INCREMENTAL GENERATION OF ALTERNATIVE PROCESS PLANS FOR INTEGRATED MANUFACTURING

A Thesis Presented to
The Faculty of the
Fritz J. and Dolores H. Russ
College of Engineering and Technology
Ohio University
In Partial Fulfillment
Of the Requirement for the Degree
Master of Science
By
Shridharan Thiruppalli
November 2002
I take this opportunity to acknowledge the support and guidance of everyone who helped me through the learning and challenging phase of my Masters thesis. At the outset, I would like to thank my advisor Dr. Dusan Sormaz for sharing his ingenuity, and his excellent guidance as mentor, which enabled my successful completion of the thesis. I would like to thank Dr. Judd for his advice that helped me acquire a deeper perspective of my work. Also, I would like to thank Dr. Koonce and Dr. Hale for their invaluable support and guidance. Last but not the least, I thank Dr. Dinesh Dhamija for his motivation and encouragement.

Also, I need to recognize my teammates for their cooperation and thoughts that enriched my thesis. Finally, I thank my parents Mrs. Janaki Krishnan and Mr. Thiruppalli Krishnan and my fiancée Ms. Devika Thayanban for their moral support.

I sincerely believe that each one of you played an important role during this phase, which proved to be a worthwhile and an excellent learning experience.
# Table of Contents

1 Introduction .................................................................................................................. 1

1.1 Research Objective ................................................................................................. 1

1.2 Motivation .................................................................................................................. 2

1.3 Computer-Aided Process Planning (CAPP) ............................................................. 3

1.4 Problems with Current CAPP Systems ................................................................. 5

1.5 Research Goals ......................................................................................................... 8

1.6 Research Scope ........................................................................................................ 10

1.6.1 Completeness ....................................................................................................... 10

1.6.2 Optimization ......................................................................................................... 11

1.6.3 Assumptions ......................................................................................................... 12

1.7 Outline of the Research Approach ............................................................................ 12

1.7.1 Feature Precedence Constraint Analysis ............................................................ 13

1.7.2 Incremental Process Plan Generation .................................................................. 13

1.7.3 Finding Least Cost Alternative Process Plan ..................................................... 14

1.8 Thesis Organization .................................................................................................. 14

2 Previous Work .............................................................................................................. 16
2.1 Process Plan Representation Models .................................................. 17

2.2 Process Sequencing and Alternative Process Plans......................... 19

2.3 Incremental Process Planning............................................................ 21

2.4 Process Plan Network Optimization .................................................. 22

3 Feature Precedence Network Generation ............................................. 24

3.1 Feature Precedence Network Representation ...................................... 24

3.2 Analysis of Feature Precedence Network .......................................... 29

3.2.1 Petri Net .................................................................................. 30

3.2.2 Definition of a Petri Net ............................................................. 32

3.2.3 Operation of a Petri Net ............................................................ 33

3.2.4 Feature Precedence Based Petri Net ............................................ 33

3.2.5 Examples ................................................................................. 35

4 Process Plan Network ......................................................................... 39

4.1 Process Planning ................................................................................ 39

4.2 Process Plan Network Generation .................................................... 42

4.2.1 Network Generation Algorithm .................................................. 47

4.3 Complexity Analysis of Network ...................................................... 55

4.4 Optimization of Process Plan Network .............................................. 61

4.4.1 Cost Model .............................................................................. 61
List of Figures

Figure 1: (a) Part Model, (b) Feature Precedence Network ........................................... 26
Figure 2: The use of Petri nets for modeling and analysis of a system ....................... 31
Figure 3: FP-Petri Node used in FPN Petri net ............................................................... 34
Figure 4: Validating Petri net .......................................................................................... 36
Figure 5(a to e): Step-by-step transition of validating Petri net ................................ 36
Figure 6: (a) FPN with cyclic constraint, (b) Validating Petri net ............................... 37
Figure 7: A set of process plan nodes and arc ............................................................... 42
Figure 8: Arc showing the cost ....................................................................................... 45
Figure 9: A portion of the process plan network ............................................................ 47
Figure 10: Process plan network showing all explored nodes ....................................... 53
Figure 11: Process plan network with optimal alternative process plans ................... 54
Figure 12: Process plan network for 3 features with 2 machines each showing all alternatives ................................................................. 57
Figure 13: PPN generated using space search approach ............................................... 60
Figure 14: Process planning models – a) Conventional process planning model, b) Concurrent process planning model, c) Incremental process planning model ........................................................................................................ 67
Figure 15: Cases of feature additions. (a-d) .................................................................. 68
Figure 16: (a) FPN before feature addition (b) FPN after feature addition ............. 70
Figure 17: (a) FPN with 4 features (b) FPN with 5 features ........................................ 71
Figure 18: PPN with one optimal process plan.................................................. 71
Figure 19: Incremental PPN. ............................................................................. 72
Figure 20: Cases of feature deletions (a-d). ...................................................... 73
Figure 21: FPN before (a) and after (b) feature deletion................................. 74
Figure 22: System architecture......................................................................... 77
Figure 23: Graphical user interface ................................................................. 83
Figure 24: Feature precedence network with 4 features................................. 88
Figure 25: FP-Petri net node ............................................................................ 89
Figure 26: Part of validating Petri net showing connection between FP-Petri
(nodes........................................................................................................ 90
Figure 27: FPN with cyclic precedence constraints. ....................................... 90
Figure 28: (a) A valid FPN, (b) Validating Petri net for the FPN..................... 91
Figure 29: FPN for the simple example.......................................................... 96
Figure 30: FPN at various stages of simple part design................................. 97
Figure 31: Incremental process plan generation for simple example............. 99
Figure 32: FPN for Bendix Example............................................................... 100
Figure 33: FPN at various stages of Bendix part design................................. 102
Figure 34: PPN for each step of modified Bendix example. ......................... 103
List of Tables

Table 1: Precedence constraints on features ......................................................... 29
Table 2: Parent-child representation of a FPN..................................................... 51
Table 3: Process candidates for features with costs in dollars .............................. 52
Table 4: Feature attributes - next, previous .......................................................... 87
Table 5: Parent and child feature nodes................................................................. 88
Table 6: Feature-machine matrix for simple example(cost in dollars)...................... 98
Table 7: Feature-Machine matrix for Bendix example. Cost in dollars................. 101
1 Introduction

1.1 Research Objective

The objective of this research is to develop a feature based computer-aided process planning (CAPP) system to automatically generate process plan network, incrementally, whenever part model undergoes a design change. The system has been designed to be an integral part of IMP process model: Intelligent Manufacturing Planning, currently under development at the IMP Laboratory in Ohio University.

The CAPP system has been designed to take in a feature based part description and its output would produce a detailed process plan specifying the sequence of processes along with corresponding machines needed to produce the part. The first phase of the research has focused on developing a design interface 1) capable of extracting precedence information from the feature based model and 2) organizing and validating the extracted information so that it could be used for process planning. The second phase of the research has rested on obtaining process and machine details from a knowledge base for producing each feature. In finality, the third phase of the research has focussed on developing a system that would search for an optimal/near optimal process plan from the space of feasible plans.
1.2 Motivation

Global competition has significantly amplified the need for manufacturing companies to produce high quality competitively priced products both quickly and efficiently. Within such an environment, it has become a necessity to deliver products to the market in the shortest time. Current developments in computer integrated manufacturing (CIM) focus on linking different automated activities within the enterprise, in order to efficiently utilize information and significantly reduce the production cost and time. One of the most critical links is the link between design and process planning activities. In addition, shortening the product development cycle and responding to the changed market demand are highly imperative.

The application of computers to product design and manufacturing process planning has brought changes in the traditionally separate disciplines of design and process planning. Though advancements have been made in computer-aided design (CAD) and computer-aided process planning (CAPP), there has been a lack of achievement in integrating them. One difficulty in integrating CAD/CAPP has arisen from the fact that design and manufacturing require different models of the same part.

Two major benefits can be expected from automating process planning. First, automated process planning can quickly generate efficient plans. Secondly,
automated process planning can help designers perform design for manufacturing (DFM). Thus, designers can focus more on functionality of designs than on manufacturing. In fact, designers lack knowledge of manufacturing, which often causes manufacturing problems and redesigns of the part. Therefore, similar to other design-analysis tools such as FEA, automated process planning can help designers reduce time spent on correcting manufacturing problems, and it can increase efficiency of product development.

1.3 Computer-Aided Process Planning (CAPP)

Although process planning is well known to the manufacturing engineering profession, for a clear discussion, it may be worth to state its definition:

"Process planning is the systematic determination of the detailed methods by which workpieces or parts can be manufactured economically and competitively from initial stages (raw material form) to finished stages (desired form)"[2].

Computer-aided process planning (CAPP) systems support this activity, providing the process planner with different tools to improve her/his performance. In computer-integrated manufacturing (CIM), CAPP is the link between computer-aided design (CAD) and computer-aided manufacturing (CAM).
Functions of a process planning system may include several or all of the tasks [6]:

1) Selection of manufacturing processes,
2) Sequencing of manufacturing processes,
3) Selection of tools and machines,
4) Determining set up requirements,
5) Selection/design of jigs and fixtures.

CAPP systems have traditionally been classified in two categories: variant and generative process planning [7]. The variant approach has been based on group technology (GT) concepts. GT consists of grouping parts in part families, which are geometrically and/or technologically similar. In the variant approach, a standard process plan is associated with each part family. Process planning begins with the classification of the part, and this classification indicates whether or not it belongs to a certain family. If a family is found, the family standard process plan is used as a reference for the new parts in the family.

For many years generative process planning has been defined as the approach for generation of fully automatic process plan [5]. This approach generates process plans according to the experiences and knowledge of the planners. This kind of system establishes form features or manufacturing features that represent the shapes and specifications of a work piece. Form
features are fundamental elements such as slot, pocket, hole, etc. which can be combined into forming the rough work piece design. These form features also have the standard processes. Therefore, by combining the form features, a generative CAPP can generate a process plan for a work piece.

1.4 Problems with Current CAPP Systems

As manufacturing companies increase the level of customization in their product offerings and move towards smaller lot production, they increasingly require the ability to respond quickly, accurately and competitively to customer requests for new products. This in turn requires the ability to swiftly translate standard-based product specifications into process plans.

Automated process planning systems are almost secluded from the design phase. Most CAPP systems accept part description, and plan the required machining operation to manufacture the part. CAPP systems perform the process-planning task at different levels of detail. Some process planners produce only a sequence of feasible processes for manufacturing each feature. Others consider machines and tools, and evaluate manufacturing cost, while many others do not perform cost evaluations.

One of the most common weaknesses of the computer-aided process planning system has been that they act as a separate function and do not have a
linkage with the CAD system. These CAPP systems are inapt for real-time integration with the CAD system to update changes in the design. A CAPP system, which accommodates this instantaneous integration with CAD system, is an incremental process planning system.

