An Integrated Framework for Feature Recognition and Co-Evaluation of Process Planning Through Kinematics Configurations for Flexible and Changeable Manufacturing Systems

Ausama Nawaz
NUST201464466MCEME35114F

Supervisor
DR. SAJID ULLAH BUTT

DEPARTMENT OF MECHANICAL ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY
ISLAMABAD

AUGUST, 2018

An Integrated Framework for Feature Recognition and Co-Evaluation of Process Planning Through Kinematics Configurations for Flexible and Changeable Manufacturing Systems

Ausama Nawaz

NUST201464466MCEME35114F

A thesis submitted in partial fulfillment of the requirements for the degree of MS Mechanical Engineering

Thesis Supervisor:

DR. SAJID ULLAH BUTT

Thesis Supervisor's Signature: ______________________

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

AUGUST, 2018

Declaration

I certify that this research work titled "*An Integrated Framework for Feature Recognition and Co-Evaluation of Process Planning Through Kinematics Configurations for Flexible and Changeable Manufacturing Systems*" is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

Signature of Student

Ausama Nawaz

NUST201464466MCEME35114F

Language Correctness Certificate

This thesis has been read by an English expert and is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the university.

Signature of Student

Ausama Nawaz

NUST201464466MCEME35114F

Signature of Supervisor

Copyright Statement

Acknowledgements

"All praise belongs to Allah, Lord of all the worlds [1:2]"

Doing job and studying at the same time required a really good struggle and motivation to complete this task. Thus, a lot of people helped and guided me directly or indirectly in the completion of this research work. So, they entirely deserve my sincere appreciation.

My special thanks to Dr. Sajid Ullah Butt, my research supervisor; for his patience, kind attitude and above all the guidance and help throughout the hostile times of my research work. It was an immense pleasure and honor for me to work with him. Without his efforts, this task might not have been accomplished. Further, many thanks to the advisory committee member Dr. Hasan Aftab. Special thanks to my external supervisors Col. Dr. Aamer Ahmed Baqai and Dr. Wasim Akram, who encouraged and motivated me to do research in this area. They guided me throughout the research phase.

My sincere thanks to my parents and siblings, for their prayers and backing throughout my work because it would not have been possible without their support.

All that I am, and hope to be, I owe it to my parents…

Abstract

In modern design, products are often designed and simulated in special software environments to test and see the effects in real world usage. That data is used to refine the design and make required changes to achieve objectives set forth beforehand. After a design is finalized, a computer file is generated which contains 3D data about the product or part. Then this file is analyzed by an operator and 2D drawings are generated from different views with dimensions. These 2-D drawings are ones which are sent to planning and then subsequently manufacturing facilities.

The aim of this research is to provide a framework for data extraction from the 3D model of the newly designed part and feed it to the later stages of manufacturing. In addition, the framework will also let the operator enter Machine details (Specifications, Tools, DoF) right after the part data is entered. This will allow the framework to eliminate all process plans irrelevant to the available machining setup and avoid the hassle of doing so later.

Weighted Genetic Algorithm (WGA) is proposed to be used for optimization and selection of best process plan(s). The priorities of different optimization objectives can be set using the weights assigned to each objective. A novel method of applying the genetic operators has been developed which allows the diversity to be ensured during the entirety of the life cycle of the algorithm.

The framework has been developed in MATLAB® and has been designed form the ground up to be fully generic allowing the process to be seamless and highly intuitive.

