque equilibrium.

In order to speed up strip layout design, and reduce the experience and skill required, researches on the computer-aided progressive die design systems have been widely reported since 1970s, providing productivity improvements, cost reductions and design automation.

From mid 1970s to mid 1980s, the first generation of CAD/CAM systems (Fogg *et al.* 1975, Nakaham *et al.* 1978, Murukami *et al.* 1980, Bergstrom *et al.* 1988) for progressive die design were developed, though few of them were based on feature modeling and AI techniques. These early systems are characterized by basic computer graphics facilities, standardization of die components, and standardization of design procedures, resulting in reduced design and drafting time. However, as these systems represent design know-how in the form of conventional procedural programming languages, only generation of the die part list and drafting of the assembly and part drawings are done by computer. The designer still needs to decide most of the important decisions including strip and die layouts interactively.

Since late 1980s, significant efforts have been made by worldwide researchers to integrate a wide variety of feature modeling, AI and traditional CAD approaches to develop dedicated progressive die design automation system including strip layout design automation. In particular, feature-based modeling technique was used to create a direct link between stamping features and stamping processes.

As one of the earliest work on feature modeling to stamping metal parts, researchers from the Department of Mechanical Engineering, University of Massachusetts, USA have described a knowledge-based system for design of progressive stamping dies for a simple hinge part (Duffey and Sun 1991). For this particular simple sheet metal part that is modeled with a feature-based approach, the system generates the flat pattern geometry and develops a strip layout automatically. However the proposed approach is limited to a narrow application domain, i.e., door hinges, and is difficult to extend to general progressive die design.

Researchers from the Department of Mechanical and Production Engineering, National University of Singapore have been developing intelligent progressive die (IPD) design automation system since late 1980s. Although, their initial work (Nee *et al.* 1992) only attempted to examine a specific problem in the strip layout design process, i.e., automatic staging and punch shape determination, the use of feature concept to represent internal and external profiles of stamping parts provides a renewed focus in the automation of progressive die design. Later their work was extended to the use of a variety of knowledge-based approaches to extend the capabilities to include automatic punch shape selection, strip layout development and 3-D die configuration (Cheok *et al.* 1994, Cheok *et al.* 1996, Cheok *et al.* 1998). Based on a feature-relationship tree that describes the stamping metal part and its topological information, model-based reasoning

and spatial reasoning techniques have been employed to reason out certain stamping processes and guide the overall planning process to develop the strip layout automatically.

Researchers at Indian Institute of Technology have developed a computer-aided die design system, CADDs, for sheet-metal blanks (Prasad *et al.* 1992, Prasad *et al.* 1994). Based on the heuristic rule-based reasoning and parametric programming techniques, the greatest advantage achieved by the system is the rapid generation of the most efficient strip and die layouts, including information about the tool configuration.

Researchers from the Department of Industrial Studies, University of Liverpool have worked on design automation for progressive piercing and blanking dies (Huang *et al.* 1996, Ismail *et al.* 1996). Their work is based on applying a coding technique to characterize the stamping part geometric features, which is subsequently used to generate the type and layout of the die punches, and then develop the strip layout automatically (Ismail *et al.* 1996). However the proposed approach is confined to straightedge sheet metal parts.

Researchers at Huazhong University of Science and Technology, China have been working on intelligent progressive die design system for many years (Xiao *et al.* 1993, Li *et al.* 2001). Features are represented in an object-oriented approach to facilitate both feature modeling to stamping metal parts and feature-based stamping process planning. An integrated intelligent CAD/CAM system for progressive die design, HPRODIE (Li *et al.* 2001), was developed, where feature mapping methods are employed to transfer the design information between different design stages at high-level semantics. With feature modeling and feature mapping, most of design processes including strip layout design can be carried out automatically.

Researchers at the National Taiwan Institute of Technology have adopted various AI techniques including fuzzy reasoning, pattern recognition, rule-based reasoning, back-propagation neural network, genetic algorithms and petri net for the design of progressive shearing cut and bending dies (Lin and Hsu 1996, Lin and Chang 1997, Lin and Deng 2001a, Lin and Deng 2001b). In order to achieve optimal strip layout design, genetic algorithms have been used as a solution tool in the analysis of the optimal strip working sequence processing a smaller difference between the right and left torque in the progressive die (Lin and Deng 2001a). The petri net model has been used to plan the work conditions (e.g., strip message communication) of each strip workstation of the progressive die (Lin and Deng 2001b).

Researchers at Shanghai Jiao Tong University, China have developed an object-oriented blackboard model which can realize hybrid knowledge representation, such as production rules, frameworks and procedures to support intelligent strip layout design to progressive stamping technology (Chi *et al.* 2001). The strip layout process knowledge is stored at different levels of knowledge sources and is activated by the blackboard event-driven messages.

Another researchers keen in progressive die design are from Department of Mechanical Engineering, Pusan National University, Korea. They have developed compact CAD/CAPP/CAM systems for the shearing cut & bending (Choi *et al.* 1999), drawing (Choi *et al.* 2000), and forging (Choi *et al.* 1996) progressive dies, in particular, in the application of lead frame for semiconductors (Choi *et al.* 2001). Based on knowledge-based rules, the work is capable of carrying out an intelligent progressive die design process with automatic development of blank layout, strip layout and die layout.

A summary of the research for sheet metal stamping process planning is given in Table 2.1. It can be seen that very little work on process planning for operation sequencing has been carried out using graph theoretic methods. This conclusion is somewhat surprising as graph theoretic methods have been widely used for operation sequencing in machining (see discussion later on).

References in progressive die design	Stamping process planning				Techniques used						Remarks	
	Unfolding	Nesting	Interactive strip layout	Automatic strip layout	Petri net	Heuristic	Genetic algorithm	Graph Theory	Constraint-based	Feature-based	Knowledge-based	
Yeh <i>et al.</i> (1994)	✓											limited to wireframe models
See Toh <i>et al.</i> (1995)	✓											using the orthographic projections as the input
Shunmugan <i>et al.</i> (2002)	✓											
Prasad <i>et al.</i> (2004)	✓											
Chow (1979)		✓										Single or double row
Qu and Sanders (1987)		✓										Two-stage approach for standard shape
Tang and Rajesham (1994)		✓										Use a polygon to approximate blank
Singh and Sekhon (1996)		✓						✓				Diagram and matrix
Choi <i>et al.</i> (1998)		✓										Incremental angles of tilt of 2°
Zhao and Peng (2002)		✓							✓			manufacturing constraints are considered
Hoffmann <i>et al.</i> (1992)				✓		✓						Bending (Press-brake)
De Vin <i>et al.</i> (1994)				✓		✓						Bending (Press-brake)
Wang <i>et al.</i> (1995)				✓		✓						Bending
Radin <i>et al.</i> (1997)				✓		✓						Bending, A*(Press-brake)
Gupta <i>et al.</i> (1998)				✓		✓						Bending, distributed planning(Press-brake)
Inui and Terakado (1999)				✓		✓						Bending(Press-brake)
Duflou <i>et al.</i> (1999)				✓		✓		✓				Bending, constraint solving, TSP(Press-brake)
Thanapandi <i>et al.</i> (2001), Thanapandi <i>et al.</i> (2002)				✓			✓					Bending(Press-brake)
Markus <i>et al.</i> (2002)				✓					✓			Bending(Press-brake)
Li <i>et al.</i> (2002)				✓								Bending, case based reasoning
Nee and Foong (1992)			✓								✓	Blanking
Ngoi and Chua (1994)			✓								✓	Piercing, blanking, bending and trimming

References in progressive die design	Stamping process planning				Techniques used						Remarks	
	Unfolding	Nesting	Interactive strip layout	Automatic strip layout	Petri net	Heuristic	Genetic algorithm	Graph Theory	Constraint-based	Feature-based	Knowledge-based	
Lin and Deng (2001)												Torque balancing Computer-aided drafting
Fogg <i>et al.</i> (1975), Nakaham <i>et al.</i> (1978), Murukami <i>et al.</i> (1980), Bergstrom <i>et al.</i> (1988)	✓	✓	✓	✓			✓			✓		
Duffey and Sun (1991)			✓							✓	✓	limited to a narrow application domain interactive rule- based operation sequencing
Cheok <i>et al.</i> (1994), Cheok <i>et al.</i> (1996), Cheok <i>et al.</i> (1998)	✓	✓	✓							✓	✓	
Prasad <i>et al.</i> (1992), Prasad <i>et al.</i> (1994)	✓	✓	✓			✓				✓		
Huang <i>et al.</i> (1996), Ismail <i>et al.</i> (1996)	✓	✓	✓							✓	✓	Confined to straightedge sheet metal parts
Xiao <i>et al.</i> (1993), Li <i>et al.</i> (2001)			✓			✓				✓	✓	feature mapping methods are employed
Lin and Hsu (1996), Lin and Chang (1997), Lin and Deng (2001a), Lin and Deng (2001b)			✓		✓		✓			✓	✓	Fuzzy reasoning, pattern recognition, rule-based reasoning, back-propagation neural network, genetic algorithms and petri net
Chi <i>et al.</i> (2001)			✓							✓	✓	Hybrid knowledge representation
Choi <i>et al.</i> (1996), Choi <i>et al.</i> (1999), Choi <i>et al.</i> (2000), Choi <i>et al.</i> (2001)			✓							✓	✓	Integrated CAD/CAM package

Table 2. 1 References in unfolding, nesting, strip layout and computer-aided progressive die design systems

2.3 Graph theory

Graph theory is a branch of mathematics that has wide practical application. Numerous problems arising in such diverse fields as psychology, chemistry, industrial and electrical engineering, transportation planning, management, marketing, and education can be posed as problems from graph theory. Graph theory is not only an area of interest in its own right but also a unifying basis from which results from other fields can be collected, shared, extended, and disseminated (Evans and Minieka, 1992). Here, some terminologies related to graph theory and used throughout this thesis are defined.

Graph: A graph G consists of two types of elements, namely vertices and edges that are said to connect pairs of vertices. The set of edges is usually defined as a set of two-element subsets of the set of vertices. Edges may be given direction. The number of vertices in a graph G is called its **order**, and the number of arcs in G is its **size**.

Simple graph: A simple graph is an un-weighted, undirected graph containing no self-loops. Unless stated otherwise, the unqualified term "graph" used in this thesis refers to a simple graph.

Undirected graph: An undirected graph or graph G is an ordered pair $G = (V, E)$ where V is a set of vertices or nodes, E is a set of edges. In progressive die design, the stamping operations that are "close" to each other can not be arranged at the same station. This closeness constraint can be easily modeled as an undirected graph. Nodes in this graph represent stamping operations, while an edge incident to a pair of nodes indicates these two stamping operations (represented by the corresponding nodes), can not be arranged at the same station. In another words, they have a closeness constraint between

them.

Directed graph: A directed graph or digraph G is an ordered pair $G = (V, A)$ where V is a set of vertices or nodes, A is a set of ordered pairs of vertices called directed edges, arcs or arrows. A variation on this definition is the oriented graph, which is a simple graph with an orientation assigned to all of its edges. The distinction between a directed graph and an oriented graph is that if x and y are vertices, a directed graph allows both (x, y) and (y, x) as edges, while only one is permitted in an oriented graph. A directed graph may or may not be allowed to have loops, that is, edges where the start and end vertices are the same. By definition, this is forbidden in an oriented graph. Unless stated otherwise, the unqualified term "directed graph" or "digraph" used in this thesis refers to an oriented graph. In progressive die design, the precedence constraints imposed by geometrical or technological requirements can be naturally modeled as a directed graph (Precedence Graph). If (u, v) is an arc of digraph G , then u is said to be adjacent to v , and v is adjacent from u . Further, the arc (u, v) is incident from u and incident to v . The **outdegree** $od(v)$ of a vertex of v is the number of vertices adjacent from v . The **indegree** $id(v)$ of v is the number of vertices adjacent to v .

Cycle: A cycle in a graph consists of a sequence of successively incident edges and their end vertices, where the terminating vertices are identical. A graph is acyclic if it contains no cycles.

Directed Acyclic Graph (DAG): Directed acyclic graphs are directed graphs with no cycles. Called DAGs, they are an important class of graphs, which have many applications, such as those involving precedence among "events". Figure 4.2 is an example of DAG. Using DAGs, many problems in graphs become simpler, such as

critical path analysis, expression tree evaluation, and game evaluation. The property of DAGs which is optimal for these problems is their ability to be topologically sorted using depth-first search.