The systems such as GARI, TOM, PROPLAN, HI-MAPP use generative process plan approach and systems such as MULTICAPP, MICAPP and Intelli-CAPP use variant approach for their process planning [2]. Invariably, most of these systems run in a batch mode. That is, they generate process plan for a given part model. These systems are basically concerned with automation of manufacturing process planning rather than acting as a channel for design improvement. Conventional computer aided process planning systems operate in a batch mode. The input to such a process plan is just the final part design. This approach creates a huge search space, in which the various process and sequence possibilities have to be searched. When the part undergoes a design change, this process of searching for sequence possibilities is done again from the start. Some of the sequences obtained from intermediate process plan may be used for re-generating process plan in the event of the design change. This advantage of re-using the intermediate process plan has not been utilized in batch mode process planning. Since the batch mode process planning works on final part design, it cannot be used to provide feedback to the designer during design phase.
During the early design stages, the designers were never concerned about the manufacturability of the part. Rather, the design engineer's concern has been on the capability of the part to support desired load and avoid interference with the mating part. The result has been a poor engineering design, just because the designers did not have adequate knowledge about the impact of design variations on the manufacturing processes.

Thus the research has emphasized on concurrent engineering and has contributed solutions to these problems. The purpose of the research has been to develop a computer aided incremental process planning system with the capability to develop alternative process plans. Incremental refers to the fact that the manufacturing plan is updated after each feature is added. The system has been designed to be capable of generating process plan at each phase of the design process. The manufacturing cost at each design phase has been evaluated and reported to the designer. This feature would aid the designer to make an appropriate decision in selecting the least cost design alternative. Consequently, it would act as a tool in concurrent engineering, where it can be used for simultaneous design of parts and their manufacturing process plans. Thus, the design process would be an ongoing negotiation between the designer and the planner, with the planner making suggestions to the designer on how to improve the manufacturability of the part. The resulting design can be manufactured during any step of its development.
Alternative process plan has been one of the solutions for integrating the process planning with shop floor scheduling. It has been achievable because a process planning system should be able to make alternative process plans according to the machine/machine tools available in the shop floor. The alternative process plans may be used to make appropriate scheduling decisions.

As mentioned earlier, some of the process planning systems, have not been concerned about the manufacturing cost evaluation and optimization of the network. However, the focus of this research has been cost optimization of the process plan network, where the optimum manufacturing cost evaluation acts as a feedback to the designer.

1.5 Research Goals

The main objective of the thesis has been the development of process sequencing module for CAPP system. The system will be capable of generating process plan network, with alternatives, for any given part model. The process plan network consists of least cost alternatives, which will be useful in making scheduling decisions. This process plan network will work incrementally and modify all the alternative process plans in response to the design change. Further, the CAPP system will also optimize the process plan network to obtain the least cost alternative process plan.
The proposed CAPP system has been based on feature-based model for process planning. A feature-based representation has been used for describing the part. A feature, which is informally defined as a geometric form unit and has been commonly used for reasoning in the design or manufacturing activities, is the fundamental element of the part description. Process planning, like any other computer aided engineering applications, requires the definitions of product geometry as input. Among the several theoretically possible representations, the feature-based product model is the one that has been adopted. Indeed, features have several advantages in the process planning systems:

- Features are suitable input data representation for the process planning system.

- Process knowledge required for planning can be associated with features.

- Features aid geometric reasoning for various tasks of process plans.

- Downstream operations such as NC code generation can be supported.

The design methodology of the proposed implementation has been based on incremental, hierarchical planning and manufacturing cost optimization. The
incremental planning makes use of the association between the designer's sequence of feature addition and the manufacturing process sequence. Hierarchical planning involves two levels: the first level deals with validation of the precedence constraints of the features, and the second level deals with sequencing of features. Subsequently, the overall process plan with minimal machine runtime or manufacturing cost is selected. The runtime, setup changeover time, and handling time are converted in terms of cost. Thereby the goal of optimizing manufacturing costs takes care of the time factor too.

The proposed CAPP system will be able to perform process planning incrementally. The process plan will update whenever the designer modifies the design. The design changes that will be handled are addition, deletion, and modification of feature properties that might change the precedence constraint. A real-time feedback about the cost of production of the current intermediate design will also be provided to the designer. The system also produces alternate process plans that might be useful in the production scheduling.

1.6 Research Scope

1.6.1 Completeness

Process plan comprises information needed to produce the parts from given designs. The information includes manufacturing feature precedence constraints
and machine-feature details. As mentioned earlier, process planning has included tasks like selection of manufacturing processes, sequencing of manufacturing processes, selection of tools and fixtures, and determining set up requirements. In this research the main focus has been on sequencing of manufacturing processes, finding the optimal process sequence and incremental process planning. Furthermore, information about machine selection and manufacturing cost will be available in hand along with the feature model.

1.6.2 Optimization

The process planning methodology adopted here will produce alternative plans. Optimization involves determining the least cost process plan alternative from the set of process plans generated by the process plan network. The research will report number of best alternative process plans with respect to their manufacturing costs. The optimal\textsuperscript{1} plans generated by the method of this research are optimal only within the assumptions postulated in this thesis. The assumptions in this thesis are believed to be common and reasonable in process planning. The assumptions are listed below.

\textsuperscript{1} ‘Near-optimal’ or ‘optimal’: Due to the fact that most sequencing problems are NP-hard, it is generally not possible to produce optimal results in a reasonable time for realistic problem sizes. Therefore, if this thesis uses the term ‘optimal’, it does not mean ‘optimal’ in the strict sense, but ‘near-optimal’.
1.6.3 Assumptions

The following assumptions are postulated in generating alternative process plans:

- Manufacturing processes are associated with part features.
- Processes are selected for individual manufacturing features,
- Machines are selected for the associated processes,
- Costs for each manufacturing process has been estimated or calculated.
- Test cases on which the system has been verified are completely machinable using milling and hole-making operations.

1.7 Outline of the Research Approach

The approach of this research consists of three steps: extraction and analysis of feature precedence constraints from the feature based part model, incremental process plan generation and finding least cost process plan alternative.
1.7.1 Feature Precedence Constraint Analysis

In this step, the precedence constraints imposed by the feature modeler have been analyzed.

- A feature precedence network (FPN) has been generated to represent the precedence constraints.

- FPN has been analyzed using Petri net.

1.7.2 Incremental Process Plan Generation

In this step, process plan for a part model has been generated incrementally whenever the part undergoes a design change. The incremental process plan has been generated by:

- Determining the design change: whether it is feature addition, deletion or modification.

- Identifying the features that were affected by design change.

- Identifying the process plan nodes that process these features.
• Generating a part of the process plan network that has been required to encompass the design change.

1.7.3 Finding Least Cost Alternative Process Plan

This steps involves finding number of cost effective alternative process plans, where it has been incorporated in the process planning methodology and only the most promising paths have been explored in the process plan generation.

1.8 Thesis Organization

This thesis consists of 7 chapters. In chapter 2, previous work related to this research has been reviewed. Chapter 3 describes representation of precedence constraints of features in the form of a network. It also proposes Petri net as a validating tool for feature precedence. The chapter provides examples to illustrate the feature precedence network (FPN) and corresponding Petri net validate the FPN. In chapter 4, generation of the process plan network has been discussed. Also, the method for the network generation has been explained, an analysis of its complexity has been performed and algorithm for optimization of the network has been provided. Additionally, the optimizing algorithm used for process planning has been explained in detail with an example. Chapter 5 describes the mechanics of incremental process plan generation in response to
design changes, where changes like feature addition, deletion and their effects on process plan network have been discussed in detail. Chapter 6 explains implementation of the prototype and reports the verification of the system and algorithms on several test cases. Finally, chapter 7 gives conclusion of the research and further research topics.
2 Previous Work

During the last two decades production technology available to companies has changed very dramatically, mostly because programmable tools were introduced to production. Highlights of this progress include CNC and DNC machines, robots, and flexible manufacturing systems. Production machines, cells and lines were organized according to the material flows. The focus was on lead-time, quality, throughput, and low work in progress. In this production environment, variant process planning systems based on GT coding have various deficiencies.

Current generation of process planning systems tries to address the problems on the basis of new tools and techniques from information technology, including features and other types of advanced product models. Armed with these weapons, developers and researchers on process planning systems can also explore a range of novel approaches to process planning.

These variant approaches have not been mutually exclusive. Process planning performed early in the life cycle of a product should concern about overall production process alternatives, and therefore it is preferably carried out at a relatively high level by a generative approach.
2.1 Process Plan Representation Models

Researches on CAPP systems have been numerous. A large volume of material has been published, for example, Leung [15] has collected over 200 references, Marri et al. [17] reviewed the literature available from 1989 to 1996, Alting and Zhang [3] reviewed about 200 references and Cay and Chassapis [5] reviewed over 170 references. Resultantly, two directions seem to have evolved. The first one focuses on improving the traditional structures, variant and generative systems. The second one focuses on utilizing Expert System techniques for enhancing the CAPP reasoning ability (Kiritsis [14], Sormaz and Khoshnevis [25]).

Several process plan representation models as a part of a modular process planning system have been presented (Catron and Ray[4]). ALPS, developed at the National Institute of Standards and Technology, is a process specification language based on a directed graph representation, which allows alternative plans and parallel activities (Carton and Ray [4]). The ISO process plan model, which dictates set-based process plan specifications, is still under development. In the ISO process plan model, the form features are represented as sets of entities. Each entity, called process plan activity, has 10 different attributes to define a form feature. A STEP (The Standard for the Exchange of Product Model Data) file is used to represent the information of a feature graph. This STEP
A related area of ongoing research has been the representation of plans for mechanical assembly using robots. AND/OR graph representations have been used for feasible assembly sequences of assembly. The nodes in the directed graph represent stable set partitions of the set of feasible assembly sequences, and the arcs represent the assembly precedence between the sequences (De Mello and Sanderson [9]).

A process planning system called AMOPPS has been developed and described [32]. The system integrated the concepts from design to manufacturing for rotational components, but lacks in research on alternative operations and/or plans and optimization approaches to obtaining operation sequences.

Several research reports propose some model for process plan representation. Common for all of them is their attempt to provide representation schema that will include most general process plan, including hierarchical representation and alternatives. Early work on process representation is ALPS [4] as a language for (graphical) representation of process plans as they are used in other manufacturing planning activities. Since that report, the representation issue is usually dealt as a part of process planning prototype systems, and many process planning systems develop special representation
implemented in particular prototype (hierarchical representation as in [4], rules [10], frames[18], or objects[25]).

Petri nets have been used extensively for modeling Discrete Event systems and FMS [21]. Use of Petri nets for modeling of manufacturing systems have been under research in the past decade. Use of Petri nets as a tool for process planning has been described in [33]. This paper has proposed Petri net model for a dynamic process planning system and as a representation technique for modeling and analyzing process-planning activities as they are specified in the Process Specification Language (PSL) project by NIST.