Key Words: *CAPP*, *CAM*, *CAD*, *Genetic Algorithm*, *Feature Recognition*, *Process Planning*, *Graphical User Interface*

Table of Contents

List of Figures

List of Tables

Nomenclature

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
DMS	Dedicated Manufacturing System
FMS	Flexible Manufacturing System
RMS	Reconfigurable Manufacturing System
RMT	Reconfigurable Machine Tool
GA	Genetic Algorithm
WGA	Weighted Genetic Algorithm
MO-WGA	Multi Objective - Weighted Genetic Algorithm
APP	Alternate Process Plan
PP	Process Plan
GUI	Graphical User Interface
TAD	Tool Approach Direction
DoF	Degrees of Freedom
CAPP	Computer Aided Process Planning
CDV	Couvercle De Vileberequin [1] – Shaft Cover

Chapter 1

1 Introduction

CAD and CAM are very familiar terminologies in today's time and age. The ability to communicate seamlessly between these those down has been the target of research for a considerable amount of time. CAPP has is the part which serves as an intermediary between CAD and CAM processes. There are a few steps involved during the transition from CAD to CAPP and finally to CAM. Each of the involved steps and methods are explained in the following sections

1.1 CAD Systems and Feature Extraction

The place to start the process of manufacturing a part is often with extraction of features from a CAD computer file which can be saved in one of many formats. CAD files usually contain information about the model in 3D space in terms of edges, faces, primitive volumes and/or vertices which are not related to CAM. Because of this, usually a user is required to input the parameters for different features in a part to be manufactured into the CAM system in order to be able to reliably produce the part. This is where Automatic Feature Recognition (AFR) systems come in. These are intended to translate the low level data in CAD file into data understandable by CAM systems.

AFR systems usually employ one of many methods which can be used to extract feature information. Some subtract the model from a solid 3D object (also called volume decomposition technique) or follow edges and ridges on a model to figure out the shape and size (Attributed adjacency Graph) to name a few. The end goal is the same; to provide output in terms of the features and their dimensions. There however are still limitations to the identification prowess of such systems but these shortcomings are the aim of many currently in progress researches.

1.2 Machine Kinematics

In any established manufacturing facility, there are bound to be machines which are already there available for production. Not all machines are made similar. Even within a same class of machines, there are fundamental differences in types of jobs a machine can handle. These difference stem from the field of study called Kinematics. Kinematics deals with motion without the consideration of forces responsible for causing the motion. A very basic quantifiable unit in Kinematics is DoF, a

measurement of ways in which a linkage can move. DoF can be calculated on a joint or for a whole mechanism. In total, there are 6 Degrees of Freedom; 3 rotational and 3 translational.
The three translational freedoms are along the three axes (traditionally x, y, z) measured in distance units and rotational freedoms are about the same three axes measured in angles (traditionally θ_x, θ_y, θ_z) (Figure 1.1 below). Degrees of freedom are checked and calculated for every linkage inside a mechanism. The DoF for a whole mechanism can be calculated using the data from all the linkages inside. Hence each machine with multiple linkages receives a DoF capability rating reflecting its capabilities in an objective way.

