2007

Process planning for reconfigurable manufacturing systems

Amr Ibrahim Mohamed Shabaka
University of Windsor

PROCESS PLANNING FOR RECONFIGURABLE MANUFACTURING SYSTEMS

by

Amr Ibrahim Mohamed Shabaka

A Dissertation
Submitted to the Faculty of Graduate Studies and Research through
Industrial and Manufacturing Systems Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy at the
University of Windsor

Windsor, Ontario, Canada
2007

Your file Votre référence
ISBN: 978-0-494-35081-2
Our file Notre référence
ISBN: 978-0-494-35081-2

ABSTRACT

Today's dynamic market requires fast and responsive manufacturing systems, which led to the development of Reconfigurable Manufacturing Systems (RMS); a new manufacturing paradigm. The machine capabilities, in RMS, change with each configuration. RMS technological enablers allow machines to be designed around products and process plans to be reconfigured in response to changes in these products.

In order to achieve the goal of this work, "RMS Process Planning Approach" was developed. It consists of four stages; the first of which clusters operations that have to be performed together. The second step introduces a new procedure that maps product features to their required machine capabilities, which are represented by a kinematic chain-like format. Accordingly candidate capable machines and their corresponding configurations are identified. Optimal process plans are generated in the third stage using Genetic Algorithms (GAs) based on a novel constraint satisfaction procedure that ensures the feasibility of all produced plans. A novel rule-based semi-generative Computer Aided Process Plans Reconfiguration (CAPPR) approach is introduced in the final stage. It reconfigures existing process plans to accommodate for changes in product requirements and/or availability of system resources. The CAPPR approach minimizes the required hard-type reconfiguration on both system and machine levels by performing less costly soft-type reconfiguration to existing process plans.

The developed approach was demonstrated and validated using two case studies based on examples from literature. It was applied to both RMS and Flexible manufacturing Systems (FMS) environment. The results showed that developed RMS Process Planning Approach is not limited to RMS and can be applied to other manufacturing systems as exemplified by FMS.

This research work advances the existing knowledge about process planning in the RMS domain with regards to macro-level process planning (sequencing, operation selection and selection of machines and their configurations). This work supports the process planner in the decision making activity of the machine assignment / selection and sequencing activities at the initial stages of manufacturing systems design and subsequent changes in products scope.

DEDICATION

To my parents, Ibrahim and Hoda Shabaka;

wife, Naglae.

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation and gratitude to my advisor, Dr. Hoda ElMaraghy, for her contributions, guidance and support which enabled me to complete my research work and present it in an acceptable form. It is her inspiration and guidance that helped me throughout this work. My committee members, Dr. Waguih ElMaraghy, Dr. Michael Wang and Dr. Nader Zamani, are to be highly recognized as well for their time and valuable suggestions.

I would like to thank my colleagues in the Intelligent Manufacturing Systems (IMS) Centre who helped me and shared with me various research problems.

Many thanks to Ms. Jacquie Mummery, the secretary of the Industrial and Manufacturing Systems Engineering Department, and Ms. Zaina Batal, the research administrative assistant of the IMS Centre.

My utmost gratitude to my parents for their continuous sacrifice, care and encouragement throughout my life. A very sincere appreciation to my wife who gave me all the time and support I needed during my study period and for my son who has been a source of inspiration.

I would like to extend my gratitude to all my friends specially Ayman Youssef who have always been a source of encouragement and support in my academic and social life.

Special thanks to Dr. Ashraf Nassef who introduced me to the field of academic research and was a source of encouragement for.

A final appreciation to all who helped me to create my mission to continue seeking knowledge throughout life.

TABLE OF CONTENTS

CHAPTER FIVE

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CIM	Computer Integrated Manufacturing
CM	Cellular Manufacturing
C	Configuration
CAPP	Computer Aided Process Planning
CAPPR	Computer Aided Process Plans Reconfiguration
d	index for TAD
D	Datum tolerance precedence constraint
DB	Database
DB_{Cap}	Database containing the Operation ID the list of candidate machines capable of performing these operations
DB_{MC}	Database containing all possible candidate machines, their configurations and capabilities and the cost of using each machine
DML	Dedicated Manufacturing Lines
FMS	Flexible Manufacturing Systems
GA	Genetic Algorithm
i,j:	index for OC number, i,j = 1,..., NOC
ILP	Integer Linear Programming
L	Logical precedence constraint
m	index for machine number m =1,..., number of machines
M	Machine/station
MC	Machine Configuration
NC	Number of selected Configurations
NOC	Number of OCs
NOP	Number of OPs
OC	Operation cluster
OP	Operation
PCM	Precedence Cost Matrix

PG	Precedence Graph
PP	Process Planning
RMS	Reconfigurable Manufacturing System
RmS	Reconfigurable machining System
RMT	Reconfigurable Machine Tool
RPP	Reconfigurable Process Planning
SA	Simulated Annealing
t	index for tool number, t =1, ..., number of tools
TAD	Tool Approach Direction
TS	Tabu Search
x,y	index for OP number, x,y = 1,..., NOP

CHAPTER ONE

INTRODUCTION

This chapter gives a brief review of the current types of manufacturing systems and RMS, the motivations behind the presented research work, the objectives and the research procedure followed during the research and an overview of the dissertation.

1.1. Evolution of Manufacturing Systems

Manufacturing is an industrial activity that changes the form of raw material to create products. Manufacturing systems were developed to use different inputs through particular processes to maintain desired output or product. The evolution of manufacturing systems through history could be traced through the relation between man and technology. The manufacturing systems passed through different paradigms to respond for the increasing size and dynamics in the market of today, which is full of competitiveness. Shorter product life-cycles, unpredictable demand, and customized products have forced manufacturing systems to operate more efficiently and effectively in order to adapt to changing requirements. Tougher competitive situations have led to increasing attention being paid to customer satisfaction.

Manufacturing systems started from job shops that contain general purpose machines. Job shops are characterized by low volume and high variety and have evolved into Dedicated Manufacturing Systems (DMS) driven by economy of scale. DMS are characterized by high volume and low variety. The needs for mass customization and greater responsiveness to changes in products lead to the concept of Flexible Manufacturing Systems (FMS). FMS address mid-volume, mid-variety production needs [H. ElMaraghy 2005].

1.2. Reconfigurable Manufacturing Systems (RMS)

Today's global markets are characterized by shorter product life cycle and unanticipated change in the demand. For companies to compete in the market, they must reduce drastically the manufacturing system lead-time, which includes the time to design, build or reconfigure, and start production. This led to the concept of Reconfigurable Manufacturing Systems (RMS). RMS was introduced by Koren *et al.* [1999]. They define Reconfigurable Manufacturing Systems as:

> "A Reconfigurable Manufacturing System (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements"

RMS will contain customized flexibility enabling cost-effective reconfiguration when new products (or product changes) are introduced, or when demand fluctuates. This customized flexibility is realized by adjustable resources at the system level (e.g., adding/removing machines, changing system layout) and the machine tool level (e.g., adding/removing an axis of motion and/or a spindle, integrating new process monitoring technology) that allow for quick and reliable (i.e., cost-effective) reconfiguration. Reconfigurable manufacturing systems will be characterized by [Koren *et al.* 2001]:

- Production capacity that is readily scalable to accommodate fluctuations in market demand,
- Production functionality that is rapidly adjustable to new products, and

- Structures that are designed to be up-gradable with new process technology.

The type of reconfiguration falls into one of two categories, it could be either logical (soft) or physical (hard) reconfiguration [H. ElMaraghy 2002 and 2005]. Table 1.1 shows some examples of the different levels and types of reconfiguration.

Table 1.1: Examples of the Different Levels and Types of Reconfiguration for an RMS.

		Type of Reconfiguration	
		Logical (Soft)	**Physical (Hard)**
Level of Reconfiguration	**Machine-Level**	G-code Machine Controller (Open Architecture)	Machine Structure Number of Spindles
	System-Level	Factory Software Process Planning Rerouting Rescheduling	Machine Layout Machine Addition/Removal Material Handling

The main components of RMS are Computer Numerical Control (CNC) machines and Reconfigurable Machine Tools (RMT), a new type of machines that have a modular structure enabling reconfiguration of its components [Landers *et al.* 2001]. This manufacturing system will require many new approaches to be considered that are different from traditional manufacturing systems, for example; the design of new modular RMTs, the design of the controllers, which can control these new machines that have variable capacity (e.g. addition/removal of spindle) and capabilities (e.g. addition/removal of axes). In addition, since the system capacity and capabilities change, there should be a different approach for process plan generation. In RMS reconfiguration affects quality of part, machine stiffness and many other aspects, however, the main focus of this work is on macro-level process planning (sequencing, operation selection and selection of machines and their configurations).

1.3. Motivation

The new RMS needs an appropriate Computer-Aided Process Planning (CAPP) system to cope with the new characteristics of the RMS where the system is subject to different configuration changes in both the hardware and software levels. As mentioned earlier, these configuration changes include changes of/within machines or tools, add/remove machines or tools, changes in resources and the location or layout of machines.

Process plans are developed according to the current manufacturing system capabilities. However, in RMS the capabilities of the machines and manufacturing system change with each configuration and may result in changes in process plans. For this reason there is the need for methodologies that achieve concurrent process planning and machine selection. To achieve this there will also be the need for a generic representation of the required machine capabilities to manufacture different features.

1.4. Objectives and Approach

The objective of the proposed work is to develop an approach for process planning that makes use of the high reconfiguration capabilities of the machines within a reconfigurable manufacturing system environment. These capabilities enable the process plans to be reconfigured and the machine configurations to be tailored according to the product feature requirements.

The purpose of this thesis is:

To show that RMS technological enablers allow machines to be designed around products and process plans to be reconfigured in response to changes in these products.

The goal of this thesis is achieved using a novel four-stage approach within the following scope:

1. The RMS consists of CNC machines and RMTs that have modules that could be added or removed to change the RMT machining capabilities.

2. The generated output process plan considered has multiple-aspects on the machine level (selection of machine types and their corresponding configurations) and the operational level (clustering of operations, sequencing of operations clusters, sequencing of operations, assigning operation clusters to machines, tool used for each operation and the tool approach direction used for each operation).

3. At any point in time a machine might be unavailable or a new part might be introduced to the manufacturing system resulting in process plan reconfiguration.

Figure 1.1 shows an overview of the proposed "RMS Process Planning Approach" which contains four major stages discussed in more detail in the following chapters.

- Inputs to the proposed approach are; part specification, operations precedence graph and the operations required which include operation type, tool approach direction (TAD), the candidate tools for each operation and a database containing all the currently available CNC machines, RMTs and their modules (configurations).

- Stage I (Operation Clustering Stage) uses the input operations precedence graph to form operation clusters based on the type of tolerance and logical constraints between operations.

- Stage II (Generating Machine Structure Stage) utilizes the capabilities of the RMT to generate the machine structures that are capable of producing each operation cluster using the available machines from DB_{MC} [Shabaka and H. ElMaraghy 2004, 2005 and 2006c]. A kinematics like representation of the machine structure is used to generate the capable machine structures [Bohez 2002]. All generated machine structures are stored in a database (DB_{Cap}) accompanied by the corresponding operation cluster. This process is useful for part changes with similar operations because it makes use of previously generated machine structures.

- In Stage III (Generating Optimum Process Plan Stage) a new developed macro process planning optimization model [Shabaka and ElMaraghy 2006a and 2006b] using GAs to obtain the optimum machining sequence, machine assignment and assigning the tools to the operations is proposed. The model generates the optimum process plan based on the part data used as input to stage I. A new process planning representation method is used to represent the new reconfiguration aspect of the RMTs. Genetic Algorithms (GAs) are used for optimization to minimize the manufacturing cost. A constraint satisfaction procedure is developed that guarantees the feasibility of all the generated process plans during the optimization process. Also the proposed method introduces a new concept of using continuous domain GAs to overcome drawbacks of previously used methods. The output of this stage will be the machine to use, its configuration, the operation clusters assigned to each machine, the sequence of operations, the TAD used for each operation, the tool used for each operation and the cost of production.

- Stage IV (Process Plan Reconfiguration Stage) generates the new reconfigured process plan in case a change occurs in the current manufacturing system [Shabaka and ElMaraghy 2007] due to change in part being produced or a machine becomes unavailable.

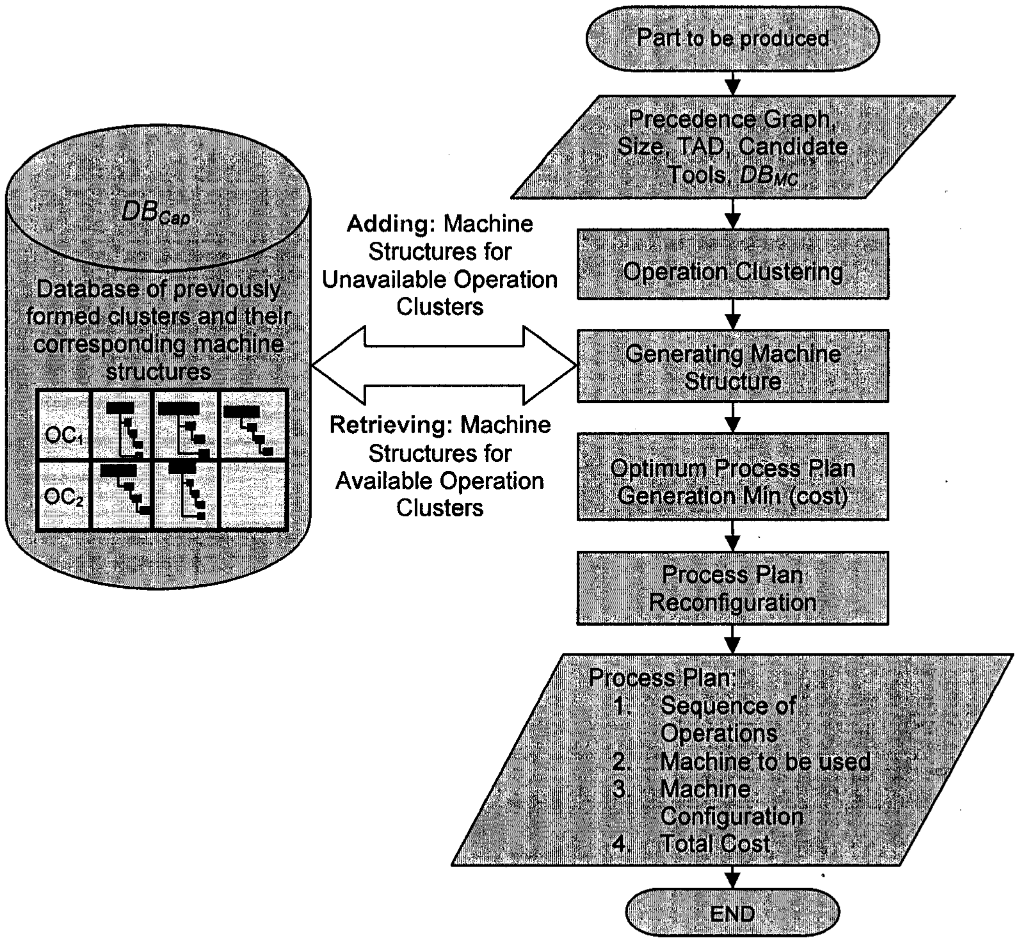


Figure 1.1: Overview of Approach [Shabaka and H. ElMaraghy 2004].

- Two case studies are presented to demonstrate the use of the developed RMS Process Planning approach and verify the results obtained in each of above-mentioned steps.

- All procedures and algorithms were developed using MATLAB® software and plotted using Excel.

1.5. Overview of the Dissertation

The dissertation is composed of eight chapters and five appendices:

- Chapter one includes the motivation, research objective, thesis and approach.

- Chapter two presents a review of the related literature highlighting the gaps in this area of research.

- Chapter three presents the RMS Process Planning Approach. The chapter starts with the basic assumptions related to the problem definition, the inputs and outputs, and ends with an overall description of the approach.

- Chapter four presents both stages I and II. It provides detailed steps of how the clustering procedure and the generation of the required machining requirements is carried out. The kinematics like structure representation of the machine tools is also presented. RMT assumptions are also discussed. An example is provided for demonstrating both stages by their application to a case study.

- Chapter five presents stage III of the proposed approach. A mathematical formulation of the process planning model is presented. A constraint satisfaction procedure is presented and the use of GAs to solve the optimization problem is described. The model is verified using a case study based on an example part from the literature. The result of the optimization technique is presented. The developed procedure is applied to a traditional manufacturing system (containing no RMT) to illustrate the generality of the developed algorithm.

- Chapter six presents stage IV of the proposed approach and two scenarios are investigated; i) if a machine becomes unavailable, and ii) if a new part is introduced. The model is illustrated using a case study.

- Chapter seven presents the application of the overall approach to a new case study.
- Chapter eight concludes the dissertation, highlights the scientific contributions and provides suggestions for future research.
- The dissertation has four appendices. Appendix A that contains the input information for the first case study. Appendix B contains the machine database used in most of the examples and it describes the elements in the database fields. Appendix C contains the RMT cost used. And Appendix D provides a brief description of GAs and its operators.

CHAPTER TWO

LITERATURE REVIEW

This chapter provides a review of the literature related to process planning and highlights the gaps in this area.

2.1. Introduction to Process Planning

The planning process is the act of preparing detailed operating instructions for turning an engineering design into an end product i.e. the part [Gu and Norrie 1995]. Process planning is a multi-decision making activity that determines the operation selection and operation sequencing which involves a great deal of manufacturing data. Operation selection and sequencing is one of the most critical activities for manufacturing [Reddy *et al.* 1999].

Computer Aided Process Planning (CAPP) is the use of software tools to ease in storing, retrieving, generating and updating of process plans. CAPP was first introduced by Neibel in 1965 when he had the idea of using the speed and consistency of computers to aid in the development of process plans. Alting and Zhang [1989] provide a good overview of the CAPP systems for the period 1965-1988, they based their review on more than 200 technical papers and 156 CAPP systems. H. ElMaraghy [1993a] provided perspectives on CAPP systems, classified process planning activities and outlined challenges that require further research. In addition, Kirtisis [1995] presents a review of knowledge-based expert systems for process planning. His review was based on a questioner he carried out. He also gave a summary and the main characteristics of 52 prototype systems developed in the period between 1981 and 1992.

H. ElMaraghy [2006] focuses on process plans and planning functions as the important link between the features of generations of products/product families and the features, capabilities and configurations of manufacturing systems and components throughout their respective life cycles. The evolution of manufacturing paradigms and the manufacturing system life cycle were discussed and the evolution of products was illustrated and classified.

Figure 2.1 shows the IDEF-0 representation of process planning. The IDEF representation shows the Inputs, Controls, Mechanisms and Outputs of process planning.

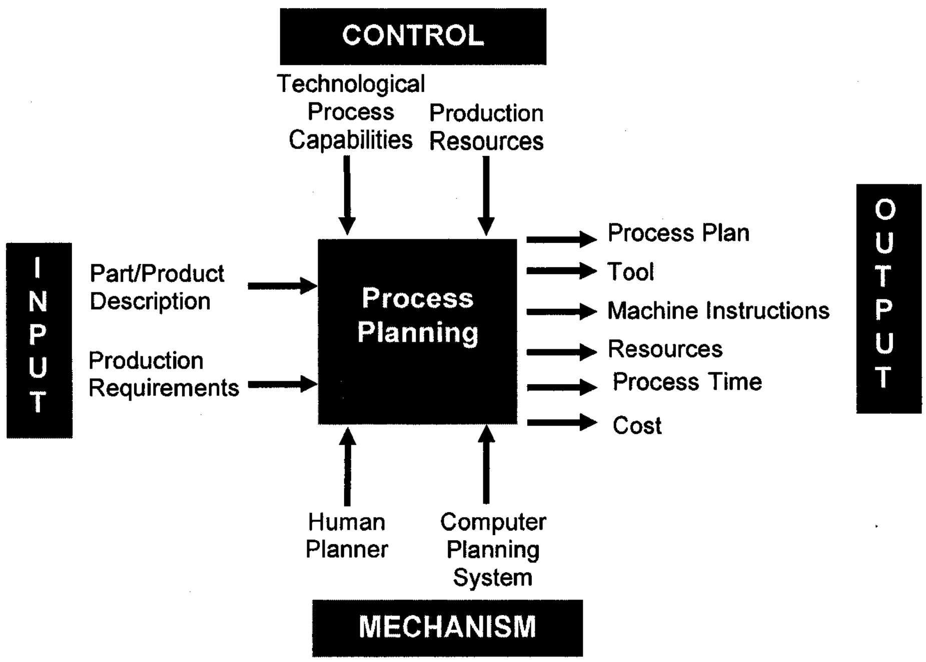


Figure 2.1: IDEF-0 Representation of Process Planning Activity [ElMaraghy 1993a].

Process planning has many functions that are different from one author to another. Figure 2.2 shows the functions of process planning according to different authors [Ciurana *et al.* 2003]. H. ElMaraghy [1993a] illustrated the different planning activity levels and the planning output for each level (Figure 2.3).

Process planning is not applied only in the field of metal removal, but it has many applications (Figure 2.4). The focus of process planning in this thesis is concerned with metal removal for prismatic parts on the macro level.

Author point	Weill et al. 1982	Alting and Zhang 1989	Gu and Norrie 1995	Hallevi and Weill 1995
1		Interpretation of product design data	Identification of design requirements	Preliminary analysis of a mechanical part
2	Selection of operations and tools	Selection of machining processes	Selection of machining processes	Seleection of machining processes (operations), tools and cutting parameters (cutting speed)
3	Selection of machine tools	Selection of machine tools	Selection of machine tools to perform the required machining operations	
4		Determination of fixtures and datum surfaces	Selection of fixtures and setups	Selection of fixtures
5				Grouping of processes into JOBS
6				Selection of Machine tools
7			Selection of cutting tools	
8	Sequencing of operations	Sequencing the operations	Selection of operation sequences	Sequencing the operations according to precedence relationships
9	Grouping of operations			
10	Selection of holding devices and datums			Selection of workpiece holders and dimensional data references
11	Selection of inspection devices	Selection of inspection devices		
12	Determination of production tolerances	Determination of production tolerances		
13	Determination of machining conditions	Determination of the proper cutting conditions	Selection of cutting conditions	Final preparation of the process planning file
14			Planing of cutting trajectory	
15	Determination of cutting times and costs	Calculation of the overall times		
16	Editing of process sheet	Generation of process sheets including NC data	Generation of CNC programs	
17			Verification of the CNC programs	

Figure 2.2: Functions of Process Planning According to Different Authors [Ciurana *et al.*, 2002].

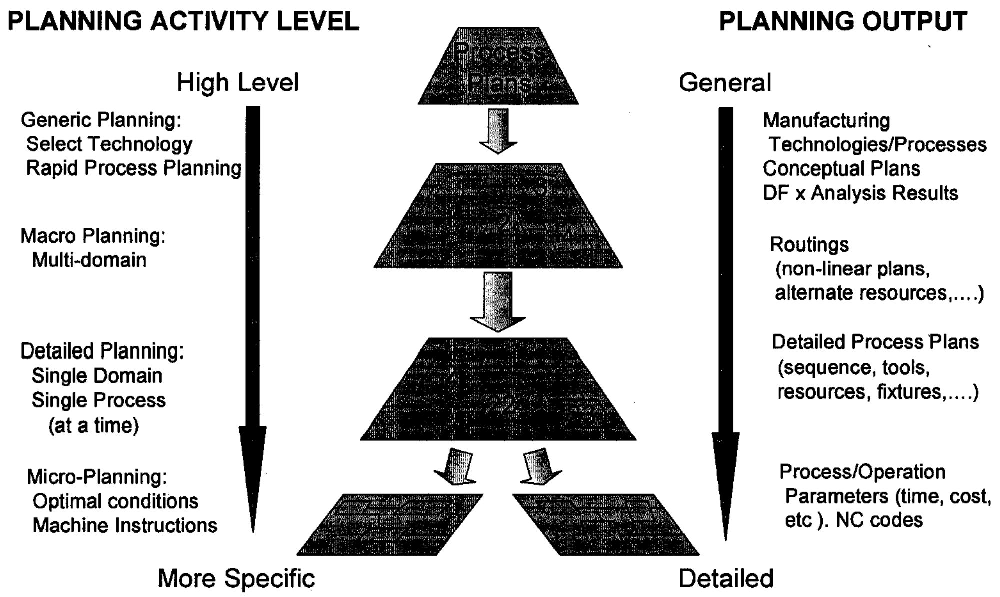


Figure 2.3: Hierarchy of Process Planning [H. ElMaraghy 1993a].

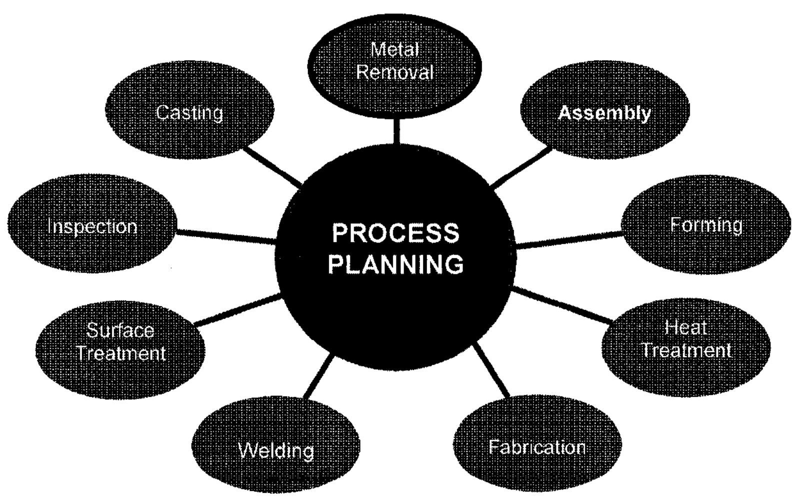


Figure 2.4: Applications of Process Planning.

Benefits of CAPP systems include:

- Reduced process planning and production lead time; faster response to engineering changes.
- Greater process plan consistency; access to up-to-date information in a central database.
- Improved cost estimating procedures and fewer calculation errors.
- More complete and detailed process plan.
- Improved production scheduling and capacity utilization.
- Improved ability to introduce new manufacturing technology and rapidly update process plans to utilize the improved technology.

2.2. Approaches to CAPP

Alting and Zhang [1989] classified the approaches to CAPP systems into Variant, Generative or Semi-Generative. ElMaraghy, H. [2006] presented classification of the various process planning concepts.

2.2.1 Variant Approach

The Variant approach is comparable with the traditional manual approaches. In the variant approach, process plans for new parts are generated by retrieving a master plan for a similar product and making the necessary modifications for the new part. This method of planning is based on the idea of grouping parts into families using Group Technology (GT) method. When using the variant approach, a coding and classification system is needed for the parts. It also requires human intervention to edit the master plan. Table 2.1 summarizes the advantages and disadvantages of the variant method.

Table 2.1: Advantages and Disadvantages of the Variant Method.

Advantages	Disadvantages
• It is well suited to medium to low production mixes • It takes little time to develop (setup), compared to the generative approach. • It can be used with other CIM modules	• While it's fast to setup, it's considered slow compared to the generative approach • More error prone

2.2.2 Generative Approach

In the generative approach, process plans are generated from scratch, (i.e. no retrieval of plans or editing takes place). The Generative system uses the engineering specifications given to the system in the form of graphical and textural information. Regarding the objectives of an ideal generative CAPP system, a truly generative process planning system in any domain is yet to be realized [H ElMaraghy 2006]. Generative Systems consists mainly of algorithms, heuristics, manufacturing knowledge in the form of rule based systems or any other form like decision tables or trees. The generative approach has the advantage that it runs fast when planning, but its draw back is that it requires a more extensive setup.

2.2.3 Semi-Generative Approach

The Semi-Generative (Hybrid) approach is not a fully generative system but rather a hybrid between both the variant and the generative approaches. It can be characterized as an advanced application of variant technology employing generative type features. Different methods for semi-generative process planning are:

- The variant method can be used to develop the general process plan, and then the generative method can be used to modify it.

- The generative method can be used to create as much of the process plan as possible then the variant method can be used to fill in the details.

2.3. Process Planning for Different Manufacturing Systems

The method or approach of process planning relies upon understanding of the manufacturing system, which is an important step to create relevant plans. Traditionally, in CAPP systems, when generating process plans, the manufacturing system was viewed as a static system and only one process plan is developed. The term static system means that the system has a fixed configuration, e.g. Dedicated Manufacturing System (DMS), which in turn implies fixed capability and capacity. However in the past decade there has been research carried out for developing CAPP systems that generate alternative process plans because of the dynamic nature of the manufacturing system (Flexible Manufacturing Systems (FMS) and RMS).

Although, no previous publications have tackled the influence of the type (FMS, RMS, ..etc.) of manufacturing systems on the process planning activities, there are different trends in process planning research suitable or designed for various available manufacturing systems. In this section a brief explanation of different manufacturing systems will be provided, and a review of different approaches of process planning for each system will follow. Manufacturing systems included in this review are Dedicated Manufacturing Systems (DMS), Cellular Manufacturing systems (CM), Flexible Manufacturing Systems (FMS) and Reconfigurable Manufacturing Systems (RMS).

2.3.1 Overview of Different Manufacturing Systems

In DMS, the manufacturing system is customized and built for a specific product at a fixed capacity. There could be multiple tool engagement, which results in high productivity. The DMS has the disadvantage that it is built for a specific product, and any change in the product will result in a change in the structure of the manufacturing system.

Cellular manufacturing is all about grouping the production equipment into machine cells, where each cell specializes in production of a part family [Groover 2001]. In the design of a CM system, similar parts are grouped into families and associated machines are gathered into groups so that one or more part families can be processed within a single machine group.

On the other hand, in FMS a variety of parts are produced on the same system. The system will include CNC machines and other programmable automation. In FMS, the CNC machines are general purpose and not manufactured around the part. This has the disadvantage of making the system expensive because the CNC machines will have more capabilities than that required to manufacture the part. Table 2.2 gives an overview of the different features of DMS, FMS, and RMS to have a better understanding of the different manufacturing systems.

Table 2.2: Characteristics of Different Manufacturing Systems [Adopted from Koren 2005].

	DMS	RMS/RmT	FMS/CNC
System Structure	Fixed	Adjustable	Adjustable
Machine Structure	Fixed	Adjustable	Fixed
System Focus	Part	Part Family	Machine
Flexibility	No	Customized	General
Scalability	No	Yes	Yes
Simultaneous Operating Tools	Yes	Yes	No
Cost	Low	Intermediate	High

2.3.2 Process Planning for DMS

Process planning for this specific manufacturing system is quite unique due to its overlap with the system design activity. This is due to the fact that in a DMS, the system is designed according to the workpiece being produced. For this reason, in a DMS, the process plan is developed for the part being produced and the manufacturing system is built according to the required plan. There were no publications found that dealt with the process planning issues with regards to DMS in specific.

2.3.3 Process Planning for CM

Cellular manufacturing is one of the most important wide spread manufacturing systems. Cellular manufacturing depends heavily on the application of Group Technology (GT) principles to manufacturing. In CM systems, the Variant approach, mainly a variation of a master plan, is used to perform high and low level process planning. For each part's family, a master plan is developed for a master part. When a part is input to the system, the family to which the part belongs is determined, and then the master plan is retrieved and edited according to the part's details. In some instances, Semi-Generative process planning is used for tackling CM to overcome the disadvantages of the Variant approach.

Reynolds *et al.* [1993] argued that group technology should be combined with process flow analysis and featured-based part matching to provide the user with intelligent assistance in selecting previously created process plans for use in variant process planning. This is the determining factor in the prevention of process plan proliferation and in the achievement of a high degree of process plan standardization.

Joshi *et al.* [1994] discussed a Group Technology (GT) and Computer Aided Process Planning (CAPP) installation which is applied to CM system at an international manufacturer and marketer of specialized hydraulic machinery. A GT software shell is described, and for the architecture that is defined, a formal model of the process planning system is developed. The model is used as the basis for the creation of a generic Semi-Generative CAPP system.

Britanik and Marefat [1995] and Marefat and Britanik [1998] proposed an approach, which is Semi-Generative. They utilized techniques, which they named case-based for multi-level process planning selection. The case-based methodology involves retrieving old feature plans generated from past experiences, modifying them to fit the part at hand and abstracting and storing the new plan for future use. A hierarchical method and networks for merging feature sub-plans into a global plan for the part has been presented. The resulting global plan is efficient because the number of fixtures and tool changes is minimized. In addition, the planner has the capability to use multiple cases in the process of constructing a new plan, providing more effective utilization of the planner's previous experiences. The planning provides a formal approach to case-based process planning.

Marefat and Britanik [1997] focused on the development of an object-oriented case-based process planner, which combines the advantages of the Variant and Generative approaches to process planning. An advantage of object-oriented design includes structured and explicit representation of the knowledge of the system. In an object-oriented process-planning system, classes are created to represent the declarative knowledge of the system. The procedural knowledge is captured by the protocols in the created classes. A process plan for a part is generated by message-passing among these classes.

Yu *et al.* [2001] presented the concept of feature variation and analyzed the geometry variation. They discussed the data structure of variant features and its definition in detail and the principle of feature-based modeling system. Using variant-feature-based modeling system, the problems of features with complex process planning such as T-slot, dovetail-groove and pocket were solved successfully.

2.3.4 Process Planning for FMS

The context of process planning for FMS falls into two categories:

- Flexible Process Planning
- Integrating Process Planning and Scheduling

2.3.4.1 Flexible Process Planning

Since FMS provides an increasing capability of the system for product variability and the system flexibility, there arose the need for developing process plans that are capable of representing alternative processing sequences and alternative manufacturing resources. Such plans that provide alternative process plans are called flexible process plans or non-linear process plans. Hutchinson and Pflughoeft [1994] defined three classes of process plan flexibility. The three classes are Sequence Flexibility, Process Flexibility and Machine Tool Flexibility.

ElMaraghy, H. [1993a] indicated that dynamic process planning in a reactive environment as one feature of the future perspectives of CAPP. In this perspective, the necessity of CAPP to consider alternative resources, alternative routes and alternative processes is mentioned as well as user defined evaluation methodologies for these alternatives. The work shows that CAPP is in an essential key for achieving a computer-integrated manufacturing (CIM) system, where design, CAPP, and production planning and control (PPC) integrated into one.

Gupta and Gali [1993] stated that advanced manufacturing systems such as an FMSs normally offer alternate feasible routes for part production. Hence, CAPP systems should be able to generate cost-efficient alternate process plans that can be adopted to solve such a problem. They presented the design and development of a CAPP that generates least-cost alternate plans. The developed system takes production capacity, and operation cost into account to determine feasible process plans for a product mix. Their proposed methodology integrates other production planning and control functions with process planning functions. A Variant approach based CAPP system and an interface with a simulation model have been developed. The system generates alternate process plans and evaluates them for a set of performance criteria.

Kruth *et al.* [1996] explained some methods that were used to improve the response time of a newly developed CAPP system that is capable of generating non-linear process plans. They introduced a method to improve the performance, which they called Opportunistic Process Planning. This approach consists of dividing the features into two sets; one set, called the important features, is used to generate a process plan; the other set, called non-important features, are added to the process plan afterwards. This generation method highly resembles the reasoning pattern of a human process planner. They introduced another method for performance improvement, which they called Feature Grouping. This method combines features that have strong resemblance and considers them as only one feature during the process plan generation. Other reasoning methods, described in their work, are: combined Variant/Generative planning and constraint-based search. They claim that these methods reduce the search space significantly.

2.3.4.2 Integrating Process Planning and Scheduling

A common problem faced by shop floor personnel is that the schedules generated at the planning level are infeasible most of the time. In normal practice, process planning is performed before scheduling. The operational decisions reached at the process planning stage limit the alternatives that might be used to make improvements during scheduling. This is because there is a gap between process planning and scheduling which needs to be filled in order to react to unexpected events (e.g. machine failure, bottlenecks, material shortage, etc.) on the shop floor. In accordance with this view, there arose the need for an integrated process planning and scheduling system for generating more realistic and flexible process plans and schedules to be used on the shop floor.

ElMaraghy,W. [1992] justified the need for integration between CAPP and Production Planning and Control (PPC) and proposed an approach to CAPP and PPC integration. The integrator module addresses the time dependant issues related to event handling, communications, database updating and response time (short, medium and long). They also discussed reactive planning environment (RPE) module implemented by Stranc [1992].

Eversheim and Schneewind [1993] mentioned that although the traditional CAPP approaches are linking Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM), they are usually done separately. ElMaraghy [1993a] pointed out that CAPP is an essential key for achieving a Computer-Integrated Manufacturing (CIM) system, where design, CAPP, and Production Planning and Control (PPC) are integrated in one. Also Gerard *et al.* [1999] presented some major issues relating to the integration of process planning and PPC for FMSs. Their work showed that the performance of an FMS can be significantly improved and FMS capabilities more effectively utilized by integrating process planning and PPC functions.

Kempenaers *et al.* [1996] presented a new collaborative approach that is based on production constraints as a means to realize a feedback from scheduling to process planning. They also describe the results of the ESPRIT project COMPLAN, which aims at the implementation of an integrated automatic process planning and scheduling system based on the concept of non-linear process plans.

Saygin and Kilic [1999] and Saygin [2000] highlighted the importance of integration between process planning and scheduling in FMS. Saygin and Kilic [1999] presented a four-stage framework that integrates flexible process planning with off-line scheduling with the objective of minimizing completion time. They also discussed flexibility in process planning, including process flexibility, sequence flexibility, and alternative machine tools.

Saygin [2000] stated that future CAPP systems are required to provide not only internal flexibility in order to successfully cope with the changing product specifications, but also external flexibility to facilitate easy integration with other manufacturing planning and control functions, and associated heterogeneous software platforms not only with a single manufacturing enterprise but also among various manufacturing enterprises under a virtual enterprise. Internal flexibility includes mainly sequence flexibility, process flexibility and machine tool flexibility. It also includes ease of substituting existing system building blocks such as cutting tool selection module in a CAPP system with enhanced modules, ease of adding new building blocks and finally ease of customization in terms of re-defining the parameters, decision making logic, and the relation among them.

Table 2.3 shows extra papers in addition to those mentioned above with regards to the two approaches used in FMS.

2.3.5 Process Planning for RMS

In the literature, there has been very little research observed with regards to the application of process planning to the new paradigm of reconfigurable manufacturing systems.

In RMS Ling *et al.* [2000a and 2000b] investigated operations that could be carried out using gang drilling and related them to process planning for RMS because the machine tool is designed around the part. Also they stated that similarities across a part family should be recognized to specify reconfigurable capabilities of the machines, and also alternative solutions for the operation patterns should be identified to account for system design constraints. Their focus is only on identifying clusters to be drilled together for gang spindle drilling. The solution process comprises three major steps; (i) Single part operation clustering, (ii) multi-part operation clustering and (iii) Setup planning and Machine selection. In multi-part operation clustering, operation clusters are grouped across a part family, the parts of which have similar machining requirements. The work demonstrated mainly focused on hole pattern identification of the same size and steps (ii) and (iii) were not shown.

This approach is a good starting step in the field of process planning in RMS but is only limited to the clustering operation which is only one step in process planning. In addition, the work is confined to clustering of drilling operations of the same hole size which is also a very specific and limited domain.

Table 2.3: Process Planning Approaches in FMS.

Author	Flexible PP	Integrating PP & Scheduling
ElMaraghy,W. [1992]		√
ElMaraghy,H.and ElMaraghy,W. [1993b]	√	√
Stranc [1992]	√	√
ElMaraghy,H.[1993a]	√	√
Eversheim and Schneewind [1993]		√
Gupta and Gali [1993]	√	
Hutchinson and Pflughoeft [1994]	√	
Sormaz and Khoshnevis [1995]		√
Kruth *et al.* [1996]	√	
Kempenaers *et al* [1996]		√
Gerard *et al.* [1999]		√
Saygin and Kilic [1999]		√
Saygin [2000]	√	√
Kim *et al.* [2003]		√
Usher [2003]	√	√
Muljadi and Ando [2005]	√	
Fuqing *et al.* [2006]	√	√
Jain *et al.* [2006]	√	√
Wang *et al.* [2006]	√	√
ElMaraghy, H. [2006]	√	√

2.4. Flexible and Dynamic Process Plans

2.4.1 Flexible (Nonlinear) Process Plans

The objective of flexible process plans is to generate in advance a number of manufacturing alternatives from which the production planning function can select to create good schedules. A flexible process plan can be decomposed into several conventional linear process plans (figure 2.5). Though the process planning may operate in a just-in-time mode (i.e., the process planning starts just before manufacturing a part), process planning is not based on the real shop state. Resource planning then uses the set of alternative process plans generated in advance.