2.2 Process Sequencing and Alternative Process Plans

The issue of process sequencing and generation of alternative process plans have received attention in process planning research. Prabhu et al [20], proposed a method for generation of operations network for rotational parts based on feature precedence constraints. They proposed a tree representation of allowed feature sequences for turning operation. However, their approach did not address issues where more than one machine may be used for processing of different features. Moreover, their research had been limited to rotational parts only.
Integration of CAD, CAPP and shop floor control for rotational parts has been discussed in Cho et al [8]. The authors have proposed a two-step process plan: off-line and on-line. They represent the precedence constraints as a feature graph and decompose it to a manufacturable task graph for the process planning. ISO process planning model is adopted for process planning from task graph. Each node in the task graph is a process planning activity and associated with each node is a list of manufacturing attributes like tools, fixtures speed, feed, etc. This information is implicit in the ISO process planning model. Once the possible task sequences are generated, it is evaluated for best operation sequence by considering tool machining time, tool change time and tool travel time.

Irani et al [11], demonstrates a strategy for CAPP in the single machine case using a feature precedence graph to represent the relative costs of setup changes required between consecutive operations. They identify the optimal process plan as the Hamiltonian path of least cost in the process plan network. However, in an earlier paper, Prabhu, et al [20], remarks that, though the Hamiltonian method enumerates all the possible paths, it is inefficient. This is because the Hamiltonian paths might not be a valid when precedence constraints are considered.
2.3 Incremental Process Planning

Recent work by Sormaz et al [26] described the process plan network as a hierarchical model, which allows other functions to select the most appropriate alternative representation for given conditions. The procedure for generation of the process plan network has been explained and two algorithms for the selection of process plans in various stages of product development have been presented. The process plan network consists of several levels: feature level, process level, tool direction level, machine level and plan level. Generation of network starts at the bottom level with features. This network is built towards the top considering alternative processes, tools, machines and by clustering processes into manufacturing activities.

Park and Khoshnevis [18] proposed a real-time CAPP system (RTCAPP) as a support tool for economic product design for prismatic parts with internal features. The design methodology of RTCAPP has been based on incremental and hierarchical planning with manufacturing cost optimization. RTCAPP uses a dynamic programming algorithm for machine selection and sequencing. The system also performs incremental process planning. When there are additions to the part design, the process planner considers only new features for process planning and backtracks the existing process plan only to the level that is required by the partial preconditional relationship tree.
2.4 Process Plan Network Optimization

In the area of network optimization, there have been many developments, finding shortest path, and k-shortest paths. There have been few methodologies in literature that exploit a fairly strong analogy between the solution to the general K-shortest path problem and the solution of a system of ordinary linear equations. One of these methods known as “double-sweep method” [19] simultaneously calculates the K-shortest path lengths from a particular source node to all other nodes in the network. The literature [19] compares different algorithm and shows that this method is one of the best methods for computing K-shortest paths of a network.

Once a process plan network has been built to mark the desired goal, an optimization procedure would be required to extract the optimal process plan. Several methods of network optimization have been available in the literature. Two procedures process plan network optimization have been described in [23] and [24]. One has been based on space search technique and the other has been based on network flow theory. The space search technique has an evaluation function that includes actual cost of a state from the start and estimates the cost from the current state to the end. Details of this method have been described in [23]. This evaluation function has satisfied the A* algorithm and has guaranteed an optimal process plan.
In the network optimization method, each arc in the network has assigned the cost of machining on its head node and cost of transfer from its tail node to the head node. Once these costs were available, the problem of finding the optimal plan could be formulated as a shortest path problem [19]. The paper has used Dijkstra's [1] algorithm for finding the shortest path due to the existence of non-negative costs on the arcs.
3 Feature Precedence Network Generation

For a given part, the machining operations cannot be performed in any arbitrary order. A number of geometric and technological constraints require that some operations be performed before or after certain other operations. This information is captured in the feature model as feature precedence constraints. The first and foremost step involved in process planning is obtaining these feature precedence constraints and translating it to a usable form. Here, the feature precedence constraint has been translated into a network form. This network has been analyzed before it has been used for process planning. This chapter provides an insight into 1) feature precedence network, 2) the methodology that has been adopted in extracting and representing it from the feature based part model, and 3) validation of precedence constraints.

3.1 Feature Precedence Network Representation

A part model is converted to a feature model by use of feature recognition techniques. Feature recognition techniques extract manufacturing features from a part model. These features serve as a model for process planning. The feature modeler determines a set of precedence relationships among the features according to the technological and geometric production constraints. The precedence of a feature by another indicates that the preceding feature should
be machined or produced before the other in order to satisfy the constraints. These constraints among features in a part model can be captured in the form of a graph. Such a graph in which each node represents a feature and each arc represents a precedence relationship is a feature precedence network.

Relations between features have been captured in feature graph, PPR tree and in predicate forming previous researches. Here a more rigorous model, the feature precedence network has been used. Feature precedence network (FPN) is defined as a directed graph representing the precedence constraints imposed on the features of a part. The formal definition of a FPN is given in Equation 3-1.

\[ FPN = \{ N, A \} \]

where, \( N = \) set of nodes
\[ A = \) set of arcs.
\[ N = F \cup \{ S, E \} \]
\[ F = \) set of features,
\[ S, E = \) terminal nodes, start and end respectively.

\( A_{ij} \) is an arc from feature node \( N_i \) to feature node \( N_j \). \( A_{ij} \in A \) if it is required that for any feasible feature processing sequence (which is permutation of set \( F \)), feature on node \( N_i \) must be processed before feature on node \( N_j \). Requirements that each feature is processed after node \( S \) but before node \( E \) are represented in Equation 3-2.
As, 
E A Iff not(∃F_k, A_{kj} ∈ A),
A_{ie} ∈ A Iff not(∃F_k, A_{ik} ∈ A)
for each F_k ∈ F
and,
A_{se} ∉ A and
A_{ii} ∉ A

Equation 3-2

In other words, FPN is a graph in which the nodes represent the set of primitive features while the edges represent the temporal precedence between two nodes. The feature precedence network has two terminal nodes S and E. The nodes mark the start and end of the network traversal. Figure 1(b) shows an example of a FPN for the part shown in Figure 1(a).

Figure 1: (a) Part Model, (b) Feature Precedence Network

A machining feature contains much more information than just dimensions, access direction and contour geometry. In general, a feature has parent features and child features [22] and this information is also stored in the feature as next and previous attributes. For instance in Figure 1, feature F4 has F2 and F3 as it
parents. Such information is critical for operation sequencing because it sets up precedence constraints between different features. One of these constraints is that if feature $F_a$ provides machining access to feature $F_b$ then, feature $F_a$ must always be machined before feature $F_b$. Thus a parent feature should be machined before any of its children. The set of parent features for feature $F_j$ may be represented as shown in Equation 3-3.

$$\text{Set of parent features } f_j \in \{ f_k \in F, \text{ } A_{kj} \in A \}$$  \hspace{1cm} \text{Equation 3-3}$$

Constraints between features can also arise due to feature interactions. A feature interaction can be defined as the intersection of the boundary and/or volume of the feature with that of other features so that the geometry of the feature becomes different from what the designer originally intended [22]. Such interactions are sometimes important during the process planning stage. An important interaction arises when holes and pockets intersect. The geometry and the mechanics of drilling and milling require that the holes be drilled before the intersecting pocket can be machined.

The feature precedence information is extracted from the manufacturing feature model of the part. While we have not addressed the algorithm for this task as they were beyond the scope of the thesis, we require that the information be available in feature model as precedence relation. This precedence
information is available in the feature model as next and previous attributes. This information is represented in the form of a network. This network is referred to as feature precedence network.

Nodes are generated for each feature in the part model. Features which are without parents (or if their previous attribute is empty) are qualified to be processed first. These feature nodes are connected to start node “$S$” in the FPN to denote that these features can be machined first. All the other feature nodes are connected through arcs based on the precedence constraint imposed on each feature in the part model. The features that can be processed at the end (or the features that have empty next attribute) are connected to the end node “$E$”. Such a network has a start and an end node as terminal nodes with all feature nodes as intermediate nodes. The resulting network is a formal representation of a FPN.

An example of FPN for the part model shown in Figure 1 (a) is represented in Figure 1 (b). The part consists of two slabs $F_1$ and $F_3$, a hole $F_2$, and a slot $F_4$. The slabs, $F_1$ and $F_3$, should be machined before hole $F_2$ and slot $F_4$ respectively in order to reduce machining time for hole and slot and the hole $F_2$ should be drilled before the slot $F_4$ is milled to avoid tool deflection. This information is obtained from part model. The Feature $F_2$ has $F_1$, and $F_4$ has $F_2$ and $F_3$ as its previous attribute and it forms the basis for the feature precedence
network generation. The features that qualify to be processed first are \( F1 \) and \( F3 \). Feature nodes corresponding to \( F1 \) and \( F3 \) are connected to the start node “S”. The precedence constraints imposed on the features are shown in Table 1. Thus, the feature node \( F1 \) is connected to \( F2 \); \( F2 \) and \( F3 \) connected to \( F4 \), and finally \( F4 \) is connected to the end node “E”.

Table 1: Precedence constraints on features

<table>
<thead>
<tr>
<th>Previous</th>
<th>Feature</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>( F1 )</td>
<td>( F2 )</td>
</tr>
<tr>
<td>( F1 )</td>
<td>( F2 )</td>
<td>( F4 )</td>
</tr>
<tr>
<td>-</td>
<td>( F3 )</td>
<td>( F4 )</td>
</tr>
<tr>
<td>( F2,F3 )</td>
<td>( F4 )</td>
<td>-</td>
</tr>
</tbody>
</table>

### 3.2 Analysis of Feature Precedence Network

Feature precedence network has been obtained from the feature modeler. Before using FPN in process planning, it would be necessary to check the validity
of the feature precedence constraint to assure a well-formed process plan network. We propose that the feature precedence network obtained from the part model should be validated using the Petri net methodology described in the following section. For a given FPN, we generate the corresponding Petri net and verify the deadlock. The feature precedence network formed from the feature constraints has been checked for cyclic constraints and they were identified as deadlocks on Petri net.

### 3.2.1 Petri Net

Petri net is a tool for the study of dynamic systems. Petri net theory has emerged from the initial seminal PhD thesis of Carl Adam Petri entitled "Kommunikation mit Automaten" [Communication with automata] presented in Bonn in 1962. In his doctoral dissertation, Petri formulated the basis for a theory of communication between asynchronous components of a computer system.

Petri net theory allows a system to be modeled by a Petri net, a mathematical representation of the system. Analysis of the Petri net can then reveal important information about the structure and dynamic behavior of the modeled system. This information can then be used to evaluate the modeled system and suggest improvements or changes. Thus, the development of a theory of Petri nets has been based on the application of Petri nets in the modeling and design of
systems. In this research, the Petri net has been proposed as a modeling tool to analyze the validity of the feature precedence network.

The practical application of Petri nets to the design and analysis of systems can be accomplished in several ways. One approach considers Petri nets as an auxiliary analysis tool. For this approach, conventional design techniques are used to specify a system. This system is then modeled as a Petri net and this Petri net model is analyzed. Any problems encountered in the analysis point to flaws in the design. The design must be modified to correct the flaws. This modified design can then be modeled and analyzed again. This cycle is repeated until the analysis reveals no unacceptable problems. This approach is diagrammed in Figure 2. Similar modeling technique is developed for analysis of the feature precedence network using Petri nets.