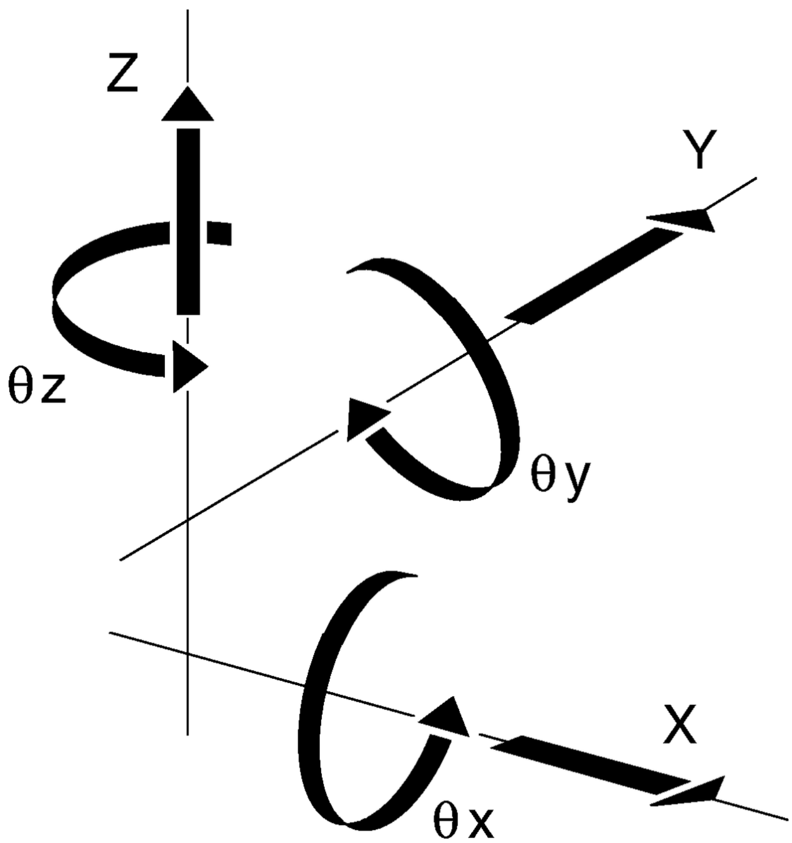


Figure 1.1 6 DoF illustrated

1.3 Manufacturing Systems

Any manufacturing setup consists of one or more machines. Furthermore, each machine consists of multiple subsystems contributing to the overall functionality of the machine. The addition of value to raw materials by transforming into finished goods is known as the process of manufacturing. As such it adds a lot to the economical prowess of an area and it provides a way of converting natural resources to profitable earnings. In its very early days, most manufacturing processes were manual and needed skilled craftsman to perform them. With time, machines evolved to take over. Machines offered added speed and efficiency as compared to their human counterparts. Many manufacturing systems exist to cater for specific production needs but here we will discuss the ones which cater to a wider production style. The most relevant and modern ones (Figure 1.2 below) are discussed in the following sections.

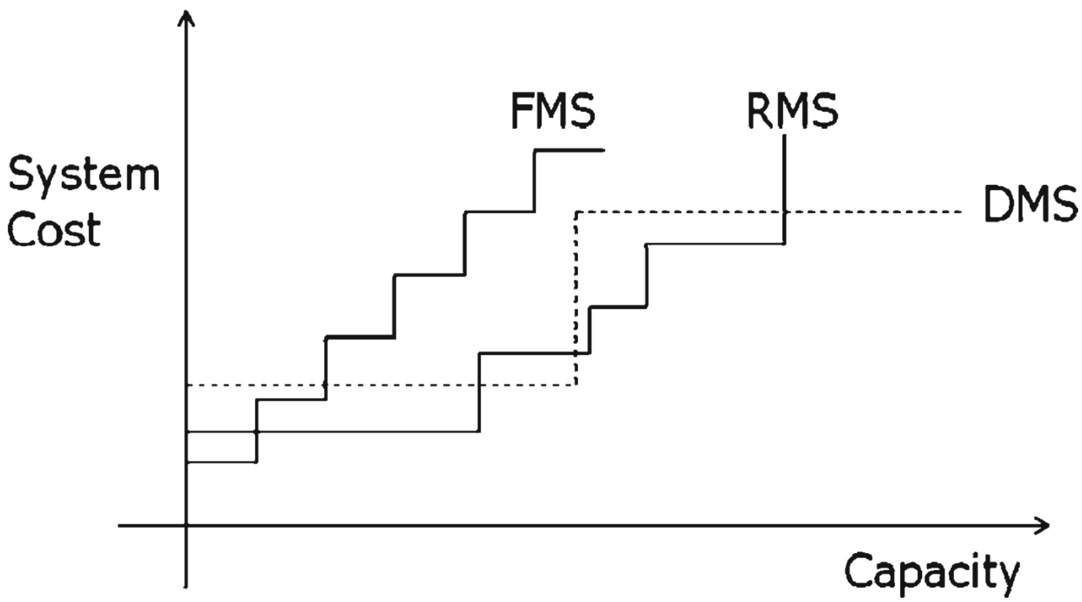


Figure 1.2 Cost vs Capacity FMS, RMS & DMS[2]

1.3.1 Dedicated Manufacturing System

In instances where same operations and thus parts are manufactured in great repetition, a DMS is just what is required. With the ability to offer highest tool utilization of any manufacturing system, these systems are the ones adopted for mass production of goods. Per item production costs are much lower as compared to other systems. Like any system, DMS is not without flaws, the most important of which is inherent difficulty in adapted changes in part design. Any change is very time consuming and costly. Other systems have been adopted to account for shortcomings but they in turn have some of their own as well.