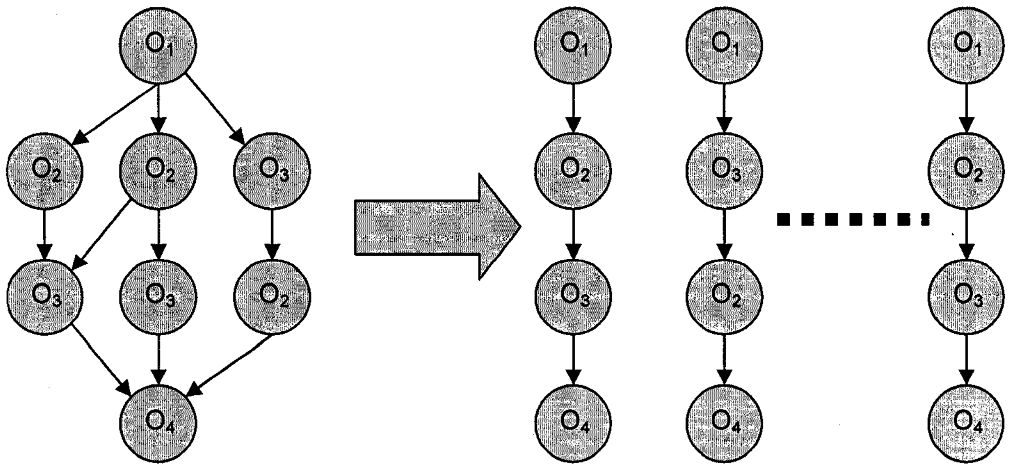


Figure 2.5: Decomposing of a Flexible Process Plan into Equivalent Linear Process Plans.

Table 2.4: Advantages and Disadvantages of Flexible (Nonlinear) Process Plans.

Advantages	Disadvantages
• Relates to the possibility of improving (off-line) scheduling performance and reacting quickly to disturbances on the shop floor.	• The length of time needed to generate a number of process plans and the added complexities for the scheduling function to handle alternatives. • Process planning is not based on the real shop state.

2.4.2 Dynamic (Closed-loop) Process Plans

In dynamic process plans, a process plan for a part, is generated on request of the scheduling function. When an operation has been completed on a machine, the scheduling function selects one or more parts from a list of current parts awaiting their next operation on that machine. The process planning function then is applied to generate a list of possible operations. Subsequently, the scheduling function selects one of the operations, and the corresponding NC program is generated. In this approach, the complete operation sequence of a part is not determined in advance but in parallel to manufacturing.

Table 2.5: Advantages and Disadvantages of Dynamic (Closed-Loop) Process Plans.

Advantages	Disadvantages
• It enables the generation of optimal process plans and schedules. • Avoids the generating of alternative process plans (i.e., a sequence of operations) that are not used on the shop floor.	• Adoption of a step-by-step local view, which limits the solution space for subsequent operations • It is difficult to organize the reapplication of process plans

2.5. Clustering

In this work, the operations clustering activity groups the operations that have to be performed together in one cluster so that they are carried out on the same machine. A group of operations performed on the same machine without changing the machine is called a set-up. In traditional manufacturing systems the machine tools are fixed and not modular, for that reason the grouping of operations depended on the machining tools available. Normally the main objective behind set-up planning is to maximize the number of machining operation which are performed in a single set-up (i.e. minimize the number of set-ups to manufacture a part). An important concern in setup planning is maintaining the accuracy required for the part being manufactured. In literature, most research in this area focused on:

- Preserving feature tolerance relationship.
- Feature interactions.
- Tool approach directions.
- Good machining practice.

Table 2.6 shows the work carried out in this area and clustering method used.

Table 2.6: Review on Operation Clustering.

	Clustering Criteria				
Authors	**Tolerance Based**	**Minimum # of Setups**	**Other criteria**	**Method**	**TAD**
Delbressine *et al.* [1993]	√			Feature Based	
Macchiaroli & Riemma [1994]			Avoid tool conflict	Heuristic	
Chu & Gadh [1996]		√	Group same TAD in one setup	Knowledge-Based	√
Demey *et al.* [1996]	√	√		Feature Based	
Ozturk *et al.* [1996]	√			Feature Based	√
Zhang, H. [1999]	√			Graph Theory	√
Ling, *et al.* [2000a,b]	√			Pattern Identification	
Zhang *et al.* [2001]	√		Minimize Location Error	Graph-Based	
Contini,& Tolio [2004]		√		Graph-Based Approach	√
Stampfer, M. [2005]		√		Expert Systems	

Contini and Tolio [2004] proposed a method to define near-optimal set-up plans for prismatic workpieces when multiple parts can be mounted on the same pallet. Set-ups are determined taking into account TAD and the constraints among the required operations. Starting from the results of the set-up planning, the configuration of the pallet can be defined and taking into account the pallet configuration, the optimal machining centre for specific manufacturing needs is selected.

2.6. Optimization Techniques Used In CAPP

Development of various feasible plans and identifying the best solution has been proven to be NP-complete [Reddy *et al.* 1999]. For this reason in the past 10 years there has been an increasing interest in solving the process planning problem using GAs.

Process planning normally is carried out in a linear manner. The operation selection (Machine, Tool and Tool Approach Direction (TAD) selection) is carried out first then followed by operation sequencing. Zhang *et al.* [1999] stated that the decision-making tasks involved in operation selection and operation sequencing have to be carried out simultaneously to achieve an optimal process plan. They introduced an approach using SA that models process planning in a concurrent manner to generate the operation selection and sequencing simultaneously.

When considering GAs for process planning there have been two major areas of consideration; the choice of objective function and how to obtain a feasible process plan.

2.6.1 Objective Function

When considering the objective function Dereli and Filiz [1999] based their objective function using a reward/penalty matrix based on three rules between each two features in sequence. Their work was concerned with sequencing of operations based on features and didn't include machine assignment. Reddy *et al.* [1999] used a similar approach but based the values in the penalty matrix (which they named as precedence cost matrix (PCM)) by setting different penalty values for operations in sequence depending on the different machining requirements between each two operation. The parameters they included were tool change, setup change and machine change. This work was also concerned with sequencing only. Also another disadvantage of the first approach is that for each new part a new PCM has to be developed.

Another approach which is used in this thesis was used by Zhang and Nee [2001], Li *et al.* [2002], and Ong, *et al.* [2002] to minimize cost based on a multi-objective function including the following five criteria; Machine usage cost, tool usage cost, machine change cost, setup change cost or TAD change cost and tool change cost. Ong, *et al.* [2002] added a sixth objective to minimize fixture cost. The second approach provides a more comprehensive and general solution than that of using the penalty function because it not only provides the sequence of operations, but the output will also include operation selection (Machine, Tool and TAD selection).

2.6.2 Process Plan Feasibility

One of the main problems faced with GAs, is generating a feasible process plan due to the randomness used in GAs. Most work [Dereli and Filiz 1999, Reddy *et al.* 1999, Zhang and Nee 2001, Li *et al.* 2002, Ong, *et al.* 2002] generates the first initial population randomly, which results in a number of infeasible process plans. A test is then carried out to select the feasible process plans [Reddy *et al.* 1999] or repair the infeasible process plans [Li *et al.* 2002]. This approach has a drawback of generating infeasible process plans especially if there is a complex part which affects the efficiency of the algorithm.

When the crossover or mutation operators are used, testing and repairing is carried out but the child chromosome will contain different characteristics than both parents which defeats the purpose.

Tang *et al.* [2004] used the partial precedence graph sorting technique to translate a given chromosome into a feasible sequence. This technique guarantees a feasible sequence. However, the problem with this method when using discrete variables is knowing the limits for each variable since the limits change from one sequence to another.

ElMaraghy and Gu [1987] first introduced the concept of clustering parts and features according to their tolerance datums and inspection requirements for task planning of CMM machines. Shabaka and ElMaraghy [2005] proposed a method for clustering operations with tolerance and logical constraints into operation clusters. Although some research used tolerances to create precedence relations between operations, forcing operations, which have tight tolerance to be processed on the same machine has not been proposed earlier in process planning. This is important because it is much cheaper to perform operations with tight tolerances on the same machine as it would reduce the required number of highly capable machines.

Other optimization techniques apart from GAs are Simulated Annealing [Ong *et al.* 2002], Petri nets [Kiritsis and Prochet 1996], Graph theory [Ciurana *et al.* 2003] and Taiber [1996] applied a set of modified algorithms from the field of combinatorial search problems.

2.7. Summary Literature Review

As can be observed from the above sections in previous manufacturing systems, the process plan was generated according to the manufacturing system capabilities. However, in RMS, since the capabilities of the machines and manufacturing system change with each configuration there should be a different approach for process plan generation. Upon reconfiguring the system for newly introduced part(s), new process plans have to be developed for the new part(s), as well as possible changes in the old process plans for the old part(s).

The review showed that different manufacturing systems require different approaches to process planning depending on the nature of the system. RMS combines features from several types of systems such as Flexibility from FMS and Multi-tool operation from DMS. The RMS also introduces the new concept of reconfigurability, which presents new challenges for process planning. It is obvious, based on the few research publications found in the field of process planning for RMS, that there remain many gaps. There is a need for a new process planning approach for RMS that achieves dynamic process plan generation where machine configuration commensurate with the performed operations in each configuration.

The machine tools in traditional manufacturing systems are not modular. For that reason, the process plans depended on the capabilities of the available machine tools. However, machine tools in the RMS domain are reconfigurable. Therefore, their structures may be generated to best suit the processing requirements of the parts.

CHAPTER THREE

OVERVIEW OF RMS PROCESS PLANNING APPROACH

This chapter presents an overview of the general approach, RMS Process Planning Approach that was developed in order to accomplish the objective of developing a process planning approach that makes use of the high reconfiguration capabilities of the machines within a reconfigurable manufacturing system environment.

3.1. Input Description

This section provides a brief description of the input parameters and information that are assumed to be available. There are several types of inputs; part information, tool information, machine information and cost information inputs. Also the database DB_{cap} is an input that is updated for each new part or machine. DB_{cap} contains previously formed clusters and their corresponding machine structures.

3.1.1 Part Information

3.1.1.1 Part Dimensions (L, W and H)

This data is required to help define the work envelope (Figure 3.1).

3.1.1.2 Operations and Operation Precedence Graph

The operations precedence graph (PG) captures the precedence constraints, which define order of succession among operations (figure 3.2). Each node (circle) represents an operation and the arcs (arrows) show the direction of operation precedence. An operation that has an arrow pointing towards it cannot be done until the node that the arrow comes from is done first. The letters next to the arcs indicate the type of precedence constrain (discussed in more detail in the chapter four).

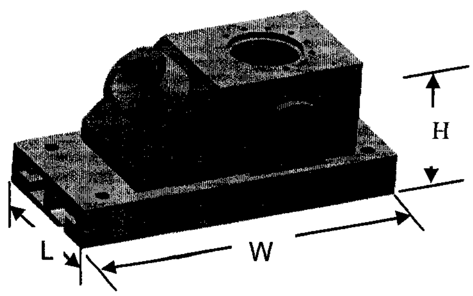


Figure 3.1: Main Part Dimensions (Part from Li *et al.* 2002).

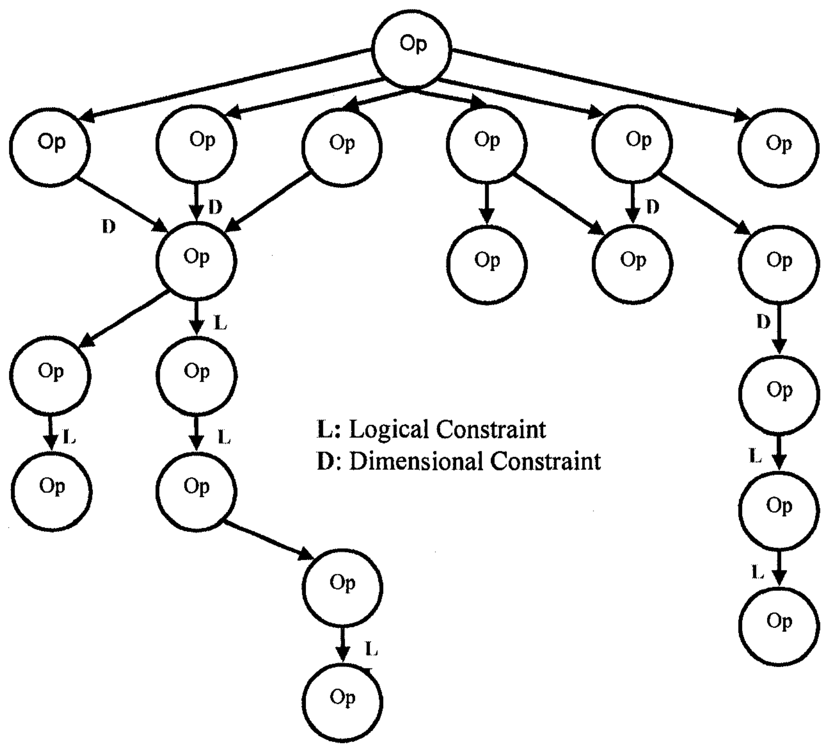


Figure 3.2: Example of Operations Precedence Graph with Constraints.

3.1.1.3 Tool approach directions (TAD)

TAD defines the cutting tool approach relative to the work piece for each operation. It helps in clustering operations and building the machine structure capable of approaching the work piece along these directions. Figure 3.3 illustrates examples of different TAD.

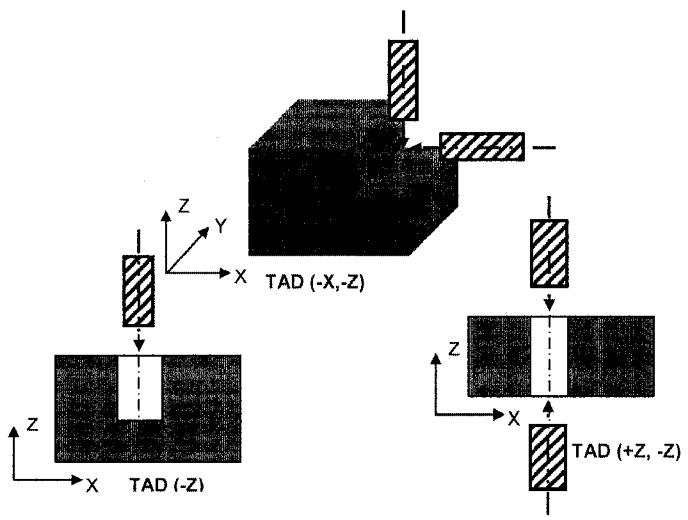


Figure 3.3: Example of Different TADs.

3.1.1.4 Candidate Cutting Tools

This is the list of cutters that could be used for each operation.

3.1.2 Machine Information

This is information about all the available machines, their capabilities and their different configurations. Information includes the machine cost of each configuration. Appendix B provides details about DB_{MC} which contains machine information.

3.1.3 Cost Information

Cost input information contains all information about that is concerned with cost. Cost information include; Machine configuration cost, machine cost, machine change cost, tool cost, tool cost index, tool change cost and setup change cost. The cost model used is similar to that used by Zhang *et al.* [1997].

3.1.4 Database DB_{Cap}

DB_{Cap} is a database containing a list of operation clusters and the corresponding machine structures that are capable of producing each cluster. The database (DB_{Cap}) is automatically updated with the new clusters after the candidate machine structures are generated for clusters not in the database. Each record in the database contains the following fields.

OC_{ID}: The OC ID number

CMC_s [1..nCMC]: An array containing the list of candidate machine capable of performing the OC. nCMC is the number of different available capable machines

3.2. Input Data Structures

This section provides the detailed data structures giving information about the inputs discussed in section 3.1.

3.2.1 Part Information (*OPs, TADs, TL_s and CT*)

OPs are the operations required to produce the part. OPs must be accompanied by operations precedence graph (PGs) that define sequential constraints between the different OPs and subsequently between different OCs. The following are the data structures giving information about *OP*s, *TAD*s and *OPFT*s.

3.2.1.1 Part Dimensions (L, W and H)

L, W, and H are the workpiece's main dimensions. This data is required to help define the work envelope.

3.2.1.2 Operations (*OPs*)

NOP: Number of operations (OPs) required to produce the part

OPID [1... *NOP*]: Array of the ID's of the OPs required to produce the part where:

$OPID(x)$ = ID of OP_x

OPP [1...*NOP*][1... *NOP*]: Matrix to represent operations precedence relations

where:

$$OPP(x,y)=\begin{cases} 1 & \text{operation } OP_x \text{ must be performed before operation } OP_y \\ 2 & \text{if } OP_x \text{ must be performed (clustered) with } OP_y \text{ due} \\ & \quad \text{to dimensional constraint} \\ 3 & \text{if } OP_x \text{ must be performed (clustered) with } OP_y \text{ due} \\ & \quad \text{to logical constraint} \\ 0 & \text{otherwise} \end{cases}$$ where:

x, y: indices for operations, $x, y = 1, \ldots, NOP$

3.2.1.3 Tool approach directions (TADs)

NTAD [1...*NOP*]: Array containing the number to possible TADs for each operation, where:

NTAD(x): Number of possible TADs for op_x

OPTAD [1...*NOP*][1... 6]: Matrix that contains the possible TADs for each operation where:

$$OPTAD(x,d)=\begin{cases}1 & \text{if for } OP_x \text{ there is a feasible TAD in } d \text{ direction} \\ 0 & \text{otherwise}\end{cases}$$ where:

x: index for operations, x = 1, ..., *NOP*

d: index to represent the TAD, d= 1, ..., 6, where the values of d indicate the following TADs:

d=1: TAD in +ve x direction

d=2: TAD in –ve x direction

d=3: TAD in +ve y direction

d=4: TAD in –ve y direction

d=5: TAD in +ve z direction

d=6: TAD in –ve z direction

3.2.1.4 Candidate Cutting Tools (TLs)

NT: Number of available tools

TLID [1... *NT*]: Array of the ID's of all the feasible TLs that can produce the part where:

$TLID(t)$ = ID of TL_t

NOT [1... *NOP*]: Array containing the number of tools that can be used for each operation, where:

NOT(x) = number of tools that can be used to produce OP_x

OPFT [1...*NOP*][1... *NT*]: Matrix that represents the feasible tools for each operation where:

$$OPFT(x,t) = \begin{cases} 1 & \text{OP}_x \text{ can be performed by if TL}_t \\ 0 & \text{otherwise} \end{cases}$$ where:

x: is the index for operations, *y* = 1, ..., *NOP*

t: is the index for the tool number, *t* = 1, ..., *NT*

3.2.1.5 Tool Cost (CT)

CT[1... *NT*]: Array containing the cost of using each cutting tool, where:

CT(t) = cost for cutting tool TL_t, where *t*: index for tool number, *t* = 1, ..., *NT*

3.2.2 Machine Information *(Ms, CM)*

This is the set of alternate machine types that are available/obtainable for use in the system. These Ms should be associated by the machine costs (CMs). The following are the data structures that describe machines (Ms) information.

3.2.2.1 Machines (Ms):

NM: Number of available/obtainable machine types

MID [1... *NM*]: Array of the ID's of available/obtainable Machines where:

MID(*m*) = ID of M_m, where *m*: index for machine, *m* = 1, ..., *NM*

3.2.2.2 Machines Configurations (MC):

NMC[1...*NM*]: Array of the number of possible machine configurations (MCs) that can be used with each machine type where:

NMC(*m*) = Number of possible machine configurations (MCs) that can be used with machine type *m* where, $m = 1, \ldots, NM$

$MCSL_m$ [1...*NMC*(*m*)][1... 3]: Matrix that contains the stroke length in different axes for all MCs for machine $m \forall\ m = 1, \ldots, NM$, where:

$MCSL_m(c,l)$ = value of the stroke length for Machine configuration MC_c using machine M_m in the direction *l*

where:

c: index for machine configurations, $c = 1, \ldots, NMC(m)$

m: index for machine, $m = 1, \ldots, NM$

l: index to represent the machine stroke length, $l = 1, 2, 3$, where the values of *l* indicate the following:

l=1: Length l=2: Width l=3: Height

MCR_m [1...*NMC*(*m*)][1... 3]: Matrix that contains the rotational capabilities in the different axes for all machine configurations for machine $m \forall\ m = 1, \ldots, NM$, where:

$MCR_m(c,a)$ = the value of the rotation angle for Machine configuration MC_c using machine M_m in the *a* direction

where:

c: index for machine configurations, $c = 1, \ldots, NMC(m)$

m: index for machine, m = 1, ..., NM

a: index to represent the machine's rotational angle, a= 1, ..., 6, where the values of a indicate the following direction:

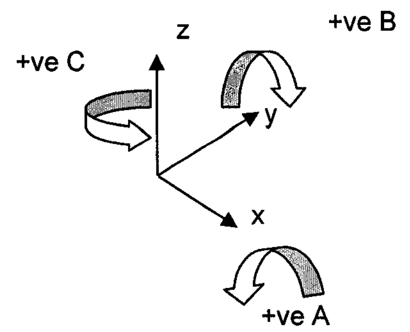


Figure 3.4: Different Directions of Rotation

a=1: Rotation around A-axis in the +ve (Counter Clockwise) direction

a=2: Rotation around A-axis in the –ve (Clockwise) direction

a=3: Rotation around B-axis in the +ve direction

a=4: Rotation around B-axis in the –ve direction

a=5: Rotation around C-axis in the +ve direction

a=6: Rotation around C-axis in the –ve direction

3.2.2.3 Machine Configuration Cost (CM):

$CM_m[1...NMC(m)]$: Array of the initial cost of all possible MCs for machine m $\forall$ m = 1, ..., NM where:

$CM_m(c)$ = Initial cost of machine configuration MCc for machine m where:

c: index for machine configurations, c = 1, ..., $NMC(m)$

CM_m includes cost of machine basic structure, modules for axes of motion, spindle modules and fixture modules

3.2.3 Cost and Cost Index Information (CM, MCI, MCC, CT, TCI, TCCI and TDCCI)

There are several costs and cost indexes included in this model. The following are the different cost and cost indexes included in the work:

3.2.3.1 Machine Configuration Cost (CM)

As mentioned earlier, this is an array containing the cost of the machines (Appendix C).

3.2.3.2 Machine Cost Index (MCI)

MCI is an index to indicate the cost of using a machine by multiplying the index by the machine cost (*CM*). It is a reasonable assumption to assume that the index is a fraction of the cost of the machine.

3.2.3.3 Machine Change Cost (MCCI)

MCCI is the cost that is incurred for every machine change in the process plan.

3.2.3.4 Tool Cost (CT)

As mentioned earlier, *CT* is an array containing the cost of each of the available tools.

3.2.3.5 Tool Cost Index (TCI)

TCI is an index to indicate the cost of using a tool by multiplying the index by the tool cost (*CT*).

3.2.3.6 Tool Change Cost Index (TCCI)

TCCI is an index to indicate the cost that is incurred for every tool change within the same setup (on the same machine).

3.2.3.7 Tool Approach Direction Change Cost Index (TDCCI)

TDCCI is an index to indicate the cost that is incurred for every TAD change within the same setup (on the same machine).

3.3. RMS Process Planning Approach

This section presents a brief description of the overall procedure performed by the developed RMS Process Planning Approach in order to accomplish the target research objective. This procedure is further detailed in the following chapters of the dissertations. The procedure has four main stages. Following this section, the output data structures representation for the different stages is provided. Figure 3.5 shows the overview of the approach and shows the chapter that provides the details for each stage.

3.3.1 Stage I (Operation Clustering Stage)

The inputs discussed in section 3.2 are taken as input to this stage and a clustering procedure is carried out using the operations precedence graph and the type of constraints between operations. The Operation precedence graph is mapped into a matrix representation which also captures the type of constraint relations between operations. The output clustered operations are then used as inputs to the next stage.

3.3.2 Stage II (Generating Machine Structure Stage)

The inputs to this stage are those discussed in section 3.2 in addition to the output from the clustering stage. This stage makes use of the configuration capabilities of manufacturing systems to design and tailor the machine tools to be capable of providing the manufacturing requirements for each operation cluster.

This stage first obtains the minimum machining requirements for each OC depending on the operations in the OC and then finds the machines in the machine data base using the different configuration to find the capable machine configurations for each OC. The Results are stored in DB_{Cap}. The capable machine configuration is stored in a structure format. DB_{Cap} is used if a new part that is introduced, to obtain the capable machine configurations for the similar OCs to avoid generating the minimum capabilities again.

The output of this stage will be an array containing all operation clusters and each operation cluster will have a list of all the machine configurations capable of being used for that OC.

3.3.3 Stage III (Generating Optimum Process Plan Stage)

The inputs to this stage are the inputs discussed in section 3.2 and the outputs of stages I and II. In this section an optimization is carried out to find the near optimal process plan. GAs are utilized to carry out the optimization and a new modified process plan model is proposed to be able to represent the OC and machine reconfiguration aspect. The output of this stage is a complete process plan for the part being manufactured. The process plan includes the operation sequencing and operation selection (machine, machine configuration, TAD, Tool) parameters for every operation and operation cluster.

3.3.4 Stage IV (Process Plan Reconfiguration Stage)

This stage is carried out when a machine that is currently used becomes unavailable or a new part is introduced. Inputs to this stage are either the machine that is unavailable or the data for the new part (same as in section 3.2). The summary of the procedure for both cases are as follows (Figure 3.5). The output of this stage is similar to that of stage III but after carrying out the necessary reconfiguration of the process plan.

3.3.4.1 A Machine is Unavailable

Input: Current process plan, available machines plus machine that breaks down.

Procedure: Look for the part of the process plan that will be affected and try to accommodate for that in either the machines in prior or post stages (1st before reconfiguration then reconfigure if there is no solution) if all fails suggest an alternative obtainable machine to be added instead of the broken machine.

Output: The new process plan after accommodating for the down-machine.

3.3.4.2 A New Part is Introduced

Input: Current process plan, available machines plus operations of new part.

Procedure: Identify the new operation clusters and try to accommodate them with the existing system if possible. Otherwise, reconfigure the minimum amount of machines as possible to accommodate for the new operation clusters. Suggest alternative machines from the list of available machines to be added to the system to accommodate for these clusters if there is no other solution.

Output: The new process plan after accommodating for the new part.

The next two sections provide the data structures of the outputs of each stage.

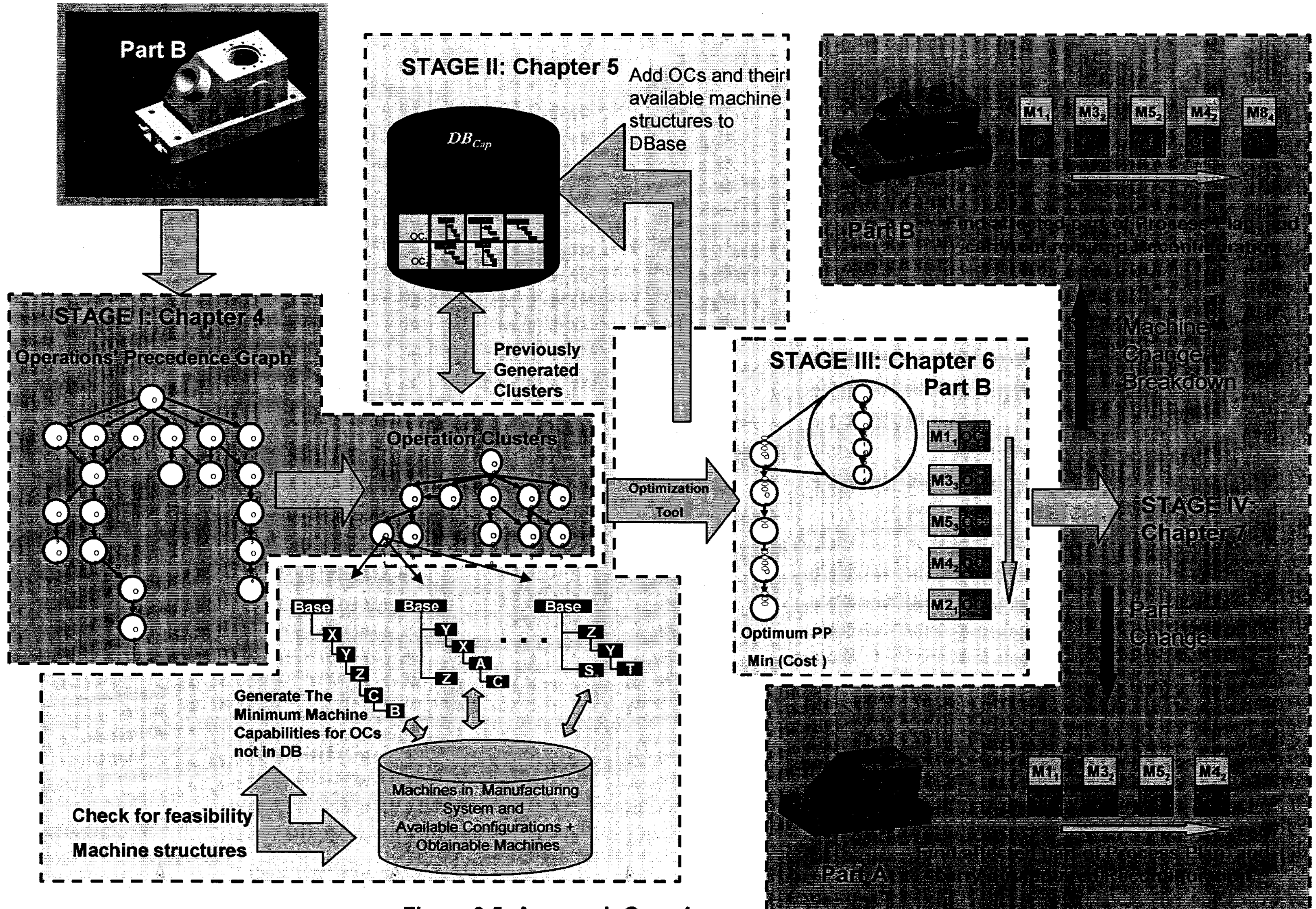


Figure 3.5: Approach Overview.

3.4. Output from OP Clustering and Generating Machine Structure Stages

This section describes the data structure of the output from stages I and II which shows the information of each operation cluster OC, the different possible TAD odds (options) for each OC, the minimum required machine capabilities for each possible TAD odd for each OC and the list of machines that are capable of producing these OCs.

3.4.1 Part Information (OCs)

3.4.1.1 Operation Clusters (OCs)

NOC: Number of operation Clusters (OCs) required to produce the part

OCID [1... *NOP*]: Array of the ID's of the OCs required to produce the part where:

$OCID(i)$ = ID of OC_i

NOPC [1...*NOC*]: Array of the number of OPs in each OC where:

$NOPC\ (i)$ = Number of OPs in operation cluster *i* where:

i: index for operation clusters, $i = 1, \ldots, NOC$

OPC [1...*NOC*][1... *NOP*]: Matrix to give information about the OPs of which each OC is composed for part where:

$$OPC(i,y) = \begin{cases} 1 & \text{if } \mathrm{OP}_y \text{ is a component of } \mathrm{OC}_i \\ 0 & \text{otherwise} \end{cases}$$ where:

i: is the index for operation clusters, $i = 1, \ldots, NOC$

y: is the index for operations, $y = 1, \ldots, NOP$

OCP [1...*NOC*][1... *NOC*]: Matrix to represent operation clusters precedence relations where:

$$OCP(i,j) = \begin{cases} 1 & \text{if } OC_i \text{ must be performed before } OC_j \\ 0 & \text{otherwise} \end{cases}$$ where:

i, *j*: indices for operation clusters, $i, j = 1, \ldots, NOC$

3.4.1.2 OC TAD requirements (OCO)

This is the data for the different possible odds for each operation cluster (Illustrated in Figure 4.28) using different feasible combinations of TAD. An OC odd is one possible combination of TADs used for the operations that are clustered in an OC.

For example if an OC contains two operations and each operation has 2 TADs then the number of possible odds (combinations for different TADs) to manufacture the OC is equal to four.

NOCO [1...*NOC*]: Array containing the number of OC odds for each OC where:

$NOCO(i)$ = Number of available odds for OC_i, where:

i is the index for operation clusters, $i = 1, \ldots, NOC$

OCO_i [1...$NOCO(i)$][1... 6]: Matrix that contains the required TADs to manufacture the different odds of OC_i for $\forall i = 1, \ldots, NOC$, where:

$$OCO_i(z_i, d) = \begin{cases} 1 & \text{if TAD in } d \text{ direction is required for odd number } z_i \text{ of } OC_i \\ 0 & \text{otherwise} \end{cases}$$ where:

i: index for operation clusters, $i = 1, \ldots, NOC$

o: index for number of odds for each operation cluster, $o = 1, \ldots, NOCO(i)$

d: index to represent the TAD, *d*= 1, …, 6, where the values of *d* indicate the following TADs:

d=1: TAD in +ve *x* direction

d=2: TAD in –ve *x* direction

d=3: TAD in +ve *y* direction

d=4: TAD in –ve *y* direction

d=5: TAD in +ve *z* direction

d=6: TAD in –ve *z* direction

3.4.1.3 Minimum Machining Capabilities (MinCAP)

The data structure in this section represents the minimum required capabilities in a machine required by each of the odds for each OC.

$MinCAP_{i,o}$[1,2][1, …, 3]: Matrix that contains the minimum required capabilities required in a machine to manufacture OC_i using OC odd *o* for that OC. for $\forall i$= 1, …, *NOC* $\forall o$= 1, …, *NOCO*(*i*), where:

$MinCAP_{i,o}$ (*r*,*a*) = Minimum rotation angle in degrees in *r* direction around axis *a* that is required in a machine to be able to manufacture OC_i using OC odd *o* for that OC

where:

i: index for operation clusters, *i* = 1, …, *NOC*

o: index for number of odds for each operation cluster, *o* = 1, …, *NOCO*(*i*)

r: index to indicate the direction of rotation axis *a*, where

r = 1 indicates counter clockwise rotation

$r = 2$ indicates clockwise rotation

a: value of rotation in degrees, where

$a = 1$ indicates rotation around A-axis

$a = 2$ indicates rotation around B-axis

$a = 3$ indicates rotation around C-axis

3.4.2 Capable Machines (MCAP)

$MCAP_{i,m}$ [1...$NOCO(i)$] [1...$NMC(m)$]: Matrix containing the list of machine configurations for machine m capable of producing the different odds for OC_i $\forall i$= 1, ..., NOC $\forall$ m= 1, ..., NM, where:

$$MCAP_{i,m}(o,c) = \begin{cases} 1 & \text{machine } m \text{ using configuration } c \text{ is capable of manufacturing } OC_i \text{ using odd } o \text{ of } OC_i \\ 0 & \text{otherwise} \end{cases}$$

where:

i: index for operation clusters, $i = 1, \ldots, NOC$

o: index for number of odds for each operation cluster, $o = 1, \ldots, NOCO(i)$

m: index for machine, $m = 1, \ldots, NM$

c: index for machine configurations, $c = 1, \ldots, NMC(m)$

3.5. Process Planning Output

The following are the data structures that provide information about the output from the stage III. The data structure for the output of stage IV is similar to stage III.

OCS [1…*NOC*]: Array of the sequence of OCs, where:

OCS(*s*) = OC that is ordered number *s* in the sequence of OCs in the process plan

s: index for sequence number, *s* = 1, …, *NOC*

MS [1…*NOC*]: Array of the sequence of the machine types used for each OC, where:

OD [1…*NOC*]: Array of the sequence of the OC odd used for each operation cluster in the process plan, where:

OD(*s*) = OC Odd used for *OCS*(*s*)

MCS [1…*NOC*]: Array of the configuration number for each machine types used for each OC, where:

MC(*s*) = Machine Configuration number for the machine type that is in sequence *s* in the machines and assigned to *OCS*(*s*)

NOPCS [1…*NOC*]: Array containing the number of operations for each OC in sequence of OCs, where:

NOPCS(*s*): Number of operations that are in operating cluster OCs

OPS [1…*NOP*]: Array of the operation in sequence of operations in the process plan, where:

OPS(*r*) = OP that is ordered number *r* in the sequence op OPs

r: index for sequence number, *r* = 1, …, *NOP*

TADS [1…*NOP*]: Array of the TAD used for each operation in sequence of operations in the process plan, where:

TADS(*r*) = Number of used tool for operation number sequenced *r* in the operations sequence of the process plan

r: index for sequence number, *r* = 1, …, *NOP*

TS [1…*NOP*]: Array of the tool number used for each operation in sequence of operations in the process plan, where:

TS(*r*) = Number of used tool for operation number sequenced *r* in the operations sequence of the process plan

r: index for sequence number, *r* = 1, …, *NOP*

The following chapters provide the detailed procedures for each stage.

CHAPTER FOUR

OPERATION CLUSTERING AND MACHINE CAPABILITY GENERATION

This chapter presents the detailed procedure for the operation clustering stage (stage I) and the machine capability generation stage (stage II). A toolbox was developed using MATLAB® software to demonstrate the use of the developed model, which is applied to a case study based on an example part from the literature. The results are presented, analyzed and compared for two parts of the same family.

4.1. Approach for Stage I and II

Figure 4.1 shows an overview chart of the proposed machine structure configuration approach. The inputs include part dimensions and tool approach directions (TAD) for each operation (Figure 3.3). The operations precedence graph (Figure 3.2)) is also an input. The method consists of a clustering stage and a machine structure generation stage. The output is the candidate machine tool configuration, machine modules, the minimum axis of motion and angles of rotation required for each cluster.

4.2. Introduction to Clustering Procedure

Operation Clustering is the first stage of the proposed approach. The operation cluster activity, sometimes referred to it in literature as setup planning, decides how the set of operations required to produce the final part are to be divided up amongst the machines that will form the system. In this work, the group of operations performed on the same machine without changing the machine is called a set-up. An operation cluster is a group of operations that must be performed together on the same machine. In traditional manufacturing systems the machine tools are fixed and not modular for that reason the grouping of operations depended on the machining tools available. However, in the RMS domain since machine tools are reconfigurable their structure should be generated to best suit the part being produced. For this reason operations clustering is a key step in determining the machines and their modules (configurations) that will be used to produce a part.

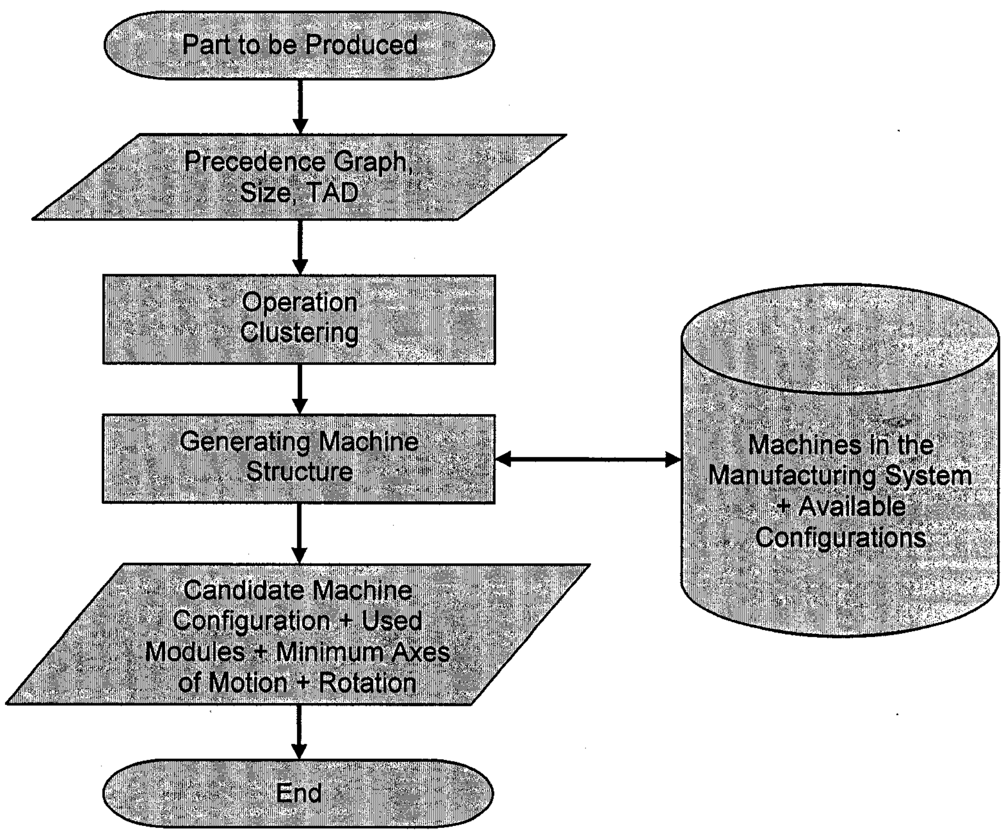


Figure 4.1: Machine Structure Configuration Steps.