Figure 2: The use of Petri nets for modeling and analysis of a system
3.2.2 Definition of a Petri Net

Petri nets are defined as a bipartite graph consisting of places graphically represented by circles, transitions represented by bars, and directed arcs connecting places to transition and vice versa.

Places represent passive system components, which store “tokens” and take particular states. Each place may have zero or more tokens. Transitions represent the active system components, which may produce, transfer or consume “tokens”. For each transition there is a set of input places and a set of output places. Arcs connecting the transitions and places represent the direction of the information (token) flow through the net. At any instant the state of a Petri net is the position of tokens at that time. Firing a Petri net moves it to a new state. Each transition has a firing action that can be performed if there are at least one token in each of the incoming places. Every firing action removes one token from each input place and adds one token to each output place. One of the important properties of a Petri net is its reachability property. This property evaluates the possibility to reach given state(s) from another state(s). A state $\mu'$ is said to be reachable from a state $\mu$, if there is a sequence of intermediate transitions leading from $\mu$ to $\mu'$. The set of all reachable transitions from $\mu$ to $\mu'$ is called the reachability set. This property of Petri net is utilized to check for the validity of feature precedence network.
3.2.3 Operation of a Petri Net

Each transition in a Petri net is either enabled or disabled. A transition is enabled if there is at least one token in each of its input places. Any transition can fire whenever it is enabled. If multiple transitions are enabled, any one of them may fire. When transition fires, one token is removed from each of the input places, and one token is added to each of the output places; this is effectively done atomically, as one action. When there are no enabled transitions, a Petri net is said to be in a state of deadlock.

3.2.4 Feature Precedence Based Petri Net

The feature precedence constraints have been obtained from the part modeler. It is necessary to check the validity of the feature precedence constraints to ensure a well-formed process plan network. So, the feature precedence constraints obtained from the part model have been first modeled as a feature precedence network and this network has been validated using Petri net-based methodology described in [27].

Petri net is generated using the information available from the feature precedence network. For each node in the FPN we have generate a connected pair of Place and a Transition directed from place to transition, this pair is called a FP-Petri node (Figure 3). This includes the start and end node in the FPN.
These FP-Petri nodes are connected according to the feature precedence constraint. The start node's place in the FP-Petri node is provided with one token to mark the start of evaluation. All the other Places have zero tokens. This forms a validating Petri net for the FPN. This Petri net, when fired, should be able to traverse till the end node without any deadlock. If the Petri net passes the evaluation, then the FPN is considered valid, otherwise, the FPN is considered to be mal-formed.

The aforementioned analysis identifies cyclic precedence constraints imposed by the feature precedence. The Petri net should fire successfully without any deadlock till the end node's transition is enabled. This state confirms a well-formed feature precedence constraint. If on the other hand, the Petri net encounters deadlock before reaching the end node, the feature precedence constraint might have cyclic constraint(s). If the above state is reached, then the system doesn't proceed further. It reports the error to the feature modeler. If the analysis doesn't report an error, the system proceeds to process planning stage.
3.2.5 Examples

Consider the feature precedence network shown in Figure 1(b). Petri net nodes are generated for each of the nodes in the feature precedence network. Each of these nodes' transition is labeled with corresponding manufacturing feature. As a result, four FP-Petri nodes are created for the four features and a node for start and end. The start node in the Petri net is provided with a token. All the other nodes have zero tokens. The Petri net is then connected according to the precedence constraints imposed in the FPN. If a node in the FPN has two or more input arcs, then each of the corresponding links in the Petri net is made through a place. In this example feature F4 in the FPN has two input arcs, one from F3 and the other from F2. The corresponding Petri net also has two inputs to the transition F4 through two places.

Figure 4 shows the validating Petri net for the FPN shown in Figure 1(b). This Petri net, when fired, should meet the evaluation criteria described in the previous section. That is, the Petri net should not encounter dead lock before the transition E is enabled. When the Petri net in Figure 4 is fired, it will successfully complete the transition without any deadlock. Figure 5 shows this Petri net in transition. The Petri net in Figure 4 complies with the above criteria and therefore confirms that the feature precedence network is valid.
Figure 4: Validating Petri net

(a)

(b)

(c)

(d)

(e)

Figure 5(a to e): Step-by-step transition of validating Petri net
Let us consider another example to illustrate an invalid feature precedence constraint. As stated earlier, the FPN will be analyzed for cyclic constraints. Figure 6(a) shows a FPN with cyclic precedence constraint and Figure 6(b) shows the corresponding validating Petri net.

![Figure 6: (a) FPN with cyclic constraint, (b) Validating Petri net](image)

It may be noted in the FPN that the features F4, F5, F2, F3, F6, F7 and F8 induce cyclic constraint. When the corresponding validating Petri net is fired, it will encounter a deadlock. Figure 6(b) shows the Petri net in the deadlock condition. The Petri net is deadlocked at transition F2 and F4, just after transition
F1 is fired. Transition F2 and F4 will not be enabled, as it will still have a zero-token input place from transition F5 and F8 respectively. This non-availability of the token in the input place leads to deadlock.
4 Process Plan Network

The current chapter describes the methodology for generation and optimization of the process plan network. The methodology includes the network generation for a given FPN and individual feature processes, described in section 4.2. The complexity issues involved in generating such PPN have been explained in section 4.3. The algorithms for the network optimization, that is to say the selection of optimal alternative process plan among all the plans represented in the PPN have been detailed in section 4.4.

4.1 Process Planning

Process planning is a task to set a plan needed to manufacture products from specified designs. Process planning includes subtasks such as selection and sequencing of manufacturing operations. In general, process planning is a complex task that requires thorough knowledge of both the design requirements of a part and the available manufacturing processes and their capabilities.

Process plans convey a large amount of processing information, such as part routings, operation sequences, material requirements, and resource requirements. Process plans must also be capable of representing alternatives in
terms of operation sequences and machine sequences in order to provide flexibility for the production control.

The approach adopted here is a feature-based process planning approach. Feature-based approaches to CAD/CAM integration have found application in design, planning, assembly, and manufacturing. A feature-based model is first obtained from the feature modeler. The feature modeler basically extracts manufacturing features from a CAD model. These features have the information about precedence constraints imposed on them. These constraints are available as next and previous attributes of a feature. As discussed in Chapter 3, the precedence constraints are first analyzed and machines are selected for processing the each feature. Process planning can be initiated, once this information is available.

Process planning produces the most suitable operation sequence for machining features in a part model. The sequence of operations depends mainly on the precedence constraints of the feature and the cost of machining. This process planning system provides the user with a number of alternative optimal/near optimal process plans.

For the purpose of process plan generation, a part model with well-defined feature precedence constraints has been obtained. For the current implementation, features were assumed to have information about the various
machines on which they may be manufactured. Later this information may be obtained from the knowledge base available in 3I-PP [29]. 3I-PP is an Intelligent, Integrated, and Incremental process planning system, built on a robust knowledge base, which consists of various machining operations and the associated machines for a feature. A decision on the machine selection for a feature will use this knowledge base. An XML interface to communicate with this knowledge base and obtain the required information has been built as a part the IMP-project group. For the purpose of implementation and testing, the information about machine selection has been assumed to be available in hand before start of process planning. The machine information will be available in the *machines* attribute of the feature model. A machine contains details about the features that can be machined on it, and its machining cost and setup cost. This provides basis for computing the cost of production.

The availability of more than one processing sequence from the feature precedence leads to alternate process plans. An estimate on the number of alternatives is discussed in Sormaz *et al* [28]. These alternatives further increase with alternative machines available for each feature. Consider the part model shown in Figure 1 (a), the hole $F_2$ may be machined either on a milling machine or on a drilling machine. Milling and drilling machines therefore form the candidate machines for the hole. When process plan is generated for such a model, it provides scope for alternative process plans.
4.2 Process Plan Network Generation

A process plan may be represented using a resource-based representation, where each node refers to a part feature that is produced using the resource as specified by the node. Such a process plan representation has been used here. Basically, the process plan node represents the machine and the feature that is processed in it.

Once the features are extracted and analyzed, the step of process planning begins. Each manufacturing feature in the part model has a node associated with it in the process plan network (PPN). This acts as a linkage between the feature model and process plan. This is the integrating link established between CAD and CAPP. This linkage helps in incremental process planning. Details about incremental process planning will be dealt with in chapter 5.

With this information in hand, a process plan with alternatives may be generated and represented as a network. The process plan network may be defined as a network with set of nodes and arcs (Equation 4-1). Figure 7 shows a typical set process plan nodes connected by an arc.

![Figure 7: A set of process plan nodes and arc](image-url)
\[ PPN = \{N, A\} \]

where \( N = \text{Set of Nodes} \),
\( A = \text{Set of Arcs} \).

Process plan node and its interaction with other nodes in the PPN may be represented mathematically by the following set of equations. Process plan node may be defined by Equation 4-2.

\[ N_i = \{M_i f_i \mid F_i\} \]
\[ \forall f_i \not\in F_i \text{ and} \]
and precedence constraint of \( f_i \) and \( F_i \) are satisfied.
where,
\( M_i \) denotes the machine on which feature \( f_i \) will be processed.
\( F_i \) is a set of features that have been processed before \( f_i \).

Every process plan node has an interaction at least with one other node in the process plan network, where the interaction is denoted by the arcs in the PPN. Equation 4-3 denotes the condition that establishes interaction between nodes.
Let \( N_j \) be another process plan node defined by,
\[
N_j = \{ M_j, f_j / F_j \}
\]
and
\[
A_{ij} \text{ be an Arc from node } N_j \text{ to } N_i.
\]
Then, \( A_{ij} \in A \),
iff \( F_j = f_i \cup F_i, f_j \notin F_i \),
\( f_j \) and \( F_j \) satisfy precedence constraints.
if \( f_i \cup F_i = F \), set of all features in the part model
then, \( A_{ij} = A_{ic} \)
where \( A_{ic} \) is a Arc from \( N_i \) to end node \( E \).

\[
\text{Equation 4-3}
\]

if \( F_i \) is empty, then \( A_{ij} = A_{sj} \)
where \( A_{sj} \) is a Arc from start node \( S \) to \( N_j \).

Nodes provide information about the feature and the machine on which they are processed. Nodes also have information about the features that have been already processed. Typically a node is denoted by “Machine Type, Current Feature / [List of completed Features]”. For instance, a node labeled, “\( \{M4, F4\}/[F2, F3] \)” denotes that feature \( F4 \) is machined at workstation \( M4 \) after both \( F2 \) and \( F3 \) have been completely machined. Arcs connect the process plan nodes with each other. These arcs are directed and represent the successive operations that have to be done to get to the final finished part. There may be more than one arc originating from a node or terminating on a node. When more than one arc originates from a node, it implies the existence of alternative operations/operation sequences from that node. Such conditions yield alternative process plans. Arcs carry information about the cost incurred on moving from
one node to another. These costs are represented on the arcs. Figure 8 shows a part of process plan network with the costs on arcs. The cost includes machining cost at the sink node and the setup cost, and transportation cost that might be incurred.