1.3.2 Flexible Manufacturing System

In the last decades of 20th Century, trend was starting to shift in the manufacturing industry. There was growing demand for the ability of manufacturing systems to adapt to changes and be rapid in implementation of those changes. These changes that FMS can respond to, fall in the small changes category and for parts belonging to the same family. The FMS system is designed with a particular part family in consideration; hence in a sense it provides customized flexibility. This customizability can be in production capacity and in the software which is running the FMS in the background. The changes required from the system can stem from multiple reasons such as change in the underlying part design, change in the demand from market and but not limited to launch of a newer version.

FMS is advantageous in the flexibility department but when compared to DMS; is a more expense solution. DMS offers very fast operational speed as compared to FMS. Initial setup cost is also higher for FMS.

1.3.3 Reconfigurable Manufacturing System

In the modern industry changeability is not the only requirement from a manufacturing system. Mass production ability is very important to be able to drive per unit cost down and subsequently for profitability to rise. This is one of the many reasons RMS was developed. RMS was not developed from scratch. It inherited desirable features from FMS (Flexibility of change) and DMS (Ability to use multi-tool operations). This also gives rise to a sub structure within RMS, RMT.

RMT is based on the ability to change structures. This in turn allows the system to the scalable in accordance with the market demand and changes in the part design. It also allows the manufacturing of new parts within the same part family by change of required tools. In addition, it allows new machines to be added to the existing systems with the minimum changes required to the underlying software. This ability to add machines in the system is exclusive to RMS as it did not exist in the FMS.

1.4 Process Plans

There are multiple sequences of operations a part might take to reach its final shape. From all the possible sequences, some operations are selected based on part specifications and others based on tools already available in the manufacturing facility. Each one of these sequences is referred to as a process plan. A process plan basically dictates which operations are to be undertaken on which machine and most importantly in what order. The step that comes between manufacturing and design is called process planning, this is what links both. There are different types of process planning (Figure 1.3 below) where level of automation and re-configurability sets them apart.

The most basic way of process planning is doing it manually. Advent of cheaper and cost effective computing paved the way for CAPP and automated process planning later. Automated process planning also is based on CAPP. It is further divided into two types. In variant approach, new parts are first categorized with respect to similarities from the parts already in the system. This saves on the full effort of generating a new process plan for each part. For this purpose, a database of parts already made needs to be maintained. Generative approach is opposite to variant approach. It is based on generation of new process plan for each new part regardless of similarities to parts already made. Reconfigurable process planning provided a better way to work with RMS having the notable ability to generate alternate process plans. It offers a way to produce process plans which evolve with the part itself and with its variants in time. All the parts are classified in families. Whenever a new part is introduced in the system, it is matched with the existing families. The most relevant master process plan is then loaded. The features which are different from the ones already in the system are identified and then these features are added to the precedence table in iterations to best determine the place for latest features.