Figure 2.4 An example of DAG

Topological sort: A topological sort is an ordering of vertices in a DAG such that, if there is path from node u to node v , then v appears after u in the ordering. Therefore, a cyclic graph cannot have a topological order. A topological sort of a graph can be viewed as an ordering of its vertices along a horizontal line so that all directed edges go in one direction. Referring to the example of the DAG depicted in Figure 2. 4, a corresponding topological sort could appear as follows (Figure 2. 5):

Figure 2.5 A topological sort of example DAG

Note that all of the edges are pointing in the same direction. In general, the vertex order produced by a topological sort is not unique. We could reorder the nodes while maintaining the property of a topologically sorted DAG (for example, we could switch the positions of A and D).

Tree: A tree is a connected graph without cycles. A digraph is a **directed tree** if its underlying graph is a tree. If u and v are distinct vertices of a tree T , then T contains exactly one $u - v$ path.

2.3.1 Data structures for graphs

Adjacency matrix and adjacency list are two data structures frequently used for graph implementation. Adjacency matrix is defined as follows. For an undirected graph, let A be an $m \times m$ matrix in which $a_{ij} = 1$ if vertices i and j are adjacent, that is, connected by an edge, and 0 otherwise. For a directed graph, we define $a_{ij} = 1$ if there exists an arc (i, j) from vertex i to vertex j and $a_{ij} = -1$ if there exists an arc (i, j) from vertex j to vertex i , and zero otherwise. An adjacency list maintains a pointer to a list of vertices that are adjacent to a given vertex. For a directed graph, we need only specify all vertices directed from a given vertex.

Adjacency Matrix Representation (AMR)	$int[][]$ <i>Graph</i> ; Graph[i][j] = 1 if there's an edge from i-th to j-th vertex.	Space Complexity: $O(n^2)$, where n is number of vertices in the graph.
Adjacency List Representation (ALR)	<div><div><div><div><div><i>List</i></div><div>...</div><div>i</div><div>...</div></div><div><div><i>Index</i></div><div>...</div><div>4→5→8</div><div>...</div></div></div></div></div> <div>means that $(i,4), (i,5), (i,8)$ are edges</div>	Space Complexity: $O(e)$, where e is number of edges in the graph.

Table 2. 2 Space complexity of AMR and ALR

From Table 2.2, we can see that ALR is preferred to AMR when the number of edges is significantly smaller than the squared number of vertices. If the number of edges is significantly greater than the squared number of vertices, AMR is a preferred implementation for graphs.

2.3.2 Cycle detection algorithms

Assume there are a set of tasks to perform and the tasks have to be performed in some sort of sequence. The precedence constraints form a directed graph. Before trying to determine the sequence of the given tasks, it is necessary to ensure that the directed graph is acyclic. A cycle in the directed graph will lead to contradiction. The standard way to recognize cycles in a tree is to do a depth-first search (DFS) marking vertices along the way. If you hit a vertex that you have already marked, then you have a cycle.

Cycle detection on a graph is a bit different from that on a tree due to the fact that a tree node (vertex) can not have multiple parents. DFS algorithm may not work properly on a graph. For example, the graph in Figure 2.6 will be falsely reported to have a cycle, since node D will be seen twice in a DFS starting at node A.

Figure 2. 6 An acyclic graph on which the tree cycle detection algorithm would fail

In order to detect cycles, we use a modified depth first search called a colored DFS (Kamil, 2003). All nodes are initially marked white. When a node is encountered, it is marked grey, and when its descendants are completely visited, it is marked black. If a gray node is encountered again before all of its descendants have been visited, then there must be a cycle.

2.3.3 Graph colouring algorithms

Graph colouring is one of the oldest and best-known problems of graph theory. When people became accustomed to applying the tools of graph theory to the solution of real world technological and organizational problems, graph coloring emerged as a natural way of tackling many practical problems.

Given an undirected graph $G = (V, E)$, vertex colouring is an assignment of a label to each node. It is required that the labels on the pair of nodes incident to any edge be different. A minimum colouring of a graph is a colouring that uses as few different labels as possible. A clique of the graph is a set of mutually adjacent vertices. A maximum clique is, naturally, a clique whose number of vertices is at least as large as that for any other clique in the graph. If the vertices have weights then a maximum weighted clique is a clique with the largest possible sum of vertex weights. Colouring and Clique problems are very closely related. The size of the maximum clique is a lower bound on the minimum number of labels needed to color a graph.

Many problems of practical interest can be modeled as colouring and clique problems. The general form of these applications involves forming a graph with nodes representing items of interest. An edge connects two 'incompatible' items. The maximum clique problem is then to find as large a set of pairwise incompatible items as

possible. The minimum colouring problem is to assign a color to each item so that every incompatible pair is assigned different colors. The minimum number of colors in a colouring of G is called the *chromatic number* of G .

The general question of determining the chromatic number of a graph is a NP-complete problem (Garey and Johnson 1979). Some exhaustive search algorithms have been developed (Brelaz 1979, Christofides 1971, Yuster 1998, Blum 1994, Mehrotra and Trick 1996). Michael Trick's page, <http://mat.gsia.cmu.edu/COLOR/color.html>, provides a nice overview of applications of graph colouring, and a collection of over seventy graph colouring instances arising in applications such register allocation and printed circuit board testing. Finally, it contains a C language implementation of an exact coloring algorithm, DSATUR. While it is probably not the fastest code, it provides a reasonable benchmark. In the developed prototype system, DSATUR algorithm is implemented in C++. Figure 2.7 shows how this algorithm works.

Figure 2. 7 An exact graph coloring algorithm (DSATUR)
(<http://mat.gsia.cmu.edu/COLOR/color.html>)

2.3.4 Graph based operations sequencing methods for machining

Operations sequencing is always a major concern of process planning and the sequence is generally obtained according to a particular objective function, such as minimizing time or cost within specified constraints. The potential for using Graph Theory for operation sequencing was addressed by Sundaram (1986) and Prabhu *et al.* (1990). Two important issues mentioned in this work are elimination of infeasible machining operation sequences and the use of a tree structure for enumerating all the paths to weed out infeasible sequences. The Hamiltonian path (HP) analogy for the process planning problem based on the precedence graph and operation cost matrix was reported by Iranni *et al.* (1995). The need for using heuristic approaches for randomly generating alternate sequences and thereby alternative process plans is stressed in their research work. Lin *et al.* (1998) proposed a framework for integrating artificial intelligence techniques and mechanistic models in the task of generating process plans for prismatic parts. The informed graph-search strategy in AI is used to achieve the function of operation sequence planning. A hybrid method combining an unsupervised learning algorithm and a graph manipulation algorithm was proposed by Chen and LeClair (1994). The sequence of the features in each set-up is constrained and determined by the interacting feature relationships. After the process of feature sequencing, several graph manipulation algorithms are proposed to obtain the optimal tools sequence for creating the features in a set-up by minimizing the tool changes. Borenstein (2000) described a graph-based model for representing manufacturing operation sequences. Three kinds of representation are proposed to model operation sequences at different levels: (i) Precedence Graph of operations; (ii) Tree of Feasible Operations; and (iii) Projected Reachability Graph of Operations.

Operation sequencing in process planning is quite similar to strip layout design in progressive die design; however, they are not so similar that the operations sequencing methods can be directly used in process planning for strip layout design. Usually, it is assumed in process planning that only one operation can be executed at a given time, but this is not true in progressive dies. In a progressive die, stamping operations are clustered and distributed to a series of sequential die stations. To minimize the die construction cost, the minimum number of die stations is always an optimization objective, which entails arranging as many operations on the same die station as possible as long as inherent constraints are not violated.

2.4 Summary

This chapter presented a preliminary survey of the techniques that can be applied to aid in developing a die design automation system. Research work in operations sequencing was also reviewed. From the review, the following observations can be made:

- Many of the progressive die design systems built have demonstrated the technical feasibility of employing knowledge-based methods to solve design and manufacturing problems
- Recent advances in the field of feature modeling and Artificial Intelligence (AI) have given rise to the possibility to construct feature- and AI-based systems that incorporate built-in intelligence and apply it to solving progressive die design problems including strip layout design automation.
- The strip layout design process is accomplished interactively in nearly all the computer-aided progressive die design systems, which is error-prone and relies

heavily on the process planner's experiences.

- Almost all the research efforts on the automation of strip layout design have focused on bending operation, other operations are seldom involved.
- Feature-based and rule-base technologies have been shown to be efficient technologies for operation sequencing.

In conclusion, the literature survey shows that in spite of the great advances in various AI and traditional computer approaches, much remains to be done, especially in strip layout design. Automating strip layout design is a valuable research topic and an important step towards automating the whole progressive die design process.

CHAPTER 3 Feature-based Sheet Metal Part and Strip Modelling

3.1 Introduction

This chapter describes the feature-based representation scheme that was developed to support the graph theoretic process planning algorithm. Section 3.2 discusses feature-based modelling of sheet metal parts. Section 3.3 describes how stamping features are created from part features. Section 3.4 discusses the mapping of stamping features to feature operations. Finally, section 3.5 describes the representation of stamping operation relations.

3.2 Feature-based modeling of sheet metal parts

The first step in process planning is to understand the design. In manual process planning the trained human process planner sifts through the geometric data of the engineering drawing to group geometric elements together to form entities that are meaningful for manufacturing. Similarly in generative process planning a trained human being translates the part drawings into a part description in terms of entities relevant to manufacturing. These entities are called features. In both cases process planning proceeds only after the part is understood in terms of its constituent features. Thus it can be said a feature based description of the part is a required input to start process planning.

As a promising tool, feature-based modeling has been employed to facilitate design and manufacturing applications by associating shape macros with application-specific semantic information. Design features created in a design-by-feature system in terms of additive or subtractive volumes pave the way to embody the function of a component as

well as to reflect designer's intent. On the other hand, manufacturing features are normally created from material stock in terms of subtractive volumes and linked to process planning activities such as tool selection, machine selection, and operations sequencing. Process planning proceeds only after the part is understood in terms of manufacturing features. Similarly, sheet metal parts are usually designed using standard sheet metal features (design features) such as flat, hole, slot, bend, emboss and so on. Then sheet metal features are mapped or converted to stamping features (manufacturing features) interactively or automatically. Stamping features are information carriers that are used to model a stamping part with a set of manufacturing information including geometric and non-geometric attributes. Each of these stamping features can be manufactured with a specific stamping operation or a combination of stamping operations.

A sheet metal part can consist of the following standard sheet metal features:

- Primary feature: flat, tab, hem, drawing, etc.
- Add-on feature: hole, cutout, slot, extrusion hole, emboss, countersink, etc.
- Connective feature: bend, jog, etc.

A sample part represented using standard sheet metal features and its feature tree is shown in Figure 3.1. This feature tree is formed by introducing new design features to it incrementally and it incorporates the design process information which can be utilized to interpret the part in terms of manufacturing features during the design evolution process.

Figure 3. 1 A sample part represented using standard sheet metal features

Sheet metal features are related to one another through spatial interaction and precision requirement in a stamping part. A sheet metal feature can relate to one or more other sheet metal features. Although there can be numerous relations among various sheet metal features, it is not practical or necessary to take all of them into account, because some of them may not be relevant to the stamping process planning task. Three critical types of relations among sheet metal features - *is-in*, *adjacent-to*, and *precision-associated* - are defined here. A similar work can be found in Chen *et al.* (1992). They defined three kinds of relations for machining parts, i.e., *is-in*, *is-on*, and *adjacent-to*. *Is-in* relation indicates a negative feature is *in* another feature, while *is-on* relation indicates a positive feature is *on* another one. However, from the view point of stamping, all the local features are produced by either punching (material removal) or forming (material shape change) and there is no positive feature. So we use the *is-in* relation to describe the spatial interaction that an add-on feature is in another sheet metal

feature (e.g., primary feature). The *adjacent-to* relation is borrowed from Chen *et al.* (1992) to indicate a spatial interaction that a connective feature is adjacent to another sheet metal feature (e.g., primary feature). The *precision-associated* relation is new and is introduced to indicate the case that a sheet metal feature does not directly connect to, but is associated with another sheet metal feature by a precision requirement. The precision requirement stated here refers to specific tolerance need, not the general part tolerance.