Process sequencing is a result of a search algorithm from initial state (raw stock), in which no feature is machined, to the goal state, the finished part. The transition through the state space is governed by feature precedence network. The space search starts from the raw stock and decision is made on the next set of features that can be machined where the decision is made according to the feature precedence network. Process plan nodes are formed for these features with all the available machine candidates. Alternative process plans are based on these features. The process plan for a part is represented in the form of a network. Traversing the network from start to the end node, through any one of the paths gives a feasible process plan. Thus, process plan is defined as any
path on the process plan network, from start node S to the end node E. The total manufacturing cost that will be incurred by use of a particular process plan can be computed by summation of all the costs on the arcs of a path. The path that yields the least cost is the process plan that will have least manufacturing cost, that is to say, it is the optimal process plan.

The process plan network generated has definite set of stages. The number of stages depends on the number of features in the part model. Every process plan network has \((N+2)\) number of stages, where N is the number of features in the part model, and the start and end nodes denote the two extra stages. It is because each stage in the PPN marks the number of features that have been processed. For instance, in Figure 9 it may be noted that there are four stages. The first stage has only the start node “S”, the second stage has five nodes, and third stage has eight nodes. All the nodes in a stage have a common property: the number of features machined is equal. For example in the third stage, all the eight nodes have two completely machined features. Such a type of representation of PPN into stages makes it very clear and readable.
4.2.1 Network Generation Algorithm

The main input information required for generating a process plan network is the FPN. FPN is basically a graph showing the constraints among the features of a part model. The FPN dictates the operation sequence in a process plan network because the operation sequence should satisfy the precedence constraints. While generating a process plan network, at every stage, the
algorithm makes sure that the feature constraints put forth in the FPN are satisfied.

PPN is generated by traversing the FPN from its start node. First, all the feature nodes connected to the start node in the FPN are considered. For all these features nodes, corresponding PPN nodes are created. These PPN nodes are connected to the start node. PPN nodes are created using the Feature-Machine information available in hand. Once the first set of node is generated, the PPN propagates using the algorithm that will be described in this section.

A hybrid of space search, Dijkstra's and k-shortest path algorithm is proposed. The algorithm explores all possible results as in space search, at the same time, expands only the nodes that might lead to the goal. While expanding from a node, in order to obtain “k” optimal solutions, “k” promising paths are explored at each stage.

The algorithm used to generate process plan is adopted from Dijkstra's shortest path algorithm. Dijkstra's shortest path algorithm finds the shortest path on a network by evaluating the distance of a node from the start node and continuously checking with the other nodes for shortest distance. The nodes that form the candidates for shortest path are labeled as permanent and all the other nodes are labeled as temporary. Upon reaching the end of the network, the permanent labeled nodes are traced to get the shortest path. This is one of the
popular algorithms for finding the shortest path on a network. But it generally works on a pre-formed network. Here, the same methodology is used to find the least cost path, but the entire network is not generated. Only the most promising paths on the network are generated and explored.

A similar approach is used in generating the optimal process plan. Instead of exploring the entire search space created by various feature and machine combination in the process plan network, only the most promising process plan states are explored. In order to generate required number of alternative optimal/near optimal process plans, specific number of alternative paths is explored at each stage. At this point the algorithm differs from Dijkstra’s shortest path algorithm. The algorithm adopted here expands only "k" promising states at each stage, where “k” is the required number of optimal plans. Promising paths are the paths that will eventually yield least manufacturing cost. These paths are explored at each stage until the required number of (“k”) process plans is obtained.

This methodology dramatically reduces the search space because we do not explore all the possible state space. It is well know that sequencing problem is NP-hard, in order to avoid getting into issue of combinatorial explosion, only promising process plan nodes are explored. This significantly avoids getting into combinatorial issues. It is to be noted that the worst case is still going to be NP-
hard. As this research mainly focuses on incremental process plan generation, it has been reasonable to focus only on problems that don’t get into combinatorial explosion. In order to accomplish the task, a well constrained feature precedence network has always been considered to keep the focus on the incremental process planning.

In order to explain the process planning in detail, let us consider the part model shown in Figure 1(a) with 4 features. The feature precedence network for the same part is shown in Figure 1(b). This part model has 2 slabs (F1, F3), a hole (F2) and a slot (F4). It is assumed that the process/machines for respective features are selected and the feature precedence network is validated before the process planning is started.

Process plan network generation starts from the node “S”, which marks the start the planning process. This node may be considered as the raw stock available for the machining. From this node, a search is made to determine the list of features that can be machined next. Table 2 provides a list of parents and their possible children for part model shown in Figure 1(a).
This information is obtained from the precedence constraints. Once this list of next features (child features) is found, all the machine candidates for each of the features are obtained. These details are assumed to be precise and available in hand during the process planning stage. The machine candidates for the features are shown in Table 3, where the cost of producing the feature on the machine is also specified. If a feature cannot be manufactured in a machine it is indicated as a "-" in the table. The table indicates that there are 3 machines each for feature F1, F2 and 2 each for feature F3 and F4. It might be noted that during process planning, part handling cost and machine setup cost will also be considered. Handling cost is added for every operation whereas; setup cost is added whenever the operation requires change of machine.
Table 3: Process candidates for features with costs in dollars

<table>
<thead>
<tr>
<th>Features</th>
<th>Milling machine M1</th>
<th>Drilling machine M2</th>
<th>Milling machine M3</th>
<th>Milling machine M4</th>
<th>Milling machine M5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slab – F1</td>
<td>10</td>
<td>-</td>
<td>3</td>
<td>15</td>
<td>-</td>
</tr>
<tr>
<td>Hole – F2</td>
<td>-</td>
<td>5</td>
<td>1</td>
<td>-</td>
<td>2</td>
</tr>
<tr>
<td>Slab – F3</td>
<td>-</td>
<td>-</td>
<td>2</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>Slot – F4</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>5</td>
</tr>
</tbody>
</table>

A portion of progressive expansion of the process plan network is shown in Figure 9. The start node expands features that may be machined first. That is, these features do not have any required predecessor defined in the FPN. In this example it may be noted that feature F1 and F3 qualify to be produced first. Nodes with F1 along with its machine candidates and F3 with its machine candidates form the children of the start node. Thus five nodes are generated at this stage. These nodes are connected to the start node “S”, which is the parent node at this stage. The cost of manufacturing each of the features is represented on the arc connecting their respective process plan nodes with their parent node.

Let us assume that the number of optimal alternative process plan required is 2. Hence, only two least-cost alternatives are selected for expansion at each stage. Nodes M3F3/[ ] and M3F1/[ ] are selected for expansion as the total machine cost is least on them (8 and 9 respectively). These nodes when expanded would produce 3 and 5 nodes respectively. Further expansion is again
decided by considering the next possible features that can be machined along with their machine candidates. At this stage only the least cost nodes are selected again. During this selection, even the un-expanded nodes in the previous stage are considered. For example, if cost on node $M1F1/\{\}$ were 10.00 instead of 16.00, then this node would be expanded at this stage. Two least cost nodes are selected for expansion. These nodes are node $M3F2/[F1]$ and node $M3F1/[F3]$, and $M3F3/[F1]$. Three nodes get selected because the two nodes ($M3F1/[F3]$ and $M3F3/[F1]$) have the same cost ($12 each) at this stage. This process is repeated until the goal/end node is reached. The network generated for the example under consideration is shown in Figure 10.

Figure 10: Process plan network showing all explored nodes
This network has many un-expanded nodes. So, a path-tracing algorithm is used to find the least cost path on the generated network by eliminating the unexpanded nodes. Finally, only two alternative process plans are represented from the generated network. Figure 11 shows the two least cost alternative process plans.

This algorithm promises a least cost process plan. As mentioned earlier, the algorithm is similar to Dijkstra's shortest path algorithm but only expands the portion of the network that will lead to optimal/near optimal solution. It can be considered as Dijkstra's algorithm working on incremental network generation. Apart from the above stated difference, the algorithm also has a feature that is not available in Dijkstra's algorithm. This process-planning algorithm finds “k” least cost operation sequences unlike Dijkstra's algorithm that finds only one least cost sequence or shortest path.

Figure 11: Process plan network with optimal alternative process plans.
This algorithm was tested on various feature precedence network configurations and was found to yield good results. A detailed description of the algorithm and various examples has been provided in chapter 6.

This methodology provides for a well-formed process plan network for a part model. The manufacturing cost for each of these alternatives may be calculated by the costs on their respective process plan arcs. This cost analysis may help the designer to make appropriate decisions during design evolution. This process plan could be considered a batch type process plan generation, where the process plan is generated from either a fully or partially completed part model. In the next chapter, incremental process plan network generation for feature addition and deletion has been discussed in detail.

4.3 Complexity Analysis of Network

The best-known difficulty of any sequencing problem is its computationally hard nature. Here in process planning we deal with operation sequencing. In the worst case, a part may have an exponential number of possible feature sequences, and hence an exponential number of operation sequences.

Consider a part with \( n \) features and \( m \) candidate machines for each of the \( n \) features. Let all features have equal precedence. This means that there are at least \( n! \) different possible feature sequences. These sequences further
increase by the $m$ machine candidates for each of these features. In general the number of sequences may be given by $n!m^n$.

Let us estimate the number of process plan nodes that might be generated in such a case. Since there are $n$ features there will be $n$ stages in the process plan network. Figure 12 shows a process plan network for 3 features and 2 machines respectively. It is evident that there are three stages in the process plan network. Let us estimate the number of nodes in each stage first, and finally estimate the total number of nodes for this problem. Equation 4-4 estimates the number of nodes in a process plan network with $n$ features and $m$ machine each, and also shows the estimate for a 3-feature, 2-machine problem.
Figure 12: Process plan network for 3 features with 2 machines each showing all alternatives
The above expression yields 24 nodes for the problem in hand, which is exactly the same as seen on the network. It may be noted that the factor $n^* m^* 2^{n-1}$ is exponential and thus, the number of process planning node increases exponentially with number of features. So it is for sure that the worst case is exponential in nature.

Thus, a blind space search approach to generate all the possible operation sequences would take exponential time. In our approach, the process planning technique will prevent most of these operation sequences from being generated. But it still seems possible that the proposed approach will generate and evaluate an exponential number of alternatives in the worst case. It is doubtful that any other approach can do any better, because the problem of finding an optimal operation sequence is likely to be NP-hard for the worst case. If the size of the problem grows, the time required for the computation of the optimal solution grows rapidly beyond reasonable bounds. Formally speaking, most sequencing
problems are NP-hard or NP-complete. A NP problem — “non-deterministically polynomial” — is one that, in the worst case, requires time polynomial in the length of the input for solution by a non-deterministic algorithm [1]. The practical consequence of NP-hardness is that the required calculation time for finding the optimal solution grows at least exponentially with the problem size.

This section explains the impact of complexity in using the proposed algorithm. Consider the same example illustrated above in Figure 10. If the process plan network is expanded using space state techniques, it would result in a very large explosion of the network. Figure 13 shows the same network generation using breadth first space search technique.
In the blind space search approach, at each stage all the possible machine candidates for the features are expanded until the goal is reached. Since it does not have any evaluation criteria, the algorithm searches for all possible states. Finally, after all the possible states are expanded, k-shortest path algorithm has to be used on the generated network to find the k-optimal solutions. This involves searching the huge network to find least cost alternatives.