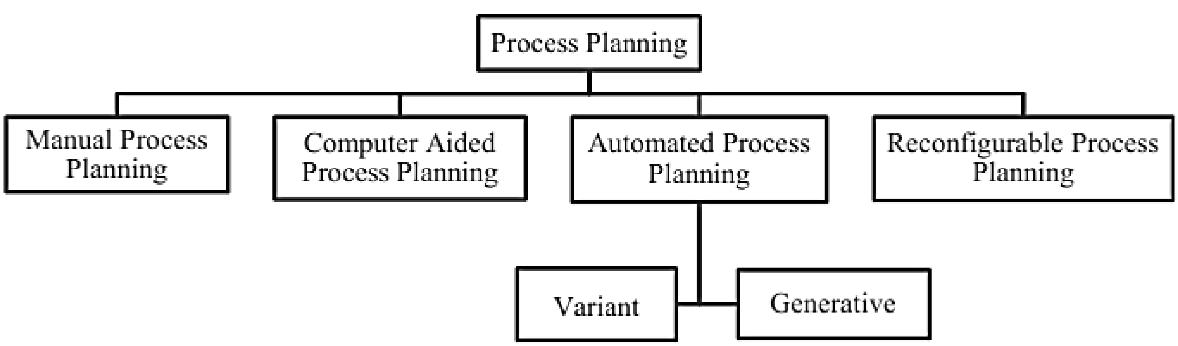


Figure 1.3 Types of Process Planning[3]

The main concentration in this theses will be on CAPP, which basically introduced the usage of computer resources in the traditionally manual task of process planning.

1.5 Scope and Motivation

This thesis aims at filling the following gaps found in the current manufacturing environment.

- Absence of a standard format for input of features data to optimization systems
- During crossover and mutation, the TADs aren’t involved
- Automatic selection of crossover type and points on a pair by pair basis
- Focus on existing machine capacities during the optimization process

There are two levels that these systems need to be developed and implemented; factory floor level and software optimization. In this thesis, with a concentration on software optimization; a new framework will be suggested, implemented and results shown. This current chapter is meant to give an overview of the topic and current industrial situation. Specifics will be delved into in the next chapter of literature review.

Chapter 2

2 Literature Review

As the suggested framework encompasses the whole process from CAD to the final process plan selection, individual areas will be discussed along with the current state of technology.

2.1 CAD Feature Recognition

The process of AFR begins by parsing the input CAD file and looking for features in the form of but not limited to holes, slots and cavities. Then the recognized features are turned into delta volume (Volume of material to be removed from stock to achieve final part shape and dimensions). Henderson et al [4] proposed such a method which recognizes features in a given CAD file, extrapolates the volume of material to be removed on a per feature basis. The delta volume is then represented in a graphical form by listing occurrences of each feature accordingly. This method was implemented in PROLOG programming language. Common features were defined as a set of logical rules. The input file is of the B-rep (Boundary representation) type which describes the part as a collection of facts. For complex features it was proposed to decompose them into smaller constituent features.

A concept known as Attributed Adjacency Graph (AAG) was developed by Joshi et al [5] for the extraction of machinable features from a given part's CAD model. This proposed technique works with models saved in B-rep format. The proposed methodology to implement AAG was only applicable on recognizing features of polyhedral nature which include but are not limited to slots, pockets, blind slots (Figure 2.1 below), blind steps, steps and polyhedral holes. The proposed method was compared by an algorithm which scans the entire model for features one by one by performing an exhaustive search. AAG showed promise in not only reducing the computational effort required but also in areas of types of features recognized. The main negative aspect of the proposed method was inability to cope with features of types other than polyhedral. Similarly intersecting features presented a challenge to the AAG algorithm proposed.

Neural networks due to their superior parallel processing capability have been gaining traction for some time now. One prerequisite for utilizing neural networks is parallel input of data. Such a technique was proposed by Prabhakar et al [6] which feeds B-rep solid models to a neural networks based algorithm. In order to accomplish this, they had to develop a new format for model representation consisting of description of faces as well as relationships between faces. They also designed a net architecture for the neural network which functionally resembles multilayer perception. This remedied the slow speed of recognition of earlier attempts at AFR systems as well as inability to recognize intersecting or partial features and also having to define library of features beforehand. By converting a solid model to an adjacency matrix, patterns are recognized on each row. The most desirable aspect of using neural networks based AFR is that all features of the solid model are recognized in parallel simultaneously. Similarly, the orientation of the input part is not a factor to be considered reducing preprocessing complexity.