Clustering is carried out in two steps based on the type of precedence constraint. The first step clusters operations that have tolerance datum constraints. The second step clusters operations based on logical constraints. The clustered operations formed are represented in a precedence graph format with each node representing a cluster of one or more operations. The new clustered operations will be used as the input to the second stage (generating the machine structure). Figure 4.2 shows the overview of the clustering operations.

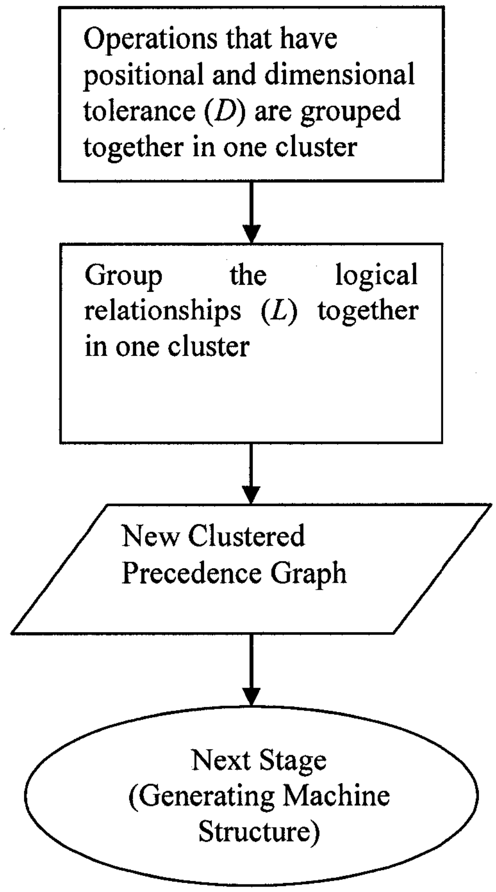


Figure 4.2: Clustering Overview.

4.2.2 Tolerance Datum Constraints

Tolerance datum constraints are operations that must be carried out on the same machine and with the same set-up to preserve the relative positioning tolerance and accuracy requirements. This is carried out to group operations that have tight tolerances together. Based on this rule operations that have positional and dimensional tolerance are clustered together so that they could be produced on the same machine because if they are separated on different machines there will be an increase in cost because this will require high setup tolerance of the material handling system.

Operations with tolerance datum constraints (Indicated by the letter D in Figure 4.3) are clustered together. This step is repeated until all operations with tolerance datum constraints are clustered together.

In general:

```
Procedure: Dimensional_Tol_Clustering
IF Tolerance pallet > Tolerance part
THEN
     Operations are carried out on the same setup
ELSE
     Operations could be carried out on different machines
END
```

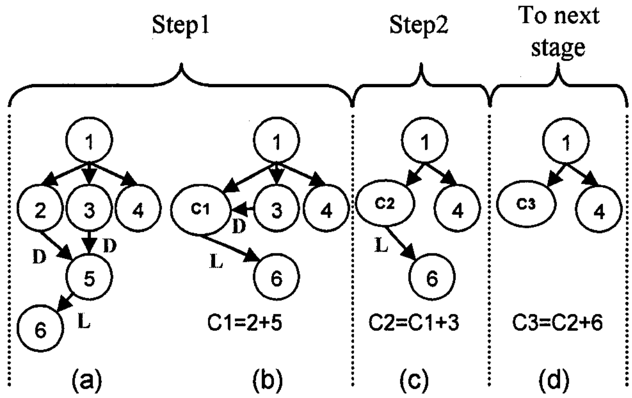
4.2.3 Logical Precedence Constraints

All the operations that have logical constraints (Indicated by the letter L in Figure 4.3) are clustered together. An example of logical constraints is the clustering of drilling, reaming and possibly boring operations together when producing a hole.

4.3. Clustering Procedure

The precedence graph is represented in a matrix format as shown in Figure 4.4(a), which corresponds to the precedence graph in Figure 4.3(a). Figure 4.3 shows the two steps in the clustering procedure and the final clustered precedence graph that will be used as input to the second stage. Figure 4.4(a), (b), (c) and (d) are the matrix representations for Figures 4.3(a), (b), (c) and (d) respectively.

In Figure 4.4(a), the matrix is a square ($n \times n$) matrix where n is the number of operations required to produce the part. The row is the predecessor and the column is the successor. The rows and columns represent the operation numbers. Each cell a_{ij} can take a value of 0, 1, 2, or 3, where i and j = 1, 2,..., n (Table 4.1).



Step 1: Tolerance Datum Constraints
Step 2: Logical Precedence Constraints

Figure 4.3: Example of Clustering Procedure.

Table 4.1: Numbers used in Precedence Matrix.

a_{ij}	Meaning
0	There is no constraint between operation *i* and operation *j*
1	Operation *i* is before operation *j* due to a constraint other than dimensional and logical constraint
2	Operation *i* is before operation *j* due to a dimensional constraint
3	Operation *i* is before operation *j* due to a logical constraint

	1	2	3	4	5	6
1	0	1	1	1	0	0
2	0	0	0	0	2	0
3	0	0	0	0	2	0
4	0	0	0	0	0	0
5	0	0	0	0	0	3
6	0	0	0	0	0	0

(a)

	1	C1	3	4	6
1	0	1	1	1	0
C1	0	0	0	0	3
3	0	2	0	0	0
4	0	0	0	0	0
6	0	0	0	0	0

(b)

	1	C2	4	6
1	0	1	1	0
C2	0	0	0	3
4	0	0	0	0
6	0	0	0	0

(c)

	1	C3	4
1	0	1	1
C3	0	0	0
4	0	0	0

(d)

Figure 4.4: Matrix Representation of Clustering.

The clustering procedure is as follows:

```
Procedure: Clustering (i)
Check every link (starting from left to right and from top
to bottom)
     IF constraint on Link of type = = (i) THEN
          IF   one of the nodes of this link is already in a
               cluster THEN
               Add the other node to the same cluster
          ELSE Create a new cluster and place the two Nodes
               in it
     ELSE Move to next Link
     END
END
```

4.4. Case Study ANC-090

In this section the clustering procedure explained in section 4.3 will be illustrated with the application to the case study ANC-090 part (detailed specifications for the part are found in Appendix A). The basic part (ANC-090) was developed as a variant of the test part (ANC-101).

Figures 4.5 shows the input operations precedence graph to the clustering stage, which is transformed into matrix format and the resulting matrix is shown in Figure 4.6.

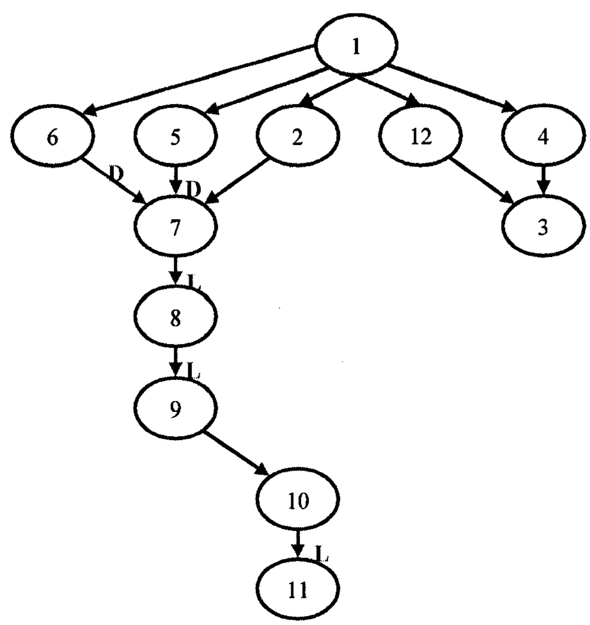


Figure 4.5: ANC-090 Precedence Graph.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	0	1	1	1	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	2	0	0	0	0	0
6	0	0	0	0	0	0	2	0	0	0	0	0
7	0	0	0	0	0	0	0	3	0	0	0	0
8	0	0	0	0	0	0	0	0	3	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	3	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0	0	0

Figure 4.6: ANC-090 Precedence Graph Matrix Representation.

The procedure starts by searching for an operation that has dimensional tolerance constraint. The first operation to have a dimensional or tolerance constrain is operation number 5 (milling of protrusion) which has to be before operation number 7 (drilling of compound hole) due to datum constraints. So both columns and rows are selected (Figure 4.7).

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	1	0	1	1	1	0	0	0	0	0	1
2	0	0	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	2	0	0	0	0	0
6	0	0	0	0	0	0	2	0	0	0	0	0
7	0	0	0	0	0	0	0	3	0	0	0	0
8	0	0	0	0	0	0	0	0	3	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	3	0
11	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0	0	0

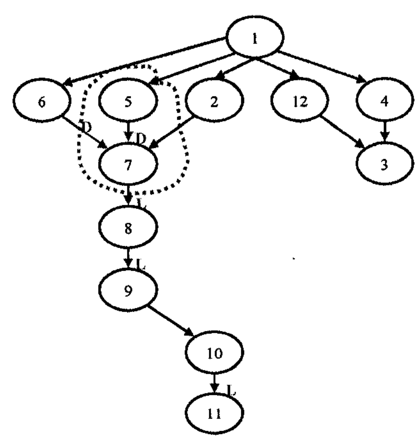


Figure 4.7: Clustering Operations 5 and 7 for ANC-090.

The next step is to cluster operations 5 and 7 together as can be seen by the dotted line (Figure 4.7). The new formed cluster is named C1. The resulting matrix is shown in Figure 4.8. It should be noted that the matrix size (rows and columns) each decreases by one since two operations are clustered together.

The clustering process does not change the sequence of operations and preserves the same constraint type between operations. This process is carried out for all dimensional and tolerance constraints (constraints that have the number 2 in the matrix). After another clustering operation all the dimensional and tolerance constraints are clustered (Figure 4.9), then the operations that have logical constraints (constraints that have the number 3 in the matrix) will be clustered together (Figure 4.10).

	1	2	3	4	C1	6	8	9	10	11	12
1	0	1	0	1	1	1	0	0	0	0	1
2	0	0	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0
C1	0	0	0	0	0	0	3	0	0	0	0
6	0	0	0	0	2	0	0	0	0	0	0
8	0	0	0	0	0	0	0	3	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	3	0
11	0	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0	0

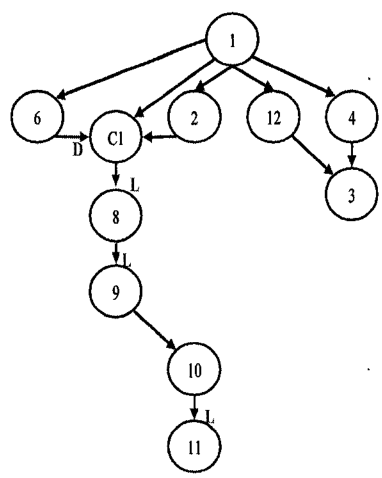


Figure 4.8: Cluster C1 for ANC-090.

	1	2	3	4	C2	8	9	10	11	12
1	0	1	0	1	1	0	0	0	0	1
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0
C2	0	0	0	0	0	3	0	0	0	0
8	0	0	0	0	0	0	3	0	0	0
9	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	3	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0

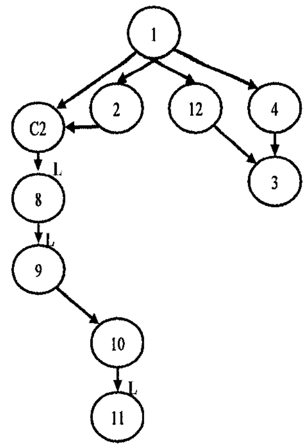


Figure 4.9: End of Clustering Due to Dimensional and Tolerance Constraints for ANC-090.

Figure 4.10 shows that there is a logical constraint between the second formed cluster (C2) and operation number 8 (reaming of compound hole). Cluster C3 will result from clustering the already formed cluster C2 and operation 8. Figure 4.11 has the final formed matrix and the final precedence graph. As an output with the developed program is the list of arrays shown in Figures 4.12. The list of arrays shows the operations that are within each cluster.

	1	2	3	4	C2	8	9	10	11	12
1	0	1	0	1	1	0	0	0	0	1
2	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0
C2	0	0	0	0	0	3	0	0	0	0
8	0	0	0	0	0	0	3	0	0	0
9	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	3	0
11	0	0	0	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0	0	0	0

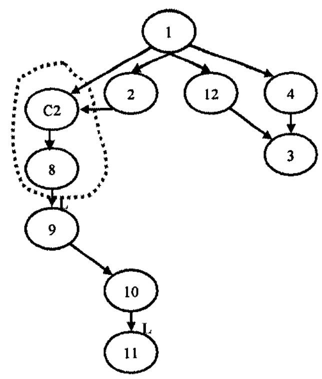


Figure 4.10: Starting of Clustering Due to Logical Constraints for ANC-090.

	1	2	3	4	C4	C5	12
1	0	1	0	1	1	0	1
2	0	0	0	0	1	0	0
3	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0
C4	0	0	0	0	0	1	0
C5	0	0	0	0	0	0	0
12	0	0	1	0	0	0	0

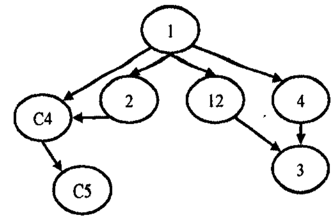


Figure 4.11: Final Matrix and Precedence Graph for ANC-090

1	1	0	0	0	0
2	2	0	0	0	0
3	3	0	0	0	0
4	4	0	0	0	0
C4	5	7	6	8	9
C5	10	11	0	0	0
12	12	0	0	0	0

Figure 4.12: Arrays Containing Operations in Each Cluster ANC-090.

4.5. Generating Machine Tool Structure

The second stage is related to generating a machine structure capable of producing the obtained operation clusters. It represents a mapping process between the processing capabilities of reconfigurable machines and machining requirement of parts features as shown in Figure 4.13.

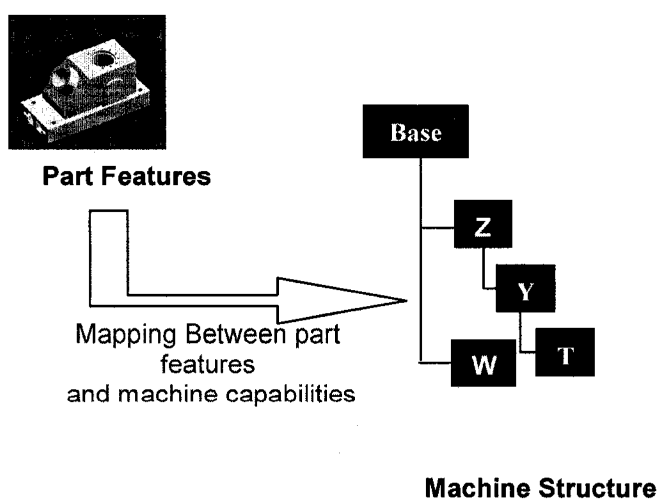


Figure 4.13: Mapping Between Part Features and Machine Capabilities.

4.6. Machine Structures

A kinematic chain-like diagram that shows the machine's axes of motion and the degrees of freedom was used by Bohez [2002] to represent the machine tool structure. This structure is generic, descriptive and can be applied to any machine configuration. It has been adopted in this research.

The structure of a machine tool composed of several axes can be viewed similar to that of a robot structure. Figure 4.14 shows a schematic diagram of a five-axis machine tool and Figure 4.15 shows the kinematic-like structure for representing it.

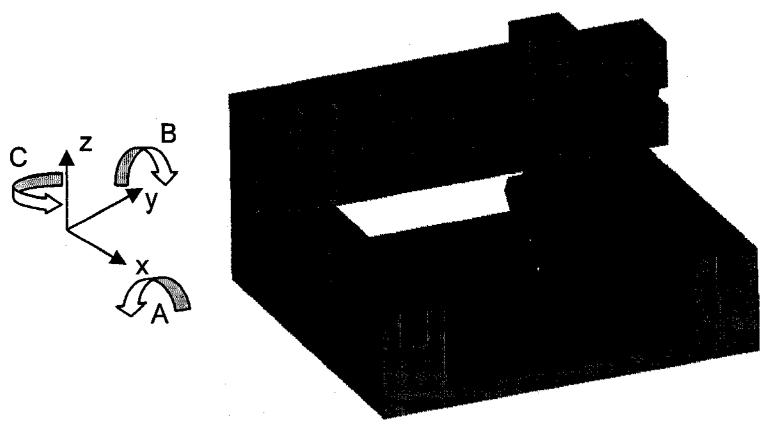


Figure 4.14: 5-Axis Machine Tool.

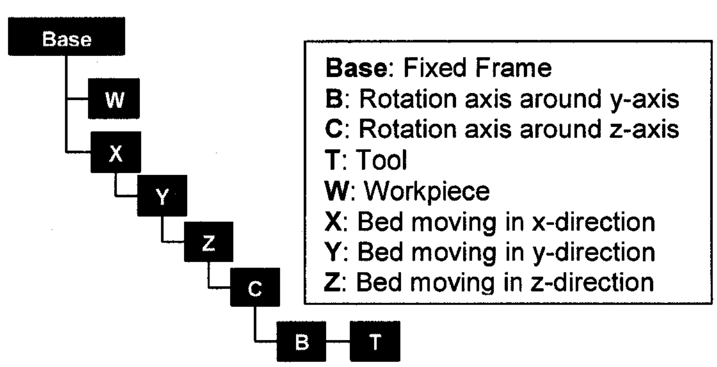


Figure 4.15: Kinematic-Like Structure For A 5-Axis Machine Tool.

From Figure 4.14, the work piece is fixed to the machine tool frame. The tool is attached to a table that rotates in the B direction (around the Y-axis). This is connected to a rotary table that rotates in the C direction (around the Z-axis) and is mounted on a bed that moves in the Z-direction. This bed is mounted on another bed that moves in the Y-direction, which in turn is mounted on an X-direction bed which is attached to the fixed frame of the machine. This information could also be inferred from Figure 4.15 where: *Base* indicates the machine tool frame, *T* indicates the tool end, and *W* indicates the work piece end. Figure 4.16 shows another representation to that illustrated in Figure 4.15.

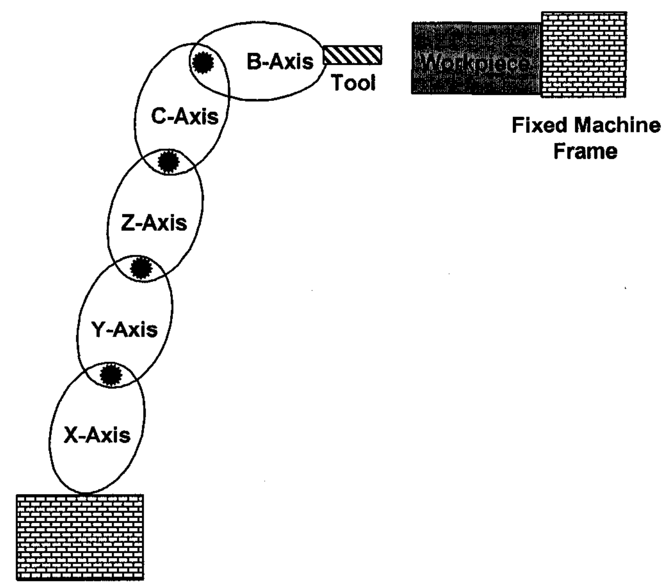


Figure 4.16: Another Kinematic-Like Structure Representation for Machine Tool Shown in Figure 4.14.

4.7. Machine Capability

Machine structures are different from one machine to another even for machines with the same number of axis. The different structure will result in different machine capabilities. The following example will illustrate how the gradual complexity of the part will require more complex machines (machines with more axis of motion).

Figures 4.17 to 4.19 present the axis of motion for the different machines used in the example. The example will start with a part that could be produced on a 3-axis machine (Figure 4.20) and then a part that would require a 4-axis machine to be manufactured (Figure 4.21) and finally a part that would require a 5-axis machine to be manufactured (Figure 4.22).

4.7.1 Three-Axis Machine:

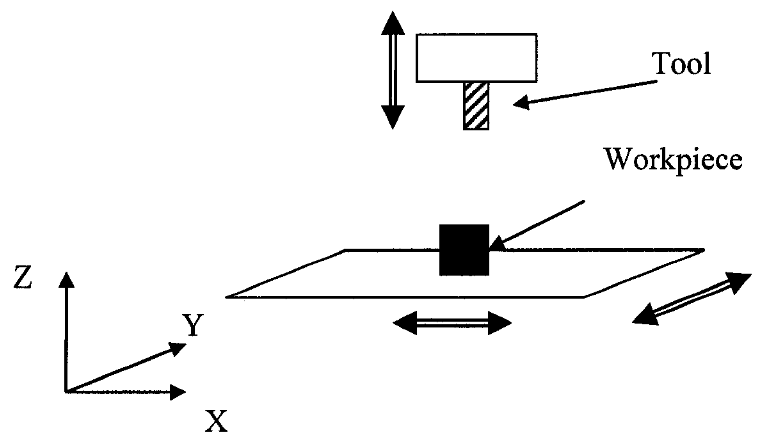


Figure 4.17: Three-Axis Machine.

4.7.2 Four-Axis Machine:

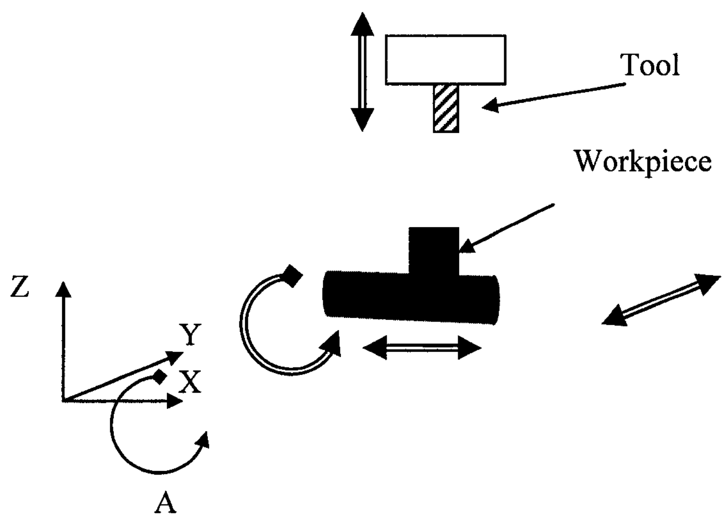


Figure 4.18: Four-Axis Machine.

4.7.3 Five-Axis Machine:

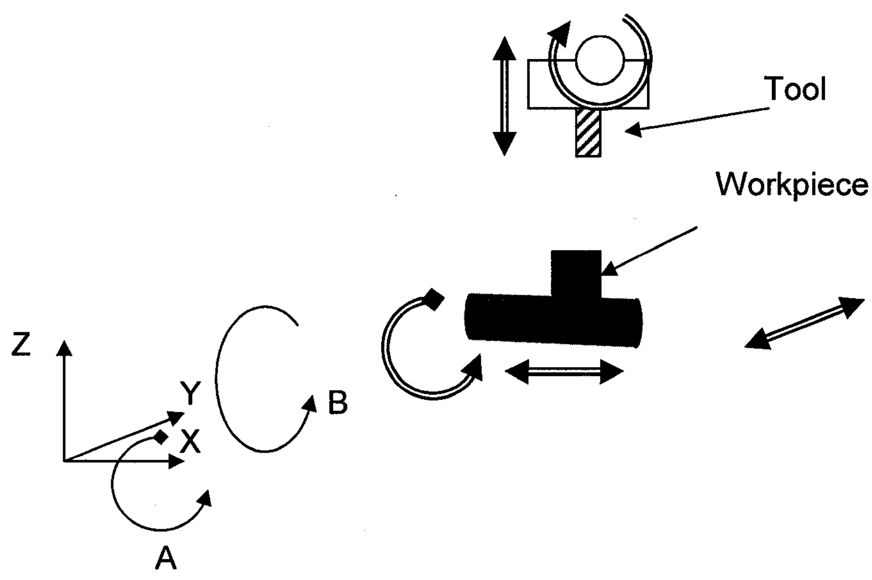


Figure 4.19: Five-Axis Machine.

Some of the 5-axis machines have the fifth axis as C (rotation around the Z axis). As mentioned above, such a machine will have different capabilities than that shown in Figure 4.19 which has the fifth axis as B (rotation around the y axis).

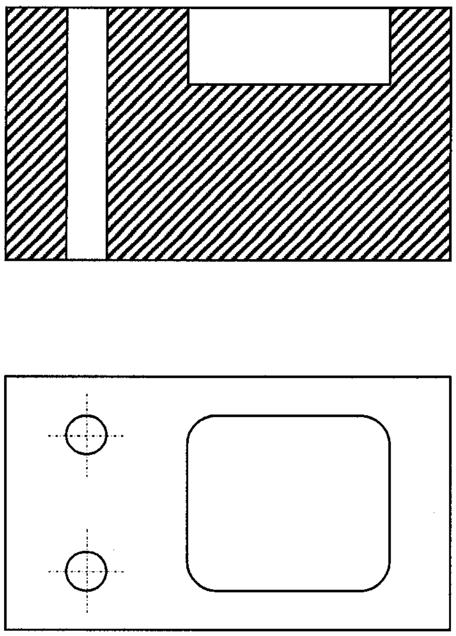


Figure 4.20: A part that can be Manufactured by a 3-Axis Machine.

The part shown in Figure 4.20 requires only 3-axis of motion for it to be manufactured but if another feature is added to the side of the part as that shown in Figure 4.21 then another axis of motion (rotation in the direction of A around the X-axis) is required to manufacture the part. It should be noted that not any 4-axis machine will do the job because for example if the 4th axis was a rotation of angle C around the Z-axis then this extra axis will not help in manufacturing the part.

The part shown in Figure 4.21 can be manufactured on a 3-axis machine but it will require 2 setups. Therefore, there will be a compromise between setup time and having a more expensive 4-axis machine.

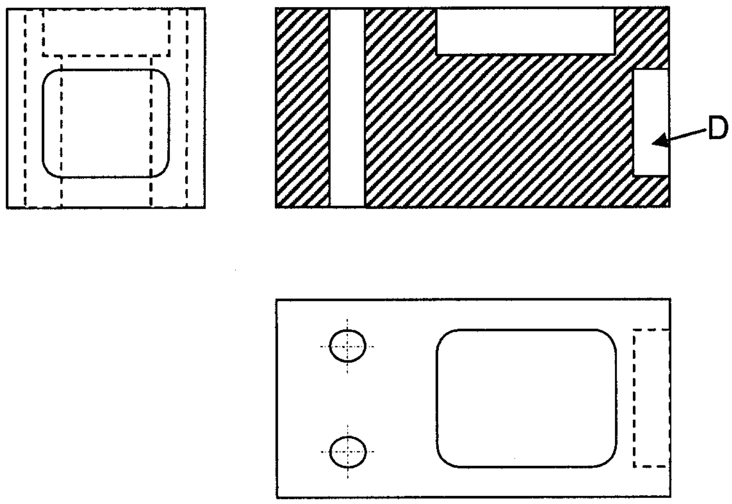


Figure 4.21: A part that Needs a 4-Axis Machine to Produce.

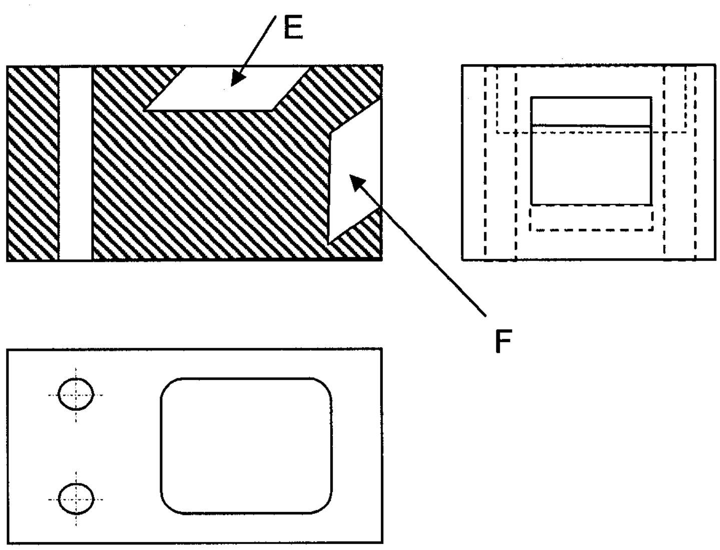


Figure 4.22: A part that Needs 5-Axis Machine to be Manufactured.

The part shown in Figure 4.22 will require the 5-axis machine shown in Figure 4.19 for it to be produced. For the feature labeled "E" the machine will require only 4-axis (X, Y, Z, and the rotation angle B) and for the area labeled "F" all five axis will be needed. Another alternative is to use two setups in a 4-axis machine that has the following axis of motion (X, Y, Z, and B). It should be noted that this machine is different then the 4 -axis machine in Figure 4.18.

4.8. Different Structures Have Different Capabilities

Different axes configuration produce different machining workspace. If the axes are of the same type rotation (R) or translation (T), then the sequence of the axes in the tool or work piece carrying kinematic chain is not important because they will result in the same workspace regardless of the sequence. Also different uses for machines require different location of the axes of motion. For example one of the earliest types of 5-axis milling machines has all the axes carrying the tool and the work piece is fixed on the table. This structure is used to handle very heavy work pieces. On the other hand, for very small work pieces, the best structure for a 5-axis milling machine is one where all the axes are carrying the work piece and the tool are fixed [Bohez 2002].

4.9. Number of Different Structure Configurations

The number of different configurations is very large if the sequence of axes of motion is taken into consideration. If all 6 axes of motion (X, Y, Z, A, B, C) are used, then the number of different configurations equals 720 [Bohez 2002]. For example, for a 5-axis milling machine the tool is carried by all axes, the first axis fixed to the machine frame can take any value of the 6 different coordinates. The second axis will take any value of the remaining 5 axes and so on. This will result in a 6! = 720 different configurations when all axes are carrying the tool. This value is the same for all other groups. There are five different possible groups where 1, 2, 3, or 4 axes are carrying the tool and the remaining axes are carrying the work piece. The total number of different combinations for a 5-axis machine would be equal to 5 x 720 = 3600.

It should also be noted that the sequence of the axes in the tool or work piece carrying kinematic chain is not important if the axes are of the same type (i.e. rotation or translation). This will result in the same workspace volume and machine capabilities. The number of configurations per group in this case is reduced a total of 120 combinations for all the groups.

4.10. Assumptions

From the information shown in the previous section, the following assumptions were made to reduce the problem size:

1. Machines are reconfigurable and consist of several fixed bases. There is a wide variety of modules for each of the axes to be added to the machine base to obtain the required machine capability.

2. All the machine structures have the basic three translation movements X, Y, Z. This reduces number of machine structure configurations by half for each group (i.e., 360 different configurations and 5 x 360 = 1800 over all the groups in the case of a 5-axis machine tool).

3. Since the sequence of the axes in the tool or work piece carrying kinematic chain is not important if the axes are of the same type (rotation or translation), then the number of configurations over all groups is reduced to 60 in the case of a 5-axis machine tool.

4. Since machines start with the three translational axes, additional rotational axes are as follows:

    a. For 4-axis machines, the additional axis could be: A, B, A', or B'

    b. For 5-axis machines the additional axes could: AB, AC, BC, A'B, A'C, B'A, B'C, C'B, A'B', B'C', C'A', or C'A

The additional apostrophe ' indicates that this axis is on the work piece side. Letters without the apostrophe indicate the additional axis is on the tool side.

The above assumptions reduce the search space to only 4 different configurations for a 4-axis machine tool and 12 different configurations for a 5-axis machine tool.

4.11. Procedure for Machine Capability Generation Stage

The procedure starts by taking the cluster precedence graph from the last stage as input. For the formed clusters a search is carried out in DB_{Cap} to look for similar clusters. If the cluster is found in DB_{Cap} then the machine structures in the database will be automatically used. If the Cluster is not found in the DB_{Cap} then the machine structure is generated depending on the required operations for that cluster. The database DB_{Cap} makes use of previously generated structures and is automatically updated if new structures are generated for a new cluster. This has the advantage of saving the time of generating a machine cluster that has already been produced before.

For all the clusters not found in DB_{Cap} a candidate of machine structures are generated. All the newly developed machine structures are run through the database (DB_{MC}) which contains all the machines available. If the manufacturing system has the machines and its configurations then the database (DB_{Cap}) is updated with the new cluster and its candidate machines. This is repeated for all new clusters that were formed in the previous stage and not found in DB_{Cap}. The following sections will give an in-depth on the procedure used to generate the machine structures that are capable of carrying out the required operations for each cluster.

The procedure for this step is as follows (Figure 4.23 shows the flowchart):

```
Procedure: Dbase Cluster Lookup (i)
FOR every cluster formed
    IF  Cluster  if found in DBCap THEN
        •   Return from Dbase the similar Cluster name
        with candidate machine structures
    ELSE Return cluster not found
        •   Generate possible structures for cluster
        •   For every structure generated check
        availability in system by checking DBMC
    END
END
END
```

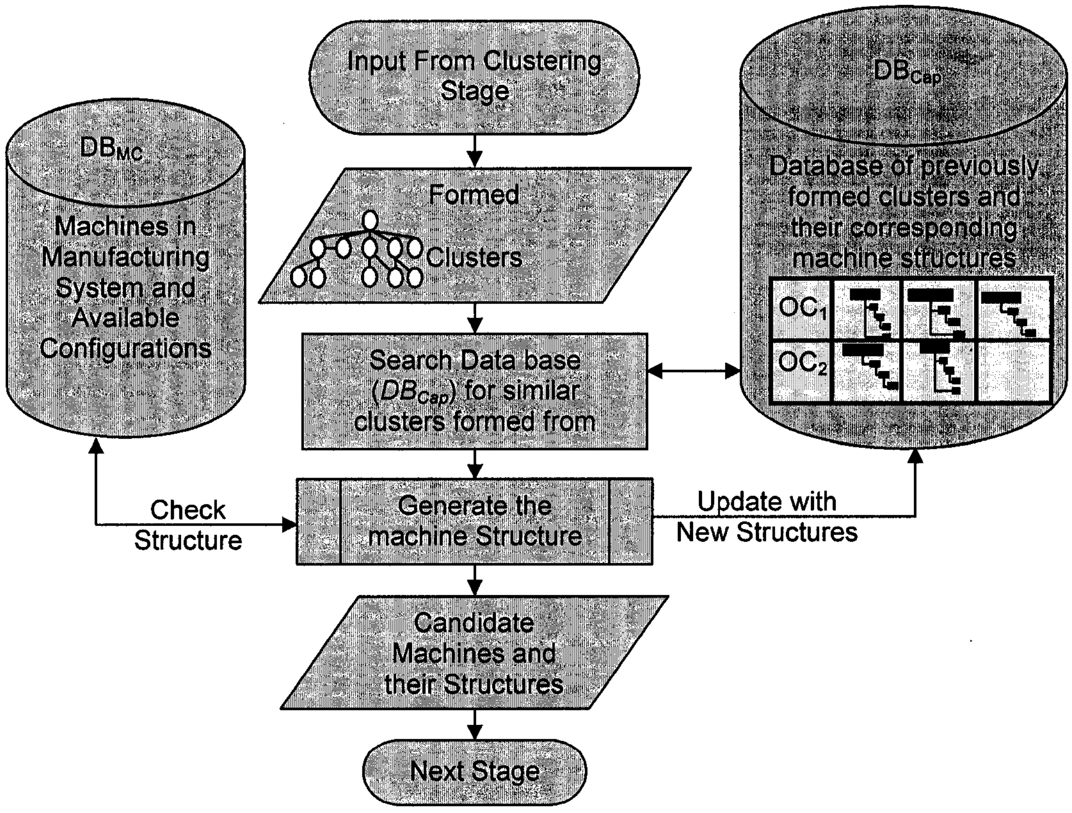


Figure 4.23: Flowchart for Stage II.

4.12. Algorithm for Generating Machine Tool Structures.

The machine structure is generated based on the capabilities required to generate the desired operation cluster. The minimum capability required to produce each operation cluster is obtained and then the structure satisfying this capability is generated. The flowchart in Figure 4.24 shows the algorithm used to obtain the minimum machining capabilities required by each cluster. This algorithm is repeated for each operation cluster.

In the developed algorithm, the coordinates are all with respect to the workpiece coordinates. From the flowchart:

NC is the number of operation clusters for the part

N_k is the number of operations in a cluster k, $k=1, 2, ..., NC$

A_k= is an (N_k x 6) matrix representing the following [$a_{i,1}$ $a_{i,2}$ $a_{i,3}$ $a_{i,4}$ $a_{i,5}$ $a_{i,6}$], i =1, 2,..., N_k.

$A_{k,}$= is a matrix representing a cluster and the TAD for each operation in that cluster.

j=1, 2, 3...6 refer to X, Y, Z, A, B, C axes

$a_{i,j}$ = 1 if there is a possible TAD in jth axis

$a_{i,j}$ = 0 if there is no TAD in jth axis

if $a_{i,j} \neq 0$ or 1 then a is the value of the angle rotation around the j axis.

OP_k= is an (N_k x 1) column vector representing the operation number corresponding to $A_{i,k}$

OC= (NC x max(N_k)) matrix, where, max(N_k) is the number of operations in the cluster that contains the largest number of operations. Each row represents a different cluster and the columns represent the operations in each cluster. This matrix is the output of the previous stage.

ROC_k = Is an (m x7) matrix the 1st column of which represents the operation number and columns 2 to 7 represent the TADs for X, Y, Z, A, B, C respectively. In this matrix, operations that have more then one TAD are repeated. If an operation has 2 TADs, that operation is repeated twice. [m = N_k + number of repetitions].

NR_k = $\Pi_{i=1}^{N_k}$(number of repetition of operation i in ROC_k). It represents the number of different possible combinations (odds) for cluster k. These combinations (odds) represent all the different possible TADs for the cluster.

$MinAxis_{k,p}$ = is a matrix representing the minimum machine capability (axes of motion) required in a machine tool to produce combination p (odd p or case p) of operation cluster k. Each matrix is (2 x 3) where the 1st row represents the minimum +ve angle or rotation needed for A, B & C respectively, and the 2nd row represents the minimum -ve angle needed for A, B & C.

Table 4.2: Different TAD and the Corresponding Required Axis of Rotation.

TAD									Rotation Angle Required		
x	-x	y	-y	z	-z	Angle	Rotation Direction	Axis of Rotation	A	B	C
1	0	0	0	0	0	90	+ve	y	0	90	0
0	1	0	0	0	0	90	-ve	y	0	-90	0
0	0	1	0	0	0	90	-ve	x	-90	0	0
0	0	0	1	0	0	90	+ve	x	90	0	0
0	0	0	0	1	0	180	-+ve	x or y	180 ↔	180	0
0	0	0	0	0	1	0	0	-	0	0	0
0	-a	0	0	0	0	a	+ve	y	0	a	0
-a	0	0	0	0	0	180-a	-ve	y	0	-(180-a)	0
0	a	0	0	0	0	a	-ve	y	0	-a	0
a	0	0	0	0	0	180-a	+ve	y	0	(180-a)	0
0	0	0	-b	0	0	b	+ve	x	b	0	0
0	0	-b	0	0	0	180-b	-ve	x	-(180-b)	0	0
0	0	0	b	0	0	b	+ve	x	-b	0	0
0	0	b	0	0	0	180-b	+ve	x	(180-b)	0	0

As mentioned earlier in the assumptions, all considered machine tools have at least 3 axes of motion. Therefore, the additional axes will either be one rotational axes, either A, B, C, or any two axes depending on the required machine capabilities. Table 4.2 shows the different notations used for representing the TADs and the calculation of the required axes.