To illustrate these feature relations, Figure 3.2 shows the equivalent symbolic feature-tree representation of the part shown in Figure 3.1. For example, *Slot0* is connected to *Flat0* by a *is-in* relation, *Bend0* is connected to *Flat0* by an *adjacent-to* relation, and *Hole1* is associated with *Hole2* by a *precision-associated* relation (which is not shown in the previous figure Figure 3.1).

Figure 3. 2 Symbolic feature-tree representation of the sample part

3.3 Generating of stamping features

A feature-based part model that consists of a set of sheet metal features only contains the geometrical and technological information about what the final product will be like. It does not provide the manufacturing information on how the final product will be produced from the raw material, e.g., rolled sheet in progressive dies. Once a die maker receives the part model, the part will be unfolded and nested to produce a desirable blank layout. Once the desirable blank layout has been created, the sheet metal features in the part model will be converted to stamping features on the strip (raw material). At the same time additional stamping feature, such as piloting and scrap features, are also created. This process is referred to by the author as “generating stamping features from sheet metal features”.

Stamping features can be divided into four categories: root feature, internal features, external features, and piloting features. Root feature is the most important feature in the hierarchical structure and is defined by the direction of the normal of its surface. This is because, whatever the sheet metal part is, there must be a primary feature that remains parallel to the surface of the die block throughout the processing of the strip in the progressive die. The default root feature is mapped from the first primary feature in the part model. Internal features are local features on the sheet metal part model that will appear in the final product. External features (scrap features) are auxiliary features that are introduced to form the final product profile. Piloting features are another kind of auxiliary feature used to register the strip when it is fed through the progressive die.

Root feature and internal features in the strip model are sheet metal features taken directly from the part model, while external features and piloting features are designed by the process planner interactively. The design process of external features produces the

blank profile by creating a set of scrap pieces to form the set of flat features. This activity requires a considerable amount of image processing and cognition skills, and needs to consider a lot of factors such as the shape, material, precision, roughness, toughness, etc. of the strip. It is very difficult to develop a knowledge-based tool to automate the scrap design scheme.

Figure 3.3 shows a strip model consisting of stamping features generated from the sheet metal features of the example part shown in Figure 3.1. A two-station strip is created to avoid overlapping between the stamping features. The local features in the part model are reproduced. In addition, two piloting features and 7 scrap features have been created on the strip.

Figure 3. 3 Stamping features and its stamping feature tree of the example part

3.4 Mapping stamping features to stamping operations

Stamping features constitute a strip model, while stamping operations are selected as elements of a stamping process plan. Typical stamping operations include piercing, notching, cut-off, blanking, lancing, shaving, drawing, embossing, coining, trimming and

other miscellaneous forming operations. A stamping feature can be manufactured with one specific stamping operation or a combination of stamping operations. Feature-operation mapping means the transformation process of design and manufacturing information from stamping features to a set of stamping operations. Three types of feature-operation mappings are distinguished here: direct mapping, decomposition mapping, and composition mapping.

(1) *Direct mapping*: a stamping feature is transformed into a stamping operation directly with one-to-one mapping; an example is the transformation of a hole feature into a piercing operation whose punch shape is the same as the hole's shape.

(2) *Decomposition mapping*: a stamping feature is transformed into a set of stamping operations with one-to-many mapping; an example is the transformation of an extrusion hole feature into pre-piercing and extrusion operations.

(3) *Composition mapping*: a set of stamping features are transformed into a stamping operation with many-to-one mapping; an example is the transformation of two bend features into one single U-bending operation. Another example is the transformation of two hole features having a *precision-associated* relation into one single piercing operation.

3.5 Representation of Stamping Operation Relations

It is assumed that n stamping features of the part are to be produced by m stamping operations, and that each operation is assigned one and only one stage of the stamping die. Two types of relationship can exist between stamping operations: cluster and precedence relations. Cluster operations are performed at the same time at the same stage. For

example, assume two holes are to be pierced, they are not close to each other, and the distance between them has a high precision requirement. In this situation the two holes should be pierced simultaneously at the same stage.

A precedence relation implies that an operation or operation cluster must be performed before another operation or cluster. For example, when the location of a hole is critical and that hole is in a flange, the hole is usually punched after the bending operation. Actually, it would be less expensive from a tooling standpoint to punch the hole while the part is flat, but it is very difficult to maintain hole location with such a sequence. Operations related by precedence are performed at different stages.

The relationships can be described mathematically by two graphs and their equivalent matrices: an operation adjacency graph and an operation precedence graph. The adjacency graph is an undirected graph defined as $G_a = (V, E)$, where V the vertices define the operations and the undirected edges E define the adjacency constraints. Two operations are adjacent if they cannot be performed at the same station. In other words, the adjacency graph is a manufacturing infeasibility graph, where the infeasibility is determined by the geometric distance between the features. The operation precedence graph is a directed acyclic graph (DAG) defined as $G_p = (V, A)$, where V the vertices define the operations and the directed edges A define the precedence constraints based on manufacturing best practices. This graph must be acyclic, but is not necessarily a tree, in order to obtain a feasible solution. The pictorial form of the graphs is useful for visualizing the process constraints. The matrix form is suitable for computer implementation.

3.5.1 Representation of operation precedence relations

Manufacturability rules are used to define precedence relationships. Some rules are given below:

Rule Operation_Relation1: The piercing operations to stamp two pilot holes should be staged in the first station.

Rule Operation_Relation2: When embosses exist, if possible they are embossed in the first station to prevent deformation of other stamping features.

Rule Operation_Relation3: If holes (except pilot holes) and scraps (that don't accommodate indirect pilot holes) can not be stamped simultaneously in the same station, the process sequence is as follows: piercing operations precede notching operations.

Rule Operation_Relation4: If holes or scraps to be pierced are close to each other, then the adjacent holes or scraps should be shifted to next die stations to prevent weak die problem.

Rule Operation_Relation5: If possible, the cut-off operations on scraps that accommodate pilot holes should be staged in the final station.

CHAPTER 3 FEATURE-BASED SHEET METAL PART AND STRIP MODELLING

	L-bend bend0	U-bend bend1 & bend2	Pierce hole0	Pierce hole1	Pierce hole2	Pierce hole3	Pierce hole4	Pierce slot0	Emboss emboss0
L-bend bend0	0	-1	0	0	0	0	0	0	0
U-bend bend1 & bend2	1	0	0	0	0	0	0	0	0
Pierce hole0	0	0	0	0	0	0	1	1	-1
Pierce hole1	0	0	0	0	0	0	1	1	-1
Pierce hole2	0	0	0	0	0	0	1	1	-1
Pierce hole3	0	0	0	0	0	0	1	1	-1
Pierce hole4	0	0	-1	-1	-1	-1	0	0	0
Pierce slot0	0	0	-1	-1	-1	-1	0	0	0
Emboss emboss0	0	0	1	1	1	1	0	0	0

Table 3. 1 Partial operation precedence matrix

Using such rules, a $m \times m$ (square) operation precedence matrix can be developed that represents the precedence relationships between pairs of stamping operations (Table 3.1). The order of the operations in the rows and columns must be the same. Each element in the operation precedence matrix can have a value of either 1, -1 or 0 where:

$$a_{ij} = \begin{cases} +1, & \text{when } i \text{ precedes } j \\ -1, & \text{when } j \text{ precedes } i \\ 0, & \text{when the two operations are independent} \end{cases}$$

$$\sum_i \sum_j a_{ij} = 0$$

where a_{ij} is any element in the matrix and i and j represent the ordinal number of the corresponding row and column respectively.

3.5.2 Representation of operation cluster relations

The operation precedence matrix defines precedence relations between stamping operations arising from geometric or manufacturing constraints. Stamping operations with a precedence relationship can't be performed at the same stage. However, there are other stamping operations that cannot be performed at the same stage, for which precedence is defined by another kind of constraint. This constraint occurs when the features corresponding to the operations are close to each other and there is not enough space for the punches. The operation adjacency matrix defines the "closeness" constraints. The matrix is a square $m \times m$ matrix (Table 3.2). To determine the entries in the matrix the distances between pairs of features are compared with a pre-defined, practical, desirable minimum distance, which in turn is determined by the stamping material type and dimensions of standard dies and punches. If the actual distance is less than the minimum then a 1 is entered in the matrix, otherwise a 0 is entered. An entry of 1 indicates that the two related operations cannot be performed at the same stage. That is,

$$b_{ij} = \begin{cases} +1, & \text{when distance between two features } i \text{ and } j \text{ is less than the minimum distance} \\ 0, & \text{when distance between two features } i \text{ and } j \text{ exceeds the minimum distance} \end{cases}$$

CHAPTER 3 FEATURE-BASED SHEET METAL PART AND STRIP MODELLING

	L-bend bend0	U-bend bend1 & bend2	Pierce hole0	Pierce hole1	Pierce hole2	Pierce hole3	Pierce hole4	Pierce slot0	Emboss emboss0
L-bend bend0	0	0	0	0	0	0	0	0	1
U-bend bend1 & bend2	0	0	1	1	0	0	0	0	0
Pierce hole0	0	1	0	0	0	0	0	0	0
Pierce hole1	0	1	0	0	0	0	0	0	0
Pierce hole2	0	0	0	0	0	0	0	1	1
Pierce hole3	0	0	0	0	0	0	0	1	1
Pierce hole4	0	0	0	0	0	0	0	0	0
Pierce slot0	0	0	0	0	1	1	0	0	0
Emboss emboss0	1	0	0	0	1	1	0	0	0

Table 3. 2 A partial operation adjacency matrix

Two stamping operations can be clustered (at the same stage) only if they are not directly connected by an edge in the operation adjacency graph. Clustering is achieved by partitioning the vertices V into a minimum number of sets of mutually independent vertices. This is a vertices colouring problem of a graph, assuming one colour per vertex. This clustering problem is different from the general graph coloring problem because operations that have precedence relations (Section 3.5.1) can't be clustered together.

The ALR (Section 2.3.1) data structure is used for implementation because the number of edges is smaller than the squared number of vertices. The matrices are displayed only for user reference.

3.6 Summary

Feature-based representation is more convenient for integrating life cycle activities than feature-extraction. Design features for stamping were classified into three basic

categories: primary features, add-on features and connective features. Three kinds of features relations were defined: *is-on*, *adjacent-to*, and *precision-associated* relations.

The manufacturing representation of sheet metal features is called stamping features, which were divided into four categories: root feature, internal features, external features, and piloting features. The mapping of design features to stamping features is accomplished by the process planner interactively.

The mapping of stamping features to stamping operations assigns specific operations to each stamping feature. Three types of feature-operation mappings, i.e., direct mapping, decomposition mapping, composition mapping were distinguished in this chapter.

Finally, the representation of stamping operation relations was elaborated. Two types of operation relations, precedence relation and cluster relation, were discussed in detail.

CHAPTER 4 THE PROPOSED METHODOLOGY

4.1 Introduction

Automated operations sequencing for manufacturing has been the goal of researchers for years and computer-aided process planning (CAPP) systems for sequencing have been developed (Shan *et al.* 1992; Atami *et al.* 1995; Rocha *et al.* 1999; Bhaskara *et al.* 1999; and Kim *et al.* 2001). The work presented here differs from the knowledge-based approaches discussed in Chapter 2. The focus is solely on developing an efficient algorithm for sequencing. This chapter presents an efficient algorithm for automatic determination of the stamping operation sequence.

The next section gives the problem statement. Section 4.3 explains the algorithm. Section 4.4 illustrates the algorithm through a case study. Section 4.5 discusses the correctness and complexity of the proposed approach. A summary is presented in Section 4.6.

4.2 Problem statement

The stamping operation sequencing task transforms a set of related stamping features of a stamping part into a set of stamping operations and describes the relations between these stamping operations. The domain of possible sequences contains in principle n^m permutations, n being the number of operations for a given part and m being the number of workstations. The final aim is, after comparison of possible alternatives, to identify an optimal sequence of operations that requires the minimum number of workstations.

The relationships among stamping features, die operations, and workstations are shown in Figure 4.1 for a stamping part to be manufactured using a progressive die, for

which the stamping operations for each feature of the part have been generated, One stamping feature may need one or several die operations to punch or form it.

Figure 4. 1 Relationship between stamping features, die operations and workstations.