Let us consider a 5-feature problem with 3 machines each. It can be estimated that the blind space search would have explored 240 nodes by using the equation 4-4. On the other hand, the proposed incremental Dijkstra’s algorithm would have explored much less than 240 nodes. The reason being that
some of the nodes will be left unexplored depending on the cost at the node. The number of nodes cannot be precisely estimated as it depends on the manufacturing cost data. But it is certain that it will be less than the blind space search technique, because at each stage definite number of nodes are only expanded rather than exploring all the possible nodes. It also has an advantage of having the information of the number of least cost process plan alternatives and their total manufacturing cost at the end of the search. The process of optimizing the network is eliminated as the algorithm generates only the optimal/near optimal solution.

It may be noted that it is still an exponential problem and it is reasonable that it will entail exponential time to solve it. This combinatorial explosion is considerably reduced by the proposed algorithm. As mentioned earlier, as the focus of the research has been on incremental process plan generation, it is reasonable to assume a well-constrained feature precedence network so that combinatorial explosion may be avoided.

4.4 Optimization of Process Plan Network

4.4.1 Cost Model

Economics plays a very important role in manufacturing planning. Estimation of cost is crucial to any process planning system. The manufacturing cost
estimation includes material cost, labor cost, machine cost and tool cost. It requires a detailed analysis to precisely estimate the manufacturing cost. The detailed analysis is based on time required to complete a given manufacturing operation and the breakdown of the part flow on the shop floor. For each operation the time and resource required have to be identified and their costs have to be computed. The manufacturing time basically consists of machining time, tool change time, part-handling time, setup time and transportation time.

Equation 4-5 shows the manufacturing cost per unit work piece [6]:

\[
C_p = \sum \left\{ \frac{C_b}{N_b} + C_m \left[ t_m + t_h + \frac{t_m}{T} \left( t_r + \frac{C_t}{C_m} \right) \right] + \frac{C_{tr}}{N_b} \right\} 
\]

where,

\[ C_p \] = manufacturing cost per part ($/part),

\[ C_b \] = setup cost per batch ($/batch),

\[ C_m \] = machining operation cost ($/min),

\[ C_t \] = tool cost ($/tool),

\[ C_{tr} \] = transportation cost between two subsequent machines,

\[ N_b \] = batch size,
\[ t_m = \text{machining time for each feature (min)}, \]

\[ t_h = \text{part handling time (min)}, \]

\[ t_t = \text{tool change time for each feature (min), and} \]

\[ T = \text{average tool life (min)}. \]

All these cost components are not so much dependent on the individual feature and require accurate data or estimate technique to evaluate the manufacture cost. Due to non-availability of data, some of the factors are assumed constant, and a simpler costing model is used.

The production cost for a feature on a machine is divided into three major portions: the actual machining cost, part handling cost and transportation cost. Machining cost consists of tooling cost, machine cost and labor cost. The part handling cost involves the cost of re-orientation of the part and the tool change cost. The transportation cost is the cost of moving the part from one machine to other for machining. This cost also includes the set-up cost wherever applicable. The transportation cost is thus incurred only in case of the movement between two different machine types. For example operations that require machine change like, \( M1F1 \) to \( M2F2 \), have transportation cost added up to their manufacturing cost. If the same set of operation can be done on the same machine, like \( M1F1 \) to \( M1F2 \), then the transportation cost is not added to the manufacturing cost.
The machining cost for each feature is available with the machines they are processed in. Based on the machines used in a given process plan, transportation and handling cost are added. These costs are represented on the arcs between the process plan nodes. The summation of cost on each arc in any given alternative process plan provides the cost of production of the part using that plan.

4.4.2 Least Cost Process Plan Alternative

The optimal process plan is the process plan with least manufacturing cost, where the process planning system finds number of least cost process plan alternative. This criterion is incorporated in the process plan network generation. As a result, the process plan network will generate only the most optimal process plans for the given parameters. It may be noted that the optimal process plan is the one with minimal number of transportation between subsequent machines. The reason being when most of the operation is performed on the same machine the transportation cost is drastically reduced. Apart from choosing same machine types, choosing the machines with minimal machining cost also reduces the manufacturing cost.

The process planning system makes such decisions during process plan network generation. And only lowest cost process plan alternatives are explored and represented in the process plan network.
5 Incremental Process Plan Generation

Mechanical and manufacturing engineering design involves product design and process design (or process planning). While the former deals with satisfaction of product's functional requirements, the latter converts design specifications into manufacturing instructions. Conventionally product design and process planning are performed sequentially. One-way mapping from design to manufacturing is adopted, and feedback of manufacturing information for the complete design is provided to the designer. However, more rigorous product requirements and the need for shorter production lead-times increasingly call for a better product and process design approach, where product design and manufacturing planning are carried out simultaneously. In contrast to the sequential approach employed by traditional process planning systems, this concurrent engineering approach reduces the time from design to production by achieving parallelism in design and process planning activities.

Conventional process planning system, works by re-generating the process plan network from the start whenever new features are added. It involves re-calculation of all the parameters, some of which might not have undergone a change due to this feature addition. The method is a non-value added activity, as it is redundant. In order to eliminate this non-value-added activity, we have proposed a methodology in this chapter to execute the update incrementally.
Figure 14 shows the linkage of design and process planning in conventional, concurrent and incremental process planning system.

For incremental process plan generation, a link between the part design and process plan needs to be established. Here, this linkage is established by associating the manufacturing features in the part model with the features on the process plan nodes. The incremental process planning approach should be able to identify the features that underwent a change in the design evolution. It should also be able to identify process plan nodes that are affected by the design change. Once these nodes are identified, changes may be incorporated in the process plan to reflect the design change. Feedback about the change in manufacturing cost for the optimal process plan is then provided to the designer. This will help the designer to make appropriate decisions on the design alternatives. The design changes that will be handled in the following section include feature addition, feature deletion and other changes that will lead to the modification of precedence constraints of features.
5.1 Feature Addition

As mentioned earlier, design evolution is an on-going process and there will be addition and deletion of features in a part model. Whenever a new feature is added to the part model, it certainly affects its existing process plan. If the part
already had a process plan, then it has to be changed to accommodate the new feature. In this section we discuss aspects of incremental feature addition.

Addition of a feature to a part model will fall under one of the cases shown in Figure 15. In this figure, $F_i$ and $F_j$ are features on the FPN before addition of the new feature $F_n$. The new feature $F_n$ may be added in one of the following ways illustrated in Figure 15.

![Figure 15: Cases of feature additions. (a-d)](image-url)
In Figure 15(a), the $Fn$ is added with no precedence constraints. In this case, all the features in the FPN will be affected by this addition. Figure 15(b) through (d), represents cases in which the feature is added in such a way that there exists a well-defined precedence constraint. All the feature additions that fall under these cases are taken into consideration during incremental process planning.

Incremental process planning on feature addition basically involves four major steps:-1) identification of the feature(s) that have been added to the existing part model, 2) finding the features in the part model that have undergone a change in their precedence constraints due to this new feature addition, 3) identifying the corresponding process plan nodes that needs to be modified to accommodate the new feature in the process plan, 4) re-generate a portion of the process plan network. Procedure that has been described here will work on both the complete PPN or on partially generated PPN during optimization procedure.

As mentioned in the previous chapter, the process plan network has been generated only for the required number of optimal plans. Thus, only these plans are represented in the network and the feature addition will be effected only on these optimal alternatives and by doing so there is a possibility that the global optimal may be overlooked. Since we are updating a pre-optimal solution, a near optimal solution may be guaranteed.
On feature addition, the list of affected features is found from the feature precedence network. This may be illustrated with the FPNs shown in Figure 16.

![Figure 16: (a) FPN before feature addition (b) FPN after feature addition](image)

Assuming the new feature is added in such a way that it changes the FPN from Figure 16(a) to Figure 16(b). The new feature $F_4$, then affects feature $F_2$, $F_3$.

These features form the set of modified features. Once this set is obtained, the next step is to find the list of nodes that are affected. These nodes are the nodes, in which these modified features are machined. Decision is made whether the modified nodes are still valid for the new FPN. All the valid nodes are retained and the rest of the modified nodes are deleted from the network. The retained modified nodes form the parent nodes and process plan network is generated from these nodes as explained in chapter 4. With this methodology, only a portion of the network is generated and is augmented with the pre-existing process plan to accommodate the new addition.
Let us consider a simple FPN shown in Figure 17(a), and add a feature F5 to the end of the FPN. This changes the FPN to the one shown in Figure 17(b).

Figure 17: (a) FPN with 4 features (b) FPN with 5 features.

Let the intermediate process plan for the FPN in Figure 17(a) be as shown in Figure 18 with one optimal solution.

Figure 18: PPN with one optimal process plan.

The features affected by addition of F5 are F3 and F4. The corresponding modified nodes are $M3F3/\{F1, F2\}$ and $M4F4/\{F1, F2, F3\}$. And these nodes are valid nodes as per the new FPN (Figure 17(b)). This validity is checked by evaluating the set of completed features and the current feature on the node for adherence to precedence constraints. Generation of the process plan network starts from this node and yields the process plan network as shown in Figure 19.
The newly formed node is highlighted in Figure 19. It clearly illustrates that only the affected nodes are re-generated and not the entire network. As a result, the non-value adding activity of re-generating the entire process plan as in a conventional process planning system is eliminated.

It might be noted that if the start node is the affected node due to feature addition, then most of the nodes have to be re-generated. In such a case, there isn't much of a difference between the batch mode process planning and the incremental process planning. But in the cases such as the one discussed, the incremental process plan is surely an efficient method.

5.2 Feature Deletion

Possibly, the designer could change his mind or the technical specification requires a part feature to be deleted. The proposed incremental process planning system handles this change and as explained in the previous section updates only a portion of the network.
Similar to feature addition, feature deletion also falls into one of the cases illustrated in Figure 20. The FPNs shown in each of the cases (Figure 20 (a) to (d)), are the original FPNs from where feature $F_d$ will be deleted. Deleting a feature that has no precedence constraints may affect all the features in the FPN. At the same time deleting a well constrained feature could affect only the features associated with it.

Incremental feature deletion involves the following major steps. First, all the nodes that contain the deleted feature have to be removed. Then a list of modified features has to be found by comparing the previous FPN and the new
FPN. Based on the modified features, a list of modified process plan nodes has to be found. These nodes will act as parent nodes and the process plan network will be re-generated from these parent nodes.

This feature can be explained with an example. Consider an FPN shown in Figure 21(a), let feature F5 be considered as deleted. This will yield a FPN as shown in Figure 21(b). The process plan network for the feature precedence network in Figure 21(a) is shown in Figure 19. Due to deletion of a feature, the process plan network will undergo a change.

![Figure 21: FPN before (a) and after (b) feature deletion.](image)

The list of modified features will include feature F3 and F4, and their corresponding process plan nodes will be $M3F3/\{F1, F2\}$ and $M4F4/\{F1, F2, F3\}$. These nodes act as parents and process plan network is generated from these nodes.