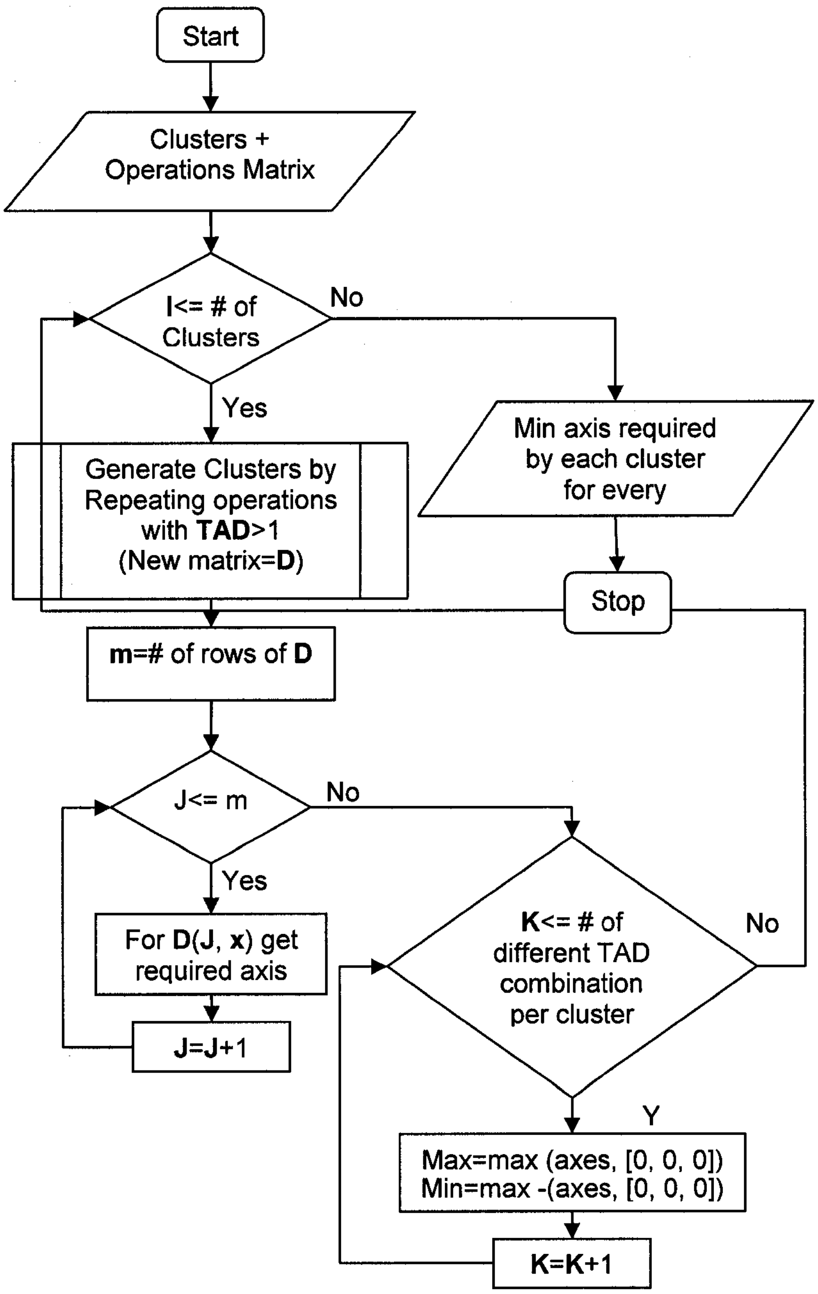


Figure 4.24: Calculation of Minimum Axes of Motion Required.

4.13. Case Study and Results

The developed machine structure configuration approach was applied to two test parts ANC-101 and ANC-090 (Figures 4.25 (a) and 4.25 (b) respectively). ANC-101 is the CAM-I, 1986 test part ANC-101 which is widely used in literature [Li *et al.* 2002, Ong *et al.* 2002, Kiritsis and Porchet 1996, Henderson *et al.* 1994, Gupta *et al.* 1994 and Hummel and Brown 1989]. The basic part (ANC-090) was developed as a variant of the test part (ANC-101). This part is similar to the part ANC-101 but with five fewer features. The case study will first be applied to the ANC-090 part then the ANC-101 because the ANC-090 is simpler and has less features.

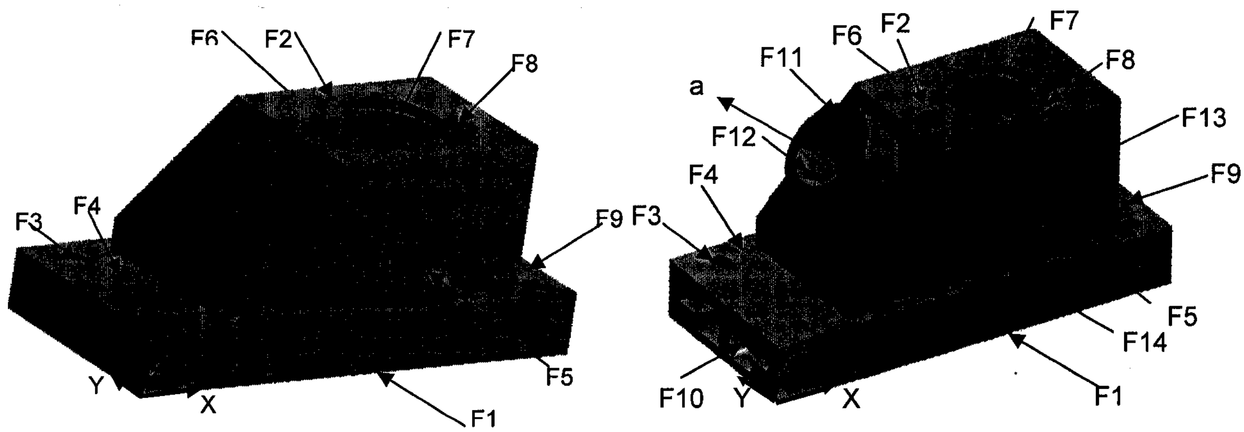


Figure 4.25: Example Parts ANC-090 (a) and ANC-101 (b) with Features.

Table 4.3 provides the details for the features and operation data for part ANC-101. ANC-101 contains 14 Features. It consists of two compound holes, one at an angle of 45°. There are two symmetric protrusions. The top surface contains 9 symmetric holes and the lower surface contains 4 holes at the corners. There is also two pockets arranged as a replicated feature on one of its sides. There are several datum and logical constraints. For example, Planer surface (F1) has to be milled before all operations because that surface is used as a datum and supporting face for all other operations. Compound hole (F7), drilling is before reaming which is before boring due to logical constraints. Same rule applies to the second compound hole (F12). Milling top surface (F2) is before drilling compound hole (F7) and drilling the nine holes (F8) for the material removal interactions. The step (F4) is before the four holes (F3) and the boss (F11) for the datum constraint and material removal interactions. Protrusion ribs (F5) and (F6) are before compound hole (F7) for the datum constraints. Boss (F11) is before compound hole (F12) for the datum and material removal interactions. Step (F9) is before the four holes (F3) and the pocket (F13) for the material removal interaction. Op10 is before Op11 due to logical constraints. Op19 is before Op20 due to logical constraints also.

Table 4.3: Operations Data for Part ANC-101.

Feature	Description	Operation	Op. ID	TAD candidates	Tool candidates
F1	Planar surface	Milling	Op1	+Z	C6, C7, C8
F2	Planar surface	Milling	Op2	-Z	C6, C7, C8
F3	Four holes arranged as a replicated feature	Drilling	Op3	+Z, -Z	C2
F4	A step	Milling	Op4	+X, -Z	C6, C7
F5	A protrusion (rib)	Milling	Op5	+Y, -Z	C7, C8
F6	A protrusion	Milling	Op6	-Y, -Z	C7, C8
F7	A compound hole	Drilling	Op7	-Z	C2, C3, C4
		Reaming	Op8		C9
		Boring	Op9		C10
F8	Nine holes arranged in a replicated feature	Drilling	Op10	-Z	C1
		Tapping	Op11		C5
F9	A step	Milling	Op12	-X, -Z	C6, C7
F10	Two pockets arranged as a replicated feature	Milling	Op13	+X	C6, C7. C8
F11	A boss	Milling	Op14	-a	C7, C8
F12	A compound hole	Drilling	Op15	-a	C2, C3, C4
		Reaming	Op16		C9
		Boring	Op17		C10
F13	A pocket	Milling	Op18	-X	C7, C8
F14	A compound hole	Reaming	Op19	+Z	C9
		Boring	Op20		C10

4.13.2 ANC-090

Table A.2 provides the different features' description, the operations required to produce these features, their IDs and the tool approach direction (TAD) for each operation for the ANC-090.

4.13.2.1 Clustering

After applying the clustering algorithm, the 12 operations were grouped into 7 different clusters as shown in Figure 4.26 and Table 4.4. This along with Table A.2 will be used as input to the second stage of generating the required machine structure. The 12 operations were clustered into 7 different clusters because of the dimensional constraints and logical constraints. It is desirable to group operations that have datum relationship or tight tolerances into one operation cluster so that it is performed on the same machine tool. If the operations are carried out on different machine tools, then very high accuracy would be required which can be avoided if the two operations are performed in the same setup. Also with logical constraints, it is desirable to perform the operations in the same setup because normally the only difference between the operations would be to just change the tool (e.g. drilling, reaming and boring for the same hole).

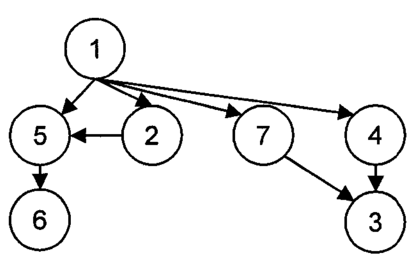


Figure 4.26: Operation Clusters Precedence Graph for Part ANC-090.

4.13.2.2 Generating Machine Tool Structure

Figure 4.27 shows the result of applying the generating machine structure algorithm shown in Figure 4.24 to the data obtained from the clustering stage. In Figure 4.27, each rectangle represents an operation cluster where the 0 or 1 indicates the possible TAD.

Table 4.4: Operation Clusters for Part ANC-090.

Operation Cluster	Operations
OC1	[OP1]
OC2	[OP2]
OC3	[OP3]
OC4	[OP4]
OC5	[OP5, OP6, OP7, OP8, OP9]
OC6	[OP10, OP11]
OC7	[OP12]

On the bottom right of each rectangle is the number of different possible TAD combinations in each cluster. For example, OC5 has 5 different operations (5, 6, 7, 8, and 9). Operations 5 and 6 each have two possible TADs. This means that this operation could be machined from two different angles. It also means that two machines with different capabilities could perform this operation. In the case of Op5 a machine with +ve Y TAD or –ve Z TAD can be used. If +ve Y is used then a 4-axis machine will be required and if –ve Z is used then a three-axis machine will do the job. For this reason all possible combinations of TADs for a cluster should be considered. Figure 4.28 shows the different combinations for OC5.

Each one of the four different TAD combinations (Figure 4.28) requires different minimum machine capabilities. If taking the –Z TAD for both operations 5 and 6 (case 4), then only a 3-axis machine is required to produce the cluster. On the other hand if +Y for Op5, and –Y for Op 6 are taken, then a module rotating around the X direction is required with at least a minimum range of rotation between -90 and 90 degrees. Figure 4.29 shows the different machine structures that could be used for each of the four cases.

		TAD					
OC	OP	X	-X	Y	-Y	Z	-Z
OC1	1	0	0	0	0	1	0
						NR_1=1	
OC2	2	0	0	0	0	0	1
						NR_2=1	
OC3	3	0	0	0	0	1	0
	3	0	0	0	0	0	1
						NR_3=2	
OC4	4	1	0	0	0	0	0
	4	0	0	0	0	0	1
						NR_4=2	
OC5	5	0	0	1	0	0	0
	5	0	0	0	0	0	1
	6	0	0	0	1	0	0
	6	0	0	0	0	0	1
	7	0	0	0	0	0	1
	8	0	0	0	0	0	1
	9	0	0	0	0	0	1
						NR_5=4	
OC6	10	0	0	0	0	0	1
	11	0	0	0	0	0	1
						NR_6=1	
OC7	12	0	0	0	0	0	1
	12	0	1	0	0	0	0
						NR_7=2	

OP: Operation Number
OC: Operation Cluster Number

Figure 4.27: Required Machine Capabilities.

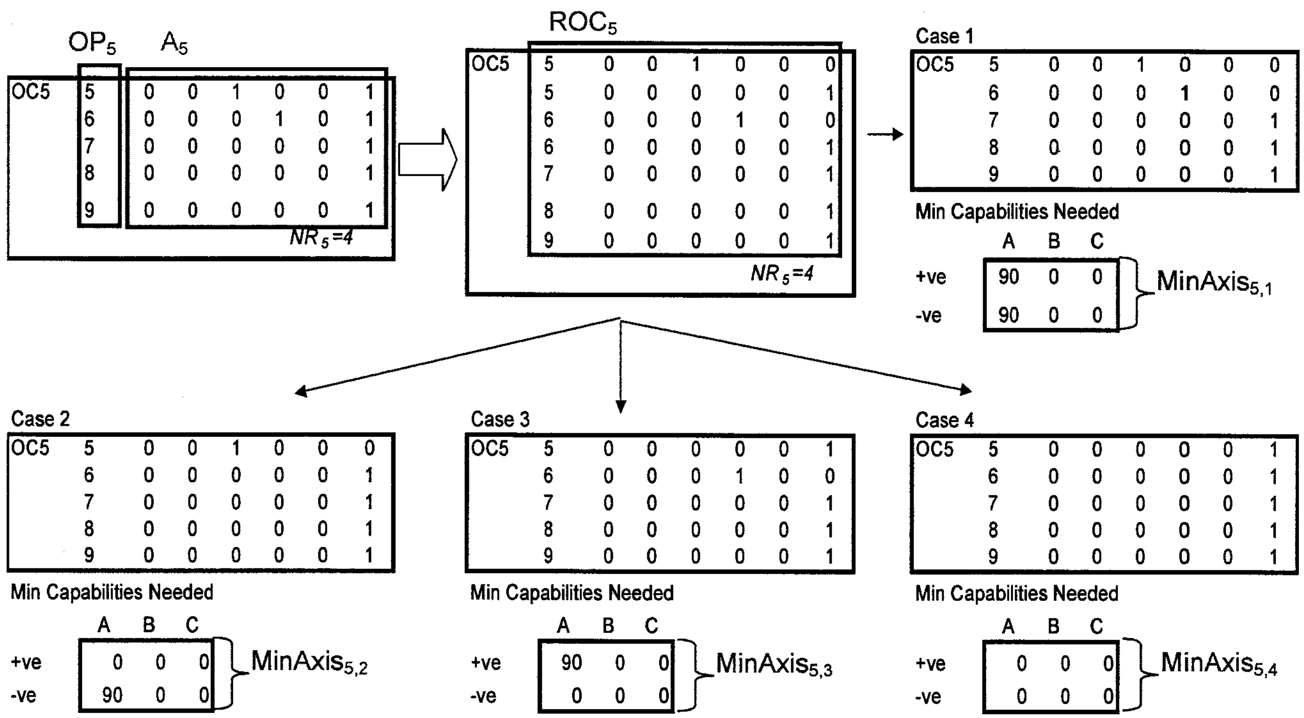


Figure 4.28: Different TAD Combinations of Operation Cluster 5.

The value shown next to the translation axis shows the minimum stroke length needed for the machine to be capable to perform the part and the value next to the rotational angle shows the minimum degrees or rotation needed for the machine to be capable of performing the OC.

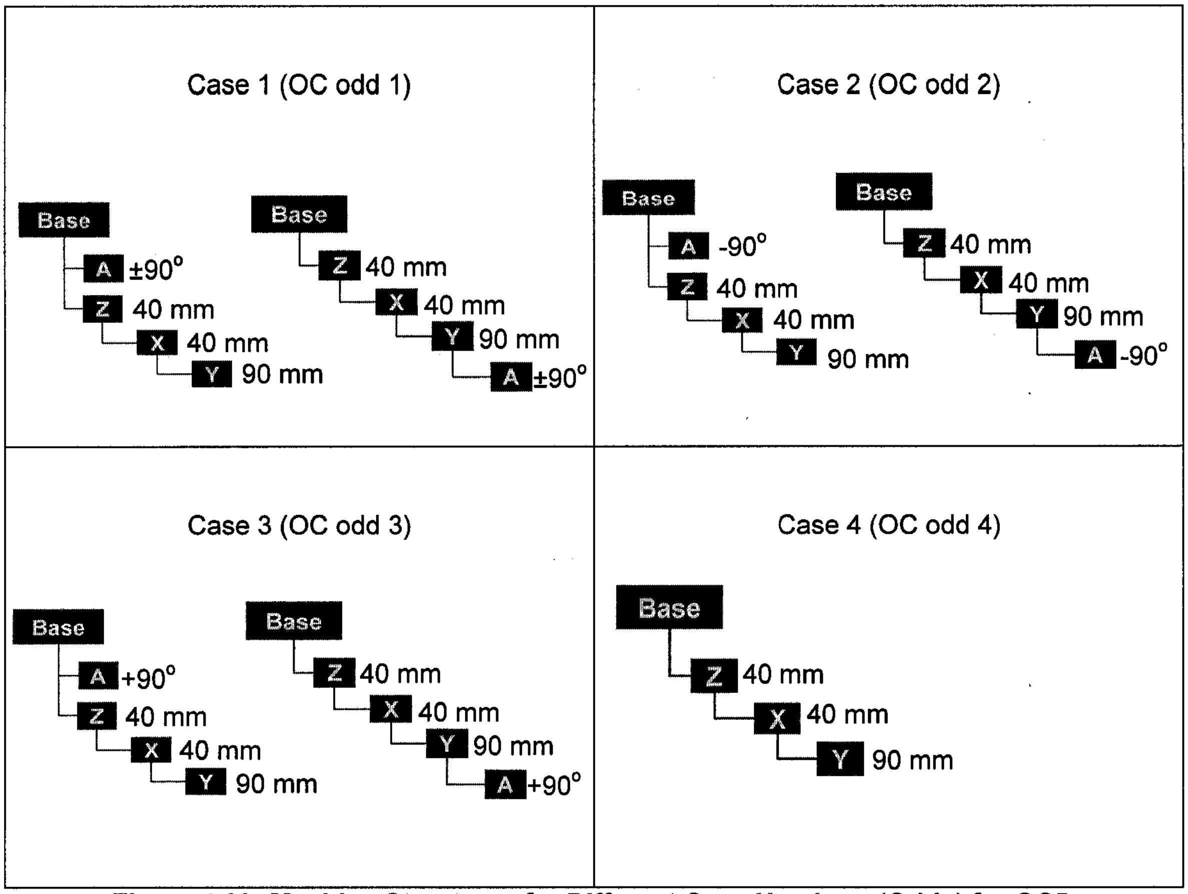


Figure 4.29: Machine Structures for Different Case Numbers (Odds) for OC5.

Figure 4.30 shows the output of the algorithm showing the minimum axes of rotation (capability) needed for each cluster including the different combinations. Figure 4.31 shows an example of the machine that could be used and also the odd (case) number chosen from Figure 4.30 based on the minimum rotation angle.

OC	Case 1	Case 2	Case 3	Case 4
1	0 180 0 0 0 0			
2	0 0 0 0 0 0			
3	0 180 0 0 0 0	0 0 0 0 0 0		
4	0 90 0 0 0 0	0 0 0 0 0 0		
5	90 0 0 90 0 0	0 0 0 90 0 0	90 0 0 0 0 0	0 0 0 0 0 0
6	0 0 0 0 0 0			
7	0 0 0 0 90 0	0 0 0 0 0 0		

Figure 4.30: Minimum Axes of Rotation Required for Every Operation Cluster and their Different Cases for ANC-090.

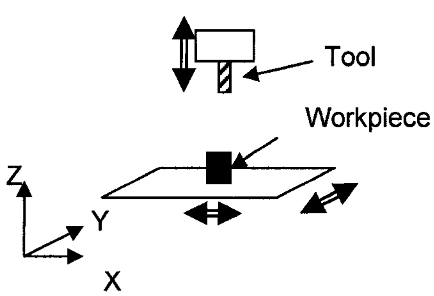
Machine Description	Machine Example	Operation Clusters	Case (Odd) Number
3-axis Machine Tool		OC1	1
		OC2	1
		OC3	2
		OC4	2
		OC5	4
		OC6	1
		OC7	2

Figure 4.31: Examples of Machine with Minimum Capabilities that could be used for ANC-090.

4.13.3 ANC-101

Table A.1 provides descriptions of the different features, the operations required to produce them, their IDs and the tool approach direction (TAD) for each operation for the ANC-101 (Figure 4.32).

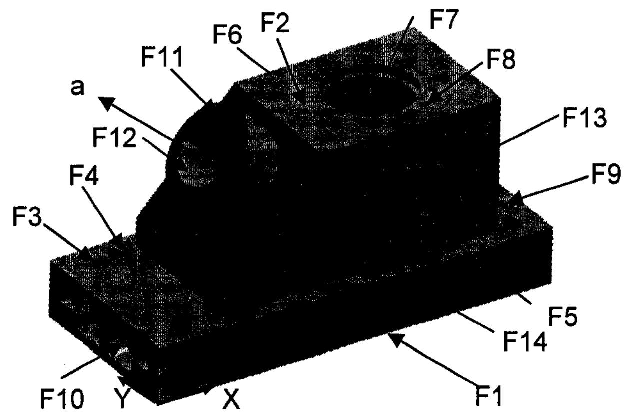


Figure 4.32: ANC-101 Part.

4.13.3.2 Clustering

After applying the clustering algorithm, the 20 operations were grouped into 11 different clusters shown in Figure 4.33 and Table 4.5. This along with Table A.1 will be used as input to the second stage of generating the machine structure.

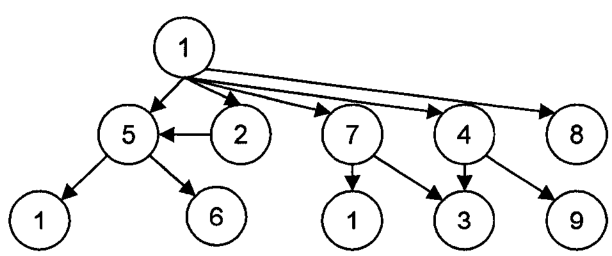


Figure 4.33: Operation Clusters Precedence Graph for Part ANC-101.

Table 4.5: Operation Clusters for Part ANC-101.

Operation Cluster	Operations
OC1	[OP1]
OC2	[OP2]
OC3	[OP3]
OC4	[OP4]
OC5	[OP5, OP6, OP7, OP8, OP9]
OC6	[OP10, OP11]
OC7	[OP12]
OC8	[OP13]
OC9	[OP14, OP15, OP16, OP17]
OC10	[OP18]
OC11	[OP19, OP20]

4.13.3.3 Generating the Machine Tool Structure

Figure 4.34, shows the result of applying the generating machine structure algorithm shown in Figure 4.24 to the data obtained from the clustering stage. In Figure 4.34, each rectangle represents an operation cluster where the 0 or 1 indicates the possible TAD. On the bottom right of each rectangle is the number of possible TAD combinations in each cluster.

Figure 4.35 shows the minimum axis of rotation (capability) needed by the machine for each cluster including the different TAD combinations. Figure 4.36 shows an example of the machine that could be used and also the case number chosen from Figure 4.35 based on the minimum rotation angle.

OC	OP	X	-X	Y	-Y	Z	-Z
		TAD					
OC1	1	0	0	0	0	1	0
						$NR_1=1$	
OC2	2	0	0	0	0	0	1
						$NR_2=1$	
OC3	3	0	0	0	0	1	0
	3	0	0	0	0	0	1
						$NR_3=2$	
OC4	4	1	0	0	0	0	0
	4	0	0	0	0	0	1
						$NR_4=2$	
OC5	5	0	0	1	0	0	0
	5	0	0	0	0	0	1
	6	0	0	0	1	0	0
	6	0	0	0	0	0	1
	7	0	0	0	0	0	1
	8	0	0	0	0	0	1
	9	0	0	0	0	0	1
						$NR_5=4$	
OC6	10	0	0	0	0	0	1
	11	0	0	0	0	0	1
						$NR_6=1$	
OC7	12	0	0	0	0	0	1
	12	0	1	0	0	0	0
						$NR_7=2$	
OC8	13	1	0	0	0	0	0
						$NR_8=1$	
OC9	14	0	0	0	a	0	0
	15	0	0	0	a	0	0
	16	0	0	0	a	0	0
	17	0	0	0	a	0	0
						$NR_9=1$	
OC10	18	0	1	0	0	0	0
						$NR_{10}=1$	
OC11	19	0	0	0	0	1	0
	20	0	0	0	0	1	0
						$NR_{11}=1$	

OP: Operation Number
OC: Operation Cluster Number

Figure 4.34: Required Machine Capabilities.

OC	Case 1	Case 2	Case 3	Case 4
1	0 180 0 0 0 0			
2	0 0 0 0 0 0			
3	0 180 0 0 0 0	0 0 0 0 0 0		
4	0 90 0 0 0 0	0 0 0 0 0 0		
5	90 0 0 90 0 0	0 0 0 90 0 0	90 0 0 0 0 0	0 0 0 0 0 0
6	0 0 0 0 0 0			
7	0 0 0 0 90 0	0 0 0 0 0 0		
8	0 90 0 0 0 0			
9	0 0 0 45 0 0			
10	0 0 0 0 90 0			
11	0 180 0 0 0 0			

Figure 4.35: Minimum Axes of Rotation Required for Operation Cluster Every and their Different Cases for ANC-101.

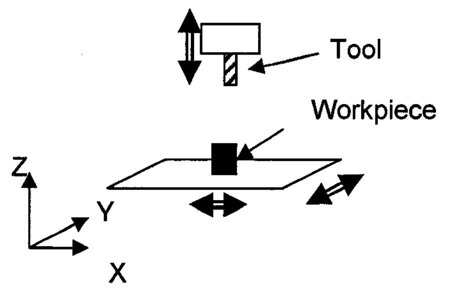
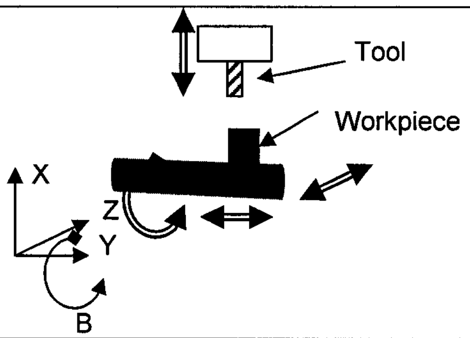
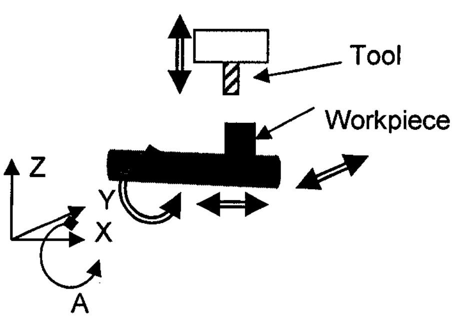
Machine Description	Machine Example	Operation Clusters	Case (Odd) Number	Rotation Angle
3-axis machine		OC1	1	180
		OC2	1	
		OC3	2	
		OC4	2	
		OC5	4	
		OC6	1	
		OC7	2	
		OC11	1	180
4- axis machine (rotation around y-axis)		OC8	1	+90
		OC10	1	-90
4- axis machine (rotation around x-axis)		OC9	1	-45

Figure 4.36: Examples of Machine with Minimum Capabilities that could be used for ANC-101.

4.14. Discussion of the Results

A 3-axis machine can produce the whole part in case of ANC-090. On the other hand, and based on the results shown in Figure 4.36, it is sufficient to use a 3-axis machine for most of the operation clusters of part ANC-101 except for three OCs. OC8 and OC10 need another rotation axis B around the Y-axis while OC9 needs a 4-axis machine with the fourth axis being a rotation A around the X-axis.

Both parts have to be rotated 180 degrees before fixturing when carrying out OC1 for both parts and OC10 for part ANC-101. This is illustrated by the 180 indicated in Figures 4.30 and 4.35 which means machining in the –ve z direction of the X-Y plane (the plane of part fixation). Therefore, machining of both parts can be achieved by fixing the part on a pallet and moving it from one machine to another without re-fixuring the part except for the machines that are assigned to the OCs which require 180 degrees of rotation.

From the results shown in Figures 4.30 and 4.35, a single operation cluster could have various sets of machine requirements, corresponding to the different combinations of TADs, which increases the flexibility of machine selection.

In traditional methods, operations are assigned to machines that are capable of carrying out every operation, which could result in assigning an operation to a machine that has excess capabilities. Although this provides a temporary solution, it is not necessarily the best if there are frequent changes in the product demand requirements, which is the case in today's market. This is because a problem might occur if a new more complex part that needs machines with more capabilities is introduced to the system and the capable machines were already assigned to the simpler part that did not need all those capabilities. The machine structure configuration approach, introduced, solves this problem. It determines the machines with minimum required capabilities to corresponding operation clusters to achieve better part/machine assignments. In addition, this fits with the concept of RMS that is meant to provide the capacity and functionality needed when needed by configuring the most appropriate machine for the task(s).

4.15. Summary and Conclusions

Different approaches are required for process planning and machine selection for different manufacturing systems depending on their nature. The challenges in RMS were outlined and showed that RMS needs a new concept of process planning that makes use of the reconfiguration capabilities that allows the machine structures to be tailored to the parts machining requirements.

A machine structure configuration approach was proposed. It introduced the concept of mapping between the processing requirements of parts and the structural requirements of reconfigurable machine tools capable of producing these parts. Given a part with its features and design specifications, operation clustering is performed. This guarantees that operations with dimensional or logical constraints will be assigned to the same machine. The minimum required machining capabilities are then generated. This can help in automatically determining/configuring machines that are capable of performing the required operations based on their kinematic structures. This is one of the main advantages/contributions of the proposed procedure when compared to traditional methods which require manual determination of candidate machines for each operation as a prerequisite for process planning. This will help in automating the process of machine selection in commercial CAM systems because for current CAM systems to generate a process plan for a given part, the machine has to be manually selected as a prerequisite. The procedure selects the capable machines for performing each individual OC depending on the capabilities required. If the machines in the machine database are traditional non reconfigurable machine tools, the approach will also select the capable machines.

The proposed procedure was applied to a case study for illustration and the machine capabilities needed for different operation clusters were demonstrated. The output showed that for a single operation cluster there could be more than one minimum machine configuration required. This increases the flexibility in selecting/configuring a suitable machine tool and reduces the risk of not finding a capable machine if a new part is introduced which is another major contribution of the presented work.

The proposed approach relies and builds upon the kinematic structure representation of machine tools. The approach is general, and can be applied to any manufacturing system, not only RMS, where dynamic and flexible process planning and machine assignments are required. This can be an important tool in aiding the process planner at the initial stages of manufacturing systems planning and design. This work will be taken a step further towards complete generation of reconfigurable dynamic process plans in the next chapters.

CHAPTER FIVE

MODELING AND OPTIMIZATION OF PROCESS PLANS FOR RMS

This chapter provides a novel model for optimizing the manufacturing cost of process plans for RMS by choosing the following parameters: machine assignment, machine configuration, operation sequencing, operation cluster sequencing and assigning the tools and tool approach directions (TAD) to the operations. The mathematical model and a novel constraint satisfaction procedure are presented and the optimization problem is solved using GAs. A toolbox was developed using MATLAB® software to demonstrate the use of the developed optimization model, which is verified using a case study based on an example part from the literature. The model was also validated by solving the same problem used in literature and comparing the results. The results are presented and analyzed.

5.1. Process Plan String Representation

A new representation is needed to represent variables on OC level (OC, M, and MC) and operation level (OP, TAD and Tool). This representation requires a new formulation to capture both levels. In literature a typical process plan is illustrated in Figure 5.1. A typical process plan provides the sequence of operations, the TAD and tool used for each operation, and the machine assignment. There is now a need for the new process plan to represent the assignment of OCs and the set of operations assigned to each OC. Also there is a need to represent the machine configuration.

Sequence of Operations

Operation Seq	P_7	P_4	P_8	P_2	P_1	P_6	P_3	P_5
Machine	M_3	M_3	M_3	M_2	M_2	M_4	M_1	M_1
TAD	-z	-z	+x	+x	+x	-y	-z	-z
Tool Used	T_1	T_2	T_2	T_3	T_3	T_1	T_1	T_4

Figure 5.1: Illustration of a Typical Process Plan Representation.

For the proposed process planning approach a new representation is proposed (Figure 5.2). The representation has three variables (Operation cluster sequence, Machine Sequence and machine configuration corresponding to each machine) having a string length equal to the number of operation clusters. And three variables (Operations Sequence, TAD for each operation and the Tool Used for each operation) having a sting length equal to the number of operations.

Sequence of Operations

Oper Clust Seq.	**OC_7**		**OC_8**			**OC_6**	**OC_3**	
Machine	M_3		M_3			M_4	M_1	
Configuration	C_1		C_3			C_1	C_2	
Operation Seq	P_7	P_4	P_8	P_2	P_1	P_6	P_3	P_5
TAD	-z	-z	+x	+x	+x	-y	-z	-z
Tool Used	T_1	T_2	T_2	T_3	T_3	T_1	T_1	T_4

Figure 5.2: Illustration of a New Process Plan Representation.

5.2. Mathematical Model

This section presents the optimization mathematical model based on the parameters and data structures defined in Section 3.1, 3.2 and 3.4 for input and in Section 3.5 for output.

5.2.1 Decision Variables

Operation Cluster Sequence (OCS)

OCS $=\{oc_1, oc_2, \ldots, oc_{NOC}\}$, where;

oc_i is Operation Cluster taking the i^{th} position in the sequence,

NOC is the number of clusters.

OC Odd Used (OD)

Refer to section 3.4.1.2 for the difinition of an OC odd.

OD$=\{ od_1, od_2, \ldots, od_{NOC} \}$, where;

od_i is the TADs OC odd used for oc_i

Machines Sequence (MS)

MS$=\{ m_1, m_2, \ldots, m_{NOC} \}$, where;

m_i is the machine type assigned to the OC in the i^{th} position of the sequence

Machines Configuration Sequence (MCS)

MCS$=\{ c_1, c_2, \ldots, c_{NOC} \}$, where;

c_i is the machine configuration used for machine m_i

Operation Sequence (OPS)

$OPSC=\{ op_1, op_2, \ldots, op_{NOP}\}$, where;

op_x is the operation taking the x^{th} position in the sequence of operations

TAD Sequence (TADS)

$TADS=\{ td_1, td_2, \ldots, td_{NOP} \}$, where;

td_x is the TAD assigned to operation op_x

Tools Used (TS)

$TS=\{ t_1, t_2, \ldots, t_{NOP} \}$, where;

t_x is tool type assigned to operation op_x

5.2.2 Objective Function and Constraints

The objective function it to minimize the total cost. The cost function used is similar to that used by Zhang *et al.* [1997] which was modified to add the reconfiguration aspect in the model.

$$\text{Min } TC= MUC+TUC+MCC+TCC+SCC \qquad (5.1)$$

Where the cost elements are:

Machine Usage Cost (MUC)

Cost of using each machine in the process plan.

$$MUC = \sum_{i=1}^{NOC} CM_{m_i}(c_i) \times MCI \;, \qquad (5.2)$$

where MCI is the machine cost index

Tool Usage Cost (TUC)

The cost of using the cutting tools for each operation.

$$TUC = \sum_{x=1}^{NOP} CT(t_x) \times TCI , \tag{5.3}$$

where TCI is the tool cost index

Machine Change Cost (MCC)

Cost of changing a machine or machine configuration in a process plan sequence (a configuration change is considered a machine change).

$$MCC = MCCI \times \sum_{i=1}^{NOC-1} \left[1 - \left(1 - \Omega(MS(i), MS(i+1))\right) \times \left(1 - \Omega(MCS(i), MCS(i+1))\right)\right], \tag{5.4}$$

where; $\Omega(x,y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases}$

Tool Change Cost (TCC)

Cost of changing cutting tools. Changing a tool between operations in the same OC is considered a tool change. Also a tool change takes place when two consecutive OCs that use the same machine and machine configuration use different cutters.

$$\begin{aligned} TCC = {} & TCCI \times \sum_{i=1}^{NOC-1} \sum_{x=1}^{NOPCS_i - 1} \Omega\left(TS\left(\left(\sum_{z=1}^{i} NOPCS_i \right) - NOPCS_i + x \right), TS\left(\left(\sum_{z=1}^{i} NOPCS_i \right) - NOPCS_i + x + 1 \right) \right) \\ & + TCCI \times \sum_{i=1}^{NOC-1} \left((1 - \Omega(MS(i), MS(i+1))) \times (1 - \Omega(MCS(i), MCS(i+1))) \times \Omega\left(TS\left(\sum_{z=1}^{i} NOPC_i \right), TS\left(\sum_{z=1}^{i} NOPC_i + 1 \right) \right) \right) \end{aligned} \tag{5.5}$$

where;

The 1st part of the equation counts the number of tool changes within the same OC for all the OCs

The 2nd part of the equation counts the number tool changes for every two consecutive OCs that use the same machine and machine configuration.

Number of Setup Change Cost (SCC)

SSC represents the setup cost or the cost of changing the TAD. A setup change takes place if there is change in the TAD between operations in sequence within the same OC. Also a setup change takes place when two consecutive OCs that use the same machine and machine configuration use different TAD.

$$
\begin{aligned}
TDCC = {} & TDCCI \times \sum_{i=1}^{NOC-1} \sum_{x=1}^{NOPCS_i - 1} \Omega\left(TADS\left(\left(\sum_{z=1}^{i} NOPCS_i \right) - NOPCS_i + x \right), TADS\left(\left(\sum_{z=1}^{i} NOPCS_i \right) - NOPCS_i + x + 1 \right) \right). \\
& + TDCCI \times \sum_{i=1}^{NOC-1} \left((1 - \Omega(MS(i), MS(i+1))) \times (1 - \Omega(MCS(i), MCS(i+1))) \times \Omega\left(TADS\left(\sum_{z=1}^{i} NOPC_i \right), TADS\left(\sum_{z=1}^{i} NOPC_i + 1 \right) \right) \right)
\end{aligned} \tag{5.6}
$$

5.2.3 Constraints

Subject To

Precedence Constraint for Clusters

All operation satisfy the precedence constraints of the clusters

$OCP\ (oc_i, oc_j) = 0\ \forall\ i>j,\ \forall i,\ i=1,\ldots,\ NOC$ (5.7)

Clusters Are Assigned Only Once

Operation Clusters should only be assigned once

$OC_i \neq OC_j,\ \forall i \neq j$ (5.8)

Machine Configuration Capabilities

Machines should be capable of manufacturing OC assigned to it.

$$MCAP_{i,m_i}(od_i, c_i) = 1 \forall i,\ i=1,\ldots, NOC \tag{5.9}$$

Precedence constraints for operations

$$OPP\ (op_x,\ op_y) = 0\ \forall\ x>y, \text{ where } x,y = 1\ \ldots,\ NOP \tag{5.10}$$

Operation should only be assigned once

$$OP_x \neq OP_y,\ \forall x \neq y \tag{5.11}$$

Operations with Tolerance or Logical Constraints are Assigned to the same Operation Cluster

Assign every two operation that have tolerance constraints (value of 2 in OPP) between them to the same OC.

$$OPC\ (oc_i,\ op_x) = OPC\ (oc_i,\ op_y)\ \forall\ OPP\ (op_x,\ op_y) = 2\ \forall\ i,\ x,\ y. \tag{5.12}$$

Where $i = 1\ \ldots,\ NOC$, x and $y = 1,\ \ldots,\ NOP$

Assign every two operation that have logical constraints (value of 3 in OPP) between them to the same OC.

$$OPC\ (oc_i,\ op_x) = OPC\ (oc_i,\ op_y)\ \forall\ OPP\ (op_x,\ op_y) = 3\ \forall\ i,\ x,\ y \tag{5.13}$$

Where $i = 1\ \ldots,\ NOC$, x and $y = 1,\ \ldots,\ NOP$

Decision Variable Domain Constraints

Operation Cluster Sequence: *String* represents the sequence of OCs.

$$oc_i \in \{1,2,\cdots,NOC\}\ \forall\, i = 1,2,\cdots,NOC \tag{5.14}$$

OC Odd Used: String representation for the TAD odd used corresponding to the Operation Cluster Sequence.

$$od_i \in \{1,2,\cdots,NOCO(oc_i)\} \ \forall\, i = 1,2,\cdots,NOC \tag{5.15}$$

Machines Sequence: String Representation of Machine Sequence Corresponding to the Operation Cluster Sequence.

$$m_i \in \{1,2,\cdots,NM\} \ \forall\, i = 1,2,\cdots,NOC \tag{5.16}$$

Machine Configuration Sequence: String Representation of Machine Configuration Sequence Corresponding to the Machine Sequence.

$$c_i \in \{1,2,\cdots,NMC(m_i)\} \ \forall\, i = 1,2,\cdots,NOC \tag{5.17}$$

Operation Sequence: String representing the sequence of operations used in sequence.

$$op_x \in \{1,2,\cdots,NOP\} \ \forall\, x = 1,2,\cdots,NOP \tag{5.18}$$

TAD Sequence: String representing the TAD sequence of used for each operation in sequence.

$$td_x \in \{1,2,\cdots,NTAD(x)\} \ \forall\, x = 1,2,\cdots,NOP \tag{5.19}$$

Tools Used: String representing the Tool sequence of used for each operation in sequence.

$$t_x \in \{1,2,\cdots,6\} \ \forall\, x = 1,2,\cdots,NOP \tag{5.20}$$

5.3. Genetic Algorithm Method

Genetic Algorithms (GAs) introduced by Holland [1975] have been broadly used as a powerful meta-heuristic global optimization method that can solve NP-complete problems.