The operation sequencing problem is usually modeled as a combinatorial optimization problem with precedence constraints. The constraints may come from technical requirements, past experience, and rules of thumb. The complexity of solutions to this problem is highly dependent upon the geometric shape and the number of features of the stamping part. Actually, aside from precedence constraints, the optimal sequence should also satisfy the simplest die construction requirement. That is, the sequence should give the smallest number of die workstations possible. Therefore, it is desirable to arrange as many operations in the same workstation as possible.

The problem of operation sequencing in die design tackled by this research can be stated as follows: given a feature-based description of the part and the manufacturing

knowledge about the ordering of operations, find an operation sequence of the part utilizing the available manufacturing resources that is optimal/near optimal with respect to a predefined global function, e.g. minimum number of die stations.

4.3 Proposed graph based operation clustering and sequencing approach

The proposed graph based operation sequencing approach for progressive die design will be described in this section. The following assumptions are made in this approach: i) the stamping part has been designed on a feature-based CAD system and hence a feature-based tree exists for all part design features; ii) the process planner has manually determined the orientation of the part on the strip and has determined the number and dimensions of the scrap pieces. The scrap pieces are also part of the design feature tree, but are not ordered; and iii) if two operations have a precedence relationship then they must be performed at different stations. The flowchart of the proposed approach is shown in Figure 4.2. It consists of five steps, which are described below.

Step 1 (Generate stamping features)

Input: Feature-based CAD model of the stamping part to-be-manufactured.

Operation: Generate the stamping features from the sheet metal features consisting of the CAD model of stamping part to-be-manufactured. In this process, the blank orientation is determined using a nesting algorithm to obtain the optimal material utilization. The sheet metal features in the CAD part model are mapped to stamping features in a strip model. Auxiliary stamping features (pilot features and scrap features) are devised by the process planner.

Output: Stamping features set (including pilot features and external features).

Step 2 (Map stamping features to stamping operations)

Input: Stamping features set.

Operation: Map stamping features to stamping operations. The current algorithm is limited to direct (one-to-one) and composition (many-to-one) mapping between stamping features and stamping operations

Output: Stamping operations set.

Step 3 (Generate operation precedence and adjacency matrices)

Inputs: Stamping operations set, pre-defined operation ordering (from manufacturing rule database), and design constraints from the CAD model.

Operation: Generate the operation precedence and adjacency matrices. Verify the operation precedence graph is acyclic using a colored DFS. Cyclic graphs must be corrected manually by the process planner by modifying the rule database and/or the design-to-operation feature assignment.

Output: Operation precedence and adjacency matrices.

Step 4 (Macro-plan: solve precedence constraints)

Inputs: Operation precedence matrix.

Operation: Cluster the operations into partially ordered sets using a modified topological sorting algorithm.

Output: Clusters of partially ordered operations.

Step 5 (Micro-plan: solve closeness constraints)

Inputs: Clusters of partially ordered operations and the corresponding sub-graphs of operation adjacency graph.

Operation: For each cluster use the graph colouring algorithm to sub-colour the clusters – this will carry out additional sorting to take into account the closeness relationships.

Output: Fully ordered operations clusters.

Figure 4. 2 Proposed graph based operation clustering and sequencing for progressive die design

4.4 A case study

Here how the proposed approach works for the example part, which is used in the previous chapters, will be illustrated as a case study.

Step 1 and Step 2 actually are already shown in Chapter 3.

Step 3. The operation precedence and operation adjacency graphs are shown in Figure 4.3 and Figure 4.4 respectively. The operation precedence and operation adjacency matrices are shown in Table 4.1 and Table 4.2 respectively. The two graphs — precedence and adjacency graphs — are automatically generated using WinGraphviz API. WinGraphviz is a free software base on Graphviz project, which is a Project of AT&T Labs research (Tsen 2005).

Step 4. Macro-planning using a modified topological sort algorithm (MTS). In determining the operations sequence, the operations to be performed in the first step are selected first. Then these nodes are deleted from the operation precedence matrix. The second step follows. This process is repeated until no nodes are left in the operation precedence matrix. In this step we only consider the precedence constraints among the operations. Operations that have no edges adjacent to them (in-degree is zero) can be performed at the first step. The algorithm searches for all the operations whose in-degrees are zero and arranges them in the same step and delete them from the operation precedence matrix. The process repeats until no operations left.

Figure 4. 3 Precedence graph of the example part (automatically generated using WinGraphviz API)

Figure 4. 4 Operation adjacency graph for the example part (automatically generated using WinGraphviz API)

	PierceHole0	PierceHole1	PierceHole2	PierceHole3	PierceHole4	PierceHole5	PierceHole6	PierceHole7	PierceSlot0	NotchScrap0	NotchScrap1	NotchScrap2	NotchScrap3	NotchScrap4	NotchScrap5	NotchScrap6	UBendBend0_Be nd1	UBendBend2_Be nd3	PiercePilotHole0	PiercePilotHole1
PierceHole0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
PierceHole1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
PierceHole2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	-1	0	0
PierceHole3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	-1	0	0
PierceHole4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	-1	0	0
PierceHole5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
PierceHole6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	-1	-1	0	0
PierceHole7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
PierceSlot0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
NotchScrap0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
NotchScrap1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	-1	-1
NotchScrap2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	-1	-1
NotchScrap3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
NotchScrap4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
NotchScrap5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
NotchScrap6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	-1	-1
UBendBend0 _Bend1	-1	-1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	1	0	-1	-1	-1
UBendBend2 _Bend3	-1	-1	-1	1	1	-1	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	0	-1	-1
PiercePilotHol e0_PilotHole1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	0

Table 4. 1 Operation precedence matrix of the example part

	PierceHole0	PierceHole1	PierceHole2	PierceHole3	PierceHole4	PierceHole5	PierceHole6	PierceHole7	PierceSlot0	NotchScrap0	NotchScrap1	NotchScrap2	NotchScrap3	NotchScrap4	NotchScrap5	NotchScrap6	UBendBend0_Bend1	UBendBend2_Bend3	PiercePilotHole0	_PilotHole1
PierceHole0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
PierceHole1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PierceHole2	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PierceHole3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
PierceHole4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PierceHole5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PierceHole6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
PierceHole7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PierceSlot0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NotchScrap0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1
NotchScrap1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1
NotchScrap2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	1
NotchScrap3	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
NotchScrap4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
NotchScrap5	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1
NotchScrap6	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
UBendBend0																				
_Bend1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	0	0
UBendBend2																				
_Bend3	0	0	0	1	1	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0
PiercePilotHole0																				
e0_PilotHole1	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	0

Table 4. 2 Operation adjacency matrix for the example part

Suppose that the operation precedence matrix has been established. The procedure for determining the operations sequence (step 4 above) is given below:

Step 1. Initialization. Copy the operation precedence matrix for use in this process; let the stamping step (i) be 1; and initialize a operations list, L_i , for the current stamping step i .

Step 2. Get all the nodes whose in-degrees are zero and store them in list L_i .

Step 3. The corresponding row and column of operations in the list L_i are then deleted from the precedence matrix and the remaining rows and columns are updated. This forms a new precedence matrix.

Step 4. Repeat Step 2, 3, until there is no operation left in the operation precedence matrix.

Step 5. Use the resulting operation lists to construct the operation sequence and stop.

For the example part (Figure 3.1), the resulting operations sequence is as below:

Step list 1: PierceHole0, PierceHole1, PierceHole2, PierceHole5, PierceHole7, PierceSlot0, NotchScrap0, NotchScrap3, NotchScrap4, PiercePilotHole0_PilotHole1;

Step list 2: NotchScrap1, NotchScrap2;

Step list 3: UBendBend2_Bend3;

Step list 4: UBendBend0_Bend1;

Step list 5: PierceHole3, PierceHole4, PierceHole6;

Step List 6: NotchScrap5, NotchScrap6.

Step 5 Micro-planning using DSATUR graph colouring algorithm. The result of Step 4 is an ordered partition of the operation precedence graph (OPG) that gives the minimum number of die stations needed to manufacture the given stamping part. As stated in Section 4.2, precedence constraints are not the only constraints that need to be considered in progressive die design. Another type of constraint, closeness constraint is considered after the partitioning of OPG. Due to the similarity between operation precedence graph and operation adjacency graph (OAG), the solution of above modified topological sort algorithm applied to OPG also partitions OAG into smaller operation adjacency graphs (S-OAGs) and each of them is a sub-graph of the original operation adjacency graph. The vertices of these S-OAGs are the operations stored in the same step list. If an edge exists between two operations, it means these two operations can not be arranged in the same station due to the closeness constraint. The S-OAGs resulting from Step 4 are shown in Figure 4.5.

Figure 4. 5 S-OAGs of the example part

The above partition divides the edges in OAG into two categories: inner edges of a certain S-OAG and inter edges of two different S-OAGs. Due to the precedence constraints, the operations clustered in different S-OAGs cannot be arranged in the same station. Inter edges between two different S-OAGs carry the same information as the precedence constraints between the two S-OAGs, which is redundant. We just need to consider the inner edges of S-OAGs. In order to satisfy the closeness constraints, we need to partition S-OAGs into smaller clusters. In each such cluster, there must be no edge between vertex pairs, as for example, between PierceHole2 and PierceHole0 in S-OAG1. In other words, all the operations in the cluster can be arranged in the same station. To solve this problem, a graph colouring (vertex coloring) algorithm is used.

The coloration of the vertices in a graph G is often used for solving scheduling problems of the following type. We are given a set E of jobs (all of them have the same processing time) which have to be performed by using some machines. The constraints to be taken into account are that for each job j in E there is subset of jobs that can not be performed by the same machine. The task of finding an optimal vertex colouring of a graph is an NP-hard problem, that is — informally speaking — a problem with no known polynomial solution. The high computational complexity of the graph colouring problem necessitates the use of heuristic approximation methods for the determination of suboptimal solutions in polynomial time. DSATUR was first presented by (Brelaz 1979) based on the observation that the constraints in color assignment do not directly result from the degree of the colored vertex, but from the number of its uniquely colored neighbors. Saturation degree of a vertex is defined as the number of different colours to which it is adjacent (coloured vertices).

DSATUR Algorithm (so called because it uses saturation degree)

1. Arrange the vertices by decreasing order of degree.
2. Colour a vertex of maximal degree with colour 1.
3. Choose a vertex with a maximal saturation degree. If there is an equality, choose any vertex of maximal degree in the uncoloured subgraph.
4. Colour the chosen vertex with the least possible (lowest numbered) color.
5. If all the vertices are coloured, stop. Otherwise, return to 3.

Algorithm DSATUR has become the de facto standard among graph-coloring algorithms (Janczewski 2001). The DSATUR algorithm has some nice properties. First of all, it optimally colours all bipartite graphs (Brelaz 1979). Next, it leads to optimal colouring of cycles, uni- and bicyclic graphs, cacti, trees, wheels, and necklaces. At last DSATUR optimally colors almost all k -colorable graphs (Turner 1988).

Algorithm DSATUR is implemented in C++ in this work to solve the micro planning sub-problem arising from closeness constraints imposed on stamping operations. The DSATUR algorithm is applied sequentially to the S-OAGs to obtain a series of independent sets for each S-OAG. All the operations in an independent set will be arranged in the same station. The independent sets obtained from the S-OAGs with smaller cardinal numbers will be arranged before those with larger cardinal numbers. The order of independent sets resulting from the same S-OAG can be sequenced arbitrarily if no other constraint exists. The minimum number of die stations is the sum of number of independent sets and each of them is determined by the maximum clique in each S-OAG.

For the example part, application of DSATUR algorithm gives the result below. S-OAG1, which has 10 vertices and 5 edges, is partitioned into two independent sets: Set 1 (PierceHole0, PierceHole5, PierceHole7, NotchScrap0, NotchScrap3, NotchScrap4, and PiercePilotHole0_PilotHole1) and Set 2 (PierceHole1, PierceHole2, and PierceSlot0). Piloting holes are always arranged at the first station of a progressive die, so Set 1 is assigned to Station1 and Set 2 is assigned to Station 2. For S-OAG2, which has 2 vertices and 0 edges, all the operations will be clustered in the same set. All the operations (NotchScrap1, NotchScrap2) in S-OAG2 are assigned to Station 3. S-OAG3 and S-OAG4, contain only one operation each, UBendBend2_Bend3 and UBendBend0_Bend1 respectively. So UBendBend2_Bend3 and UBendBend0_Bend1 will be performed in Station4 and Station 5 in that order. For S-OAG5, all three operations (PierceHole3, PierceHole4, and PierceHole6) in it are assigned to Station6 as there are no precedence and closeness constraints among them. For the same reason, NotchScrap5 and NotchScrap6 are assigned to Station 7. These final operations will cut off the stamping part from the strip. The final result is an optimal operations sequence (minimum number die stations) satisfying all the precedence constraints and closeness constraints among the constituent stamping operations. The resulting 3-D strip layout is shown in Figure 4.6.