If there were a design change without any addition or deletion of features, then it would fall into a category, which is a combination of feature addition and
deletion. It will follow the same methodology as in the above sections. First the
list of modified features is found, then the nodes. A decision on whether to keep
the node or delete is made and the remaining part of the network is generated as
earlier, following the decision.
6 Implementation and Testing

6.1 System Architecture

This section describes an overall architecture of the proposed incremental process planning system. It shows the linkage to the CAD system in the upstream and linkage with the CAM (scheduling module) system in the downstream.

The process planning architecture has been based on an object-oriented paradigm, in which features, machines, processes are modeled as objects. The system architecture of the process planner is shown in Figure 22. The complete system is currently under development at Ohio University’s Intelligent Manufacturing Planning laboratory.
Figure 22: System architecture.
An engineering drawing, after identifying features and modeling to encompass the precedence constraints and the process alternatives form the input to the process planning system. Process planning system must access the specification of the part depicted on the drawing. For this purpose a feature-based design approach may be used. In the feature-based system, the part is identified in terms of parameterized manufacturing features, and the relationships among the various features are specified at the early stages of design. One such system may be used to model features. The manufacturing features are obtained from the feature modeler as an input to the process planning system.

As process planning involves large number of variables, there is always a possibility of space search explosion during this process. This is generally a time consuming process. Certain constraints have to be applied to limit the huge search space. The limiting constraints are the constraints put forth by the feature precedence. Feature precedence constraints are created on the basis of geometric, technological, and economic factors. These constraints are depicted on the FPN for a part model. The feature precedence constraints dramatically reduce the search space during process planning. Apart from the feature precedence constraints, the manufacturing cost constraints are also a good source of reducing the search space.
The process planning system consists of two basic modules: 1) feature precedence network representation and validation module, and 2) process planning and cost optimization module. The first module accepts the manufacturing features from the part model and checks for flaws in the feature precedence constraints. For this task the module uses a Petri net-based model to evaluate the precedence constraints. As a result of this module the feature precedence network is validated and acts as an input for process planning, the next module.

The process-planning module determines a feasible set of feature sequences along with a set of machine/operation combination. Generally there are alternative manufacturing processes that can produce a feature. Likewise, several machine types are capable of performing specified processes. A feasible set of alternative machines for a feature is obtained from the machine’s attribute of a feature. These machine details are obtained from inferring from the 3I-PP knowledge base. The process-planning module checks if the processes/machines are selected for all the features in the part model. Once all the features have their machines selected, process planning by operation sequencing is started.

Operation sequencing is the task of arranging the manufacturing features, along with their machine candidates, in a proper order to attain the economical
operation sequence. The operation-sequencing module performs an evaluation on the various machine instances that are possible for each feature. The evaluation is performed based on the machining cost and transportation cost. While sequencing the operations, the feature precedence constraints are strictly adhered. These operations are sequenced to obtain a process plan for a part model with minimal cost. A least cost route search using Dijkstra’s algorithm is employed for operation sequencing. This procedure generates all the possible operation sequences that would yield a least cost process plan.

In case of incremental process planning, which occurs due design changes, all the modules work the same way as explained. Due to design change, there will be a need to modify the existing process plan. An Incremental process plan module decides on which part of the process plan network has to be modified. In the concurrent engineering environment, with the aim of reducing the product development life cycle, there is a need for rapid feedback from the process planner to the designer regarding the cost valuation of the design. As a result the part model and its process plan gets finalized after several negotiations between the designer and the process planner.

The system architecture shown in Figure 22 also shows links between process planner and other modules in the computer integrated manufacturing environment. The manufacturing cost details of optimal process plans generated
at each stage of the incremental process planning are provided to the designer. This cost detail acts as an important aid in design evaluation and selecting the most appropriate design alternative. Feed back from the feature precedence module informs the modeler about the validity of the feature precedence constraints.

The process plan for the part, generated by the process-planning module is sent to a production scheduler. The scheduler generates a schedule in which the parts will be manufactured in the shop floor depending on the availability of the resources. During scheduling the alternative process plans finds its advantage. Depending on the availability of the resources (machines), the scheduler will be able to choose one of the alternative process plans. Thus, generating alternative process plans is useful in scheduling.

The interface between process planner and the scheduling activities described are a part of integrated manufacturing environment, but are outside the scope of this thesis.

6.2 Data Structures

This section describes the implementation details of process planning system. Here a description of the various data structures used in implementation has been explained.
The implementation of the prototype of the process planning system has been achieved using Java. Java provides the object-oriented hierarchy for defining and implementing various schemas required for process planning. It also provides with a graphical user interface (GUI) tool to represent the process plan network.

The GUI created to represent the prototype is shown in Figure 23. The GUI basically contains 3 tabs, one each for PPN, FPN and the validating Petri net. The is provided with buttons for saving a PPN, opening a saved PPN, selecting nodes and arcs in FPN, PPN and Petri net, and a button to fire the Petri net to manually conform validity of the FPN. It may be noted that before starting the process planning, the validity of the FPN is automatically checked by firing the Petri net.
The process-planning module is divided into two major groups; one defines the feature precedence network and the other process plan network. The basic entities required for process planning are features and machines. Feature modeler defines the feature object. The feature object has parameters to represent precedence constraints and machine candidates. Apart from these parameters, the feature object also has geometric information. The machine object basically consists of the type of machine it represents, and the features that may be produced using them. It also has information about the machining cost for each feature.
For representation of feature precedence constraints, FPN is created which consists of a set of feature precedence nodes and arcs. Feature precedence nodes have details about the feature that they represent in the feature precedence network. Feature precedence nodes use the feature precedence information in the feature (next and previous attributes) to form a link with each other in the feature precedence network.

For the purpose of validating the feature precedence, a Petri net model of the FPN is built. The Petri net consists of two types of node; place and transition, and an arc connecting the nodes. The Petri net nodes are modeled as PlaceObject and TransitionObject, and the arcs are modeled as PetriArc. The PlaceObject provides information about the number of tokens it carries and the TransitionObject has information on whether or not it is enabled. Details about Petri net operations were explained in section 3.2.

Once the feature precedence constraints are validated, a process plan network is generated. This process plan network consists of process plan node and arcs. Process plan nodes, defined as NodeObject, basically contain information about the feature that is machined in it, the machine that is used to produce the feature and the list of features that have been machined. These components of a node are stored as attributes in the process plan node object. The manufacturing feature in the process plan node provides the linkage
between the feature model and the process planning system. Any change that happens to the feature is directly reflected on the feature in the process plan node, as they work on the same object. The arcs connecting the process plan nodes are ArcObject. These arcs have details about the cost incurred between any two connected nodes. Basically an arc contains cost of machining at the sink node and the transportation cost to get to that node. If both the sink and the source nodes have the same machine type, the transportation cost is not incurred, as there is absence of necessity for any part to be transported.

With this summary about the data structures of the implementation, let us review the algorithms that were used to create the process planning system.

6.3 Implementation of Algorithms

The process planning system consists of feature precedence network generation module, validation of feature precedence constraint, and process planning with incremental design change and optimization. Various algorithms have been used to obtain the desired result. These algorithms have been explained in this section.
6.3.1 Feature Precedence Network

Feature precedence network is generated with the precedence information available in the feature model. This information is available in the next and previous attributes. The next attribute of a feature has information about the features that can be machined only after it and the previous attribute gives details about the features that have to be machined before it can be processed. With these next and previous attributes, a network of features is generated. The algorithm for generating the FPN is as follows:

The algorithm has two main data structures, the open set and the closed set. The open set stores non-duplicate values of objects that can act as parents and generate nodes. The closed set consists of all the objects that have generated children. The open set never contains an object that is in the closed set. This is the underlying data structure for the following algorithm.

1) Create a start node “S” and put it in open list

2) Remove the first node in the open set and create children for it. The children for a feature are the set of features that can be machined next this feature. Form nodes with these child features to get the child nodes. Connect all the children with the parent node. Put the parent node in the
closed set and append the open set with all the children if they are not in the closed set.

3) Repeat step 1 and 2 until the open set becomes empty.

The children for the start node will be the nodes with features that have their previous as empty. The child for the feature with empty next attribute will be the end node. So, the network starts from the start node and finally converges at the end node.

6.3.1.1 Example

Consider a feature model with the details as given in Table 4. Nodes are generated for representation of FPN. These nodes are listed in Table 5 along with their children. The feature precedence network generated using the above-described procedure is shown in Figure 24.

Table 4: Feature attributes - next, previous

<table>
<thead>
<tr>
<th>Features</th>
<th>Previous</th>
<th>Next</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>-</td>
<td>F2</td>
</tr>
<tr>
<td>F2</td>
<td>F1</td>
<td>F4</td>
</tr>
<tr>
<td>F3</td>
<td>-</td>
<td>F4</td>
</tr>
<tr>
<td>F4</td>
<td>F2, F3</td>
<td>-</td>
</tr>
</tbody>
</table>
6.3.2 Validating Petri Net

The feature precedence network is created from the information available from the feature modeler. This information has to be validated before we make use of it in process planning. For the purpose of validating, a Petri model of the feature precedence network is created. The algorithm for the creation of the Petri net ihas beenexplained in this section.
A FP-Petri net node for our purpose may be defined as connected pair of place and transition, with the direction from place to transition as shown in Figure 25.

![Figure 25: FP-Petri net node](image)

This pair will be addressed as a FP-Petri node in this section. Two data structures, open and closed with the same functionality as described in the section 6.3.1 are also used. The algorithm for formation of a validating Petri net is as follows:

1) Create FP-Petri nodes, as shown in Figure 25, for all the nodes in the feature precedence network,

2) Connect these nodes in the same manner as their corresponding nodes in the FPN, with the arc start at the transition and terminating on the place of the next node.

3) If more than one arc tend to terminate on a place, then a new place is created to terminate the arc. This new place is connected to the sink transition as in Figure 26.
4) The start FP-Petri node is given one token in its place.

![Petri net diagram](image)

*Figure 26: Part of validating Petri net showing connection between FP-Petri nodes.*

Following this procedure produces a feature precedence validating Petri net. This Petri net is fired until the end Petri node’s transition is enabled. If a deadlock occurs, either the feature precedence is malformed or it has non-valid set precedence constraints. As mentioned earlier, Petri net evaluates the feature precedence constraints to check for cyclic constraints only. A cyclic constraint will have a FPN as shown in Figure 27.

![FPN diagram](image)

*Figure 27: FPN with cyclic precedence constraints.*

The cyclic constraint is one of the major flaws in constraints. If a cyclic precedence constraint exists in the FPN, then a process plan network cannot be generated. Thus, it is important to evaluate the feature precedence network
before proceeding to the planning process. Example of feature precedence Petri net for a valid FPN is shown Figure 28.