GAs work by mimicking the biological processes underlying classic Darwinian evolution. The implementation of GAs utilize a population of candidate solutions (called chromosomes). The value of the chromosomes in the current generation is evaluated using a fitness function and ranked. From the ranking candidates are selected from which the next generation is created. The process is repeats until a predefined number of generations.

The General GA procedure includes the following five steps:

1. Randomly generating initial solution
2. Evaluation of the fitness function for each chromosome and accordingly determine the ranking.
3. Selection operator
4. Application of the genetic crossover and mutation operators on the selected chromosomes.
5. Goto Step #2.

Traditional GAs code the independent variables into binary strings representing the chromosomes, which discretises the continuous domain variables. Coarse discretisation limits the search resolution and might lead to near to global optimal solutions. On the other hand, fine discretisation leads to long binary chromosomes and hence would increase the search space. Such increase may be drastic leading to prohibiting large search spaces [Michalewicz *et al.* 1994].

Currently, research in genetic algorithms tends to use real-coded representations for continuous parameter optimization problems [Hererra *et al.* 1998]. Such version of GAs is known as real-coded GAs and has the following advantages:

- Real parameters make it possible to use large domains for the independent variables.
- Real parameters tend to exploit the gradual changes in the objective function corresponding to gradual changes in the independent variables.

The real coded GAs was not used earlier for process plan optimization and is being introduced here for the above reasons to seek the near global optimal process plan. Appendix D provides details on the operators used.

5.4. Traditional Versus Proposed GA Approach

Figure 5.3 illustrates the traditional and proposed GA procedure for process planning. In traditional process planning the variables that are generated in the initial population are random and discrete, this results in a considerable amount of infeasible process plans. For this reason there is a need for a repair function to repair the infeasible process plan. This function is recursive and time consuming. In the proposed continuous domain GA procedure all generated random plans are feasible and for this reason there is no need for the repair function.

Also, in the traditional method after each generation, the repair function has to be executed for every infeasible process plan in that generation which reduces the efficiency of the algorithm. In the proposed procedure there is no need for the repair function even after the crossover and mutation operators because all generated process plans are feasible.

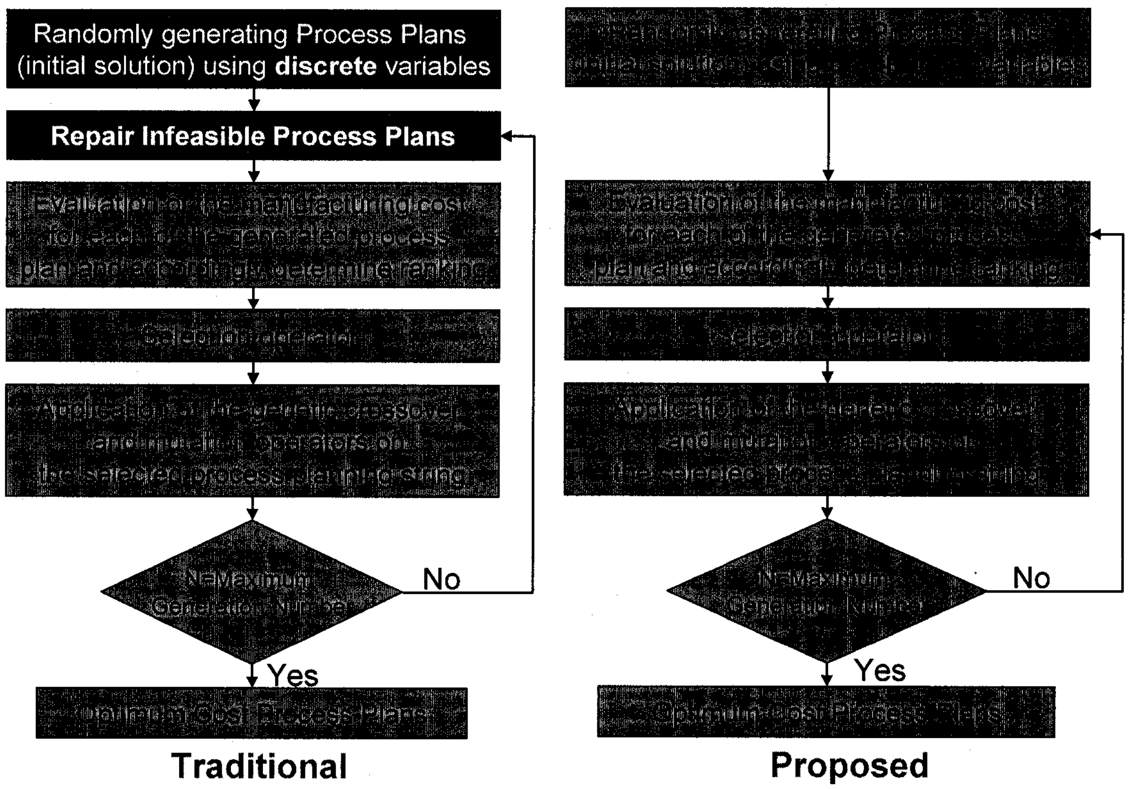


Figure 5.3: Traditional vs. Proposed GA Procedure.

5.5. String Representation and Proposed Real-Coded Approach

Previously discrete GAs where used in process planning but this had the disadvantage because the number of feasible alternatives varies depending on the operation sequence used. For example every operation has a different number of feasible machines, tools or TADs and accordingly the domain sizes of the alternatives to select from will vary. Another disadvantage occurs during crossover and mutation. When two feasible chromosomes perform crossover the resulting string could be infeasible and a repair function is needed to repair the chromosome.

The use of continuous domain variables solves these problems as it permits dealing with varying domain sizes while maintaining equal probabilities of selecting each alternative [Youssef & H. ElMaraghy 2006]. In addition, this facilitates the use of the proposed constraint satisfaction approach (discussed in the following section) for manipulation of the generated solutions in terms of crossovers and mutations for the purpose of always producing feasible solutions.

5.6. Decoding of Variables and Constraint Satisfaction Approach

To guarantee the satisfaction of specified constraints, the process plan solution shown in Figure 5.2 is expressed in a new domain of continuous variables ranging between 0 and 1.

Oper Clust Seq.	0.11		0.54			0.59	0.82	
TAD Odd Used	0.36		0.74			0.93	0.34	
Machine	0.55		0.61			0.17	0.22	
Configuration	0.23		0.54			0.86	0.37	
Operation Seq	0.13		0.42			0.37	0.9	
Tool Used	0.71	0.56	0.32	0.85	0.32	0.52	0.19	0.5

Figure 5.4: String Representation of the Encoded Process Plan.

Decoding is the translation of any of the produced encoded solution strings (Figure 5.4) to a full process plan as depicted by the solution string in Figure 5.2. The encoded string has five groups of variables (Operation Cluster Sequence, TAD Odd Used, Machine, Operation Sequence, and Tool Used) as shown in Figure 5.4. The size of the first 5 variables is equal to NOC and the size for the

Tool Used string is equal to NOP. Therefore, the number of variables for any given problem is equal to 5×NOC+NOP. The TAD used shown in Figure 5.2 for each operation is not required to be encoded because the TAD approach can be obtained from the TAD Odd Used.

The representation shown in Figure 5.4 has an advantage using less variables to represent the string shown in Figure 5.2. In Figure 5.2 the number of variables equals 3×NOC + 3×NOP as compared to the representation shown in Figure 5.4 having 5×NOC + NOP variables.

5.6.1 Decoding the Operation Cluster Sequence

The proposed constraint satisfaction approach works in the following manner. Each variable in the *Oper Clust Seq* string determines the selected feasible sequence of OCs. The number of feasible OCs at a specific point in the sequence is obtained by checking *OCP* after omitting the OCs that have already been sequenced in the *Oper Clust Seq* string to find the number of OCs that have no preceding OC. The feasible OCs are numbered in order starting from 1. This number determines the OC to use. The value of the continuous domain variable that ranges from 0 to 1 is multiplied by the total number of feasible OCs. and then rounded up to the nearest integer which will in turn represent the order of the OC to select in the current sequence. This method guarantees equal probability of selection for all the possible feasible OC sequences and guarantees that a feasible sequences is always generated.

5.6.2 Decoding the TAD Odd Used

The 0-1 value in the TAD Odd Used is multiplied by the number of possible TAD combinations for the OC corresponding to it (The OC that is in the same location in the *Oper Clust Seq* string) then rounded up to the nearest integer which will in turn represent the TAD combination number used.

5.6.3 Decoding the Machine Used

The 0-1 value in the Machine Used string is multiplied by the number of possible machines that are capable of producing the corresponding OC using the TAD odd used for that OC. The value is then rounded up to the nearest integer which will in turn represent the Machine used.

5.6.4 Decoding the Configuration Used

The 0-1 value in the Configuration string is multiplied by the number of possible machine configuration for the corresponding Machine Used String that are capable of producing the corresponding OC using the TAD odd used for that OC. The value is then rounded up to the nearest integer which will in turn represent the machine configuration used.

5.6.5 Decoding the Operation Sequence

Before decoding the OP sequence, all possible operation permutations (sequences) that do not violate precedence constraints for each of the OCs is generated and stored. The 0-1 value in the *Operation Seq* is multiplied by the number of possible OP permutations combinations for the OC corresponding to it then rounded up to the nearest integer which will in turn represent the permutation number used. This will in turn give the sequence of operations within the corresponding operation cluster. This method guarantees that within each operation cluster there is an equal probability of selection for all possible operation sequences.

5.6.6 Decoding the Tool Used

The 0-1 value in the *Tool Used* is multiplied by the number of possible cutters for the OP corresponding to it then rounded up to the nearest integer which will in turn represent the tool used.

5.7. Case Study

The proposed GA optimization model was applied to test part ANC-101.

5.7.1 Inputs

Inputs to the proposed model are divided into three input types:

a) Cost data input shown in Table 5.1. Cost data models for reconfigurable machine tools were adopted from Spicer [2002]. Table 5.2 shows the machines database containing the different machine capabilities. Figure 5.5 shows the structure for the available machines and their configurations.

b) Part data input for ANC-101 from Appendix A.

c) Output from Stage I and II. The Operation clusters for ANC-101 (Table 4.5). The OC odd showing the TAD used for parts ANC-101 (Figure 4.35) will be the same as before because they are function of the part and not the manufacturing system. From the machine database and the minimum required machine capabilities (from stage II), the capable machines for each OC odd are shown in Figure 5.6. The capable machines are obtained by selecting all the machines in the machine database that have equal or greater capabilities than the minimum required capabilities.

Table 5.1: Cost Information Used (Li *et al.* [2002] and Spicer [2002]).

ID	Type	Cost	ID	Type	Cost
1	1-Spindle 3-Axis	760	C1	Drill 1	7
2	1-Spindle 3-Axis RMT	860	C2	Drill 2	5
3	1-Spindle 4-Axis RMT	1010	C3	Drill 3	3
4	1-Spindle 4-Axis RMT	1010	C4	Drill 4	8
5	1-Spindle 5-Axis RMT	1110	C5	Tapping Tool	7
6	Drill Press	385	C6	Mill 1	10
	MCI = 0.1xMachine Cost		C7	Mill 2	15
	MCCI	160	C8	Mill 3	30
	TDCCI	100	C9	Ream	15
	TCCI	20	C10	Boring Tool	20

Table 5.2: Available Machine Tool Data.

			Stroke Length			Rotation Angles					
Mc_{ID}	M/C_{Base}	M/C_{Conf}	X	Y	Z	+X	-X	+Y	-Y	+Z	-Z
1	1	1	100	50	60	0	0	0	0	0	0
2	2	1	100	100	100	0	0	0	0	0	0
3	2	2	100	100	100	135	135	0	0	0	0
4	2	3	100	100	100	0	0	115	115	0	0
5	2	4	100	100	100	135	135	115	115	180	180
6	3	1	120	80	90	0	0	0	0	0	0

MC_{ID}	Structure	MC_{ID}	Structure

1	Base # 1
	└ X 100 mm
	  └ Z 60 mm
	    └ Y 50 mm
		2	Base # 2
			└ Z 100 mm
			  └ X 100 mm
			    └ Y 100 mm

3	Base # 2
	└ Z 100 mm
	  └ X 100 mm
	    └ Y 100 mm
	      └ A ±135°
		4	Base # 2
			└ Z 100 mm
			  └ X 100 mm
			    └ Y 100 mm
			      └ B ±115°

5	Base # 2
	└ Z 40 mm
	  └ X 40 mm
	    └ Y 90 mm
	      └ A ±135°
	        └ B ±115°
	          └ C ±180°
		6	Base # 3
			└ X 40 mm
			  └ Y 40 mm
			    └ Z 90 mm

Figure 5.5: Structure for Machines and Configurations in Database.

	Odd number for TAD Used			
OC	1	2	3	4
1	M1,M2,M3, M4,M5			
2	M1,M2,M3, M4,M5, M6			
3	M1,M2,M3, M4,M5, M6	M1,M2,M3, M4,M5, M6		
4	M4,M5	M1,M2,M3, M4,M5		
5	M3,M5	M3,M5	M3, M5	M1,M2,M3, M4,M5
6	M1,M2,M3, M4,M5, M6			
7	M4, M5	M1,M2,M3, M4,M5		
8	M4, M5	M1,M2,M3, M4,M5, M6		
9	M3, M5			
10	M4, M5			
11	M1,M2,M3, M4,M5, M6			

Figure 5.6: Capable Machines for Each TAD Odd for Every OC.

5.7.2 GA Parameters Used

Table 5.3 provides the population size, the number of generations and the number of times each operator is applied in this work. Michalewicz *et al.* [1994] may be consulted for a description of these operators.

Table 5.3: Parameters used in Real Coded GAs.

Parameter	Value
Population size	200
Number of generations	150
Number of times of cross-over operator (arithmetic cross-over, simple cross-over and heuristic cross-over)	6 times each
Number of times of mutation operator (uniform mutation, non-uniform mutation and whole non-uniform mutation)	12 times each

5.7.3 Results and Discussion

The developed MATLAB® toolbox was used, with the available machine data. Figure 5.7 demonstrates a sample of the GA convergence curves. The number of generations used was 150. The cross-over and mutation operators were each applied 6 and 12 times respectively per generation in this work.

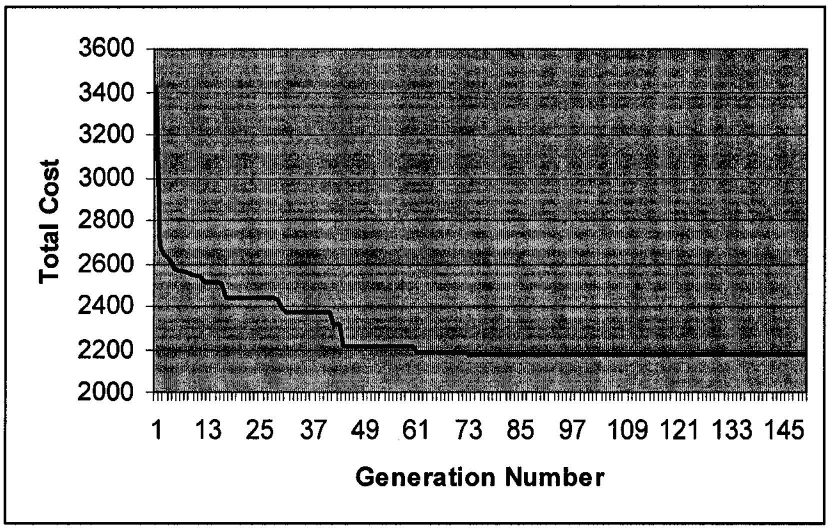


Figure 5.7: Convergence Curves Using a Population Size of 150.

The process plan with the least cost reached has a total of 2173.5 cost units. Figure 5.8 shows the corresponding optimal process plan representation. The 20 operations are grouped into 11 clusters; therefore, there are 75 variables. The encoded representation shown in Figure 5.4 reduced the original representation (figure 5.2) from 93 variables to 75 variables (i.e., problem was reduced by 18 variables).

The output shows that four machines are used. Machine ID 4 (Machine base number 2 using configuration 3) is used for the first 7 OCs (OC1, OC4, OC7, OC2, OC5, OC8 & OC10) in sequence. OC6 is then assigned to the drill press having machine ID 6 (Machine base number 3 using configuration 1). OC9 is then assigned to the RMT having machine ID 3 (Machine base number 2 using configuration 2). The remaining OCs (OC3 and OC11)are assigned to another machine having ID6. Figure 5.9 shows a diagram for the manufacturing system, showing the machine in sequence and the OCs assigned to each machine.

The number of tool (cutter) changes and TAD changes is counted for operations assigned to the same machine and not between two different machines. Therefore, the output shows the number of Tool changes is equal to 12 and the number of TAD change is equal to 3. Cost break down is as follows:

Machine Usage Cost (MUC)	=	923.5
Tool Usage Cost (TUC)	=	230
Machine Change Cost (MCC)	=	480
Tool Change Cost (TCC)	=	240
Setup Change Cost (SCC)	=	300
Total Cost	=	2173.5

<table>
<tr><td>Oper Clust Seq.</td><td>1</td><td>4</td><td>7</td><td>2</td><td colspan="5">5</td><td>8</td><td>10</td><td colspan="2">6</td><td colspan="4">9</td><td>3</td><td colspan="2">11</td></tr>
<tr><td>TAD Odd Used</td><td>1</td><td>2</td><td>2</td><td>1</td><td colspan="5">4</td><td>1</td><td>1</td><td colspan="2">1</td><td colspan="4">1</td><td>1</td><td colspan="2">1</td></tr>
<tr><td>Machine ID</td><td>4</td><td>4</td><td>4</td><td>4</td><td colspan="5">4</td><td>4</td><td>4</td><td colspan="2">6</td><td colspan="4">3</td><td>6</td><td colspan="2">6</td></tr>
<tr><td>Operation Seq</td><td>1</td><td>4</td><td>12</td><td>2</td><td>6</td><td>5</td><td>7</td><td>8</td><td>9</td><td>13</td><td>18</td><td>10</td><td>11</td><td>14</td><td>15</td><td>16</td><td>17</td><td>3</td><td>19</td><td>20</td></tr>
<tr><td>TAD</td><td>+Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>-Z</td><td>+X</td><td>-X</td><td>-Z</td><td>-Z</td><td>-a</td><td>-a</td><td>-a</td><td>-a</td><td>+Z</td><td>+Z</td><td>+Z</td></tr>
<tr><td>Tool Used</td><td>6</td><td>6</td><td>6</td><td>6</td><td>7</td><td>7</td><td>3</td><td>9</td><td>10</td><td>2</td><td>6</td><td>1</td><td>5</td><td>3</td><td>9</td><td>10</td><td>7</td><td>2</td><td>9</td><td>10</td></tr>
</table>

Figure 5.8: Representation of Optimal Reached Process Plan.

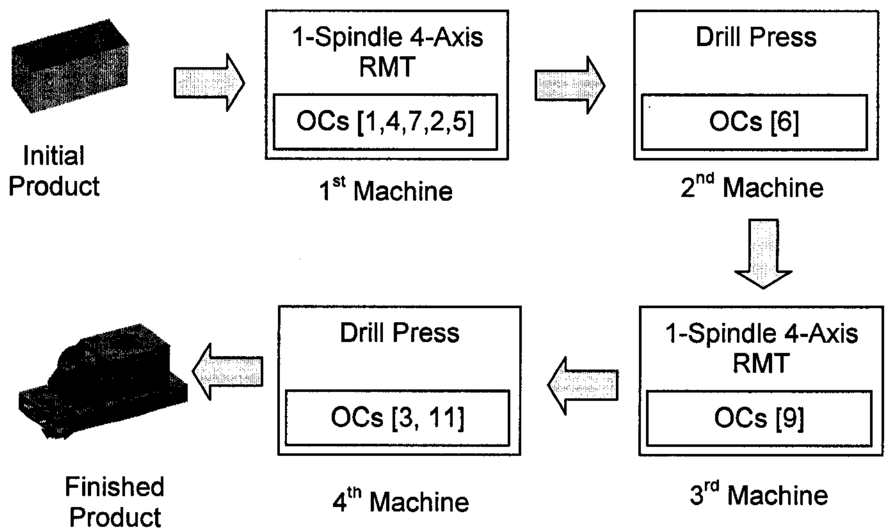


Figure 5.9: Manufacturing System Layout.

5.8. Application to Traditional Manufacturing Systems

The proposed GA optimization model was also applied to test part ANC-101 but in a traditional manufacturing system where there is no RMTs.

5.8.1 Inputs

Inputs to the proposed model are divided into three input types:

a) Cost data input shown in Table 5.4.

b) Part data input from Appendix A.

c) Output from Stage I and II. The Operation clusters (Table 4.5) and the OC odd showing the TAD used (Figure 4.35) will be the same as before because they are function of the part and not the manufacturing system.

For the new manufacturing system, the capable machine for each OC Odd is shown in Figure 5.10.

Table 5.4: Cost Information Used [Ong *et al.* 2002].

ID	Type	Cost	ID	Type	Cost
M1	3-axis machine	100	C1	Drill 1	7
M2	3-axis CNC	200	C2	Drill 2	5
M3	4-axis CNC (A rotation ±135°)	300	C3	Drill 3	3
M4	4-axis CNC (A rotation +90°)	290	C4	Drill 4	8
M5	4-axis CNC (B rotation ±120°)	320	C5	Tapping Tool	7
M6	5-axis CNC	450	C6	Mill 1	10
			C7	Mill 2	15
	MCCI	1000	C8	Mill 3	30
	TDCCI	120	C9	Ream	15
	TCCI	15	C10	Boring Tool	20

	Odd number for TAD Used			
OC	1	2	3	4
1	M1,M2,M3, M4,M5, M6			
2	M1,M2,M3, M4,M5, M6			
3	M6	M1,M2,M3, M4,M5, M6		
4	M5,M6	M1,M2,M3, M4,M5, M6		
5	M3,M6	M3,M6	M3, M4, M6	M1,M2,M3, M4,M5, M6
6	M1,M2,M3, M4,M5, M6			
7	M5, M6	M1,M2,M3, M4,M5, M6		
8	M5,M6			
9	M3, M4, M6			
10	M5, M6			
11	M1,M2,M3, M4,M5, M6			

Figure 5.10: Capable Machines for Each TAD Odd for Every OC.

5.8.2 GA Parameters Used

Table 5.5 provides the population size, the number of generations and the number of times each operator is applied in this work. Michalewicz *et al.* [1994] may be consulted for a description of these operators.

Table 5.5: Parameters used in Real Coded GAs.

Parameter	Value
Population size	200
Number of generations	150
Number of times of cross-over operator (arithmetic cross-over, simple cross-over and heuristic cross-over)	6 times each
Number of times of mutation operator (uniform mutation, non-uniform mutation and whole non-uniform mutation)	12 times each

5.8.3 Results and Discussion

The developed MATLAB® toolbox was used, this time with the new available machine data. Figure 5.11 demonstrates a sample of the GA convergence curves. Figure 5.12 shows the convergence curves using different population sizes. The number of generations used was 150. The cross-over and mutation operators were each applied 6 and 12 times respectively per generation in this work.

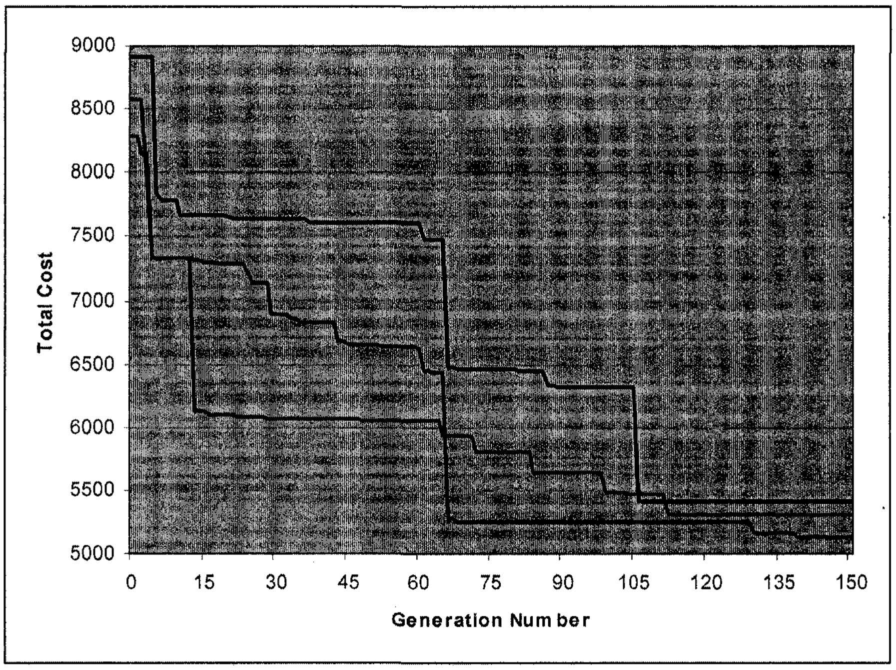


Figure 5.11: Convergence Curves Using a Population Size of 200.

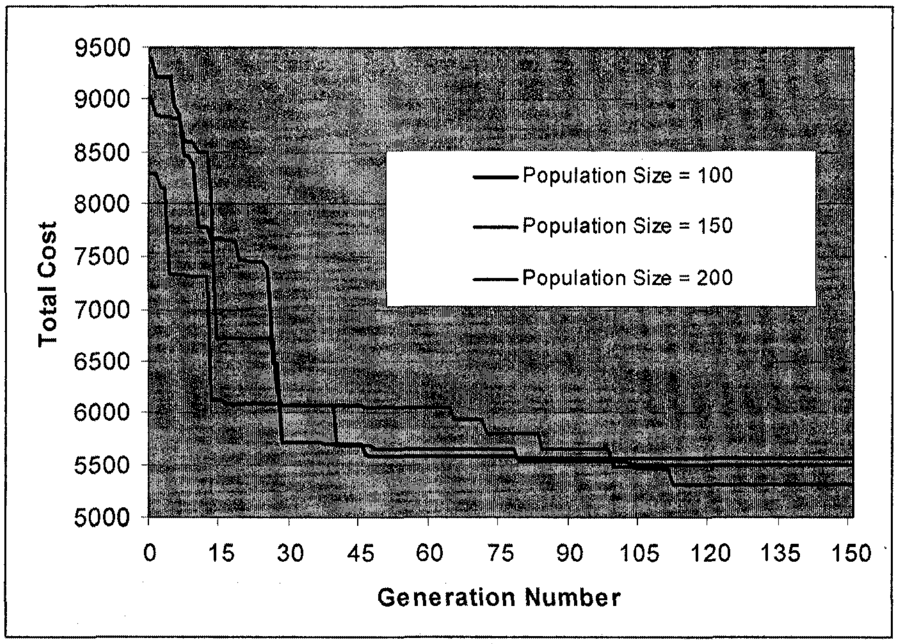


Figure 5.12: Convergence Curve for Different Population size.

The process plan with the least cost reached has a total of 5125 cost units. Figure 5.13 shows the corresponding optimal process plan representation. The 20 operations are grouped into 11 clusters; therefore, there are 75 variables. The encoded representation shown in Figure 5.4 reduced the original representation (Figure 5.2) from 93 variables to 75 variables (i.e., problem was reduced by 18 variables).

The output shows that only two machines are used. M6 is used for the 1st 6 OC (OC1, OC8, OC4, OC7, OC10 & OC9) in sequence and the remaining OCs (OC3, OC2, OC4, OC6 & OC11) in sequence are assigned to machine M1. There are two setups. The OCs assigned to machine M6 are the first setup and those assigned to M1 are the second setup. Figure 5.14 shows a diagram for the manufacturing system, showing the machine in sequence and the OCs assigned to each machine.

The number of tool (cutter) changes and TAD changes is counted for operations assigned to the same machine and not between two different machines. Therefore, the output shows the number of Tool changes is equal to 14 and the number of TAD change is equal to 4. Cost break down is as follows:

Machine Usage Cost (MUC)	=	3200
Tool Usage Cost (TUC)	=	235
Machine Change Cost (MCC)	=	1000
Tool Change Cost (TCC)	=	210
Setup Change Cost (SCC)	=	480
Total Cost	=	5125

The computation time required was on average 1 min/run on a Pentium 4 2.6 GHz PC with 512 MB memory. This is a reasonable time considering the large solution space containing 64 variables with over 860 constraints.

Although the cost for the RMT (Table 5.2) are higher when compared to normal CNC machines (Table 5.4) which is logical because of the reconfigurability features and more complicated structures and control of RMTs as compared the standard CNC machines, the results show that the cost of process planning for RMS is cheaper . The reason for this result is because the cost indices are higher in the second example (traditional manufacturing systems). This is because the cost indices are different. Although the cost of using an RMT is higher. But the cost of changing a machine in a traditional system will be much higher than that of an RMS.

Oper Clust Seq.	1	8	4	7	10	9				3	2	5					6		11	
TAD Odd Used	1	1	1	1	1	1				2	1	4					1		1	
Machine	6	6	6	6	6	6				1	1	1					1		1	
Operation Seq	1	13	4	12	18	14	15	16	17	3	2	6	5	7	8	9	10	11	19	20
TAD	+Z	+X	+X	-X	-X	-a	-a	-a	-a	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	+Z	+Z
Tool Used	6	2	6	6	6	3	9	10	7	2	7	7	7	3	9	10	1	5	9	10

Figure 5.13: Representation of Optimal Reached Process Plan.

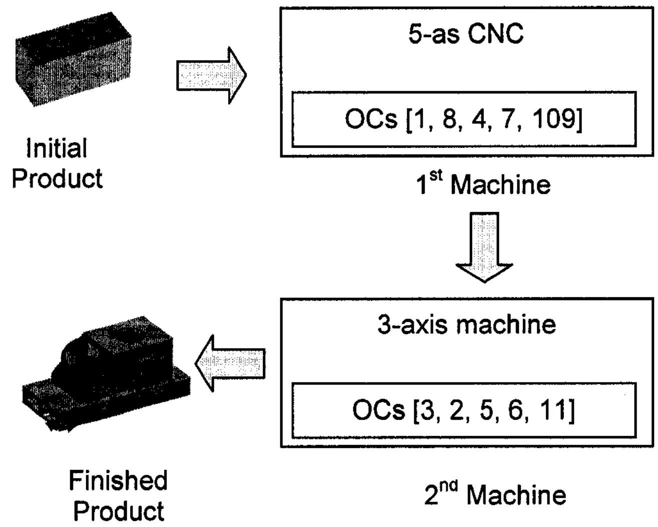


Figure 5.14: Manufacturing System Layout.

5.9. Validating Results by Comparing to Literature

The proposed GA optimization model was applied to test part ANC-101 using the objective function evaluation criteria of Li *et al.* [2002] and the same cost indices to validate the proposed model. In this scenario the logical constraints and tolerance constraints are changed to normal precedence constraints to that there will not exist any operation clustering.

5.9.1 Inputs

Inputs to the proposed model are as follows:

a) The objective function used was modified to match that used by Li *et al.* [2002]. The difference was what would be defined as a tool change and a setup change. Figures 5.15 and 5.16 illustrate what they define as a tool change and setup change respectively.

Conditions of machining two consecutive operations	Tool change
Same tool and same machine	no
Same tool and different machines	yes
Different tools and same machine	yes
Different tools and different machines	yes

Figure 5.15: Definition of Tool Change by Li *et al.* [2002].

Conditions of machining two consecutive operations	Set-up change
Same TAD and same machine	no
Same TAD and different machines	yes
Different TADs and same machine	yes
Different TADS and different machines	yes

Figure 5.16: Definition of Setup Change by Li *et al.* [2002].

b) The cost data input used by Li *et al.* [2002] is shown in Table 5.6.

c) Part data input for ANC-101 from Appendix A with a change of logical and tolerance constraints to normal precedence constraints. This will result in operations being treaded separately and not clustered into OCs

Table 5.6: Cost Information Used (Li *et al.* [2002]).

M	Type	Cost	ID	Type	Cost
M1	Drill Press	10	C1	Drill 1	7
M2	3-axis vertical Milling Machine	40	C2	Drill 2	5
M3	CNC 3-axis vertical Milling Machine	100	C3	Drill 3	3
			C4	Drill 4	8
M4	Boring Machine	60	C5	Tapping Tool	7
			C6	Mill 1	10
			C7	Mill 2	15
	MCCI	160	C8	Mill 3	30
	TDCCI	100	C9	Ream	15
	TCCI	20	C10	Boring Tool	20

5.9.2 GA Parameters Used

Table 5.7 provides the population size, the number of generations and the number of times each operator is applied in this work.

Table 5.7: Parameters used in Real Coded GAs.

Parameter	Value
Population size	200
Number of generations	75
Number of times of cross-over operator (arithmetic cross-over, simple cross-over and heuristic cross-over)	4 times each
Number of times of mutation operator (uniform mutation, non-uniform mutation and whole non-uniform mutation)	16, 8 8 times each respectively

5.9.3 Results and Discussion

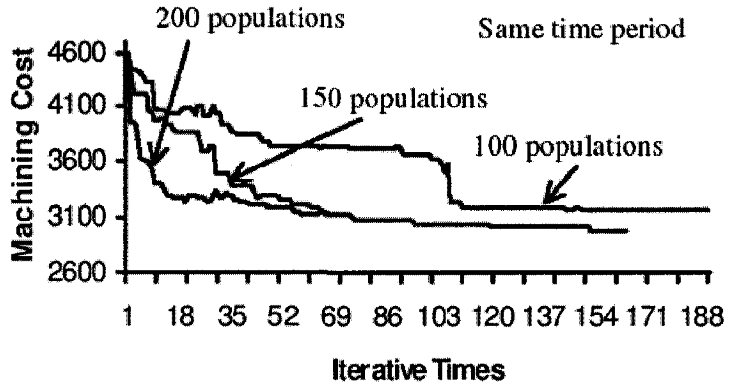


Figure 5.17: GA results Obtained by Li *et al.* [2002].

Figure 5.17 presents the results obtained by Li *et al.* when solving the same problem using a discrete GA algorithm showing the minimum cost value reached was around 3000 using a population size of 200 and 180 generations. The proposed continues domain model was used after relaxing the constraints so that the same exact problem is being solved using the continuous domain model. The GA parameters used are shown in Table 5.7. The same population size of 200 was used but the number of generations was reduced to more than half. Figure

5.18 shows the output GA convergence curves. The optimal value reached by the proposed constraint satisfaction continuous domain model is 2820 which is less than the optimal value reached by Li *et al.* when using their GA model. It should be noted that the value was reached in only 75 generations which indicates that the value was reached is less than half the number of objective function evaluations they used. These results validate the proposed approach and show that the proposed GA model has a higher and more efficient convergence rate.

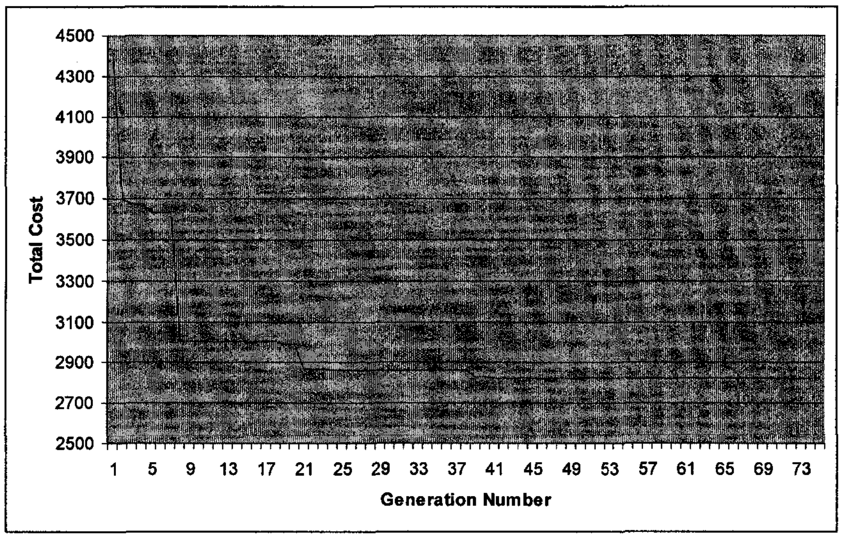


Figure 5.18: Convergence Curves Using a Population Size of 200.

In summary of the output he process plan with the least cost reached has a total of 2820 cost units. Figure 5.19 shows the corresponding optimal process plan representation. Table 5.8 shows a comparison between the two methods.

Table 5.8: : Comparison Between Li *et al.* Solution and the Proposed Continuous GA Model.

	Li *et al.* Solution	Proposed Continuous GA model
Minimum Value Reached (cost units)	3000	2820
Population Size	200	200
Number of Generations	180	75
Variables	Integer	Continuous

The number of tool (cutter) changes and TAD changes is counted for operations assigned to the same machine and not between two different machines. Therefore, the output shows the number of Tool changes is equal to 12 and the number of TAD change is equal to 10. Cost break down is as follows:

Machine Usage Cost (MUC)	=	1160
Tool Usage Cost (TUC)	=	260
Machine Change Cost (MCC)	=	160
Tool Change Cost (TCC)	=	240
Setup Change Cost (SCC)	=	1000
Total Cost	=	2820

Operation Seq	1	13	6	12	5	4	18	14	15	16	2	7	8	3	17	19	9	10	11	20
Machine Used	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	3	3
Tool Used	7	7	7	7	7	7	7	7	3	9	6	3	9	2	10	9	10	1	5	10
TAD Used	+Z	+X	-Z	-Z	-Z	-Z	-X	-a	-a	-a	-Z	-Z	-Z	-Z	-a	+Z	-Z	-Z	-Z	+Z

Figure 5.19: Representation of Optimal Reached Process Plan.

Figure 5.20 shows a diagram for the manufacturing system, showing the machine in sequence and the operations assigned to each machine.