Figure 4.6 3-D strip layout for the example part.

4.5 Proof of correctness and algorithm complexity

This section presents a proof of the correctness of the approach and discusses the computational complexity of the algorithms. The correctness is established in Claims 4.1–4.3.

Claim 4.1 *To obtain a correct operations sequence, the precedence graph must be acyclic.*

Proof: If a cycle exists in the precedence graph, there will be at least one operation that has to be performed before and after another operation simultaneously according to the established precedence constraints. This is impossible. The acyclic check prevents this by informing the process planner to modify the rule database and/or the design-to-operation feature assignment.

Claim 4.2 *The number of S-OAGs obtained using MTS algorithm is the lower bound of die stations.*

Proof: The minimum number of die stations can not be smaller than the number of S-OAGs obtained using MTS algorithm unless the precedence constraints are violated.

Claim 4.3 *The chromatic number obtained using a graph colouring algorithm is the lower bound of die stations for each S-OAG.*

Proof: The chromatic number obtained using a graph colouring algorithm is the minimum number of colours needed to colour the corresponding S-OAG. So the chromatic number obtained using a graph colouring algorithm is the lower bound of die stations for each S-OAG. Otherwise, the adjacency constraints will be violated.

Claim 4.4 *The MTS algorithm runs in $O(n^2)$ time using $O(n^2)$ space, where n is the number of nodes (operations) in the precedence graph.*

Proof: Let $G_p = (V, A)$ be a precedence graph with n nodes. An adjacency matrix is employed to represent the precedence graph in the computer, which needs $O(n^2)$ space. The check of a node's in-degree (whether zero or not) is determined by searching for -1 in the corresponding row of the node. $O((n-m)^2)$ time is needed to finish each round of running MTS, where m is the number of nodes that have been determined in previous rounds of running (refer to page 54).

Turner (1988) showed the DSATUR algorithm could be efficiently implemented in time $O(m \log n)$. Spinrad and Vijayan (1985) constructed a family of 3-chromatic graphs for which the number of colors used by an implementation of DSATUR is $O(n)$. The implementation of DSATUR in this work follows Turner (1988), and therefore is estimated to run in time $O(km \log n)$, where k is the number of S-OAGs to be sorted.

4.6 Summary

This chapter presented an efficient algorithm for automatic determination of the stamping operation sequence. Operation relations were represented by two graphs: an operation precedence graph and an adjacency graph. The first represents precedence constraints defined by best manufacturing practice. The second represents geometric infeasibility relations between operations. A modified topological sort algorithm partitions the precedence graph into ordered independent sets, which gives the minimum number of die stations based on precedence constraints. A graph colouring algorithm was employed to obtain the fully ordered operation clusters. A case study was presented to illustrate the approach. At the end, proof of correctness and algorithm complexity were discussed. The approach has been implemented in C++ and is fully integrated with SolidWorks CAD system. The implementation of the proposed approach will be discussed in the next chapter.

CHAPTER 5 SYTEM IMPLEMENTATION

5.1 Introduction

Previous chapters have proposed a graph based operations sequencing approach to support strip layout design automation. In this chapter, implementation of the proposed approach will be discussed. The approach has been implemented in a prototype system (AutoSOS - Automatic Stamping Operation Sequencing), which is written in C++ and is fully integrated with the SolidWorks CAD system.

All computer-based design packages require a suitable modeling system, the choice of which is dictated by matching a modeling system's capabilities with the type of information that is needed. The following criteria have been used to select the modeling system:

- 1) the method of representation must be general enough to be able to represent all stamping designs;
- 2) the burden of inputting a design into the modeler and flagging key data must be minimized, as user burden is a chief factor in the package's eventual acceptance by product design engineers;
- 3) all the required data for analysis must be obtainable from the design model.

Bearing these requirements in mind, SolidWorks was chosen as the CAD modeling system. SolidWorks is a 3D Solid Modelling CAD package. However, there is a little more to it than that; in the words of its creators it is a "feature-based, parametric design tool".

Feature based – in engineering terms an assembly is made up of a number of parts. Similarly, in SolidWorks, a part is made up of a number of features. A feature is a discrete element such as a hole, a chamfer, or a fillet. Features are either sketched (as a 2D outline which is then moved through space to produce the desired 3D result) or applied (directly on to the model via a specific software function). Sketched features tend to be major items of geometry such as holes, bosses and cuts. Applied features are typically secondary items such as chamfers and fillets.

Parametric – with a parametric model the dimensions drive the geometry. This is quite the opposite of a 2D CAD system, where dimensioning is optional. With SolidWorks, dimensions are present and driving the geometry at all times. Consequently they can be revisited and adjusted to alter the model, this can be achieved within a parametric system that comes into play when relationships are built into a model or assembly.

Using its COM interface, SolidWorks exposes its objects, e.g., feature objects, to the outside world, thus enabling users and other applications to access the functionality of the SolidWorks CAD system. The SolidWorks API contains hundreds of functions that can be called from C++, which provides programmers with direct access to the SolidWorks user functions, e.g., to retrieve a related feature object through the feature relation data of an existing feature.

5.2 System architecture

Figure 5.1 shows the main modules that make up the AutoSOS system. SolidWorks provides representation and extraction support of the geometrical and topological information from stamped metal parts, the intermediate flat pattern and blank layout, and

the resulting strip layout. AutoSOS extracts sheet metal feature information from SolidWorks via SolidWorks API. Extracted sheet metal features are mapped to stamping features, which are then mapped to stamping operations interactively. The precedence and cluster relationships between operation pairs are identified from the production rules that are written in *CLIPS* (C Language Integrated Production System). The rules applied to create stamping features from sheet metal features are also written in *CLIPS*. *CLIPS* is an object-oriented expert system shell, which is interfaced with SolidWorks through C++.

Figure 5. 1 System architecture of AutoSOS

In the field of software engineering, the benefits of an object-oriented approach to software design are already well known, and object-oriented modeling (OOM) (Rumbaugh *et al.* 1991) is becoming more and more widely used. One major advantage of OOM in developing software architecture is its support to software modularity, reusability and scalability. In Figure 5.2, the graphical notation of the unified modeling language (UML) is used to represent the classes of objects and relationships between them.

Figure 5. 2 UML class diagram of AutoSOS

AutoSOS is structured as a hierarchy of abstraction levels, which represent different aspects or stages of the solution process. Each level contains various planning objects

such as sheet metal features, stamping features, stamping operations, stamping stations, and operation sequences.

Referring to Figure 5.2, an operation sequence is represented by an instance of *OperationSequence* class, while a *OperationSequence* object may be associated with multiple instances of *StampingStation* class. This arises from the observation that a progressive die has a series of die stations. In each die station, it is possible that several stamping operations are performed. So a *StampingStation* object is associated with multiple *StampingOperation* objects. It should be noted that sometimes there maybe no operation performed at a certain station. This station will be an idle station, which is used to strengthen the die block. Operation precedence and operation adjacency graphs are represented by OPG and OAG classes. Association class *FeatureOperationMapping* is used to represent the mapping relationship between stamping features and stamping operations. Similarly, association class *FeatureFeatureMapping* is used to represent the mapping relationship between sheet metal features and stamping features.

5.3 Objected-oriented classification of sheet metal features

In order to keep the memory requirement low, reduce the computational effort, and facilitate maintenance of feature validity, an object-oriented classification of sheet metal features is implemented, as shown in Figure 5.3. The approach was specifically developed within the domain of sheet metal parts. Each type of sheet metal features is defined in terms of the generic definition called “class”. The feature type *SheetMetalFeature* can be represented as the top-most generic class object, while lower level of feature classes (e.g., *PrimaryFeature*) can be represented as the child class objects of the generic class *SheetMetalFeature*. Lower level of feature classes can inherit the attributes from the higher level of feature class, and can also add specific attributes

pertinent to their own definitions. The classification inheritance property allows a more compact and structural information representation.

Primary features, such as flat, drawing, etc., are sheet metal features that form the major shape of a stamping part, while add-on features are those features that are used to modify the part shape for various functional purposes. Add-on features consist of hole, cutout, slot, emboss, etc. Connective features, such as bend, jog, etc. are those features that are used to connect two primary features.

Figure 5. 3 Object-oriented classification of sheet metal features

Sheet metal feature classes are the basic operation unit in a feature-based progressive die design system. Some important classes are given below.

```
class SheetMetalFeature : public CObject  
  
{  
  
private:  
  
    CString FeatureName;    //Feature name  
  
    CString FeatureID;      //Feature ID  
  
    ...  
  
public:  
  
    CString GetFeatureName();    //Get feature name  
  
    CString GetFeatureID();      //Get feature ID  
  
    ...  
  
}
```

```
class PrimaryFeature : public SheetMetalFeature  
  
{  
  
private:  
  
    CPtrList AddonList;        //List of addon feature pointers exist on this  
                                primary feature
```


CHAPTER 5 SYSTEM IMPLEMENTATION

```

        CPtrList ConnectiveList;           //List of connective feature pointers
                                           connected with this primary feature

        ...

public:

        CPtrList GetAddonList();           //Get the list of addon feature pointers
                                           exist on this primary feature

        CPtrlist GetConnectiveList();      //Get the list of addon feature pointers
                                           exist on this primary feature

        ...

}

```

```

class Flat : public PrimaryFeature

{

private:

        BOOL IsRoot;                       //TRUE if it is the root feature;

        ...

Public:

        BOOL IsRootFeature();              //Determine whether it is the root
                                           feature

```

```

void SetRootFeature();           //Set this flat as the root feature

...

}

```

The class declaration of other concrete planning objects, such as stamping features and stamping operations are similar to sheet metal features and is omitted here.

5.4 Knowledge representation using CLIPS

1. Mapping knowledge

It should be noted that the creation of stamping features from sheet metal features is achieved as follows. First, a flat feature is treated as a root feature by setting its “IsRoot” attribute to “TRUE”. Second, piloting holes are introduced to register the strip in the stamping process. Piloting holes may be created by selecting existing holes if they are suitable. If there are no suitable existing holes, new piloting holes will be created and located in the scrap section of the strip. Third, flat features are mapped to scrap features on the strip. These features form the external profile of the stamping part. This process is usually called scrap design. Finally, other stamping features are created automatically by selecting sheet metal features of the part.

Heuristic rules are used to reason out a set of stamping feature objects from a set of sheet metal feature objects. In the following, a piloting hole will be used as an example to illustrate the way mapping knowledge is represented.

Rule *Piloting101*: If a hole can be used as a pilot hole, then it should be circular in shape, should not have very low tolerance in diameter, and should be greater than 3 times the thickness of the stamped part. (Below is its corresponding source code in *CLIPS*.)

```
(defrule Piloting101                                     Line 1

(declare (salience 30))                                  Line 2

?Hole <- (Hole (Type THROUGH) (Shape CIRCULAR) (Piloting NO)

(Tolerance-range ?Tol-r &(> ?Tol-r ?*Tolerance-threshold*))

(Diameter ?Dia &(> ?Dia (* 3 ?*Part-thickness*))))      Line 3

=>

(modify ?Hole (Piloting CANDIDATE))                      Line 4

)
```

Line 1 is the rule name. In Line 2, a salience of 30 is declared in this rule. The property of salience is used to set the order of rule execution. The rule of higher salience will be executed before the rule of lower salience. Line 3 denotes the left-hand side (hypothesis) of the rule, indicating that a desired hole feature constrained in shape, diameter and tolerance range as above, is needed to fire the rule. Line 4 denotes the right-hand side (conclusion) of the rule, indicating that the hole feature is then selected as a piloting candidate.