Volume decomposition is one of the newer techniques being used to implement AFR. It operated on polyhedron features similar to some other methods. Sakurai [7] proposed that the polyhedron be divided into maximal cells. These maximal cells are formed by the intersection of half spaces of faces which possess convex edges. Then then half spaces are organized into sets. The formed sets are examined for any relationships occurring between the concave edges of the neighborhoods and half spaces themselves. This whole process accomplishes decomposition of the delta volume into smaller maximal cells of convex nature. Then these maximal cells are subtracted from one another in varying orders to produce more than one interpretations of features. This can lead to a large number of feature interpretations. To avoid this problem, machining sequence was generated directly off of the maximal convex cells. A weak point of the proposed method is that it cannot efficiently deal with curved surfaces and any related features.

In time when the 20th century was starting to take over the previous century, a concept was emerging which promised faster time from concept to product and between product design changes. This concept is known as Agile Manufacturing. Due to this concept, a new CAD data format labelled as being ‘neutral’ was introduced; STEP (STandard for Exchange of Product model data). Bhandarkar et al [8] proposed a methodology to extract features data from a STEP file. The output of the proposed system was a STEP files based on the AO224 format for use in process planning based on form features. The features which require milling to be produced are recognized in the proposed technique. Han et al [9] proposed a technique to integrate process planning and AFR in the domain of machining. A set of machine setups are generated which can be used to manufacture the required part. Then using a twin layered approach, machinable features are recognized by consulting the database by the upper layer. Then using the lower layer, optimal machining sequence is generated and cost is calculated. This is repeated for every setup until there exists an optimized sequence of machining for every setup. The input part file is STEP in this case.

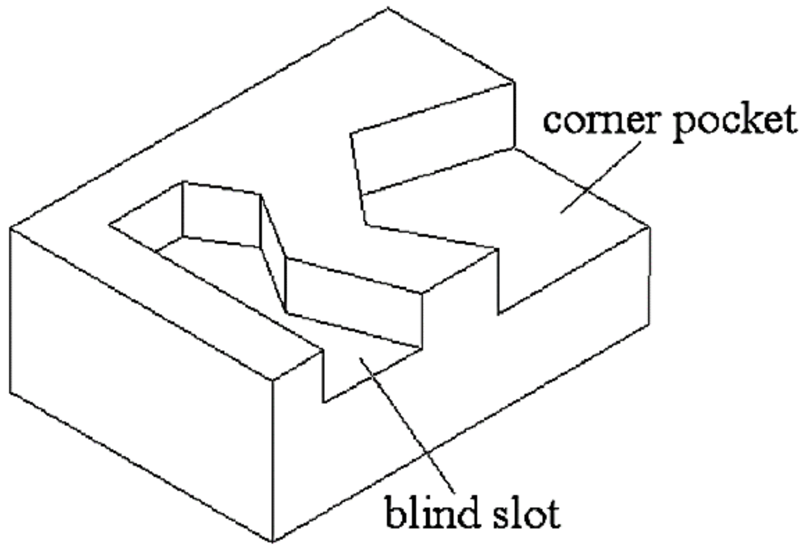


Figure 2.1 Sample part with non-standard type corner pocket and blind slot features

In a bid to recognize non-standard features in B-rep based CAD file, Ozturk et al [10] proposed a method for AFR utilizing neural networks. Their approach is based on processing of face-score values of the solid model in terms of topology and geometry. Because of the fact that the primary goal is to extract features of non-standard nature, the ability of neural networks to learn is used here. The system automatically classifies the recognized features. This method is not able to work with parts containing curved faces and surfaces.

Han et al [11] proposed a method to integrate AFR with process planning in an effort to link these two systems. Firstly, the AFR system generates a list of machinable features in the given part based on the previous knowledge of machine capabilities. At the same time, dependencies between features is also established and recorded. Then an optimization algorithm is used to generate the best machining sequence based on the given constraints. There are still some constraints which if considered would yield a much better real world implementable machining sequence.

Historically many CAD file formats have been used to implement AFR. Such an attempt was made by Ahmad et al [12] using DXF (Data Interchange Format) file format. This format was chosen owing to its availability across different commercial CAD systems.