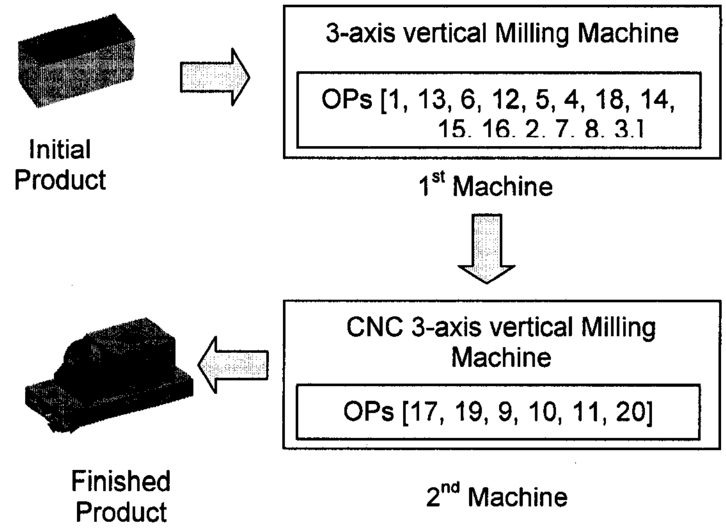


Figure 5.20: Manufacturing System Layout.

5.10. Summary and Conclusions

A new continuous process planning based on choosing the following parameters: operation selection (Machine, Machine Configuration, Tool and TAD selection) and operation sequencing. All the parameters were considered simultaneously in the optimization model in order to achieve the lowest cost.

A novel procedure was developed and utilized to ensure the generation of feasible process plans. It is based on mapping of the decision variables from their original discrete domain into a continuous domain of variables, which not only guarantees the generation of feasible process plans but also addresses the problem of having variable domain size. In addition, the proposed method produces solution strings that are easy to manipulate using different types of operators, such as crossovers or mutations, without violating the constraints or changing the size of the solution string as in traditional methods. Also the proposed method guarantees that operations that have related tolerance or logical constraints are clustered together and manufactured on the same machine. A new process plan representation was developed accordingly to represent both the OC and OP strings.

The ability to guarantee that certain operations will be clustered together on the same machine is a powerful characteristic of the proposed approach because in practice, operations that have tight tolerances should be carried out in the same setup to reduce the cost of re-setting and re-fixturing.

A toolbox was developed using MATLAB® software for implementing the proposed optimization model. A case study was presented to demonstrate the use of the developed model and the constraint satisfaction procedure. Good results were obtained compared to literature. Test on some of the GA parameters was also demonstrated. The new proposed approach was also validated by solving the same problem used in literature and it reached a better optimal solution in less than half the number of objective function evaluations. The computation time required was on average 1 min/run on a Pentium 4 2.6 GHz PC with 512 MB memory. This is a reasonable time considering the large solution space containing 64 variables with over 860 constraints. The algorithm was also tested for a part having 24 operations and the results were obtained in under 1.5 min/run. As the number of operation increase, the computational time will increase at a higher rate because the number of variables increases. Also to obtain a good solution with larger number of variables, then the numbers of generations and population size have to increase, which results in an increase in the number of objective function evaluations and thus increasing the computational time. The results also showed that process planning for an RMS will cost less depending on the different cost indices. An Advantage of using cost indices is that if a parameter is not of interest, for example the TAD, then the cost index for the TAD could be set to zero to study the effect of other parameters on the process plan.

The tool is flexible in the sense that the tolerance and logical constraints can be relaxed to produce traditional process plan with no pre-assigned OCs while taking advantage of the continuous domain method. Finally, it is important to point out that the new approach is applicable to any manufacturing system such as job shop, FMS or RMS depending on the type of available machines, for example, if the machines provided in the machine database exists are reconfigurable machine tools then the algorithm is solved taking machine configuration into account, even if some of the machines are RMT and the rest are fixed structure machines which is a typical scenario in RMS. This method could serve as a tool in aiding the machine assignment/selection activities.

CHAPTER SIX

RECONFIGURABLE PROCESS PLANNING

In Reconfigurable Manufacturing Systems (RMS) the manufacturing environment is dynamic. It requires computer-aided process planning (CAPP) systems that make use of the different capabilities of RMS as a result of both hard/physical and soft/logical reconfiguration [ElMaraghy 2002 and 2005] in response to changes in product requirements. The concept of Reconfigurable Process Planning, first introduced by ElMaraghy [2006], can be used to address this requirement. In RMS a process plan changes due to machine unavailability and/or part change. Rule based algorithms called "Process Planning Reconfiguration Rules" are introduced to aid in the decision making procedure. This chapter starts by providing a detailed description of the decision procedure used to reconfigure a Process Plan according to the developed process planning reconfiguration rules.

6.1. Process Planning in RMS

Figure 6.1 illustrates the Process Plan Reconfiguration. A change in the part or product being manufactured will in turn result in change of the current process plan. The change in process plan could result in new machining requirements resulting in machine configuration or machine addition/removal. If machine capabilities changed due to unavailability of one of its modules or the whole machine became unavailable due to breakdown then this could also result in a change in the process plan which is indicated by the bidirectional arrow. This introduces the concept of reconfigurable process planning.

As can be seen from the figure the Process Plan reconfiguration is initiated by either part change or machine change. The following sections will illustrate how the Process Plan reconfiguration is carried out in both cases in detail.

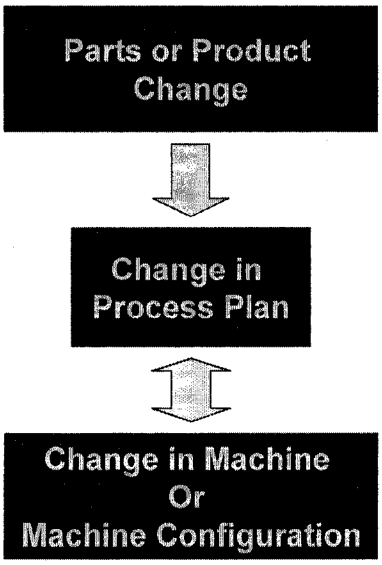


Figure 6.1: Process Plan Reconfiguration.

6.2. Process Plan Reconfiguration

CAPP systems for RMS should be able to generate cost-efficient alternate process plans because RMSs offer alternate process plans by using alternate capable machines or by even using the same machines in the process plan with a different configuration. The use of flexible or non-linear process plans that are capable of representing alternative processing sequences and manufacturing resources could be a solution but as mentioned earlier this has a drawback because of the length of time needed to generate a number of process plans and the added complexities for the reconfiguration function to take machine reconfiguration into account. Although non-linear generation of process is efficient in reacting quickly to disturbances, it is better in improving off-line process planning which would be of little value in RMS. In RMS a new dynamic (closed-loop) process planning reconfiguration approach is required that combines the advantage of dynamic process planning by enabling the generation of optimal process plans avoiding the generating of process plans that are infeasible in the current system state. The new approach should utilize parts of the already existing process plan and modify the affected parts of the process plan.

The proposed approach has to start with an initial generated feasible process plan that is to be changed due to part/machine change. Normally this initial process plan should be optimally generated utilizing the reconfiguration capabilities of the system and tailor the manufacturing system around the part. The approach used to generate this initial optimal process plan will be the same used in stage III.

A novel Computer Aided Process Planning Reconfiguration (CAPPR) approach that attempts to reconfigure an existing process plan to accommodate either of the following two scenarios is presented;

i. Modification to a current part or introduction of one to the system.

ii. A machine becomes unavailable for any reason such as breakdown.

The new semi-generative CAPPR approach utilizes a rule-based algorithm that aims at minimizing the required hard-type reconfiguration on both system and machine levels by performing less costly soft-type reconfiguration to the existing process plan. In the CAPPR approach, an existing process plan is described by a string that represents the following parameters: machine assignments, corresponding machine configurations, operation clusters sequence, operations sequence within the clusters, tool assignments and their corresponding tool approach directions (TAD) for different operations. This process plan string is reconfigured by adding or removing segments as necessary according to the required changes.

The approach (which is described in detail in sections 6.2.1 and 6.2.2), first, links the operation clusters precedence graph of the existing part to its process plan. In case of part modification, the portions of the operation clusters precedence graph that are common between the two parts are identified and the corresponding process plan segments are mapped to the reconfigured process plan of the new part (retrieval macro process planning). The remaining missing portions of the process plan are generated according to a predefined set of rules with the objective of minimizing the manufacturing system reconfiguration effort (generative macro process planning). The new set of rules was named process planning reconfiguration rules. Even if the new part is totally different from the current part, the CAPPR approach attempts to maximize the utilization of the current manufacturing system configuration and its existing machines and their corresponding configurations before suggesting physical system reconfiguration (i.e. machines reconfiguration, addition or removal). In case of a machine becoming unavailable, the CAPPR attempts to re-allocate the affected operation clusters to any of the existing machines before considering system reconfiguration. Precedence constraints are observed in all of the above-mentioned cases.

The following subsections provide a detailed procedure for the CAPPR Approach in the case of machine unavailability and part change. A MATLAB® toolbox has been developed for both approaches.

6.2.1 Machine Unavailability

Unavailability of a machine can be a result of many reasons such as failure in the machine itself or a component of a machine such as a spindle. Another reason for machine tool unavailability could be because it is halted for a scheduled maintenance. Inputs to the model are current system state, current process plan and the machine that is unavailable. The approach starts by searching for the part of the process plan that is affected. The affected portion of the process plan is checked for feasibility to be accommodated for in either the machines in prior or post stages of the unavailable machine first before reconfiguration then reconfigure if there is no solution. If there still remains no feasible solution even by reconfiguring the current machines, an alternative obtainable machine to be added instead of the broken machine is suggested. The new process plan after accommodating for the unavailable machine is the output of this approach. The pseudo-code illustrating the process plan reconfiguration rules used for the case of machine unavailability is as follows:

```
Procedure: Machine Unavailability
▪ Identify affected portion of process plan.
If   Affected portion of the process plan be accommodated in
     machines prior and/or post the affected machine without
     reconfiguration.
          ▪ Assign affected tasks to the machines prior
            and/or post affected machine.
Else If  There a solution by reconfiguring the current used
         machines?
          ▪ Reconfigure the minimum number of current
            machines.
Else
         Suggest an alternative obtainable machine to be
         added instead of the unavailable machine.
End If.
End Procedure.
```

Figure 6.2 shows the flowchart for the machine unavailability.

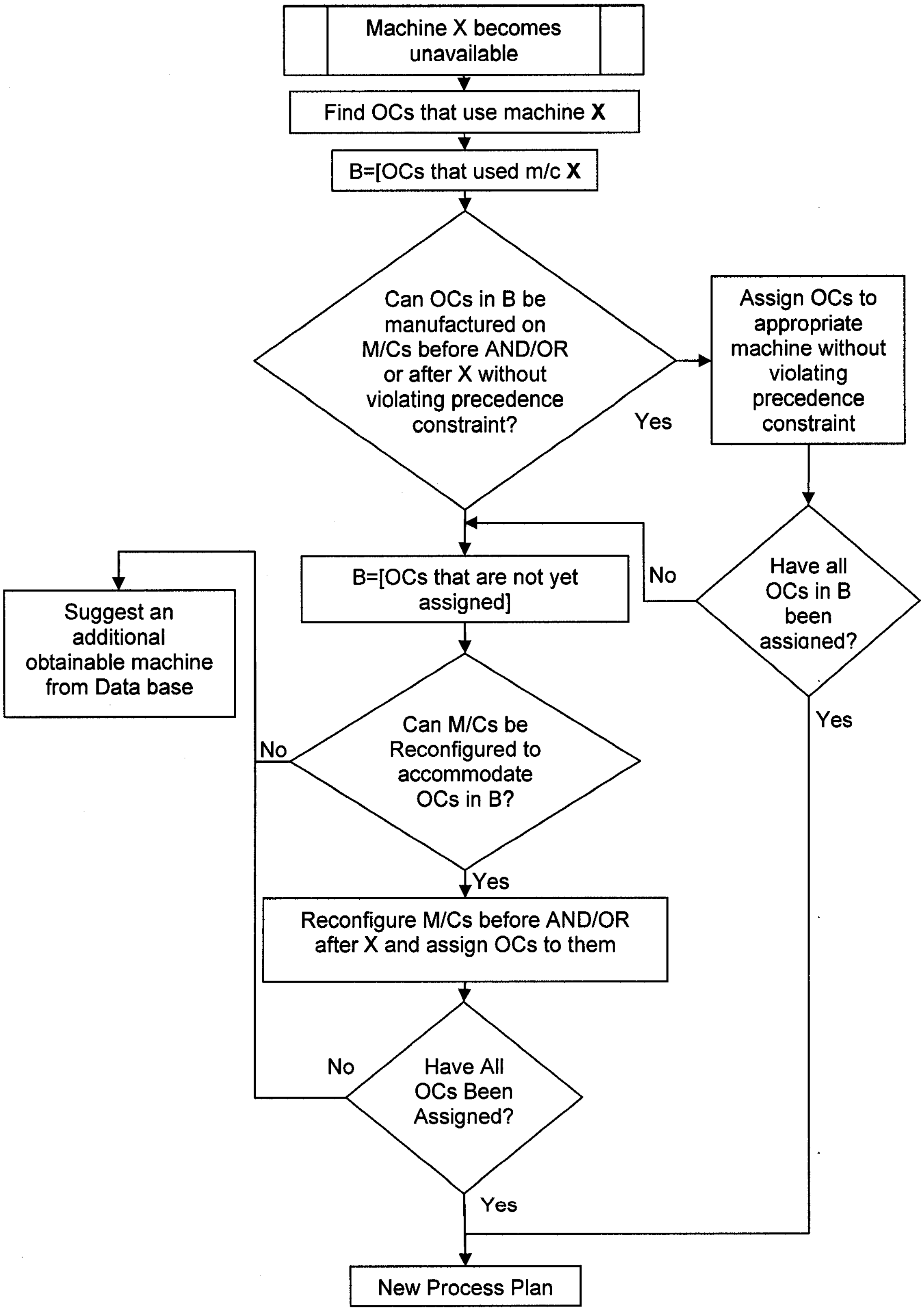


Figure 6.2: Flowchart for the Case of Machine Unavailability.

6.2.2 New Part Introduced to the Manufacturing System

Today's markets are dynamic and the products have a short life cycle and a typical scenario will be the introduction of new parts to the manufacturing system. For this reason CAPP systems dealing with reconfigurable manufacturing systems should encompass efficient techniques to quickly generate process plans according to the change in the part while at the same time make the most use of the current systems configurations and using the portions of the process plans that are similar to that of the current part.

The proposed approach can be used for a new part that has some similar features to the current part in system or even if a new part that has totally different features as that of the part being used currently.

Information entered to the model are the parts current system state including the current process plan (machine assignment, sequence of operations, current machine configuration, TAD used) and the database DB_{Cap} containing the Operation ID for the old part and the candidate machines capable of performing these operations (Output from stage II). Also taken as input are the operation precedence graphs for both the new and old part.

The procedure starts by identifying the new operations and operation clusters and accommodates them with the existing system if possible. Otherwise, reconfiguring of the minimum amount of machines as possible is carried out to accommodate for the new operation clusters. Alternative existing machines to be added to the system are suggested to accommodate for these clusters if there is no solution through accommodation. The output of the approach is the new process plan after accommodating for the new part.

Identification process of the new operation starts by identifying the similar and different operations. For example figure 6.3 shows the precedence graph of a part containing 5 operations (assumed to be old or current part being produced) and the precedence graph for a part containing 6 operations (assumed to be a new part being introduced). By checking the operation ID, operations having the same ID (O_1, O_2, O_3, O_5,) from the old and the new part are assigned the same machines. Operations in the old part and not in the new are removed but the procedure takes into account that the machine and its configuration that was used by O_4 in case it is required by the operations in the new part (O_6, O_7). O_6 and O_7 are inserted in a string called TempList that contains the operations that yet to be assigned. Before assigning O_6 and O_7 a check is carried out to validate the Operations sequence for the operations that are similar between the two parts. This check is carried out because the sequence of operations and OCs for the similar operations between the two parts could be different. Any OCs that are common between the two parts and its current machine sequence assignment violates the Operations precedence graph for the new part is removed from its current machine assignment and added to TempList. All the operations in TempList are then sorted in order of precedence so not to violate the operations precedence graph for the new part. For the example in Figure 6.3, TempList will contain O_6, and O_7 in sequence. These two operations will be allocated in sequence using the process planning reconfiguration rules shown in the procedure discussed below. The procedure is carried out for each operation in sequence. The first step is identifying the first location an operation can be located without violating the precedence constraints. Then a check is carried out to try to find a solution without reconfiguration for all the machine locations following this first location. If no solution is found then a similar procedure is carried out to try to find a solution through reconfiguration starting from the same first location. If after checking all the subsequent machines, there is no capable machine, then a machine that is capable is added to the first feasible location. This procedure is then repeated for the remaining Operations in sequence that are found in TempList.

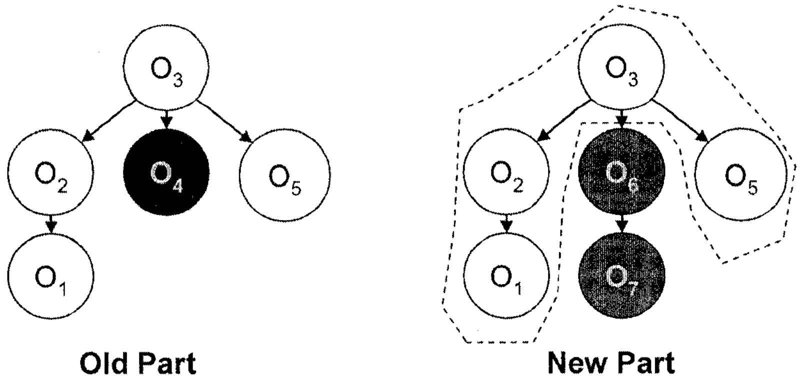


Figure 6.3: Precedence Graph of Old and New Part.

This approach combines both retrieval and generative process planning approaches. The retrieval approach is used for the portion of the process plan that is common for both old and new parts. The generative approach is used for the operations that are not found in the old part using the process planning reconfiguration rules. As mentioned earlier the generative approach tries to find a solution by minimizing the number of reconfigurations and if possible avoiding the addition of new machines. The pseudo-code used for the case of part change is as follows:

```
Procedure: Part Change
▪ Retrieve portion of process plan that is common for both
  current and new part.
▪ Generate remaining portion of the process plan for the
  new part with minimum changes (reconfigurations) in the
  existing system.
If   new operations can be carried out on existing system
     without reconfiguration
            ▪ Accommodate new operations in the existing
              system without reconfiguration.
Else Reconfigure the minimum amount of machines as possible
     to accommodate for the new operations.
End If.
If   There still remain unassigned operations.
```

- Suggest alternative machines to be added to the system to accommodate for these operations.

End If.

- Remove machine tools that were used by the old part and are no longer needed for the new part

End Procedure.

Figure 6.4 shows the flowchart for the case of part change.

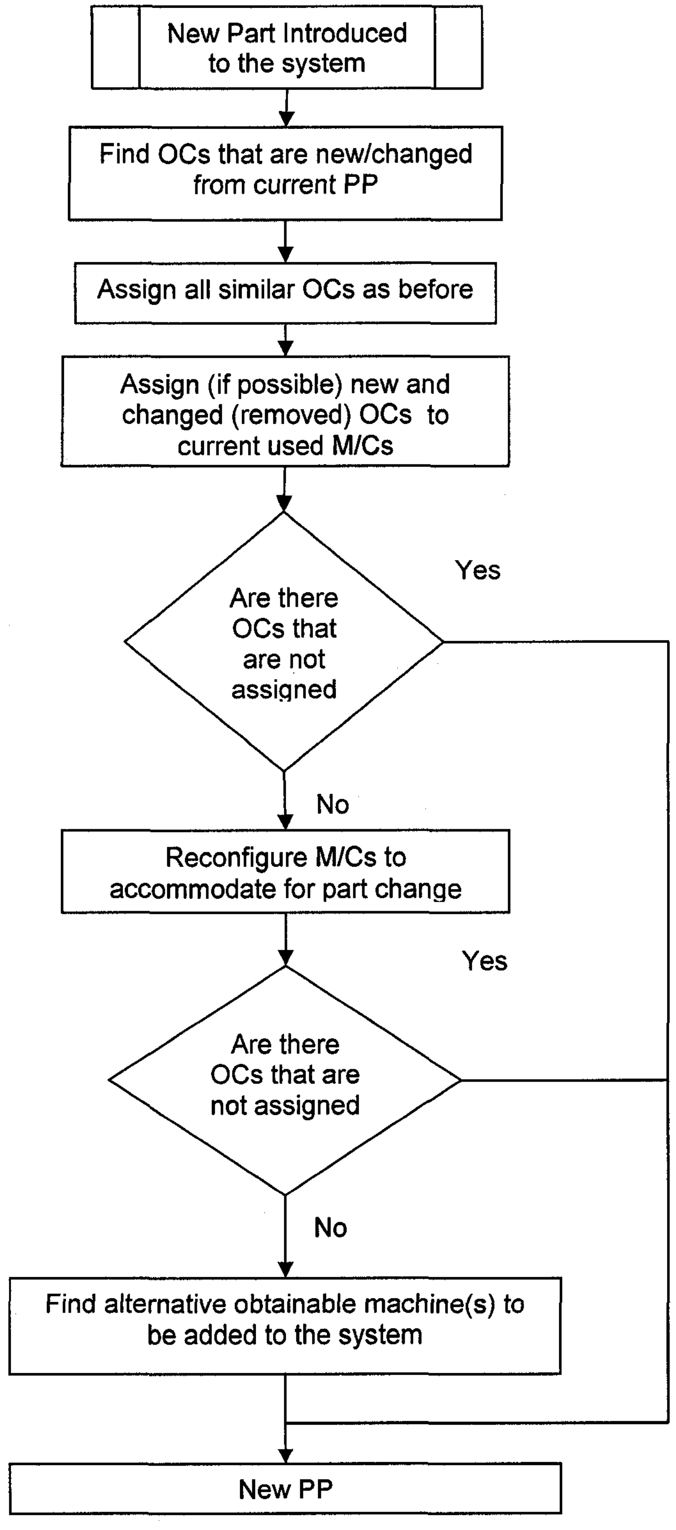


Figure 6.4: Flowchart for the Case of Part Change.

6.3. Illustrative example

This section provides a simple illustrative example to help elucidate the two procedures described in the previous section. Each example assumes a given process plan and shows the solution using the proposed procedure in the case of feasible or infeasible solution.

6.3.1 Machine Unavailability

Figure 6.5 shows an example of a machine breakdown and how the procedure proposed previously is implemented. The example is for a part containing 5 OCs. Figure 6.5 (a) shows that the five operations are assigned to 3 machines and also the configuration of each machine is shown. In this example an assumption is made that M_3 breaks down. The affected OCs are O_2 and O_4. The first step is to check if the machine prior to M_3 which is M_1 or the machine after M_3 which is M_2 are capable of performing O_2 and O_4. This procedure is carried out because it is the fastest solution to use the current system state rather than reconfigure or add an external machine. Also this solution will guarantee that the operations precedence graph is not violated. Figure 6.5 (b) represents the case where there is a solution for assigning the current affected operation clusters. The example represents a solution by assigning O_2 to machine M_1 and O_4 to M_2. If there is no solution because the current machine configurations are not capable of carrying the required operations, then the next step is to find a solution by reconfiguring the current used machines. Figure 6.5 (c) shows a case in which machine's M_2 configurations is changed from C_2 to C_3. After the reconfiguration O_2 and O_4 are assigned to machine M_2 with the new configuration C_3. If there is no configuration in the current machines that is capable of carrying out the required operations, then a new machine capable of carrying the required operations is suggested. Figure 6.5 (d) illustrates the example of addition of a new machine M_4 that is capable of performing operations O_2 and O_4.

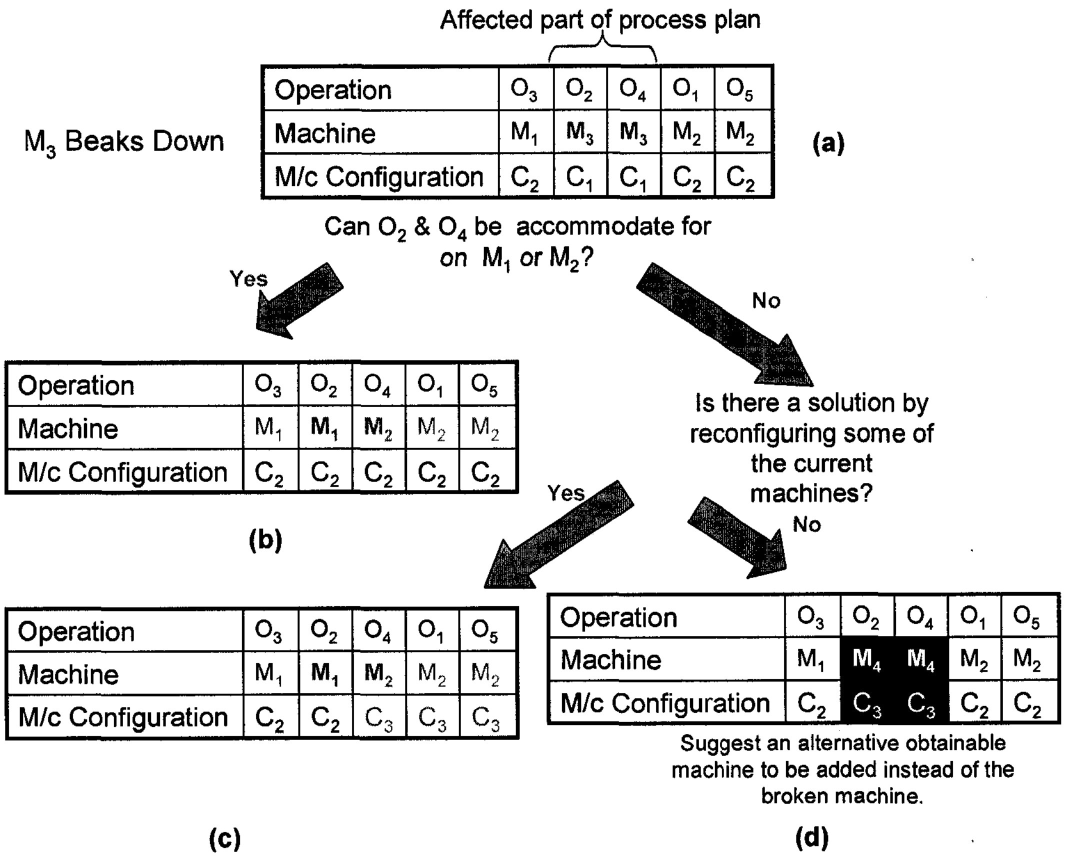


Figure 6.5: Illustrative Example of Machine Breakdown Procedure.

6.3.2 New Part Introduced to the Manufacturing System

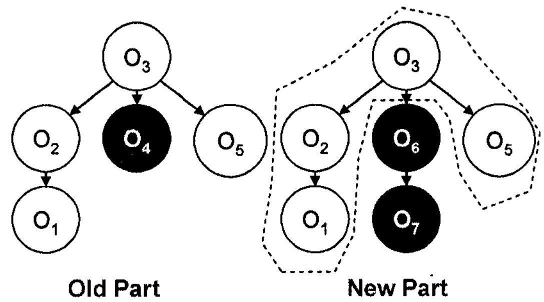
Figure 6.6 shows an example of a new part being introduced and how the procedure proposed previously is implemented. The current part contains 5 operations and the new part being introduced contains 6 operations. Figure 6.6 shows the precedence graphs of both parts and indicates the different operations between the new and the current part. Figure 6.6 (a) shows that current process plan before the new part is introduced. The approach starts by identifying the common portion of the process plan between the two parts and removes the portion of the process plan that is found in the current part and not in the new part. The removed portion of the process plan is highlighted in figure 6.6 (b). The following step will be to try to accommodate for the operations found only in the new part (O_6 and O_7). The first check will be to try to accommodate for the two new operations with the current system state. It should be noted that M_4 using configuration C_3 is still present in the current system. Assuming there is a valid process plan for O_6 and O_7 without the need of reconfiguration, figure 6.6 (c) represents an example of a solution in which O_6 is assigned to M_3 using C_1 and O_7 to M_2 using C_2. M_4 is removed since none of the operations are assigned to it. The solution represented in figure 6.6 (d) assumes there is no solution without reconfiguration and the solution is by reconfiguring M_3 from C_3 to C_2 so that the machine is capable of performing operation O_6. A check must be made to insure that the machine after reconfiguration is still capable of performing the other operations assigned to it (i.e. O_2). O_7 is assigned to M_2 without the need of reconfiguration because it is capable of performing O_7 using the current configuration C_2. Also in this case M_4 is removed because it is not used. Figure 6.6 (e) shows the case for which there is no solution except through the addition of a new machine. The solution shown in the example shows a case were both machine addition and machine reconfiguration is required. O_6 is assigned to a new added machine M_5 and O_7 is assigned to M_3 after reconfiguring from C_3 to C_2.

One of the parameters that has to be taken into consideration when reconfiguring a process plan is avoiding the generation of infeasible process plans by violating the precedence constraints.

The process plan reconfiguration can be seen through the addition and removal or change in the process plan from one part to another and how the similar portions of the process plan are mapped form the old part to the new part and how the new operations are accommodated for in the new part.

Current process Plan

Operation	O_3	O_2	O_4	O_1	O_5
Machine	M_1	M_3	M_4	M_2	M_2
M/c Configuration	C_2	C_1	C_3	C_2	C_2



Retrieve common PP (a)

Operation	O_3	O_2		O_1	O_5
Machine	M_1	M_3		M_2	M_2
M/c Configuration	C_2	C_1		C_2	C_2

O_6 & O_7 not yet assigned

Removed PP portion (PP reconfiguration)

(b) Can new operations be assigned without reconfiguration of M/C?

Yes

Added PP portion (PP reconfiguration)

Operation	O_3	O_2	O_6	O_1	O_5	O_7
Machine	M_1	M_3	M_3	M_2	M_2	M_2
M/c Configuration	C_2	C_1	C_1	C_2	C_2	C_2

O_6 assigned to M_3
O_7 assigned to M_2
M_4 is removed (c)

No

Is there a solution by reconfiguring some of the current machines?

Yes

Operation	O_3	O_2	O_6	O_7	O_1	O_5
Machine	M_1	M_3	M_3	M_2	M_2	M_2
M/c Configuration	C_2	C_2	C_2	C_2	C_2	C_2

(d) M_3 reconfigures for O_6
O_7 assigned to M_2
M_4 is removed

No

Add a new machine to the system

Operation	O_3	O_6	O_2	O_7	O_1	O_5
Machine	M_1	M_5	M_3	M_3	M_2	M_2
M/c Configuration	C_2	C_3	C_2	C_2	C_2	C_2

M_5 assigned for O_6
M_3 reconfigures fr O_7
M_4 is removed (e)

Figure 6.6: Illustrative Example of New Part Being Introduced Procedure.

6.4. Results and Discussion

In this section the above procedures will be applied to the case study part ANC-090 and ANC-101 and the output from the developed MATLAB® toolbox will be presented. Appendix A shows the details for both parts.

6.4.1 Machine Unavailability

The machine unavailability procedure is applied to the ANC-101 test part. The input to this stage is the systems current state, which will be the optimum process plan obtained from stage III. Figure 6.7 shows the process plan of the current system before a machine becomes unavailable. The cost for this process plan is 2634 cost units. Two cases will be illustrated; solution with and without reconfiguration.

Oper Clust Seq.	1	8	7	10	2	4	5					3	11		6		9			
TAD Odd Used	1	1	2	1	1	2	4					1	1		1		1			
Machine	1	1	1	1	1	1	1					1	1		2		1			
Configuration	1	3	2	2	2	2	2					4	4		1		2			
Operation Seq	1	13	12	18	2	4	6	5	7	8	9	3	19	20	10	11	14	15	16	17
TAD	+Z	+X	-Z	-X	-Z	-Z	-Z	-Z	-Z	-Z	-Z	+Z	+Z	+Z	-Z	-Z	-a	-a	-a	-a
Tool Used	6	2	6	6	6	6	8	8	3	9	10	2	9	10	1	5	3	9	10	7

Figure 6.7: Current Process Plan

6.4.1.2 Case I: Solution without Machine Reconfiguration

Assuming machine M_2 using C_1 that has OC_6 assigned to it becomes unavailable, using the developed toolbox, there is a solution without reconfiguration. The output figure 6.8 shows the current machine configurations indicating the affected machine. All the possible solutions and the process plan cost of each solution are also illustrated in the figure. The change of machine assignments is highlighted.

There are two possible solutions. The first solution is by assigning OC_6 to be carried out on machine (M_1 using C_2) as OC_9. The second solution is by assigning OC_6 to be carried out on machine (M_1 using C_4) as OC_3 and OC_{11}. The second solution is chosen because it has less process planning cost.

The output is stored in a database file to keep a record of the new current system's state so that it could be used again in case there is another unexpected machine unavailability or a new part is introduced to the system

	Oper Clust Seq.	1	8	7	10	2	4	5	3	11	6	9	
Current State	Machine	1	1	1	1	1	1	1	1	1		1	
	Configuration	1	3	2	2	2	2	2	4	4		2	
First Solution	Machine	1	1	1	1	1	1	1	1	1	1	1	2666
	Configuration	1	3	2	2	2	2	2	4	4	2	2	
Second Solution	Machine	1	1	1	1	1	1	1	1	1	1	1	2696
	Configuration	1	3	2	2	2	2	2	4	4	4	2	

Figure 6.8: Solution without Reconfiguration.

6.4.1.3 Case II: Solution with Reconfiguration

In this case, referring back to figure 6.7, M_1 using C_3 is assumed to be unavailable. OC_8 is the effected OC in this case. Figure 6.9 shows the output solutions for this case.

	Oper Clust Seq.	1	8	7	10	2	4	5	3	11	6	9	
Current State	Machine	1		1	1	1	1	1	1	1	2	1	
	Configuration	1		2	2	2	2	2	4	4	1	2	
First Solution	Machine	1	1	1	1	1	1	1	1	1	2	1	2618
	Configuration	3	3	2	2	2	2	2	4	4	1	2	
Second Solution	Machine	1	1	1	1	1	1	1	1	1	2	1	2678
	Configuration	4	4	2	2	2	2	2	4	4	1	2	

Figure 6.9: Solution Through Reconfiguration.

The first solution is carried out by assigning both OC_1 and OC_8 to M_1 after reconfiguring from C_1 to C_3. The second solution is carried out by reconfiguring the same machine to C_4. From the figure, the first solution is selected because it has the least cost.

6.4.2 New Part Introduced to the Manufacturing System

A case study was carried out assuming that the part being manufacturing in the current system is ANC-090 and a new part ANC-101 is being introduced. Figures 6.10 and 6.11 illustrates the difference in the operations and OC precedence graphs between the two parts respectively.

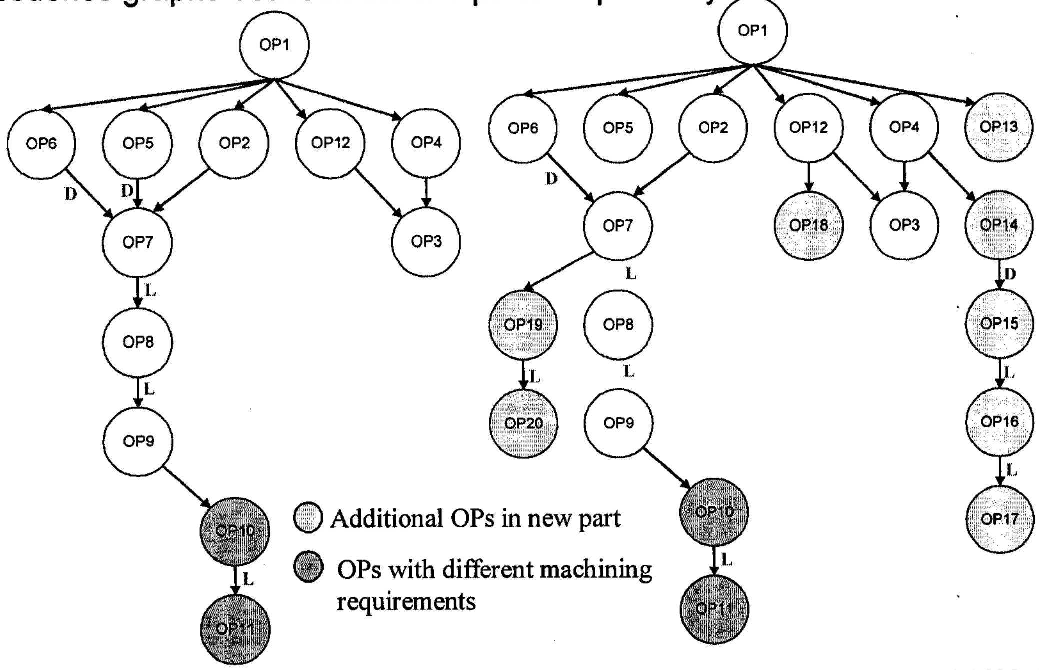


Figure 6.10: Difference in Operations Precedence Graph between ANC-090 and ANC-101.

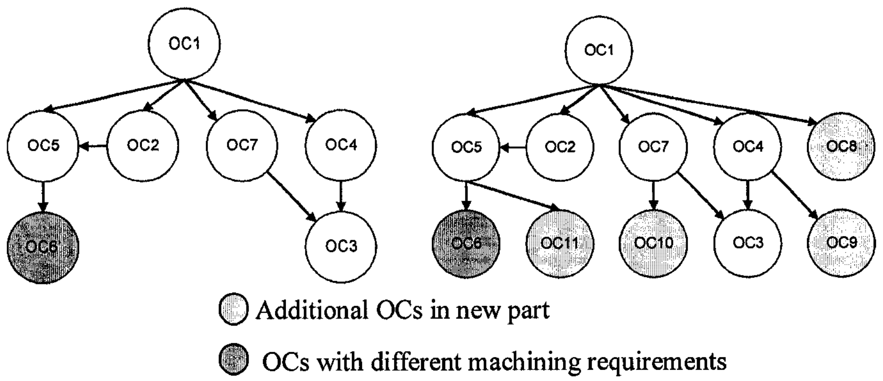


Figure 6.11: Difference in OC Precedence Graph between ANC-090 and ANC-101.

There is an additional 8 operations and 4 OCs between the ANC-101 and ANC-090 and. Figure 6.12 illustrates a sample of the current process plan for ANC-090 and Figure 6.13 shows output obtained from the developed tool box.

The current system state (figure 6.12) uses three machines. OC_6 is assigned to M_2 using C_1, OC_1 is assigned to M_1 using C_1 and all other OCs are assigned to a similar machine as that used by OC_1. The reason why OC_1 is not assigned a separate machine because it needs machining from the –Z direction.

Oper Clust Seq.	1	4	7	3	2	5					6	
TAD Odd Used	1	2	2	1	1	4					1	
Machine	1	1	1	1	1	1					2	
Configuration	1	1	1	1	1	1					1	
Operation Seq	1	4	12	3	2	6	5	7	8	9	10	11
TAD	+Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z
Tool Used	6	6	6	2	6	8	8	3	9	10	1	5

Different Ocs

Figure 6.12: Current Process Plan for ANC-090.

Oper Clust Seq.	1	8	4	9				7	3	2	5					10	11		6'	
TAD Odd Used	1	1	2	1				2	1	1	4					1	1		1	
Machine	1	1	1	1				1	1	1	1					1	1		2	
Configuration	1	3	[illegible]														[illegible]		1	
Operation Seq	1	13	4	14	15	16	17	12	3	2	6	5	7	8	9	18	19	20	10	11
TAD	+Z	+X	-Z	-a	-a	-a	-a	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-X	+Z	+Z	-Z	-Z
Tool Used	6	2	6	3	9	10	7	6	2	6	8	8	3	9	10	6	9	10	1	5

Different Ocs
Additional Ocs
Machine Addition
Machine Reconfiguration

Figure 6.13: New Output process Plan ANC-101.

The output (figure 6.13) shows that for the manufacturing system to be capable of manufacturing the new ANC-101 the following actions have to be carried out:

- OC_1 remains assigned to the same machine without reconfiguration.
- Add an M_1 machine using C_3 to accommodate for OC_8.
- Reconfigure M_1 using C_1 to C_2 to accommodate for OC_9.
- OC_4, OC_9, OC_7, OC_3, $OC_2$and OC_5 are assigned in sequence to the new reconfigured machine.

- Add an M_1 machine using C_2 to accommodate for OC_{11} because it needs machining in the +Z direction.
- Assign OC_6' to same machine as OC_6 on M_2 using C_1.