![Figure 28: (a) A valid FPN, (b) Validating Petri net for the FPN.](image)

### 6.3.3 Process Plan Network with Cost Optimization

Process plan network is generated only if the FPN for the part model is valid. A process plan network consists of a start “S” and an end node “E”. These nodes are the two terminal nodes, which mark the start and end of the process plan network respectively. A process plan node belongs to a NodeObject and the arcs connecting them are ArcObject. As described in section 6.2, the arcs contain the costing details. The process-planning algorithm also consists of open and closed set as described in section 6.3.1. The following steps are used to create a process plan network:

1) The open set starts with the start node “S” in it.

2) The first node in the open set is removed and children are created for it. The child nodes are the nodes with features that can be machined at
this stage. A child node is formed with each of these features as current feature for all of its machine candidates of the feature. For example if feature F2 is the only feature that can be machined after F1, and has two machine candidates, say M1 and M2. Then, the child nodes for the node with F1 as current feature will be M1F2/F1 and M2F2/F1. The current feature and the completed features on the parent form the completed feature list for all its children.

3) The child nodes are added to the end of the open set and the parent node is added to the closed set. Only the child nodes that are not in the closed set are added to the open set. At this stage all the children are expanded for a parent, but only a few of them will be explored to generate children in the next iteration.

4) All the nodes in the open are evaluated for manufacturing cost and they are stored in the ascending order of their cost. First “k” nodes are selected as parents and constitute the expanding set. Children are generated for all the nodes in the expanding set.

5) Step 3 is repeated until the number of required optimal paths ("k") is found.
Once the process plan network is generated, there might be number of nodes that were not expanded. So the network is pruned to obtain only the required number of optimal process plans. The process plan network at this stage has only the required number of optimal alternative plans.

6.3.4 Incremental Process Planning

Incremental process planning is referred to the method of process planning that updates the current process plan whenever a design change occurs in the part model. There are basically two types of updates that may occur in a part model: feature addition and feature deletion. The third type is the modification of features that lead to change in feature precedence may be considered as a combination of feature addition and deletion. The algorithm for incremental process planning with feature addition is as follows:

1) Identify the feature added,

2) Identify set of features that were modified because of this feature addition. This set consists of all the next and previous features.

3) Identify the set of nodes that have these features machined on them. These are the set of modified nodes.
4) Make decisions whether these nodes are valid according to the new FPN. If they do not conform to the new FPN, then the node has to be deleted along with all the arcs originating as well as terminating on the node.

5) The rest of the modified nodes that were found to comply with the new FPN are added to the open set, and all the other nodes are added to the closed set. These open and closed sets are similar to the one described in section 6.3.3. That is to say, only a part of the network will be generated.

6) From here the algorithm follows steps 3 in the section 6.3.3, which explains generating process plan network with cost optimization.

In case of feature deletion the algorithm is almost same as the one described above, with the following few changes:

1) The feature that is deleted is identified and all the nodes that are associated with these features are deleted from the process plan network.
2) The features that were modified by the deletion of the feature are identified and the corresponding nodes are selected as modified nodes from the process plan network.

3) From here the step 4 described in the above algorithm for new feature addition can be followed to generate the process plan network.

In case of modifications in the FPN without addition or deletion of features, the same procedure is followed after identifying the set of modified features and their respective process plan nodes.

6.4 Test Cases

This section provides evaluation of test results, where test cases have been presented in which the prototype system and the algorithms were applied along with an analysis of test results.

Process planning system has been verified on several part model examples. Two of those examples have been described in this section. For each example, the part geometry, its feature precedence network, validating Petri net, issues with process planning and the final optimal alternative process plans are discussed and generated.
6.4.1 A Simple Example

As a first test case, a simple part is used. The part and its FPN were obtained from [26]. This part has nine features. The FPN of the part is shown in Figure 29.

For the purpose of demonstrating the incremental process plan generation, let us build the part by adding features on to the stock. The change in feature precedence due to this addition is shown in Figure 30. Let us start with Figure 30(a), with four features without any precedence constraints.
The optimal process plan is generated and is shown in Figure 31(a). Similarly after every stage of feature addition, the FPN and its corresponding PPN are shown in Figure 30 and Figure 31 respectively. Table 6 shows the machine candidates for each feature along with its machining cost. In this example, all the
features were intentionally assigned to machine M1. So, it should be noted that the optimal process plan obtained from system reduces the manufacturing cost by economizing on setup cost and handling cost which might have been added if the features were machined in different machines.

Table 6: Feature-machine matrix for simple example (cost in dollars)

<table>
<thead>
<tr>
<th>Features</th>
<th>Machine M1</th>
<th>Machine M2</th>
<th>Machine M3</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>1.0</td>
<td>-</td>
<td>5.0</td>
</tr>
<tr>
<td>F2</td>
<td>2.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F3</td>
<td>2.0</td>
<td>-</td>
<td>3.0</td>
</tr>
<tr>
<td>F4</td>
<td>3.0</td>
<td>7.0</td>
<td>-</td>
</tr>
<tr>
<td>F5</td>
<td>2.0</td>
<td>7.0</td>
<td>-</td>
</tr>
<tr>
<td>F6</td>
<td>5.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>F7</td>
<td>3.0</td>
<td>9.0</td>
<td>2.0</td>
</tr>
<tr>
<td>F8</td>
<td>2.0</td>
<td>-</td>
<td>1.5</td>
</tr>
<tr>
<td>F9</td>
<td>4.0</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Figure 31: Incremental process plan generation for simple example
6.4.2 Modified Bendix Example

Let us consider a complex example to illustrate incremental feature addition and deletion. This example is obtained from [26]. The FPN for the Bendix example is shown in Figure 32. This part example has 12 features. Feature interactions have produced precedence constraints that are represented as FPN in Figure 32. The machines that will be used for producing these features are given in Table 7, along with their machining cost.

Figure 32: FPN for Bendix Example
Let us consider a hypothetical design evolution for this part. First the part starts with 4 features (4 slabs- F1, F2, F3 and F4) as shown in fig31 (a). These 4 slabs may be machined in any order as they do not have any precedence constraints imposed. The process plan with 2 optimal solutions for this FPN is shown in Figure 34(a). Let a pocket F7 be added to the part model. This will change the FPN from Figure 33(a) to Figure 33(b).
Figure 33: FPN at various stages of Bendix part design.
Figure 34: PPN for each step of modified Bendix example.
In may be noted that, this addition of feature F7 will modify the intermediate process plan shown in Figure 34 (a). The optimal PPN for this FPN is shown in Figure 32(b).

Now let feature F5 and F6 be added to the part model. The corresponding FPN and PPN are shown in Figure 33 (c) and Figure 34 (c). Similarly, by adding features F9, F10, F11 and F12, the part model is completed. Various stages of FPN and optimal process plan at each stage are shown in Figure 33 and Figure 34 respectively.

This method of building the optimal process plan network reduces the problem space and focuses on obtaining optimal solution. By this way, the combinatorial issue may be bypassed.
7 Conclusion and Future Research

This chapter discusses the research accomplishments and provides a brief summary of the outline of the thesis and its contribution to the area of Computer Integrated Manufacturing. This chapter also discusses the limitations of the thesis and proposes topics for future research.

7.1 Research Summary

The thesis has described a new object-oriented approach to incremental process planning. The process planning system accepts a feature model of a part from the part modeler. These feature models contain information about the precedence constraints among the features. The system starts with the representation of these precedence constraints in the form of a network, Feature Precedence Network. This network has been analyzed using a Petri net model to check for presence of cyclic precedence constraints. Once the FPN is validated, the features were sent to the process planning stage. During this stage, the features had information about the various machines on which they may be processed. This information is currently inherent in the feature, but may be obtained from process selection module.
Once the machine information was available with the feature model, process planning by way of operation sequencing was performed. If the part design underwent a change during the designing phase, then the corresponding process plan was updated incrementally to reflect the change. This type of process planning is called incremental process planning.

The state space search algorithm is the core of the process-planning module and cost optimization is performed using modified Dijkstra's algorithm. The search space was considerably reduced by the incremental network generation and by the precedence constraints specified in the FPN. Without these constraints the search space would be too large and would have lead to a combinatorial explosion.

The implementation of the process planning system using Java API has provided a very good means for object oriented definition of the models and a nice graphical user interface (GUI). The system developed provides a satisfactory proof for the ideas presented in the thesis.

The research has provided considerable contribution to the area of Computer Integrated Manufacturing. Some of the contributions are outlined here.

- Development of an operation sequencing procedure that considers the feature precedence constraints, and cost optimization.
• Application of a process planning procedure that searches and generates only the states that might lead to least manufacturing cost process plan. This considerably reduces the search space and enhances performance of the system.

• The research creates a link between the design and the process planning functions by way of feature model generated by the feature modeler.

• This link is not a one-way link as in many conventional CAPP systems. Instead, a two-way link is created and a feedback about the manufacturability and the cost of manufacturing is provided to the part designer during the design phase.

• Since the system provides alternative process plans, it helps the scheduling activity to make flexible production schedules depending on the availability of resources.

7.2 Limitations

Some of major limitations of the thesis are presented in this section.
- Due to non-availability of the interface between the knowledge base and the process planning system, machine data has been implemented within the feature model before process planning. This limitation could be overcome once the linkage with the knowledge base is established.

- Cost optimization is performed on a simpler costing model due to difficulty of collecting real cost data used in the industry.

- The feature precedence constraints and cost optimization are the only criterion for limiting the search space. The search space still has a potential for a combinatorial explosion in case of unconstrained or sparsely constrained FPN. This could over come by incorporating a heuristics for sequencing the features/operations in an unconstrained feature model.

### 7.3 Future Research

This thesis opens several research topics that may lead to significant contribution to this research area. Some of the future research area have been outlined here:
- Detailed machine and tool specification in each stage of process planning.

- Implementation of a better costing model to take into consideration the cost of tool changes and the re-orientation of the part.

- Incremental scheduling of the process plan in shop floor in accordance with the incremental process planning due to design change.

- Consideration of feature clustering as means for reducing the number of nodes in process plan network.
8 References


Abstract

SRIDHARAN, THIRUPPALLI. M.S. November 2002
Industrial and Manufacturing Systems Engineering

Incremental Generation Of Alternative Process Plans For Integrated Manufacturing (115 pp.)

Director of Thesis: Dušan N. Šormaz

Current developments in computer integrated manufacturing focus on linking different automated activities with the enterprise, in order to efficiently use the information and significantly reduce the production cost and time. The most critical link is the link between design and process planning activity. In addition, shortening the product development cycle and responding to the changed market demand are highly imperative. Traditional CAPP systems operate on a batch mode. That is, process plan is generated for a completed part model. If the part undergoes design change, then the entire process plan network is generated again to accommodate the design change. Instead, if only a part of the process plan network is updated to accommodate the design change, then the process plan is considered to be incremental.

The main objective of this research is to develop a computer-aided process planning system. This system establishes a linkage between part design and
process planning system, where the linkage helps to generate process plan network incrementally. The process plan network also accommodates alternate process plans depending on the number of process sequences and machines available for features. Focus is mainly established on process sequencing aspect of process planning. Further, the CAPP provides a number of optimal process plan alternatives based on manufacturing cost. This CAPP system has been tested for incremental process plan generation with number of example.

Approved: __________________________

Signature of Director