Rule *Piloting102*: If a hole is selected for piloting purpose, than it should be stamped in a single step of piercing operation. (For conciseness, the *CLIPS* source codes of this rule and other rules introduced later are omitted in this chapter.)

Rule *Piloting103*: If no suitable hole exists for piloting purposes, then two pilot holes should be created in the scrap sections and on the opposite sides of the stamped part.

2. Precedence knowledge

Producing a good strip layout efficiently is an art. For a given stamping part, the stamping operations cannot be performed in any arbitrary order. Geometric and technological constraints will require that certain operations be performed before or after other operations.

Chang (1990) uses heuristic techniques to determine precedence constraints among features. A number of rules based on machining practices have been defined and are used to determine precedence constraints among pairs of features. The Machinist system (Hayes and Wright 1989) is capable of handling precedences that arise because of setup considerations. In this system, precedences are generated by examining the setup interactions among features. If machining of a feature destroys the precondition for clamping required for machining of another feature, then these two features interact and a precedence constraint exists. Gupta and Nau (1995) provides a systematic method of finding precedence constraints that considers tolerances, feature accessibility, machining practices and machining time. Precedence constraints are also important in generating and evaluating alternative assembly sequences (Swaminathan *et al.* 1998, Henrioud *et al.* 2003, Lai and Huang 2004).

The most common geometric and technological constraints have been implemented, which have been extracted from existing references. The constraints are expressed as heuristic rules, which are divided into the following categories.

(1) Rules of accessibility

Rule *Precedence101*: IF there are bend features, THEN the notching operations that remove surrounding material will be arranged first. It is because the bend feature can not be formed unless the surrounding material has been removed.

Rule *Precedence102*: IF a feature needs several operations to form it, THEN the sequence of these operations will be from rough to finishing operations. For example, a hole with high tolerance requirements may be formed by a pre-piercing operation followed by a trimming operation. To obtain the required tolerance, the sequence of these two operations can not be changed.

(2) Rules of location accuracy

Rule *Precedence103*: IF there are pilot holes, THEN they will be punched first. To facilitate the location of the strip, the piercing operation that punches piloting holes should be arranged at the first station so that they can be used throughout the remainder of the operations to ensure that the part will be produced according to the specifications.

(3) Rules of datum

Rule *Precedence104*: IF a feature is referenced from a datum feature, THEN the datum feature is punched or formed first. This rule comes into play when stamping one feature removes the datum required for another. Hence, if there are datum features, stamp them first.

(4) Rules of profile accuracy

Rule *Precedence105*: IF large and small holes are located close to each other, THEN the large hole should be cut first. This is because the operation piercing large

holes will distort the small hole features if the small hole features are arranged at earlier stations.

Rule *Precedence106*: IF a hole is located near the bending line of a bend feature, THEN the hole should be cut after bending.

Rule *Precedence107*: When embosses exist, if possible they are embossed in the first station to prevent deformation of other stamping features.

(5) Rules of cost

Rule *Precedence108*: IF possible, holes should be cut when the part is flat. This is because, after bending or drawing, expensive cams might be required.

(6) Rules of closeness

Rule *Precedence109*: IF the distance between two features is smaller than a predefined minimum distance, THEN these two features can not be arranged at the same station.

(7) Rules of weak die

Rule *Precedence110*: If holes (except pilot holes) and scraps (that don't accommodate indirect pilot holes) can not be stamped simultaneously in the same station, the process sequence is as follows: piercing operations precede notching operations.

Rule *Precedence111*: If holes or scraps to be pierced are close to each other, then the adjacent holes or scraps should be shifted to the next die stations to prevent weak die problem.

(8) Rules of precision

Rule *Precedence112*: IF holes to be pierced are not close to each other, and distance among them has high precision requirement, THEN the holes should be pierced simultaneously in a same die station.

(9) Rules of parting

Rule *Precedence113*: IF possible, the cut-off operations on scraps that accommodate pilot holes should be staged in the final station.

Rule *Precedence114*: IF a blanking operation is selected to cut the part off from the strip, THEN it will be assigned to the final station.

The rules listed above depict the precedence constraints that need to be considered when designing an operation sequence. Other more special constraints may occur depending on the geometry and topology of the designed sheet metal part, but those listed are the most common.

5.5 User interface design

AutoSOS is interfaced with SolidWorks. The design scenario is as follows. The product model information of the sheet metal part (to-be-manufactured) is extracted to AutoSOS system. Instances of sheet metal features are created and stamping features are automatically generated from them. The stamping features are assigned to stamping operations. The operation precedence and operation adjacency graphs are generated by using the pre-specified production rules. Then a modified topological sort algorithm and graph colouring algorithm are applied to obtain the operation sequence for the given sheet metal part. Finally, the 3-D strip layout is generated automatically and displayed in SolidWorks.

Figure 5. 4 User interface of AutoSOS

Figure 5.4 shows the user interface of AutoSOS system. It includes 11 sub-menus, *i.e.* unfold, create a strip, strip parameters setting, scrap design, features operation mapping, operation closeness analysis, operation precedence analysis, topological sorting, micro-planning using graph colouring and strip layout.

- **Unfold:** Transform the folded 3-D sheet metal part into a flat pattern. In this process, a fixed plate needs to be selected.
- **Create a strip:** Create a blank strip by defining the following parameters: strip width, blank orientation, length margin, station pitch, number of stations. By adjusting the first

four parameters, the optimal material utilization ratio could be obtained. Figure 5.5 shows the window to adjust the strip parameters.

Figure 5. 5 Create a blank strip window

- Strip parameters setting: The User can modify the strip parameters whenever he or she wants.
- Scrap design: The user interactively designs the scraps that are cut to form the profile of the sheet metal part interactively. If there is no suitable hole for piloting, new piloting holes will be created in this module as well.
- Feature operation mapping: In this sub-module, the stamping features are mapped to stamping operations. The user has the option to combine compatible features and map them to the same operation. Currently, the algorithm is limited to direct (one-to-one) and composition (many-to-one) mapping between stamping features and stamping operations. Figure 5.6 shows the mapping window.

Figure 5. 6 Feature operation mapping window

• Operation closeness analysis: The distances between operations are obtained by calculating the minimum distance between constituent feature pairs of the operations. These distances are stored as an operation adjacency matrix, used to generate an operation adjacency graph. Figure 5.7 shows the result of operation closeness analysis, the operation adjacency matrix.

Figure 5. 7 Operation adjacency matrix window

- Operation precedence analysis: The precedence constraints are generated automatically by evaluating the sheet metal part using the predefined rule base. The resulting precedence constraints are stored as an operation precedence matrix (Figure 5.8).

Figure 5. 8 Operation precedence matrix window

- Topological sorting: A modified topological sort algorithm is executed to cluster the operations into partially ordered sets.
- Micro-planning using graph colouring: A graph colouring algorithm is executed to generate the fully ordered operation sets.
- Strip layout: Automatic generation of the 3-D strip layout for the sheet metal part.

5.6 Summary

In this chapter, the work for software implementation of the proposed graph based operation stamping operations sequencing approach was expounded. Based on the system architectural structure of the proposed system architecture, a computerized, automated strip layout system was developed. The next chapter presents several design cases to show the system's effectiveness in supporting strip layout design automation.

CHAPTER 6 CASE STUDIES

6.1 Introduction

The objective of this chapter is to demonstrate the application of the algorithm for two parts: a simple part and a relatively complex part. The results shown in this chapter were obtained using a PC running at 2.4GHz with 512M RAM.

6.2 Case 1: A sheet metal bracket

Figure 6.1 shows a sheet metal bracket. There are 6 flats, 5 holes and 5 bends in this part.

Figure 6. 1 A sheet metal bracket

As explained before, the progressive die design process begins with unfolding, followed by nesting and scrap design. After that, the stamping features are mapped to

stamping operations, which will be assigned to stamping stations using the approach proposed in chapter 4. As to the part shown in Figure 6.1, it is assumed that the strip parameters set in the nesting stage are as follows: strip width is 150mm, blank orientation is 30 deg, length margin is 20mm, and station pitch is 72mm. The material utilization ratio can be determined by formula 6.1.

$$MRU = \frac{SA}{SW \times SP} \times 100\% \quad (6.1)$$

In which: MRU = the material utilization ratio.

SA = surface area of the unfolded part.

SW = strip width.

SP = station pitch.

From formula 6.1, we know the material utilization ratio will be 63.9%. The strip parameters setting dialog is shown in Figure 6.2.

Figure 6. 2 Parameters setting dialog in nesting process

If the user accepts the above nesting result, he may click OK button to close the parameter setting dialog and begin to design the scraps and pilot holes. The finished scrap design for the part (Figure 6.1) is shown in Figure 6.3. Two pilot holes are in purple colour, while the scrap features are in red, yellow or blue colour.

Figure 6. 3 Top view of scrap design for the sheet metal bracket

The next step is to map the stamping features to stamping operations. The mapping process is shown in Figure 6.4.

Figure 6. 4 Feature operation mapping process

17 stamping features are mapped to 17 stamping operations. According to the proposed approach, two graphs – operation precedence graph and operation adjacency graph – will be generated according to precedence rules and feature proximity respectively. These two graphs are shown in Figure 6.5 and Figure 6.6. The time taken to generate the graphs was 1.0 and 1.2 seconds respectively.

Figure 6. 5 Operations precedence graph for the sheet metal bracket

Figure 6. 6 Operations adjacency graph for the sheet metal bracket

Next, the modified topological sorting algorithm is applied to generate the macro-plan base on the precedence constraints. The output is as follows:

Figure 6. 7 Macro planning result for case 1

The time taken for this algorithm was 1.3 seconds.

Then a graph colouring algorithm is applied on the obtained step lists to solve the closeness constraints. The result is as follows:

Figure 6. 8 Micro planning result for case 1

The time taken for this algorithm was 6.3 seconds.

The final 3D strip layout will be generated automatically according the above result (Figure 6.9). The time taken by SolidWorks to generate the layout was 24.7 seconds.

Figure 6. 9 Final strip layout of the sheet metal bracket

6.3 Case 2: A sheet metal electric iron part

Figure 6.10 shows a sheet metal electric iron part that is produced using a progressive die. This part is quite intricate and has 17 bend features, 18 flat features, and 4 holes. The labels Flat1 to Flat 18 denote the 18 flats in this part, which are connected by 17 bend features, *i.e.*, Bend1 to Bend17.

Figure 6. 10 A sheet metal electric iron part

Selecting the Flat1 as the root feature, the part is flattened and nested on the strip. The strip parameters are defined as follows: strip width is 100mm, blank orientation is 0 degree, length margin is 20mm, and station pitch is 140mm. The resulting material utilization ratio is 56.3%. After the strip parameters are determined, two external piloting holes are introduced to register the part in the feeding and scrap features are also

introduced to form the external profile of the part. The scrap design is shown in Figure 6.11, which has 2 piloting holes and 12 scrap features on it. The two pilot holes are shown in purple.

Figure 6. 11 Scrap design for the sheet metal electric iron component

Then, the feature-operation mapping process is followed (Figure 6.12).

Figure 6. 12 Feature-operation mapping process for the sheet metal electric iron

Now, the 35 stamping features are mapped to 35 stamping operations. The full list of stamping operations is shown in Figure 6.13.

Figure 6. 13 Operations adjacency graph for the sheet metal electric iron

According to the proposed approach, two graphs – operation precedence graph and operation adjacency graph – will be generated according to precedence rules and feature proximity respectively. These two graphs are shown in Figure 6.14 and Figure 6.15. The time taken to generate the two graphs were 6.3 and 24.7 seconds respectively.

Figure 6. 14 Operations adjacency graph for the sheet metal electric iron

Figure 6. 15 Operations precedence graph for the sheet metal electric iron

Next, the modified topological sorting algorithm is applied to generate the macro-plan base on the precedence constraints. The output is as follows:

Figure 6. 16 Macro planning result for case 2

The time taken for this algorithm was 2.4 seconds.

Then a graph colouring algorithm is applied on the obtained step lists to solve the closeness constraints. The result is as following:

Figure 6. 17 Micro planning result for case 2

The time taken for this algorithm was 6.8 seconds.

The final 3D strip layout will be generated automatically according the above result (Figure 6.18), which is a 17 stations progressive die. The time taken by SolidWorks to generate the layout was 10 minutes.