6.5. Summary

A novel Computer Aided Process Planning Reconfiguration (CAPPR) approach that attempts to reconfigure an existing process plan to accommodate for the changes in the manufacturing system by either machine and/or part change. The semi-generative approach utilizes a developed rule-based algorithm called reconfigurable process planning rules that aims at minimizing the required hard-type reconfiguration on both system and machine levels by performing less costly soft-type reconfiguration to the existing process plan. Existing process plan is described by a string. This process plan string is reconfigured by adding or removing segments as necessary according to the required changes.

The new CAPPR approach utilizes the current system state to the best possible. Even if the new part is totally different from the current part then the CAPP system makes use of the current system status in terms of machines used and machines' configuration to generate the process plan. The approach searches for a solution without machines' reconfiguration because that is least costly solution. If no solution is available then a solution is found by reconfiguring the minimum number of machines being currently used. If there still remains no solution then an expensive solution will have to be suggested, which is the addition of a new machine to the manufacturing system. The approach has been tested using the develop toolbox. In all cases, the reconfigured process plans are fully developed.

The developed approach can serve as an important tool in the decision making activity when there is subsequent changes in products scope or there is unexpected machine unavailability within a RMS environment.

CHAPTER SEVEN

OVERALL RMS PROCESS PLANNING APPROACH

In this chapter the overall RMS process planning procedure is carried out for a second part (ANC-300). Figure 7.1 shows the ANC-300 part containing 20 features. Table 7.1 shows operations data and the operations precedence graph is shown in Figure 7.2. In this chapter the results of applying the procedures and approaches from stages I to III will be presented. Following will be the results after applying the CAPPR approach containing the reconfigurable process planning rules.

7.1. Operation Clustering

Table 7.1 provides the different features' description, the operations required to produce these features, their IDs and the tool approach direction (TAD) for each operation for the ANC-300.

After applying the clustering algorithm, the 24 operations were grouped into 16 different clusters as shown in Figure 7.3 and Table 7.2. This along with Table 7.1 will be used as input to the second stage of generating the required machine structure.

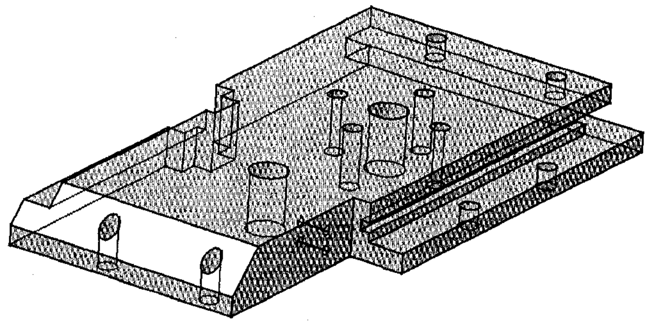


Figure 7.1: ANC-300 Part Containing 20 Features [Zhang *et al.* 1997].

Table 7.1: Operations Data for Part ANC-300 [Zhang *et al.* 1997].

Feature	Description (Old Feature Number)	Operation	Op. ID	TAD candidates	Tool candidates
F1	Planar surface	Milling	Op1	-Z	T5, T6
F2	Slot (F5)	Milling	Op2	+Y	T5, T6
F3	Slot (F10)	Milling	Op3	+X	T5, T6
F4	Step (F4)	Milling	Op4	-Y, -Z	T5, T6
F5	Slot (F6)	Milling	Op5	+Y	T5, T6
F6	Step (F17)	Milling	Op6	-Y	T5, T6
F7	Through Hall (F7)	Drilling	Op7	+Z, -Z	T2
		Reaming	Op8	+Z, -Z	T3
		Boring	Op9	+Z, -Z	T4
F8	Through Hall (F9)	Drilling	Op10	+Z, -Z	T2
		Reaming	Op11	+Z, -Z	T3
		Boring	Op12	+Z, -Z	T4
F9	Pattern Holes (F8)	Drilling	Op13	_Z	T1
F10	Step (F14)	Milling	Op14	-Y, -Z	T5, T6
F11	Blind Hole (F18)	Drilling	Op15	-Y	T1
F12	Blind Hole (F19)	Drilling	Op16	-Y	T1
F13	Through Hole (F1)	Drilling	Op17	+Z, -Z	T1
F14	Through Hole (F2)	Drilling	Op18	+Z, -Z	T1
F15	Chamfer (F3)	Milling	Op19	+Z, -Z	T7
F16	Through Hole (F15)	Drilling	Op20	+Z, -Z	T1
F17	Through Hole (F16)	Drilling	Op21	+Z, -Z	T1
F18	Slot (F13)	Milling	Op22	-Y, -Z	T5, T8
F19	Through Hole (F11)	Drilling	Op23	-Z	T1
F20	Through Hole (F12)	Drilling	Op24	-Z	T1

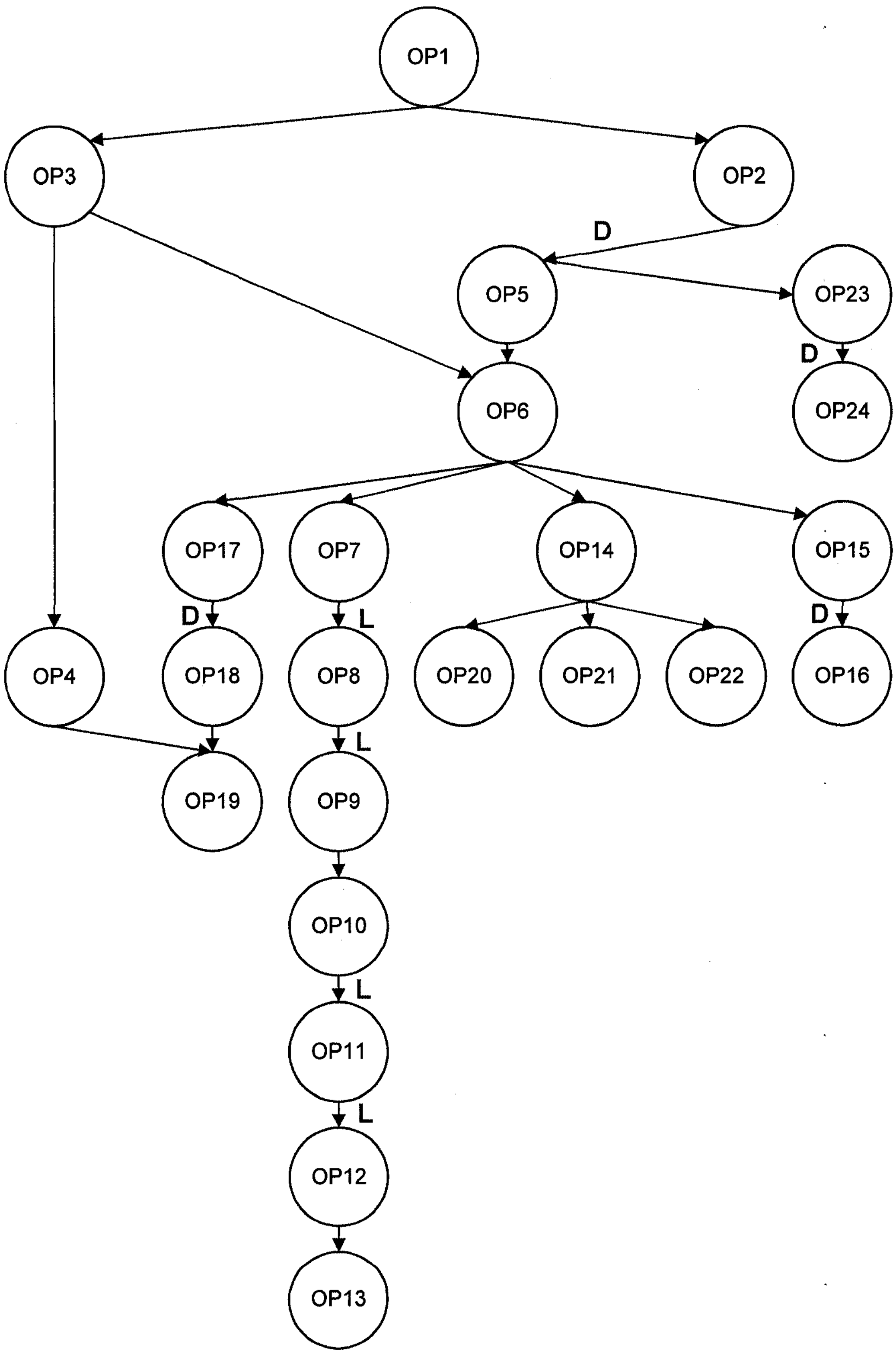


Figure 7.2: Operations Precedence Graph for Part ANC-300.

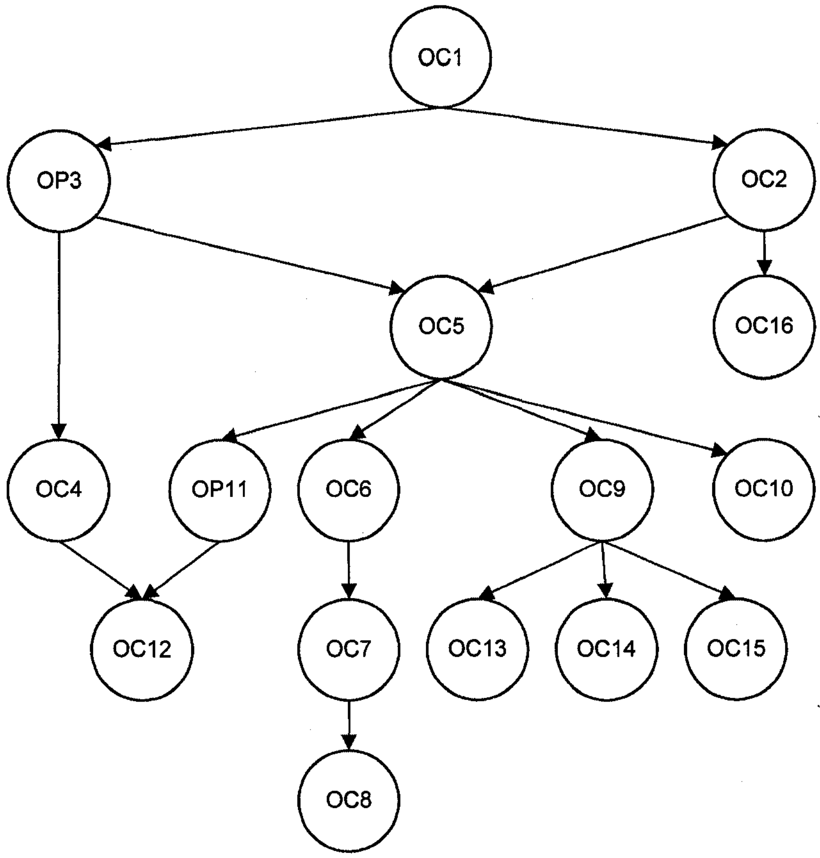


Figure 7.3: Operation Clusters Precedence Graph for part ANC-300.

Table 7.2: Operation Clusters for Part ANC-300.

Operation Cluster	Operations
OC1	[OP1]
OC2	[OP2, OP5]
OC3	[OP3]
OC4	[OP4]
OC5	[OP6]
OC6	[OP7, OP8, OP9]]
OC7	[OP10, OP11, OP12]
OC8	[OP13]
OC9	[OP14]
OC10	[OP15, OP16]
OC11	[OP17, OP18]
OC12	[OP19]
OC13	[OP20]
OC14	[OP21]
OC15	[OP22]
OC16	[OP23, OP24]

7.2. Generating Machine Tool Structure

Figure 7.4 shows the output of the algorithm showing the minimum axes of rotation (capability) needed for each cluster including the different combinations.

OC	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
1	0 0 0 0 0 0							
2	0 0 0 90 0 0							
3	0 90 0 0 0 0							
4	90 0 0 0 0 0	0 0 0 0 0 0						
5	90 0 0 0 0 0							
6	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	0 0 0 0 0 0
7	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	0 0 0 0 0 0
8	0 0 0 0 0 0							
9	90 0 0 0 0 0	0 0 0 0 0 0						
10	90 0 0 0 0 0							
11	180 0 0 0 0 0	180 0 0 0 0 0	180 0 0 0 0 0	0 0 0 0 0 0				
12	180 0 0 0 0 0	0 0 0 0 0 0						
13	180 0 0 0 0 0	0 0 0 0 0 0						
14	180 0 0 0 0 0	0 0 0 0 0 0						
15	90 0 0 0 0 0	0 0 0 0 0 0						
16	0 0 0 0 0 0							

Figure 7.4: Minimum Axes of Rotation Required for Every Operation Cluster and their Different Cases for ANC-300.

7.3. Optimum Process Plan

The data obtained from stages I and II are used as inputs to the optimization model proposed for stage III to obtain the optimal process plan for ANC-300. The cost data used are shown in Table 5.1. The GA parameters used are shown in Table 7.2:

Table 7.3: Parameters used in Real Coded GAs.

Parameter	Value
Population size	400
Number of generations	150
Number of times of cross-over operator (arithmetic cross-over, simple cross-over and heuristic cross-over)	18 times each
Number of times of mutation operator (uniform mutation, non-uniform mutation and whole non-uniform mutation)	12 times each

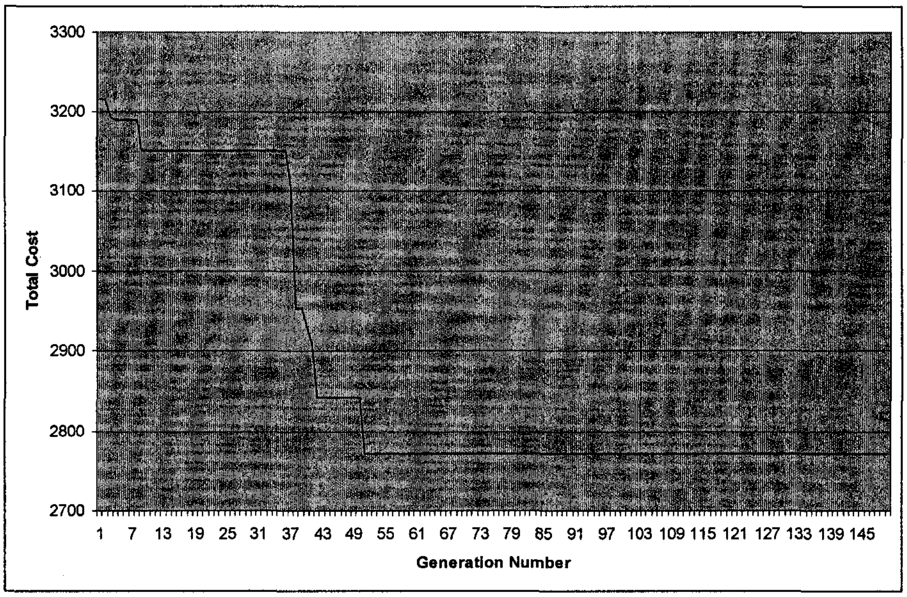


Figure 7.5: Convergence Curves Using a Population Size of 150.

Figure 7.5 demonstrates the GA convergence curves. The process plan with the least cost reached has a total of 2772 cost units. Figure 7.6 shows the corresponding optimal process plan representation. The 24 operations are grouped into 16 clusters; therefore, there are 75 variables. The encoded representation shown in Figure 5.3 reduced the original representation (figure 5.2) from 120 variables to 104 variables (i.e., problem was reduced by 16 variables).

The output shows that two machines are used. Machine ID 5 (Machine base number 2 using configuration 4) is used for the first nine OCs (OC1, OC2, OC16, OC3, OC4, OC5, OC9 & OC15) the following seven OCs are then assigned to machine ID 3 (Machine base number 2 using configuration 2). The number of Tool changes is equal to 12 and the number of TAD change is equal to 5. Cost break down is as follows:

Machine Usage Cost (MUC)	=	1706
Tool Usage Cost (TUC)	=	166
Machine Change Cost (MCC)	=	160
Tool Change Cost (TCC)	=	240
Setup Change Cost (SCC)	=	500
Total Cost	=	2772

Oper Clust Seq.	1	2		16		3	4	5	9	15	10		13	11		6			14	12	7			8
TAD Odd Used	1	1		1		1	1	1	1	1	1		1	1		1			1	1	4			1
Machine ID	5	5		5		5	5	5	5	5	5		3	3		3			3	3	3			3
Operation Seq	1	2	5	23	24	6	4	6	14	22	15	16	20	17	18	7	8	9	21	19	10	11	12	13
TAD	-Z	+Y	+Y	-Z	-Z	+X	-Y	-Y	-Y	-Y	-Y	-Y	+Z	+Z	+Z	+Z	+Z	+Z	+Z	+Z	+Z	-Z	-Z	-Z
Tool Used	5	5	5	1	1	5	5	5	5	5	1	1	1	1	1	2	3	4	1	7	2	3	4	1

Figure 7.6: Representation of Optimal Reached Process Plan.

7.4. Reconfigurable Process Planning

To apply the CAPPR approach, the following scenario is assumed:

- The current part being produced is ANC-090 and the current process plan for part ANC-090 is shown in Figure 7.7
- The machines that are available to choose from or to reconfigure to are those in the database (Table 5.2)
- A new ANC-300 part is being introduced to replace ANC-090

Oper Clust Seq.	1	4	7	3	2	5					6	
TAD Odd Used	1	2	2	1	1	4					1	
Machine	1	1	1	1	1	1					2	
Configuration	1	1	1	1	1	1					1	
Operation Seq	1	4	12	3	2	6	5	7	8	9	10	11
TAD	+Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z	-Z
Tool Used	6	6	6	2	6	8	8	3	9	10	1	5

Different Ocs

Figure 7.7: Current Process Plan for ANC-090.

The above scenario was applied to the RMS Process Planning Approach and the obtained results where as follows.

1. Similar OCs between ANC-300 and ANC-090 is OC1 only (Figure 7.8).

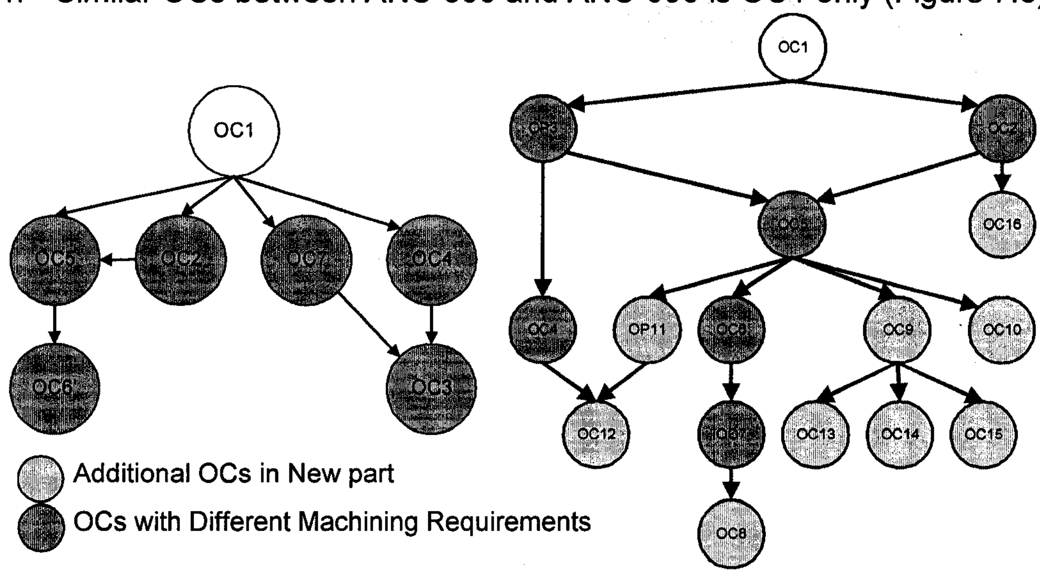


Figure 7.8: Additional and Different OCs between ANC-090 (left) and ANC-300 (right).

2. The output of the developed algorithm is stated below step by step. The process plan in the following section will be only represented by the Operation Cluster Sequence, the Machine being used and the configuration number so that the steps are easier illustrated (Figure 7.9).

Oper Clust Seq.	1	4	7	3	2	5	6
Machine	1	1	1	1	1	1	2
Configuration	1	1	1	1	1	1	1

Figure 7.9: Original Process Plan.

Oper Clust Seq.	1	0	0
Machine	1	1	2
Configuration	1	1	1

Figure 7.10: Remove Different OCs and keep Machines in Place

Before starting to reconfigure the Original Process Plan (Figure 7.9), all the similar OCs have to be Identified. The Only Similar OC is OC number 1, so it remains assigned to the same machine with the same configuration and the other 6 OCs from the original plan are removed. Machine 1 using configuration 2 has all the operation assigned to it remains in place (Figure 7.10) just in case it is needed by new OCs from the ANC-300 part.

The remaining OCs for Part ANC-300 that are not similar or violate a precedence constraint are sorted according to their precedence relationship. The remaining OCs are: OC2, OC 3, OC4, OC5, OC6, OC7, OC8, OC9, OC10, OC11, OC12, OC13, OC14, OC15, and OC16.

The reconfiguration Process starts in sequence to assign the remaining OCs, starting with OC2, then OC3, etc.

Step 1 (OC2): 1st location to Add OC without violating precedence constraints is after location 1. Reconfigure Machine 2 from configuration 1 to configuration 2. Allocate OC2 in location 3 (Figure 7.11).

	1	2	3	4
Oper Clust Seq.	1	0	0	2
Machine	1	1	2	2
Configuration	1	1	[redacted]	

Figure 7.11: Step 1-Machine Reconfiguration.

Step 2 (OC3): 1st location to Add OC without violating precedence constraints is after location 1. Reconfigure Machine 2 from configuration 2 to configuration 4. Allocate OC3 in location 3 (Figure 7.12).

	1	2	3	4	5
Oper Clust Seq.	1	0	0	3	2
Machine	1	1	2	2	2
Configuration	1	1			

Figure 7.12: Step 2-Machine Reconfiguration.

Step 3 (OC4): 1st location to Add OC without violating precedence constraints is after location 4. Add OC4 to Machine 2 using configuration 4 in location 5 (Figure 7.13).

	1	2	3	4	5	6
Oper Clust Seq.	1	0	0	3	4	2
Machine	1	1	2	2	2	2
Configuration	1	1	4	4	4	4

Figure 7.13: Step 3-Add OC to Existing Machine Without Reconfiguration.

Step 4 (OC5): 1st location to Add OC without violating precedence constraints is after location 6. Add OC6 to Machine 2 using configuration 4 in location 7 (Figure 7.14).

	1	2	3	4	5	6	7
Oper Clust Seq.	1	0	0	3	4	2	5
Machine	1	1	2	2	2	2	2
Configuration	1	1	4	4	4	4	4

Figure 7.14: Step 3-Add OC to Existing Machine Without Reconfiguration.

Step 5 (OC6): 1st location to Add OC without violating precedence constraints is after location 7. Add OC6 to Machine 2 using configuration 4 in location 8 (Figure 7.15).

	1	2	3	4	5	6	7	8
Oper Clust Seq.	1	0	0	3	4	2	5	6
Machine	1	1	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4

Figure 7.15: Step 4-Add OC to Existing Machine Without Reconfiguration.

Step 6 (OC7): 1st location to Add OC without violating precedence constraints is after location 8. Add OC7 to Machine 2 using configuration 4 in location 9 (Figure 7.16).

	1	2	3	4	5	6	7	8	9
Oper Clust Seq.	1	0	0	3	4	2	5	6	7
Machine	1	1	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4

Figure 7.16: Step 5-Add OC to Existing Machine Without Reconfiguration.

Step 7 (OC8): 1st location to Add OC without violating precedence constraints is after location 9. Add OC8 to Machine 2 using configuration 4 in location 10 (Figure 7.17).

	1	2	3	4	5	6	7	8	9	10
Oper Clust Seq.	1	0	0	3	4	2	5	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4

Figure 7.17: Step 6-Add OC to Existing Machine Without Reconfiguration.

Step 8 (OC9): 1st location to Add OC without violating precedence constraints is after location 7. Add OC8 to Machine 2 using configuration 4 in location 8 (Figure 7.18).

	1	2	3	4	5	6	7	8	9	10	11
Oper Clust Seq.	1	0	0	3	4	2	5	9	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4

Figure 7.18: Step 7-Add OC to Existing Machine Without Reconfiguration.

Step 9 (OC10): 1st location to Add OC without violating precedence constraints is after location 7. Add OC10 to Machine 2 using configuration 4 in location 8 (Figure 7.19).

	1	2	3	4	5	6	7	8	9	10	11	12
Oper Clust Seq.	1	0	0	3	4	2	5	10	9	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	4

Figure 7.19: Step 8-Add OC to Existing Machine Without Reconfiguration.

Step 10 (OC11): 1st location to Add OC without violating precedence constraints is after location 7. Add OC11 to Machine 2 using configuration 4 in location 8 (Figure 7.20).

	1	2	3	4	5	6	7	8	9	10	11	12	13
Oper Clust Seq.	1	0	0	3	4	2	5	11	10	9	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	4	4

Figure 7.20: Step 9-Add OC to Existing Machine Without Reconfiguration.

Step 11 (OC12): 1st location to Add OC without violating precedence constraints is after location 8. Add OC12 to Machine 2 using configuration 4 in location 9 (Figure 7.21).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Oper Clust Seq.	1	0	0	3	4	2	5	11	12	10	9	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	4	4	4

Figure 7.21: Step 10-Add OC to Existing Machine Without Reconfiguration.

Step 12 (OC13): 1st location to Add OC without violating precedence constraints is after location 11. Add OC13 to Machine 2 using configuration 4 in location 12 (Figure 7.22).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Oper Clust Seq.	1	0	0	3	4	2	5	11	12	10	9	13	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4

Figure 7.22: Step 11-Add OC to Existing Machine Without Reconfiguration.

Step 13 (OC14): 1st location to Add OC without violating precedence constraints is after location 11. Add OC14 to Machine 2 using configuration 4 in location 14 (Figure 7.23).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Oper Clust Seq.	1	0	0	3	4	2	5	11	12	10	9	[illegible]	13	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	[illegible]	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	[illegible]	4	4	4	4

Figure 7.23: Step 12-Add OC to Existing Machine Without Reconfiguration.

Step 14 (OC15): 1st location to Add OC without violating precedence constraints is after location 4. Add OC15 to Machine 2 using configuration 4 in location 5 (Figure 7.24).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Oper Clust Seq.	1	0	0	3	4	2	5	11	12	10	9	15	14	13	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	[illegible]	4	4	4	4	4

Figure 7.24: Step 13-Add OC to Existing Machine Without Reconfiguration.

Step 15 (OC16): 1st location to Add OC without violating precedence constraints is after location 6. Add OC16 to Machine 2 using configuration 4 in location 7 (Figure 7.25).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Oper Clust Seq.	1	0	0	3	4	2	16	5	11	12	10	9	15	14	13	6	7	8
Machine	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Figure 7.25: Step 14-Add OC to Existing Machine Without Reconfiguration.

Step 15: Remove Machines that have no OCs associated with it (Figure 7.26).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Oper Clust Seq.	1	3	4	2	16	5	11	12	10	9	15	14	13	6	7	8
Machine	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Configuration	1	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Figure 7.26: Step 15-Add Remove Empty Locations.

The Final full process plan is shown in Figure 7.27

Oper Clust Seq.	1	3	4	2		16		5	11		12	10		9	15	14	13	6			7			8
TAD Odd Used	1	1	1	1		1		1	4		2	1		1	1	1	2	8			8			1
Machine	1	2	2	2		2		2	2		2	2		2	2	2	2	2			2			2
Configuration	1	4	4	4		4		4	4		4	4		4	4	4	4	4			4			4
Operation Seq	1	3	4	2	5	23	24	6	17	18	19	15	16	14	22	21	20	7	8	9	10	11	12	13
Tool Used	6	5	5	5	5	1	1	5	1	1	7	1	1	5	5	1	1	2	3	4	2	3	4	1

Figure 7.27: Final Reconfigured Process Plan for ANC-300.

7.5. Summary

The proposed approach has been applied to ANC-300 which is from a different part family from the ANC-101 and ANC-090 to illustrate the flexibility of the developed tool. The Reconfiguration approach has been applied in case of part change from ANC-090 to another different part ANC-300 and the process plan reconfiguration had been illustrated. The results show how the CAPPR approach uses the retrieval techniques for common OCs between the two parts. The generative techniques are then used to generate the process plan for the different operations in the new part. The results show that the CAPPR approach supports the upsizing of the manufacturing system to be capable of producing the new part. Also the Results demonstrate how the approach utilizes the current system state in terms of current machines and configurations for the old part.

CHAPTER EIGHT

CONCLUSIONS AND FUTURE WORK

In RMS process plans and planning functions are important links between products features and the features, capabilities and configurations of manufacturing systems. For this reason the efficient generation and reconfiguration of process plans is an important enabler for RMS.

The objective of the research presented was to develop an approach for macro-level process planning that makes use of the high reconfiguration capabilities of the machines within a reconfigurable manufacturing system environment. These capabilities enable the process plans to be reconfigurable and the machine configurations to be tailored according to the product feature requirements. To achieve this objective, several sub-problems had to be addressed, which include:

1. The challenges in RMS were outlined and showed that RMS needs a new concept of process planning that makes use of the reconfiguration capabilities that allows the machine structures to be tailored to the parts machining requirements.

2. The development of a procedure to cluster operations according to the logical and dimensional constraints.

3. The development of a mapping approach which establishes a mapping between the features of products and machine tools to generate the set machine structures capable of producing different operation clusters.

4. The representation of machine structures in a format similar to kinematic chains which captures the number, type and order of different axes of motion.

5. The development of a machine database that contains all the current available machine tools, their different capabilities and their configurations.

6. The formulation of a model for optimizing the process plans to minimize the total manufacturing cost for reconfigurable manufacturing systems.

7. The development of a constraint satisfaction procedure based on representing the process plan decision variables in a continuous domain that guarantees the feasibility of the generated process plans.

8. The development of a set of '*process planning reconfiguration rules*' that helps determine the exact actions to be carried out in case of unavailability of machine or part change. The rules attempt to maximize the utilization of the current manufacturing system configuration and its existing machines and their corresponding configurations before suggesting physical system reconfiguration.

8.1. Observations

The following observations can be made from the presented research:

1. Different approaches are required for process planning and machine selection for different manufacturing systems depending on their type.

2. The nature of RMS allows the machine structures to be tailored according to the demand requirements of the system.

3. The clustering procedure guarantees that operations with some special types of precedence constraints will be assigned to the same machine.

4. The proposed machine generation approach relies and builds upon the kinematic structure representation of machine tools. This developed mapping approach is general in nature and not limited to RMS.

8.2. Conclusions

The following concluding remarks can be pointed out of the presented research with regards to the problem under investigation (process planning for RMS):

1. The output from the machine structure generation stage showed that more than one machine configuration is generated for a single operation cluster. This increases the flexibility in selecting/configuring suitable machine tools and reduces the risk of not finding a capable machine if a new part is introduced which is another major contribution of the presented work.

2. The presented mapping approach for selecting the different types of machine(s) and their appropriate configurations to produce different types of parts and features, according to the required machine capabilities is a fundamental building block in generative planning of manufacturing processes.

3. The developed optimization process planning model representation needs less variables to represents the full process plan depending on the number of operations clustered in an operation cluster.

4. The process planning model was compared to the results of literature and reached better results for a similar problem using discrete GA parameters.

5. The developed set of "*process planning reconfiguration rules*" obtains a quick solution by applying a set of rules to reduce the costly amount of hard reconfiguration. Whereas, obtaining an optimal process plan will

require re-planning, a computationally expensive solution that will result in large amounts of hard reconfiguration.

6. The new developed approach to reconfigurable process planning facilitates expansion, downsizing and modification to the part or system.

7. RMS technological enablers allow machines to be designed around products and process plans to be reconfigured in response to changes in these products.

8.3. Research Contributions

The reported research makes the following contributions to the fields of machining structure capabilities, process planning and machine configurations selection for Reconfigurable Manufacturing Systems:

1. A new approach for process planning for RMS is developed. It addresses most of the issues and prerequisites of process planning systems when applied in the RMS context. The approach:

    a. Generates the machine structures and selects the capable machines for each operation cluster.

    b. Generates the optimal process plan and the corresponding machine configurations for every operation cluster.

    c. Provides detailed process plan reconfiguration steps that are to be carried out in case of machine unavailability of part change.

2. Although the focus of this work was on macro process planning in the machining domain. The general concepts introduced especially the concepts related to process planning reconfiguration, can be applied to

other domains such as assembly, metal forming and others. However implementation details will still have to be modified in order to define the details of an operation in the different domains.

3. The concept of mapping between the processing requirements of parts and the structural requirements of reconfigurable machine tools capable of producing these parts is introduced.

4. The proposed approach applied the new concepts of kinematics structure representation of the machine tools in process planning which proved to be a powerful way of machine representation.

5. A machine structure configuration selection approach was proposed. In this approach, given a part with its features and design specifications, operation clustering is performed. The modules of a machine structure that are capable of carrying out all the operations in the cluster are then selected. This can help in automatically determining/configuring machines that are capable of performing the required operations based on their kinematic structures. When compared to traditional methods which require manual determination of candidate machines for each operation as a prerequisite for process planning. This will help in automating the process of machine selection in commercial CAM systems because for current CAM systems to generate a process plan for a given part, the machine has to be manually selected as a prerequisite.

6. A new process planning approach was modeled for choosing the following parameters: operation selection (Machine, Tool and TAD selection) and operation sequencing. A new process plan representation was developed accordingly to represent both the OC and OP strings. The approach is flexible in the sense that the tolerance and logical constraints can be relaxed to produce traditional process plan with no pre-assigned OCs

while taking advantage of the continuous domain process plan representation method.

7. A novel procedure is developed and utilized to overcome the constraint satisfaction challenge of generating infeasible process plans. It is based on representing the decision variables into a continuous domain of variables. The new continuous domain of variables not only guarantees the satisfaction of the specified constraints but also provides variables that are not function of the number of alternatives which have variable domain size. This produces solution strings that are easy to manipulate using different types of operators, such as crossovers or mutations, without violating the constraints or changing the size of the solution string. The model was validated using a problem in literature. The developed procedure is general and can be applied to complex parts. In addition, this novel process planning model can be applied to any manufacturing system and not limited to RMS.

8. Rules, called "*Process Planning Reconfiguration Rules*", were introduced to help determine the exact actions to be taken to minimize the required hard-type reconfiguration on both system and machine levels by performing less costly soft-type reconfiguration to the existing process plans. A procedure was developed for automatically reconfiguring the process plan by adding/removing segments to/from the existing process plan string representation as necessary according to the required changes. In the case of part families the process plan can be developed for the composite part which contains all the features for all the parts within that part family. This will result in using the machine configurations that are capable of manufacturing all the parts within the family which avoids machine reconfiguration for part change within the part family.

9. A tool implementing the developed approach was developed using MATLAB® software. This tool provides a practical means for obtaining the

required machine capabilities for a given part and selecting the capable machines from the machine database. The tool also obtains the initial optimal process plan that reduces the manufacturing cost. In addition the tool facilitates selecting a machine to be unavailable or changing the part and the steps to be carried out to reconfigure the process plans are reported.

8.4. Future Work

A number of future research topics can be drawn from the presented research. These include:

1. Adding additional constraint types other than Tolerance and Logical such as Economical and Technological constraints. Examples of both types of constraints are:

   Economical (E): If drilling a large hole and underneath it is a small hole then it will be economical to drill the larger hole then the small one so that the small drilling tool cuts through less material. This saves tool life and time.

   Technological (T): An example of this constraint will be when drilling two holes that meet at an intersection. It is preferred to drill the thinner and longer hole first.

2. Extending the machine structure generation stage to be able to generate machine structures for parallel kinematic machines and accordingly find the work envelope for those structures to be able to select the capable machines.

3. Investigating the use of a hybrid optimization in solving the continuous optimization problem when generating the optimal process plan. This

hybrid algorithm can combine both GAs and Tabu Search (TS) and have the merits of both techniques. One way to do that is to have the TS algorithm start with the end results of GAs.

4. Expanding the cost model to incorporate cost elements such as reconfiguration cost and the cost of adding a machine, or cost of machine unavailability to aid in decision making in the RMS Process Planning Approach. Although the *Process Planning Reconfiguration Rules* try to avoid machine reconfiguration, knowing the other cost elements will help making a more accurate decision.

5. Applying an optimization technique to the process plan after reconfiguration. This technique will be applied after the *Process Planning Reconfiguration Rules* are applied. The optimization will be constraint to the current system configurations (i.e. after applying the *Process Planning Reconfiguration Rules* the machines' configurations will be fixed) and all the other parameters will be optimized to have an optimal process plan without having costly hard reconfiguration.

6. Investigating the use of more test cases will give feedback on special issues of different features which could help designers to find an alternative to those features in future designs taking into consideration the difficulty of obtaining data for more test cases

REFERENCES

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

Bohez, E.L.J. (2002) " Five-axis milling tool kinematic chain design and analysis", International Journal of Machine Tools and Manufacture, Vol.42, pp.505-520.

Britanik, J., and Marefat, M. (1995) "Case-based manufacturing process planning with integrated support for knowledge sharing", Proceedings of the IEEE International Symposium on Assembly and Task Planning, pp. 107-112.

Chu, C.-C.P. and Gadh, R. (1996) "Feature-based approach for set-up minimization process design from product design", Computer-Aided Design, Vol. 28, No.5 , pp.321-332.

Ciurana, J., G.-Romeu, M.L. and Castro. R. (2003) "Optimising process planning using groups of precedence between operations based on machined volumes". Engineering Computations, Vol.20, No. 1, pp. 67-81.

Contini P., and Tolio, T. (2004) "Computer-aided set-up planning for machining centres configuration", International Journal of Production Research, Vol. 42, No. 17, 'Selected Papers from the 17th ICPR 'Enterprise Integration - a Production Perspective', pp. 3473-3491.

Delbressine, F.L.M., De Groot, R., and Van Der Wolf, A.C.H. (1993) "On the automatic generation of set-ups given a feature-based design representation", Annals of the CIRP, Vol.42, No.1, pp.527-530.

Demey, S. Van Brussel, H. and Derache, H. (1996) "Determining set-ups for mechanical workpieces", Robotics and Computer Integrated Manufacturing, Vol.12, No.2, pp.195-205.

Dereli, T. and Filiz, I.H. (1999) "Optimisation of Process Planning Functions by Genetic Algorithms", Computers & Industrial Engineering, Vol.36, pp.281-308.

ElMaraghy, H.A. (2006) "Reconfigurable Process Plans for Reconfigurable Manufacturing", 3rd International CIRP Conference in Digital Enterprise Technology, Setúbal, Portugal, 18th to 20th September 2006 (Keynote paper).

ElMaraghy, H.A., (2005) "Flexible and Reconfigurable Manufacturing Systems Paradigms," Special Issue of the International Journal of Flexible Manufacturing Systems (IJFMS), (http://dx.doi.org/10.1007/s10696-006-9028-7), Vol. 17, No. 4, pp. 261-276.

ElMaraghy, H.A., (2002) "Flexible Manufacturing Systems", 91-512 Class Notes, Industrial and Manufacturing Systems Engineering Department, University of Windsor, Ontario, Canada.

ElMaraghy, H.A. (1993a) "Evolution and Future Perspective of CAPP", Annals of the CIRP, Vol.42, No.2, pp.739-751.

ElMaraghy, H.A. and ElMaraghy, W.H, (1993b) "Bridging the Gap between Process Planning and Production Planning and Control", CIRP J. Manufacturing Systems, Vol. 22/1, pp. 5-11.

ElMaraghy, H.A. and Gu, P. (1987) "Expert system for inspection planning", Annals of the CIRP, Vol. 36, No.1, pp.85-89.

ElMaraghy, W.H. (1992) "Integrating Assembly Planning and Scheduling-CAPP-Related Issues", Annals of the CIRP, Vol.41, No.1, pp.11-14.

Eversheim, W. and Schneewind, J. (1993) "Computer-Aided Process Planning-State of the Art and Future Development", Robotics and Computer Integrated Manufacturing, Vol.10, No.1/2, pp.65-70.

Fuqing Zhao; Aihong Zhu; Zongyi Ren; Yahong Yang (2006) "Integration of process planning and production scheduling based on a hybrid PSO and SA algorithm", 2006 IEEE International Conference on Mechatronics and Automation pp. 2290-2295.

Gerard J.C., Gaalman and Nallan C. Suresh (1999) "Towards an Integration of Process Planning and Production Planning and Control for Flexible Manufacturing Systems", International Journal of Flexible Manufacturing Systems, Vol.11, No.1, pp.5-17.

Groover, P (2001) "Automation, Production Systems and Computer Integrated Manufacturing". Prentice Hall, NJ.

Gu, P. and Norrie, D.H. (1995) "Intelligent manufacturing planning", Chapman and Hall, London.

Gupta, S.K., Nau, D.S., Regli, W.C. and Zhang, G., (1994) "A Methodology for Systematic Generation and Evaluation of Alternative Operation Plans", Advances in Feature Based Manufacturing / Edited by Shah, J.J., Mantyla, M. and Nau, D.S., Elsevier.

Gupta T., and Gali, T. R. (1993) "Design and implementation scheme for an alternate process planning system- A-CAPPS", Computers in Industry, pp. 15-23.

Henderson, M.R., Srinath, G., Stage, R., Walker, K. and Regli, W., (1994) "Boundary Representation-Based Feature Identification", Advances in Feature Based Manufacturing / Edited by Shah, J.J., Mantyla, M. and Nau, D.S., Elsevier.

Herrera, F., Lozano, M. and Verdegay, J.L., (1998) "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioral Analysis", Artificial Intelligence Review, Vol. 12, No. 4, pp. 265-319.

Holland, J., (1975) “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, MI.

Hummel, K.E.; Brown, C.W. (1989) “The Role of Features in the Implementation of Concurrent Product and Process Design”, Proceedings of ASME Winter Annual Meeting, DE-Vol.21, PED-Vol.36, ASME, New York, pp.1-8.

Hutchinson, G.K. and Pflughoeft, K.A. (1994) “Flexible process plans: their value in flexible automation systems”, International Journal of Production Research, Vol.32, No.3, pp.707-719.

Jain, Ajai ; Jain, P.K.; Singh, I.P. (2006) "An integrated scheme for process planning and scheduling in FMS", International Journal of Advanced Manufacturing Technology, Vol. 30, No. 11/12, pp. 1111-1118.

Joshi, S. B., Hoberecht, W. C., Lee, J., Wysk, R. A., and Barrick, D. C. (1994) ”Design, development and implementation of an integrated group technology and computer aided process planning system”, IIE Transactions, Vol. 26, No. 4, pp. 2-18.

Kempenaers, J., Pinte, J., Detand, J. and Kruth, J.-P. (1996) “A Collaborative Process Planning and Scheduling System”, Advances in Engineering Software, Vol. 25, pp. 3-8.

Kim, Y.F.; Park, K.; Ko, J. (2003) “A Symbolic Evolutionary Algorithm for the Integration of Process Planning and Job Shop Scheduling”, Computers and Operations Research, Vol.30, pp.1151-1171.

Kiritsis, D. and Prochet, M. (1996) "A generic Petri net model for dynamic process planning and sequencing optimization". Advances in Engineering Software, Vol.26, pp.61-71.

Kiritsis, D. (1995) "A Review of Knowledge-Based Expert Systems for Process Planning. Methods and Problems", International Journal of Advanced Manufacturing Technology, Vol.10, pp.240-262.

Koren, Y. (May 2005) "What Are the Differences between FMS & RMS," Paradigms of Manufacturing – A Panel Discussion, 3rd Conference on Reconfigurable Manufacturing, Ann Arbor, Michigan, USA.

Koren, Y., Ulsoy, A.G., Landers, R.G., Mehrabi, M., Pasek, A., Yip-Hoi, D. (2001) "NSF Engineering Research Center for Reconfigurable Machining Systems," Proc. 2001 NSF Design, Manufacturing and Industrial Innovation Research Conference.

Koren, Y., Heisel, H., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G. and Van Brussel, H. (1999) "Reconfigurable Manufacturing Systems", Annals of the CIRP, Vol. 48, No. 2, pp. 527-540.

Kruth, J.-P.; Detand, J.; Zeir, G. V.; Kempenaers, J.; Pinte, J. (1996) "Methods to Improve the Response Time of a CAPP System That Generates Non-Linear Process Plans", Advances in Engineering Software, Vol.25, pp.9-17.

Landers, R.G., Min, B.–K., and Koren, Y. (2001) "Reconfigurable Machine Tools", Annals of the CIRP, Vol.50, No.1, pp.269–274.

Li, W.D., Ong, S.K. and Nee, A.Y.C. (2002) "Hybrid Genetic Algorithm and Simulated Annealing Approach for the Optimization of Process Plans for Prismatic Parts". Int J Prod Res., Vol. 40, No.8, pp.1899-1922.

Ling, C. Yip-Hoi, D. and Koren, Y. (2000a) "Hole Pattern Identification in Gang Spindle Head Design for Reconfigurable Machining Systems", Proceedings of 2000 ASMT DET00: 26th Design Automation Conferences, Baltimore, Maryland, pp.25-32.

Ling, C. Yip-Hoi, D. and Koren, Y. (2000b) “Operation Clustering in Process Planning for Reconfigurable Machining System Design”, Proceedings of the 2000 Japan-USA Flexible Automation Conference, Ann Arbor, Michigan, pp.879-886.

Macchiaroli, R. and Riemma, S. (1994) “Clustering method for production planning and scheduling in a flexible manufacturing system”, Proceedings of IEEE International Conference on Robotics and Automation, 8-13 May 1994, pp.3155 – 3160.

Marefat, M. M., and Britanik, J. M. (1998) “Case-Based Process Planning for Three-Dimensional Components”, Integrated Product and Process Development, John Wiley & Sons, Inc.

Marefat, M. M., and Britanik, J. M. (1997) “Case-based process planning using an object-oriented model representation”, Robotics and Computer-Integrated Manufacturing, Vol. 13, No.3, pp 229-251.

Michalewicz, Z., Logan, T.D. and Swaminathan, S., (1994) “Evolutionary Operators for Continuous Convex Parameter Spaces”, Proceedings of the 4th International Conference on Evolutionary Computation, pp. 84-107.

Muljadi, H.1 ; Ando, K. (2005) “Feature recognition method aiming at the generation of alternative process plans”, DAAM International Scientific Book 2005, pp. 449-464

Ong, S. K., Ding, J. and Nee, A.Y.C. (2002) “Hybrid GA and SA Dynamic Set-Up Planning Optimization”, International Journal of Production Research, Vol. 40, No.18, pp.4697-4719.

Ozturk, F., Kaya, N. Alankus, O.B. and Sevinc, S. (1996) “Machining features and algorithms for set-up planning and fixture design”, Computer Integrated Manufacturing Systems, Vol. 9, No.4, pp.207-216.

Reddy, S.V.B., Shunmugam, M.S. and Narendran, T.T. (1999) "Operation Sequencing in CAPP Using Genetic Algorithms", International Journal of Production Research, Vol.37, No.5, pp.1063-1074.

Reynolds, B. I., Fink, P. K., and Moehle, C. J. (1993) "Intelligent assistance for variant process planning", SME Technical paper (Series) MS, MS93-252, pp. 1-12.

Saygin, C. (2000) "On Flexibility Aspects of Computer Aided Process Planning", Proceedings of the 2000 Japan.USA Flexible Automation Conference, Ann Arbor, Michigan, pp.1393-1400.

Saygin, C. and Kilic, S.E. (1999) "Integrating flexible process plans with scheduling in flexible manufacturing systems", International Journal of Advanced Manufacturing Technology, Vol.15, No. 4, pp.268-280.

Shabaka, A.I. and ElMaraghy, H.A. (2007) "Computer Aided Process Planning Reconfiguration (CAPPR)", 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Institute for Machine Tools and Industrial Management, Toronto, Canada on 22-24 July 2007 (Accepted).

Shabaka, A.I. and ElMaraghy, H.A. (2006a) "A GA-based constraint satisfaction model for generating optimal process plans". (Accepted for Publication in a special issue of the International Journal of Computer Integrated Manufacturing)

Shabaka, A.I. and ElMaraghy, H.A. (2006b) "A GA-based constraint satisfaction model for generating optimal process plans". 3rd International CIRP Conference in Digital Enterprise Technology, Setúbal, Portugal, 18th to 20th September 2006.

Shabaka, A.I. and ElMaraghy, H.A. (2006c) "Generation of Machine Configurations Based on Product Features", International Journal of

Computer Integrated Manufacturing (Published Online on : 26 July, 2006 DOI: 10.1080/09511920600740627).

Shabaka, A.I. and ElMaraghy, H.A.(2005) "Mapping Products Machining Requirements and Machine Tools Structure Characteristics in RMS", 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production, Institute for Machine Tools and Industrial Management, Technical University of Munich (Technische Universität München),Garching, Germany, September 22-23 2005.

Shabaka, A.I. and ElMaraghy, H.A. (2004) "Structural mapping between operation clusters and machine configuration for RMS", Proceedings of the International Workshop on Advanced Manufacturing Technologies (AMT), NRC-IMTI, London, Ontario.

Son, S.Y., (2000) "Design Principles and Methodologies for Reconfigurable Machining Systems", Ph.D. Dissertation, University of Michigan.

Sormaz, D. N.; Khoshnevis, B. (1995) "Knowledge Representation for Automated Process Planning", Assembly and Task Planning, 1995, Proceedings., IEEE International Symposium on , pp.34-39.

Spicer, P., Yip-Hoi, D. and Koren, Y., (2005) "Scalable Reconfigurable Equipment Design Principles", International Journal of Production Research, Vol. 43, No. 22, pp. 4839-4852.

Spicer, J.P., (2002) "A Design Methodology for Scalable Machining Systems", Ph.D. Dissertation, University of Michigan.

Stampfer, M. (2005) "Integrated set-up and fixture planning system for gearbox casings", International Journal of Advanced Manufacturing Technology, Vol. 26, No. 4, pp. 310-318.

Stranc, C., (1992) "The Reactive Planning Environment", MSc. Thesis, McMasterUniversity.

Taiber, J.G. (1996) "Optimization of process sequences considering prismatic workpieces". Advances in Engineering Software. Vol.25, pp.41-50.

Tang, L., Yip-Hoi, D. M., Wang, W. and Koren, Y. (2004) "Concurrent Line-Balancing, Equipment Selection and Throughput Analysis for Multi-Part Optimal Line Design". The International Journal for Manufacturing Science & Production, Vol. 6, No.1-2, pp. 71-81.

Usher, J. M. (2003) "Evaluating the Impact of Alternative Plans on Manufacturing Performance", Computers and Industrial Engineering, Vol.42, pp.585-596.

Wang, L.; Shen, W.; Hao, Q. (2006) "An overview of distributed process planning and its integration with scheduling", International Journal of Computer Applications in Technology, Vol. 26, No. 1/2, pp. 3-14.

Youssef, A.M.A. and ElMaraghy, H.A. (2006) "Modelling and Optimization of Multiple-Aspect RMS Configurations", International Journal of Production Research. Special Issue on Advances in Evolutionary Computation for Design and Manufacturing Problems, (In Press).

Yu, F. –F., Liu, X., and Liang, X. (2001) "A variant-feature-based modeling system for mechanical parts", Mechanical Science and Technology, Vol. 20, No.3, pp. 478-480.

Zhang, F., Zhang, Y. F. and Nee, A. Y. C. (1997) "Using Genetic Algorithms in Process Planning for Job Shop Machining", IEEE Transactions On Evolutionary Computation, Vol. 1, No. 4, November 1997.

Zhang, H.-C. and Lin, E. (1999) "A hybrid-graph approach for automated setup planning in CAPP", Robotics and Computer Integrated Manufacturing, Vol.15, pp.89-100.

Zhang, Y. F. and Nee, A.Y.C. (2001) "Application of Genetic Algorithms and Simulated Annealing in Process Planning Optimization", Computational Intelligence in Manufacturing Handbook, edited by Wang, J. and Kusiak, A., CRC Press 2001.

Zhang, Y. F., Ma, G. H. and Nee, A. Y. C. (1999) "Modeling Process Planning Problems in an Optimization Perspective", Proc. of the 1999 IEEE Int. Conf. on Rob. & Aut., pp.1764-1769.

Zhang, Y., Hu, W., Rong, Y. and Yen, D.W. (2001) "Graph-based set-up planning and tolerance decomposition for computer aided fixture design", International Journal of Production Research, Vol. 39, No.14, pp.3109-3126.

APPENDIX A

ANC EXAMPLE PARTS

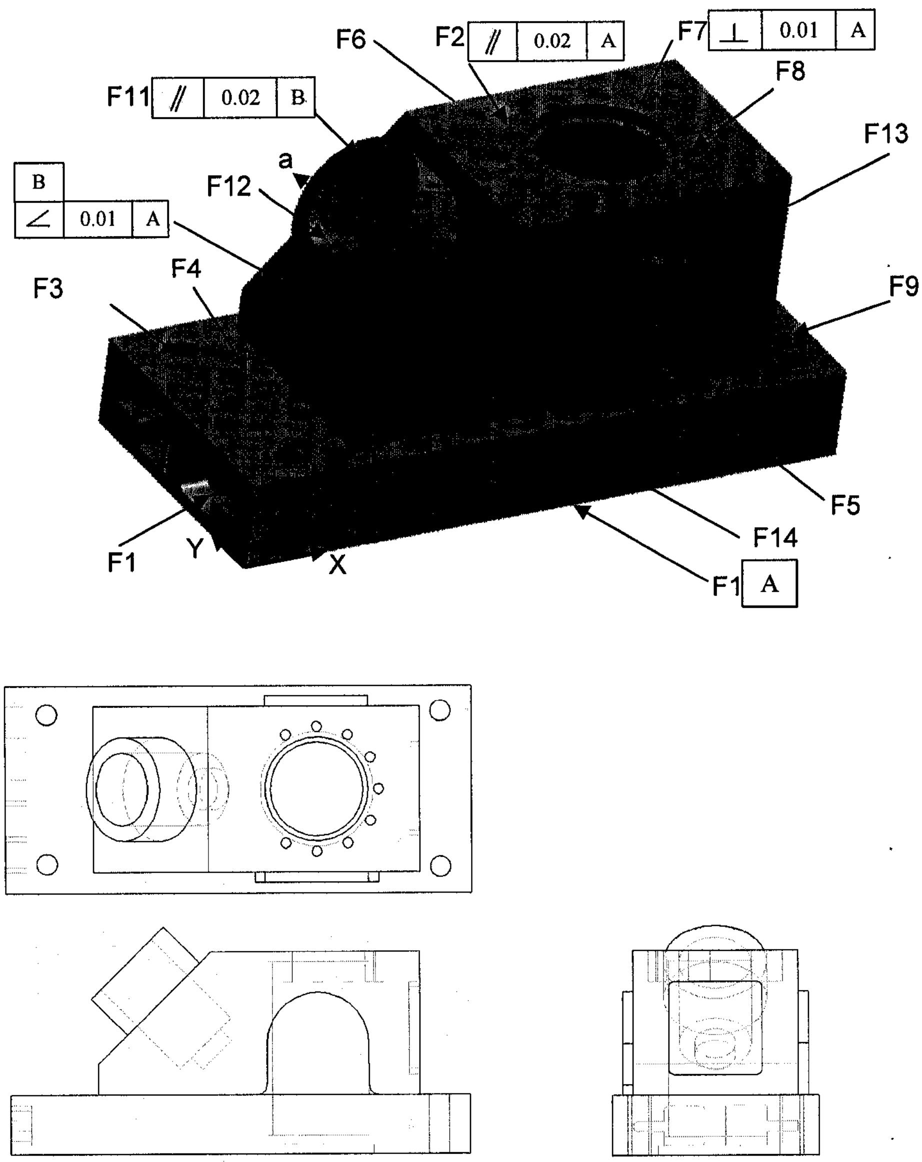


Figure A.1: Part ANC-101 and its Features.

Table A.1: Operations Data for Part ANC-101.

Feature	Description	Operation	Op. ID	TAD candidates	Tool candidates
F1	Planar surface	Milling	Op1	+Z	C6, C7, C8
F2	Planar surface	Milling	Op2	-Z	C6, C7, C8
F3	Four holes arranged as a replicated feature	Drilling	Op3	+Z, -Z	C2
F4	A step	Milling	Op4	+X, -Z	C6, C7
F5	A protrusion (rib)	Milling	Op5	+Y, -Z	C7, C8
F6	A protrusion	Milling	Op6	-Y, -Z	C7, C8
F7	A compound hole	Drilling	Op7	-Z	C2, C3, C4
		Reaming	Op8		C9
		Boring	Op9		C10
F8	Nine holes arranged in a replicated feature	Drilling	Op10	-Z	C1
		Tapping	Op11		C5
F9	A step	Milling	Op12	-X, -Z	C6, C7
F10	Two pockets arranged as a replicated feature	Milling	Op13	+X	C6, C7. C8
F11	A boss	Milling	Op14	-a	C7, C8
F12	A compound hole	Drilling	Op15	-a	C2, C3, C4
		Reaming	Op16		C9
		Boring	Op17		C10
F13	A pocket	Milling	Op18	-X	C7, C8
F14	A compound hole	Reaming	Op19	+Z	C9
		Boring	Op20		C10

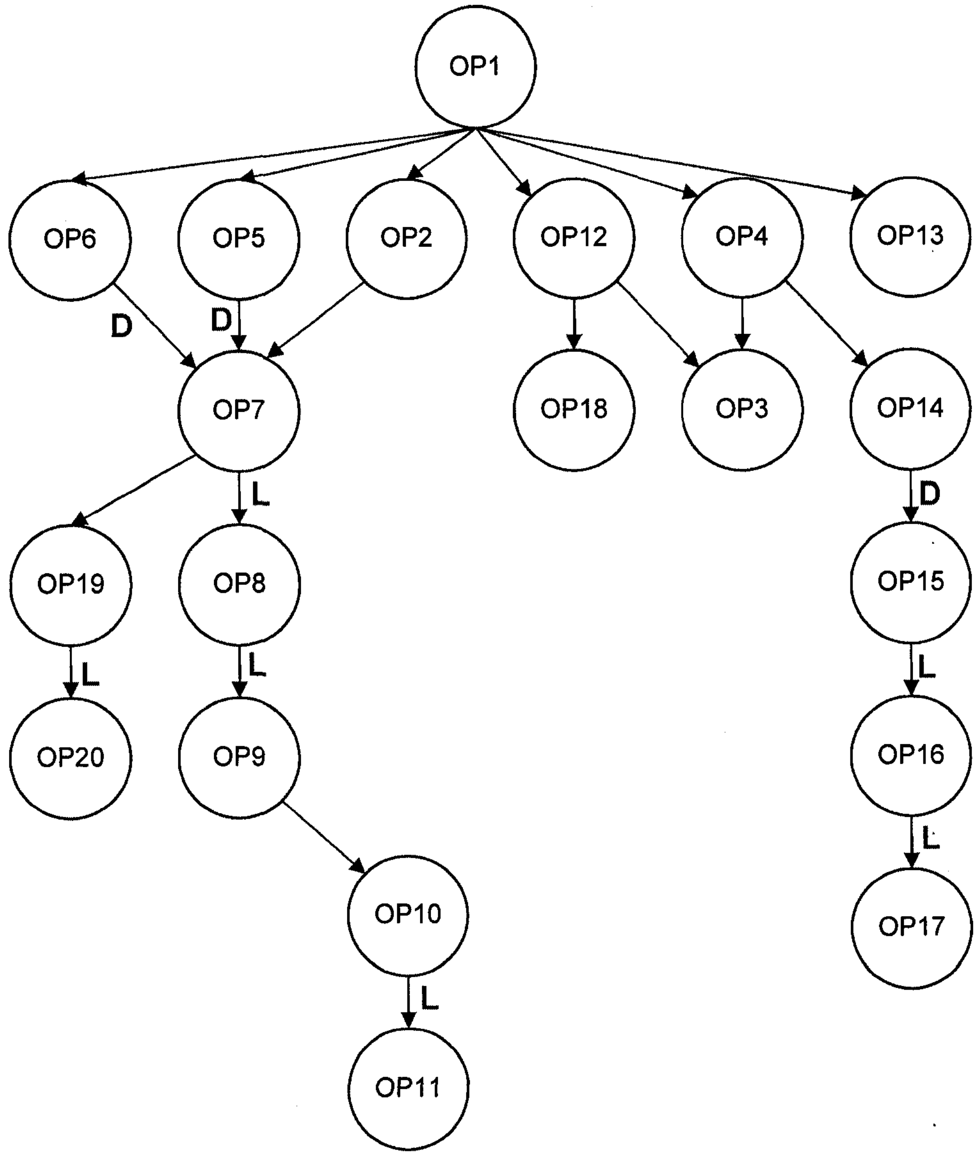


Figure A.2: Operations Precedence Graph for Part ANC-101.

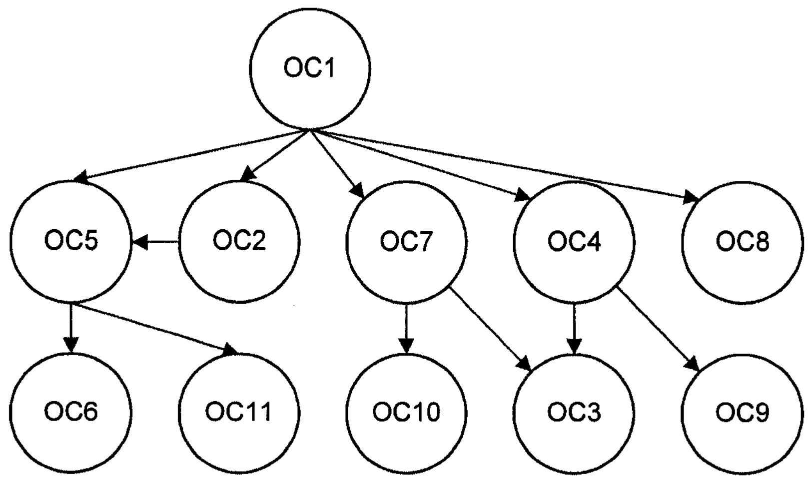


Figure A.3: Operation Clusters Precedence Graph for Part ANC-101.

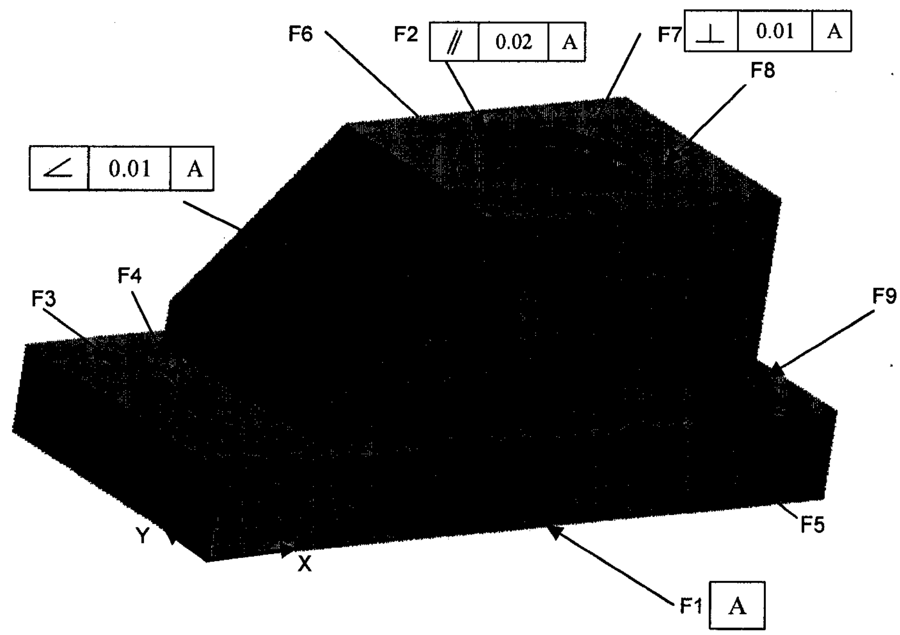


Figure A.4: Part ANC-090 and its Features.

Table A.2: Operations Data for Part ANC-090.

Feature	Description	Operation	Op. ID	TAD candidates	Tool candidates
F1	Planar surface	Milling	Op1	+Z	C6, C7, C8
F2	Planar surface	Milling	Op2	-Z	C6, C7, C8
F3	Four holes arranged as a replicated feature	*Drilling*	Op3	+Z, -Z	C2
F4	A step	Milling	Op4	+X, -Z	C6, C7
F5	A protrusion (rib)	Milling	Op5	+Y, -Z	C7, C8
F6	A protrusion	Milling	Op6	-Y, -Z	C7, C8
F7	A compound hole	Drilling	Op7	-Z	C2, C3, C4
		Reaming	Op8		C9
		Boring	Op9		C10
F8	Six holes arranged in a replicated feature	Drilling	Op10	-Z	C1
		Tapping	Op11		C5
F9	A step	Milling	Op12	-X, -Z	C6, C7

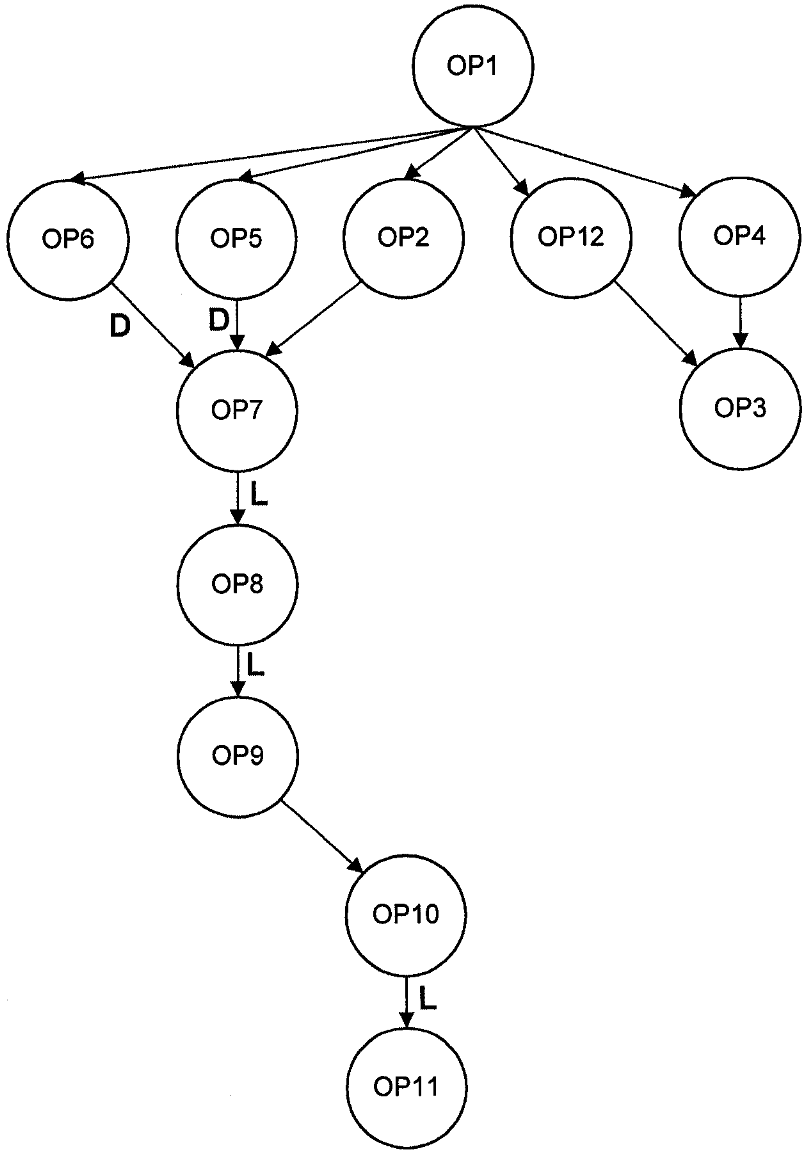


Figure A.5: Operations Precedence Graph for Part ANC-090.

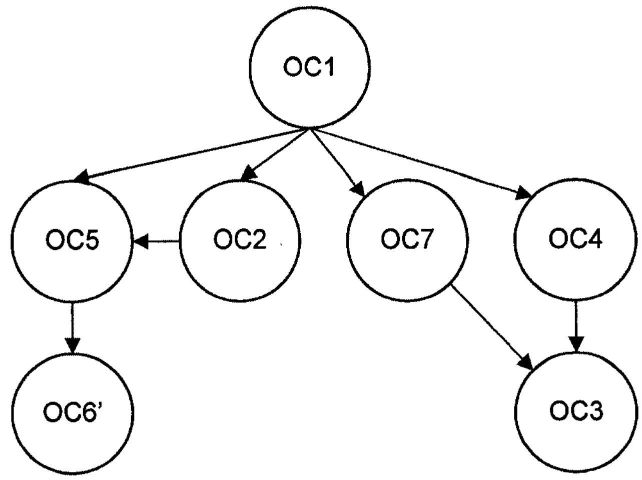


Figure A.6: Operation Clusters Precedence Graph for Part ANC-090.

APPENDIX B

MACHINE DATABASE

Table B.1 shows the machine tool database. The database has the following date:

a. Mc_{ID}: This field contains the machine ID number and it is unique for every machine with a given configuration.

b. M/C_{Base}: This field contains the machine base number, which is the machine number for a traditional machine tool or is the base number for an RMT that could have different configurations.

c. M/C_{Conf}: Is the configuration number for a given machine base number.

d. Stroke Length: This field contains the maximum stroke length in the X, Y and Z directions respectively. These values are used to validate is a machine is capable of producing a contain OC or not.

e. Rotation Angle: This contains the rotation angle limits in all three axis (+ve and -ve). These values are also used to choose the capable machines to perform a certain OC.

Table B.1: Available Machine Tool Data.

			Stroke Length			**Rotation Angles**					
Mc_{ID}	**M/C_{Base}**	**M/C_{Conf}**	**X**	**Y**	**Z**	**+X**	**-X**	**+Y**	**-Y**	**+Z**	**-Z**
1	1	1	100	50	60	0	0	0	0	0	0
2	2	1	100	100	100	0	0	0	0	0	0
3	2	2	100	100	100	135	135	0	0	0	0
4	2	3	100	100	100	0	0	115	115	0	0
5	2	4	100	100	100	135	135	115	115	180	180
6	3	1	120	80	90	0	0	0	0	0	0

MC_{ID}	Structure	MC_{ID}	Structure
1	Base # 1 └ X 100 mm └ Z 60 mm └ Y 50 mm	2	Base # 2 └ Z 100 mm └ X 100 mm └ Y 100 mm
3	Base # 2 └ Z 100 mm └ X 100 mm └ Y 100 mm └ A ±135°	4	Base # 2 └ Z 100 mm └ X 100 mm └ Y 100 mm └ B ±115°
5	Base # 2 └ Z 40 mm └ X 40 mm └ Y 90 mm └ A ±135° └ B ±115° └ C ±180°	6	Base # 3 └ X 40 mm └ Y 40 mm └ Z 90 mm

Figure B.2: Structures for Machine in Database.

APPENDIX C

RMT COST DATA

There is limited information in the literature regarding the cost estimates of the prospective reconfigurable machine tools (RMTs). Spicer [2002 and 2005] provide details about some cost figures/estimates for different RMT concepts, as extrapolated from actual cost information about present agile CNC machines. Son [2000] also provides cost estimates for RMT costs.

From Spicer [2002 and 2005] and Son [2000] the following cost estimates for different alternative configurations of different RMTs can be deduced (Only single spindle RMT costs are considered in this appendix).

C.1 Horizontal Milling RMT Limited to 3-Axis

The RMT machine base costs 480,000 USD. Each spindle module costs 170,000 USD and its corresponding fixture module costs 110,000 USD. Therefore, the total cost for this 3-axis single spindle RMT is 760,000 USD.

C.2 Horizontal Milling RMT Upgradeable to 4-Axis

The 3-axis RMT machine base that can be upgraded to 4-axis costs 580,000 USD. Each spindle module costs 170,000 USD and its corresponding fixture module costs 110,000 USD in case of 3-axis and costs 160,000 USD in case of 4-axis (rotary fixture). Adding the 4th axis of motion necessitates adding an additional pallet indexer that costs 100,000 USD. Therefore, the total cost for this

RMT with 3-axis and 1 spindle is 860,000 USD. The total cost for this RMT with 4-axis and 1 spindles 1,010,000 USD.

C.3 Drilling Press RMT

The RMT machine base with one spindle costs 385,000 USD. Each additional spindle module with its fixture costs 170,000 USD. Therefore, the total cost for this reconfigurable drilling press with 1 spindle is 385,000 USD.

APPENDIX D

GENETIC ALGORITHMS

This appendix is provided to give a brief idea about Genetic Algorithms (GAs) and the operators used for real-coded GAs.

D.1 General Overview of Genetic Algorithms

The following pseudo-code gives the general overview of a Genetic Algorithm:

2. Let $F(x_1,\ldots,x_m)$ be an objective function to be optimized, where $(x_1,\ldots,x_m)$ are the independent variables, where each variable x_i ranges between a lower and an upper limit $[v_{min}, v_{max}]_i$.
3. Convert the function F from a minimization to a maximization problem, where a new function $f(F)$ is to be maximized. The new function is known as the fitness function.
4. Generate a random population P of N instances of the independent variables (known as **chromosomes**).
5. For a pre-specified number of generations (iterations)
    a. Let the total number of offspring chromosomes due to the application of the mutation and cross over operators be denoted by M.
    b. Use the **selection** operator to fill a new population with N-M high fitness chromosomes.
    c. Use the **selection** operator along with the **mutation** and **cross over** operators to fill the remaining M locations in the population.

d. For the new population, evaluate the objective function (and fitness) value for the chromosomes changed by cross over and mutation, and retain the fitness values of the unchanged chromosomes.

6. End

D.2 Operators Used for Real-Coded Genetic Algorithms

This section provides a brief description of the operators used in the real-coded Genetic Algorithms. The same types of operators apply for integer-coded GAs with the small difference of dealing with discrete domains of decision variables rather than continuous domains. This can be achieved by splitting the [0,1] ranges of the continuous domains into a number of equal divisions representing the different values in the discrete domains for each variable. Same kinds of operators can be applied accordingly.

D.2.1 Selection Operators

The selection scheme adopted is an elitist tournament selection, where the best chromosome is retained between successive generations, to ensure that there is no loss of the best-obtained chromosome. The tournament selection is modified to accommodate the selection of low fitness chromosomes as well as high fitness chromosomes. This modification is necessary as some mutation operators operate on low fitness chromosomes.

D.2.2 Cross-Over Operators

Cross-over operators change chromosomes in a semi-local fashion to produce new chromosomes in the vicinity of the old ones, and hence should be used on chromosomes with high fitness values. Three cross-over operators were used in this work:

D.2.2.1 Arithmetic Cross-Over

Given a pair of chromosomes:

$$\underline{X}_1 = \{x_1^{\ 1}, x_2^{\ 1}, x_3^{\ 1}, \ldots, x_n^{\ 1}\}$$

$$\underline{X}_2 = \{x_1^{\ 2}, x_2^{\ 2}, x_3^{\ 2}, \ldots, x_n^{\ 2}\}$$

Generate a random number α between [0,1] and produce the new chromosomes $\underline{Y}_1$ and $\underline{Y}_2$, where:

$$\underline{Y}_1 = \alpha\, \vec{x}_1 + (1-\alpha)\vec{x}_2$$

$$\underline{Y}_2 = (1-\alpha)\vec{x}_1 + \alpha\, \vec{x}_2$$

This operator produces new chromosomes on a straight line joining the parent chromosomes. It has some kind of an averaging effect between the values of the parent chromosomes. Such operator is useful when a minima is located between the parent chromosomes.

D.2.2.2 Simple Cross-Over

Simple cross-over simulates the bit swapping found in the cross-over operator of binary coded Genetic Algorithms. Given a pair of parent chromosomes:

$$\underline{X}_1 = \{x_1^{\ 1}, x_2^{\ 1}, x_3^{\ 1}, \ldots, x_k^{\ 1}, \ldots, x_n^{\ 1}\}$$

$$\underline{X}_2 = \{x_1^{\ 2}, x_2^{\ 2}, x_3^{\ 2}, \ldots, x_k^{\ 2}, \ldots, x_n^{\ 2}\}$$

Choose a random location k, and produce the new chromosomes $\underline{Y}_1$ and $\underline{Y}_2$, by swapping the values in both chromosomes to the right of the location k.

$\underline{Y}_1 = \{x_1^{\,1},....,x_k^{\,1},x_{k+1}^{\,2},....,x_n^{\,2}\}$

$\underline{Y}_2 = \{x_1^{\,2},....,x_k^{\,2},x_{k+1}^{\,1},....,x_n^{\,1}\}$

This operator acts as an averaging search mechanism along the dimensions of the parent chromosomes.

D.2.2.3 Heuristic Cross-Over

Heuristic cross-over was introduced by Michalewicz *et al.* [1994] to add a steepest-descent search element to the genetic search, to fine-tune the solutions. Given a pair of chromosomes $\underline{X}_1$ and $\underline{X}_2$, find $f(\underline{X}_1)$ and $f(\underline{X}_2)$, where *f* is the objective function value in case of minimization. Generate $\vec{x}_3$ along the direction of the lower objective function value, where:

$$\underline{X}_3 = r.(\underline{X}_2 - \underline{X}_1) + \underline{X}_2$$

r = random number between [0,1]

If the boundaries are exceeded then repair the value of $\vec{x}_3$ to stop at the boundary.

D.2.3 Mutation Operators

Mutation operators are random search elements within the genetic search that diversify the search within the domain of the independent variables. Since there is no guarantee that the generated chromosomes will have a better objective function values, therefore the parent chromosome on which the operator is applied should be chosen from among the low fitness chromosomes. Four mutation operators were used in his work:

D.2.3.1 Uniform Mutation

Given a chromosome $\underline{X} = \{x_1,.......,x_n\}$, replace x_k with a random number between $[L_k, U_k]$, where $[L_k, U_k]$ are the bounds on the variable x_k, where the location *k* is chosen randomly between 1 and *n*. Uniform mutation diversifies the search along a randomly chosen variable within the set of independent variables.

D.2.3.2 Boundary Mutation

In many optimization problems, the global optimum value of the objective function lies near the boundary of the search space. The genetic search might miss those boundary optima if the search points become concentrated in the middle of the search space. In order to remedy this problem, Michalewicz *et al.* [1994] introduced the boundary mutation operator. Given a chromosome $\underline{X} = \{x_1,....,x_k,....,x_n\}$, a random location $k \in \{1,\dots,n\}$ is chosen, then the variable x_k is replaced with either the minimum or the maximum value of the range of the x_k. Either boundary is chosen randomly.

D.2.3.3 Non-Uniform Mutation

Non-uniform mutation is an operator that starts as a diversifying search element over large spaces around the mutated chromosome in the early stages of the search, and ends up with small variations around the mutated chromosome in the final generations. Boundary mutation is applied as follows: Given a chromosome $\underline{X} = \{x_1,....,x_k,....,x_n\}$, replace x_k by $x_k^{'}$ (k randomly chosen), where:

$$x_k^{'} = \begin{cases} x_k + \Delta(\mathrm{t}, U_\mathrm{k} - x_k) \\ x_k - \Delta(\mathrm{t}, x_k - L_\mathrm{k}) \end{cases}$$

Either of the above equations is chosen randomly.

$$\Delta(t,y) = y \cdot r\left(1 - \frac{t}{T}\right)^6$$

t = The number of the current generation

T = Maximum number of generations

R = Random value between [0, 1]

In the early stages of the search, the value [1-*t*/*T*] is large, and hence large variations from the mutated chromosome can be obtained. This value decays with generations, thus producing small variations.

D.2.3.4 Whole Non-Uniform Mutation

Given a chromosome $\underline{X} = \{x_1, \ldots\ldots\ldots, x_n\}$, apply non-uniform mutation on all variables. This operator diversifies the search along the space of all variables. It is particularly useful in the early stages of the search.

VITA AUCTORIS

NAME:	Amr Ibrahim Mohamed Shabaka
PLACE OF BIRTH	London, England
YEAR OF BIRTH	1977
EDUCATION	Cairo University, Cairo, Egypt 1993-1998 B.Sc.
	Cairo University, Cairo, Egypt 1998-2001 M.Sc.
	University of Windsor, Windsor, Ontario, Canada 2001-2007 Ph.D.