Figure 6. 18 Final 3D strip layout for the sheet metal electric iron part

6.4 Comparison between this research work and relevant previous research works

In order to demonstrate the advantage of this research work, Table 6.1 is introduced to make comparison between this research work and previously developed other systems and research works listed in table 2.1. Just some of them are selected, which are similar to this research work.

References	Stamping operations					Strip layout		Remarks
	Piercing	Notching	Blanking	Trimming	Bending	Interactive strip layout	Automatic strip layout	
Wang <i>et al.</i> (1995)					√		√	Bending
Radin <i>et al.</i> (1997)					√		√	Bending, A*(Press-brake)
Gupta <i>et al.</i> (1998)					√		√	Bending, distributed planning(Press-brake)
Inui and Terakado (1999)					√		√	Bending(Press-brake)
Duflou <i>et al.</i> (1999)					√		√	Bending, constraint solving, TSP(Press-brake)
Thanapandi <i>et al.</i> (2001), Thanapandi <i>et al.</i> (2002)					√		√	Bending(Press-brake)
Markus <i>et al.</i> (2002)					√		√	Bending(Press-brake)
Li <i>et al.</i> (2002)					√		√	Bending, case based reasoning
Nee and Foong (1992)			√			√		Blanking
Ngoi and Chua (1994)	√				√	√		Piercing, blanking, bending and trimming
Lin and Deng (2001)	√		√			√		Torque balancing
This research work	√	√	√		√	√	√	

Table 6. 1 Comparison between this research work and relevant previous research works

From Table 6.1, it is obvious that the research has been carried out mainly addresses the issue of sequencing of bending operations. Although some efforts have

been put on global stamping operation sequencing, their strip layout generation is mostly in interactive nature.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This work examined the problem of process planning for sheet metal parts formed by progressive dies. The aim of this research was, given a sheet metal component, to develop a novel, scientific strip layout approach that is capable of generating a stamping sequence automatically. Automation of strip layout would promote die design standardization and provide aid in the training of new process planning personnel. This also would assure the continuity of the knowledge involved and lead to significant increases in productivity.

The proposed approach incorporates rules and heuristics used by human process planning experts and synthesizes the reasoning process using graph theoretic algorithms. Two types of constraints between stamping operations were identified in this work: cluster and precedence constraints. Operations have cluster constraints should be performed simultaneously at the same station. While a precedence constraint implies that an operation or operation cluster must be performed before another operation or cluster. The proposed approach first generates the stamping features from the sheet metal features consisting of the CAD model of stamping part to-be-manufactured. In this process, the blank orientation is determined using nesting algorithm to obtain the optimal material utilization. The sheet metal features in the CAD part model are mapped to stamping features in a strip model. Auxiliary stamping features (pilot features and scrap features) are devised by the process planner. Then, stamping features are mapped to stamping operations, followed by generating the operation precedence and adjacency graphs. Verify the operation precedence graph is acyclic using a colored DFS. Based on the

operation precedence graph, a modified topological sort algorithm is applied to cluster the operations into partially ordered sets. Finally, a graph colouring algorithm is applied on the operation adjacency graph which formed by obtained partially ordered operation sets.

The approach has been implemented in a prototype system (AutoSOS - Automatic Stamping Operation Sequencing), which is written in C++ and is fully integrated with the SolidWorks CAD system. This system can be used in either of two ways: as an automated process planning system, *i.e.*, a tool by someone who is not an expert but requires the use of expert knowledge; or as a checking system, *i.e.*, an assistant to an expert in the field.

If it is used as an automated process planning system, the result is a sequenced cluster of operations that meet the constraints of both graphs. It can output 2 numbers, the minimum number of stations based on topological sort (the ideal minimum) and the feasible number after colouring (the feasible minimum). The 3D strip layout is automatically generated.

On the other hand, if it is used as a checking system, it can output 3 numbers: the actual number of stations set by the planner, the feasible number and the ideal number. The numbers are useful to help the process planner see the impact of constraints and improve the process plan accordingly.

7.2 Future work

As explained in the previous chapters, laying out the strip is the most critical activity in the process of progressive die design and manufacturing. From the strip layout automation, it is possible to develop an integrated progressive die design system, which can cover the whole product life cycle of sheet metal parts manufactured by progressive dies including sheet metal product modeling, unfolding, nesting, strip layout, and die structure design. The recommendations for future work to develop such an integrated progressive die design system are:

- (1) Increase the number of features handled by the system.

The current system can only handle the following features: wall, bend, jog, hole, slot, emboss. It is estimated that up to 60 percent of metal stampings used by the electronic industry can be modeled using these features. However, there are many other features found in stamping parts, such as, lancing, engraving, and indenting. These features must be incorporated in the system before it can be used by the industry. To expand the capability of the system to handle the new features, the following should be done: i) develop the corresponding sheet metal feature class to represent the new feature; ii) define the new process for the new feature if needed; iii) examine the tooling design process involved in the stamping of the new feature and develop heuristics or precedence constraints that can be coded into the computer to mimic this tooling design process.

- (2) Integrate AutoSOS with the OOBM (object-oriented blackboard modelling) system developed by our research group (Tor *et al.* 2005).

The main difficulty of existing knowledge-based systems for stamping process planning is the lack of a proper architecture for organizing heterogeneous knowledge sources (KSs) in a cooperative decision making environment. Cooperative decision making for knowledge-based stamping process planning involves a variety of KSs such as unfolding knowledge to produce a flat pattern, nesting knowledge to produce a blank layout, various types of planning knowledge for different stamping operations like piloting, piercing, notching, cut-off, blanking, bending, etc., and staging knowledge to sequence the stamping operations. These KSs may be expressed in different representation schemes such as procedures, algorithms, reasoning methodologies, frames and rules. This justifies the use of a blackboard architecture because it can manage heterogeneous KSs effectively. The KSs interact through the blackboard to develop a solution incrementally. The operations sequencing approach developed in this research can be incorporated into the OOBM system and works as the staging knowledge to sequence the stamping operations.

- (3) Develop a concurrent engineering environment for sheet metal product design, progressive die design and manufacturing.

Usually, sheet metal product design, progressive die design and manufacturing processes are carried out in a series of stages by using different software packages and separately and sequentially in different branches of the company, even different companies. With this traditional design and manufacture method, sheet metal products are frequently designed without a systematic consideration of downstream product development requirements, such as process planning, manufacturability, production scheduling and manufacturing optimization. In addition, the feedbacks from these downstream processes to the product designer can only be made after the

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

product is designed or even manufactured. This can often result in a lot of expensive and time consuming reworks. Consequently, it affects the quality, cost and delivery time of the product. A concurrent engineering environment will fill the gap between sheet metal products design and progressive die design as well as its manufacturing.

REFERENCES

- Adamowicz, M. and Albano, A. (1976). Nesting of two-dimensional shapes in rectangular modules. Computer-Aided Design, 8(1), pp. 27–33.
- Aomura, S. and Koguchi, A. (2002). Optimized bending sequences of sheet metal bending by robot. Robotics and Computer-Integrated Manufacturing, 18(1), pp. 29-39.
- Bergstrom, K., Kivivuori, S., Osenius, S. and Korhonen, A. (1988). Computer aided design of progressive die. Modelling of Metal Forming Processes, Kluwer Academic Publishers, New York, pp. 155-162.
- Blum, A. (1994). New approximation algorithm for graph colouring. Journal of Association of Computing Machinery, 41, pp. 470-516.
- Borenstein, D. (2000). A directed acyclic graph representation of routing manufacturing flexibility. European Journal of Operational Research, 127(1), pp. 78-93.
- Brelaz, D. (1979). New methods to color the vertices of a Graph. Communications of the ACM, 22, pp. 251-256.
- Chang, T. C. (1990). Expert process planning for manufacturing. Addison-Wesley Publ., Massachusetts, USA.
- Chen, C. L . P. and LeClair, S. R. (1994). Integration of design and manufacturing solving set-up generation and feature sequencing using an unsupervised-learning approach. Computer-Aided Design, 26, pp. 59-75.
- Chen, Y. M., Miller, R. A. and Vemuri, K. R. (1992). A framework for feature based part modeling. Computers in Engineering, ASME, 1, pp. 357-365.

REFERENCES

- Cheok, B. T. and Foong, K. Y. (1996). An intelligent planning aid for the design of progressive dies. Proceedings Institution of Mechanical Engineers, Part B, Journal of Engineering Manufacture, 210, pp. 25-35.
- Cheok, B. T. and Nee, A. Y. C. (1998). Configuration of progressive dies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 12, pp. 405-418.
- Cheok, B. T. and Nee, A. Y. C. (1998). Trends and developments in the automation of design and manufacture of tools for metal stampings. Journal of Materials Processing Technology, 75(1-3), pp. 240-252.
- Cheok, B. T., Foong, K. Y., Nee, A. Y. C. And Teng, C. H. (1994). Some aspects of a knowledge-based approach for automating progressive metal stamping die design. Computers in Industry, 24, pp. 81-95.
- Chi, X. Y., Ma, H. J., Zhao, Z. and Peng, Y. H. (2001). Research on hybrid expert system application to blanking technology. Journal of Materials Processing Technology, 116, pp. 95-100.
- Choi, J. C. and Kim, C. (2001). Compact and practical CAD/CAM system for the blanking or piercing of irregular shaped-sheet metal products for progressive working." Journal of Materials Processing Technology, 110(1), pp. 36-46.
- Choi, J. C., Kim, B. M. and Kim, C. (1999). An automated progressive process planning and die design and working system for blanking or piercing and bending of a sheet metal product. International Journal of Advanced Manufacturing Technology, 15(7), pp. 485-497.

REFERENCES

Choi, J. C., Kim, B. M. Kim, S. W. and Kim, C. H. (1996). Computer-aided forging-die design for two-dimensional cross-sectional components. Journal of Materials Processing Technology, 62, pp. 124-132.

Choi, J. C., Kim, B. M., Cho, H. Y. et al. (1998). An integrated CAD system for the blanking of irregular-shaped sheet metal products. Journal of Materials Processing Technology, 83, pp. 84-97.

Choi, J. C., Kim, C. and Kim, J. H. (2001). An integrated CAD/CAM system for manufacturing lead frame for semiconductors. International Journal of Advanced Manufacturing Technology, 17, pp. 840-849.

Choi, J. C., Kim, C., Choi, Y., Kim, J. H. and Park, J. H. (2000). An integrated design and CAPP system for deep drawing or blanking products. International Journal of Advanced Manufacturing Technology, 16, pp. 803-813.

Chow, W. W. (1979). Nesting of a single shape on a strip. International Journal of Production Research, 17(4), pp. 305-322.

Christofides, N. (1971). An algorithm for the chromatic number of a graph. The Computer Journal, 14, pp. 38-39.

Dagli, C. H. and Tatoglu, M. Y. (1987). An approach to two-dimensional cutting stock problems. International Journal of Production Research, 25(2), pp. 175-190.

De Vin, L. J., J. de Vries, et al. (1994). The generation of bending sequences in a CAPP system for sheet-metal components. Journal of Materials Processing Technology, 41(3): 331-9.

REFERENCES

- Dori, D. and Ben-Bassat, M. (1984). Efficient nesting of congruent convex figures. Communications of the ACM, 27(3), pp. 228–235.
- Duffey, M. R. and Sun, Q. (1991). Knowledge-based design of progressive stamping dies. Journal of Materials Processing Technology, 28(1-2), pp. 221-227.
- Duflou, J. R. (1997). Design verification for bent sheet metal parts: a graphs approach. International Transactions in Operational Research, 4(1), pp. 67-73.
- Duflou, J. R., Van Oudheusden, D., Kruth, J.-P., Cattrysse, D. (1999). Methods for the sequencing of sheet metal bending operations. International Journal of Production Research, 37(14), pp. 3185-3202.
- Evans, J. R. and Minieka, E. (1992). Optimization algorithms for networks and graphs. Second edition, Marcel Dekker, New York.
- Fogg, B. and Jaimeson, B. (1975). The influencing factors in optimizing press tool die layouts and a solution using computer aids. CIRP Annals, 24, pp. 429-434.
- Garey, M. R., and Johnson, D. S. (1979). Computers and Intractability. Freeman Publishing. San Francisco.
- Gupta, S. K. and Nau, D. S. (1995). Systematic approach to analysing the manufacturability of machined parts. Computer-Aided Design, 27(5), pp. 323-342.
- Gupta, S. K., D. A. Bourne, et al. (1998). Automated process planning for sheet metal bending operations. Journal of Manufacturing Systems, 17(5): 338-60.
- Hayes, C. C. and Wright, P. (1989). Automatic process planning: using feature interaction to guide search. Journal of Manufacturing Systems, 8(1), pp. 1-15.

REFERENCES

-
- Henrioud, J. M., Relange, L., *et al.* (2003). Assembly sequences, assembly constraints, precedence graphs. ISATP'03: 5th IEEE International Symposium on Assembly and Task Planning, Besancon, France, IEEE.
- Hoffmann, M., Geibler, U. and Geiger, M. (1992). Computer aided generation of bending sequences for die-bending machines. Journal of Materials Processing Technology, 30(1), pp. 1-12.
- Huang, K., Ismail, H. S. and Hon, K. K. B. (1996). Automated design of progressive dies. Proceedings of Institution of Mechanical Engineers. Part B., Journal of Engineering Manufacture, 210 (B4), pp. 367-376.
- Inui, M. and Terakado, H. (1999). Fast bending sequence planning for progressive press-working. Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning (ISATP'99), pp. 21-24.
- Irani, S. A., Koo, H. Y. and Raman, S. (1995). Feature-based operation sequence generation in CAPP. International Journal of Production Research, 33(1), pp. 17- 39.
- Ismail, H. S. and Hon, K. K. B. (1992). New approaches for the nesting of two-dimensional shapes for press tool design. International Journal of Production Research, 30(4), pp. 825–837.
- Ismail, H. S., Chen, S. T. and Hon, K. K. B. (1996). Feature-based design of progressive press tools. International Journal of Machine Tools and Manufacture, 36 (3), pp. 367-378.
- Ismail, H.S., Hon, K. K. B., Huang, K. (1995). Intelligent object-oriented approach to the design and assembly of press tools. CIRP Annals - Manufacturing Technology, 44(1), pp 91-96.

REFERENCES

Janczewski, R., Kubale, M. *et al.* (2001). The smallest hard-to-color graph for algorithm DSATUR. Discrete Mathematics, 236(1-3), pp. 151-165.

Kamil, A. (2003). www.cs.berkeley.edu/~kamil/sp03/041403.pdf.

Kim, C. H., Park, J. H. *et al.* (2004). Expert system for process planning of pressure vessel fabrication by deep drawing and ironing. Journal of Materials Processing Technology, 155-156(1-3), pp. 1465-1473.

Kim, C., Kim, B. M. *et al.* (2001). Development of an integrated computer-aided process planning system for press working products. Journal of Materials Processing Technology, 111(1-3): 188-192.

Kim, C., Park, Y. S. *et al.* (2002). A study on the development of computer-aided process planning system for electric product with bending and piercing operations. Journal of Materials Processing Technology, 130-131, pp. 626-631.

Kim, M. S., Choi, J. C. *et al.* (2004). The development of an automated process planning and die design system for multi former-bolt products. International Journal of Advanced Manufacturing Technology, 24(3-4), pp. 288-297.

Lai, H.-Y. and Huang C.-T. (2004). A systematic approach for automatic assembly sequence plan generation. International Journal of Advanced Manufacturing Technology 24(9-10), pp.752-763.

Li, C. Y., Li, J. J., Wen, J. Y. and Xiao, X. Z. (2001). HPRODIE: using feature modeling and feature mapping to speed up progressive die design. International Journal of Production Research, 39 (18), pp. 4133-4151.

-
- Li, J. Y., A. Y. C. Nee, et al. (2002). Integrated feature-based modelling and process planning of bending operations in progressive die design. International Journal of Advanced Manufacturing Technology, 20(12), pp. 883-895.
- Lin Z-C. and Deng, C-H. (2001b). Application of petri net in the planning of a shearing cut and bending progressive die workstation. International Journal of Materials and Product Technology, 10, pp. 579-591.
- Lin, A. C., Lin, S-Y, Diganta, D., Lu, W., F. (1998). An integrated approach to determining the sequence of machining operations for prismatic parts with interacting features. Journal of Materials Processing Technology, 73(1-3), pp. 234-250.
- Lin, Z.-C. and G.-J. Peing (1994). Investigation of an expert system for sheet-metal bending design. Journal of Materials Processing Technology, 43(2-4), pp. 165-176.
- Lin, Z-C. and Chang, Y-C. (1997). Sash sheet metal development and the application of a back-propagation neural network in the calculation of the developed length. Journal of Materials Processing Technology, 69, pp. 95-105.
- Lin, Z-C. and Deng, C-H. (2001a). Analysis of a torque equilibrium model and the optimal strip working sequence for a shearing-cut and bending progressive die. Journal of Materials Processing Technology, 115(3), pp. 302-312.
- Lin, Z-C. and Hsu, C-Y. (1996). An investigation of an expert system for shearing cut progressive die design. International Journal of Advanced Manufacturing Technology, 11, pp. 1-11.
- Markus, A., J. Vancza, et al. (2002). Constraint-based process planning in sheet metal bending. CIRP Annals - Manufacturing Technology, 51(1), pp. 425-428.

-
- Mehrotra, A. and Trick, M. A. (1996). A column generation approach for graph colouring. INFORMS Journal on Computing, 8, pp. 344-354.
- Murakami, H., Shirai, K., Yamada, O. and Isoda, K. (1980). A CAD system for progressive die. In Proceedings of 21th Machine Tool Design and Research Conference, Macmillan, London, pp. 587-591.
- Nakaham, S., Toshio, K., Tamura, K., et al. (1978). Computer aided progressive die design. In Proceedings of 19th Machine Tool Design and Research Conference, Macmillan, London, pp. 1-12.
- Nee, A. Y. C. (1984). A heuristic algorithm for optimum layout of metal stamping blanks, Annals CIRP, 33, pp. 317-320.
- Nee, A. Y. C. (1989). PC-based computer aids in sheet-metal working. Journal of Mechanical Working Technology, 19, pp. 11-21.
- Nee, A. Y. C. and Foong, K. Y. (1992). Some considerations in the design and automatic staging of progressive dies. International Journal of Materials Processing Technology, 29, pp. 147-158.
- Ngoi, B. K. A. and Chua C. K. (1994). A knowledge based system for strip layout design. Computer in industry, 25(1), pp. 31-44.
- Ong, S. K., De Vin, L. J., et al. (1997). Fuzzy set theory applied to bend sequencing for sheet metal bending. Journal of Materials Processing Technology, 69, pp. 29-36.
- Paquin, J. P., and Crowley, R. E. (1987). Die design fundamentals. Industrial Press Inc., Second Edition.

Park, S. B. (1999). Expert system of progressive die design for electron gun grid parts. Journal of Materials Processing Technology, 88(1), pp. 216-221.

Park, S. B., Choi, Y. et al. (1998). Study of a computer-aided process design system for axisymmetric deep-drawing products. Journal of Materials Processing Technology, 75(1-3), pp. 17-26.

Prabhu, P., Elhence, S., Wang, H. and Wysk, R. A. (1990). An operations network generator for computer aided process planning. Journal of Manufacturing Systems, 9(4), 283-291.

Prasad, K. S. R. K. and Selvaraj, P. (2004). Practical methods of computer-aided flat pattern development for sheet metal components. International Journal of Production Research, 42(15), pp. 3011-3039.

Prasad, Y. K. D. V. (1994). A set of heuristic algorithms for optimal nesting of two-dimensional irregularly shaped sheet-metal blanks. Computers in Industry, 24, pp. 55-70.

Prasad, Y. K. D. V. and Somasundaram, S. (1992). CADDs: an automated die design system for sheet-metal blanking. Computing & Control Engineering Journal, 3(4), pp. 185-191.

Qu, W. and Sanders, J. L. (1987). A nesting algorithm for irregular parts and factors affecting trim losses. International Journal of Production Research, 25(3), pp. 381-397.

Radin, B., M. Shpitaini, et al. (1997). Two-stage algorithm for determination of the bending sequence in sheet metal products. Journal of Mechanical Design, Transactions of the ASME, 119(2): 259-266.

REFERENCES

-
- Radin, B., M. Shpitaini, et al. (1997). Two-stage algorithm for determination of the bending sequence in sheet metal products. Journal of Mechanical Design, Transactions of the ASME, 119(2), pp. 259-266.
- See Toh, K. H., Loh, H. T., Nee, A. Y. C. and Lee, K. S. (1995). A feature-based flat pattern development system for sheet metal parts. Journal of Material Processing Technology, 48, pp. 89-95.
- Shpitalni, M. and Saddan, D. (1994). Automatic determination of bending sequence in sheet metal products. CIRP Annals, 43(1), pp. 23-26.
- Shunmugam, M. S. and Kannan, T. R. (2002). Automatic flat pattern development of sheet metal components from orthographic projections. International Journal of Machine Tools & Manufacture, 42, pp. 1415-1425.
- Singh, R. and Sekhon, G. S. (1996). A computerized diagraph and matrix approach for evaluation of metal stamping layouts. Journal of Materials Processing Technology, 59, pp. 285-293.
- Smith, D. A. (1990). Die design handbook. Society of Mechanical engineers. 3rd edition, pp. 2.1-2.60.
- Spinrad, J. P. and Vijayan, G. (1985). Worst case analysis of a graph coloring algorithm. Discrete Applied. Mathematics, 12, pp.89-92.
- Sundaram, R. M. (1986). Process planning and machining sequence. Computers and Industrial Engineering, 11, pp. 184-188.
- Swaminathan, A., Shaikh, S. A., et al. (1998). Design of an experience-based assembly sequence planner for mechanical assemblies. Robotica, 16, pp. 265-283.

REFERENCES

- Tang, C. H. and Rajesham, S. (1994). Computer aided nesting in sheet metal for pressworking operations involving bending. Journal of Materials Processing Technology, 44, pp. 319–326.
- Thanapandi, C. M., A. Walairacht, et al. (2001). Multi-component genetic algorithm for generating best bending sequence and tool selection in sheet metal parts. Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation, pp. 21-26.
- Thanapandi, C. M., A. Walairacht, et al. (2002). Preprocessor to improve performance of GA in determining bending process for sheet metal industry. Foundations of Intelligent Systems. 13th International Symposium, ISMIS 2002. Proceedings, pp. 27-29.
- Thulasiraman, K. and Swamy, M.N.S. (1992). Graphs and Algorithms. John Wiley, New York.
- Tolouei-Rad, M. (2003). An efficient algorithm for automatic machining sequence planning in milling operations. International Journal of Production Research, 41(17), pp. 4115-4131.
- Tor, S. B., Britton, G. A., and Zhang, W. Y. (2005). Development of an object-oriented blackboard model for stamping process planning in progressive die design. Journal of Intelligent Manufacturing, 16, pp. 499-513
- Tsen, O. (2005). <http://home.so-net.net.tw/oodtsen/wingraphviz/>
- Turner, J. S. (1988). Almost all k-colorable graphs are easy to color. Journal of Algorithms, 9, pp. 63 - 82.

REFERENCES

-
- Wang, C.-H. and Bourne, D. A. (1995). Using features and their constraints to aid process planning of sheet metal parts. Proceedings of the 1995 IEEE International Conference on Robotics and Automation, 1(3), pp. 21-27.
- Wang, F. and Chang, L. (1995). Determination of the bending sequence in progressive die design. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 209(B1), pp. 67-73.
- Wang, F. L., Xiao, X. Z. and Li J. J. (1995). On the development of intelligent CAD/CAM systems for progressive dies. SPIE, 2620, pp. 253-261.
- Xiao, X., Li, Z., Li, J., Li, Z. G., Ma, J. and Xiao, J. (1993). An intelligent approach to the strip layout of progressive die design. In Proceedings of the 4th International Conferences on CIM, Singapore, 1(2), pp. 22-28.
- Yeh, S., Kamran, M. and Nnaji, B. O. (1994). Unfolding sheet metal parts: a graph-based approach. International Journal of Production Research, 33, pp. 729-740.
- Yuster, R. (1998). Linear colouring of graphs. Discrete Mathematics, 185, pp. 293-297.
- Zhao, Z. and Peng, Y. (2002). Development of a practical blank layout optimization system for stamping die design. International Journal of Advanced Manufacturing Technology, 20, pp. 357-362.