As most of the other attempt at AFR utilize B-rep based CAD part models, Huang et al [13] proposed a technique which accomplishes feature recognition by using sub-graph isomorphic techniques. The basic assumption that they make in order to move forward with their proposed method is that there some features are referred to as high-level features (Figure 2.2 below) (A collection of standard features which are combined in a manner specified by the user). Then the extracted features are arranged in a graph to emphasize the relationship amongst them.

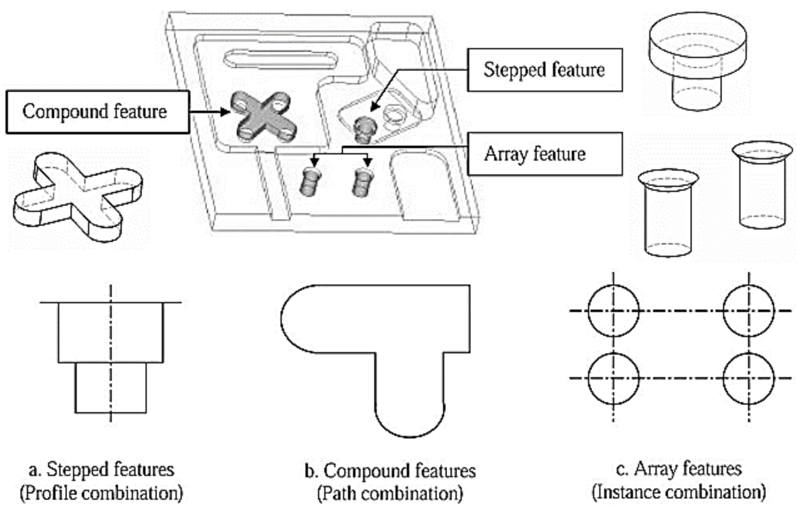


Figure 2.2 High Level Features [14]

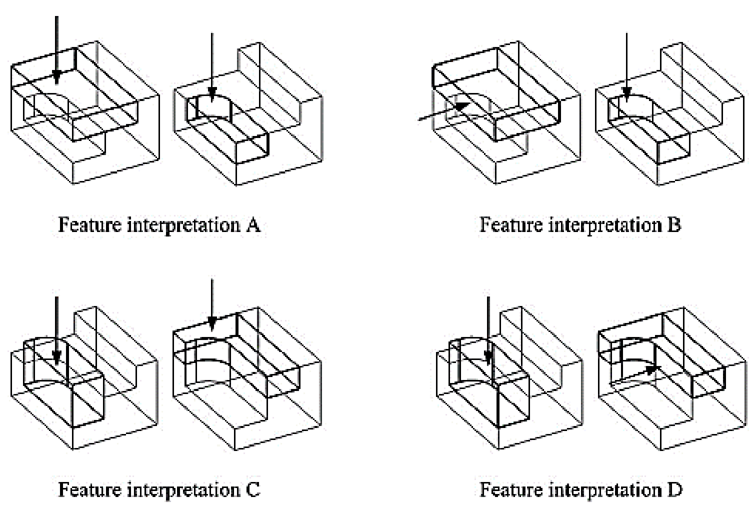


Figure 2.3 Different ways to interpret the same features [15]

The main advantage behind this technique is that it is very efficient time wise and in the sense of computation as well as it only looks for features which are specified by the user specific their requirements.

A method to tackle the issues of lack of scalability and multiple interpretations of features was taken head on by Woo et al [15]. They proposed that a delta volume be decomposed into maximal features. They then fed the maximal features to process planning phase. They solved the problem of scalability by using the approach of divide and conquer when the delta volume is being decomposed. They concluded that their technique provided satisfactory AFT results in reasonable amount of time.

Parts sometimes contain features which need more than 2 axes to be defined properly (Figure 2.3 above). One such classification is known as 2.5D (Figure 2.4 below). In order to manufacture these parts, machines having more than 2 axes are required. In order to apply AFR on such features in parts, Sundararajan et al [16] used a volume decomposition technique based on recursive descent. This results in a graph containing features with spatial relationships of the features as well as geometry. Their proposed method also works on parts with freeform features (Figure 2.4 below).

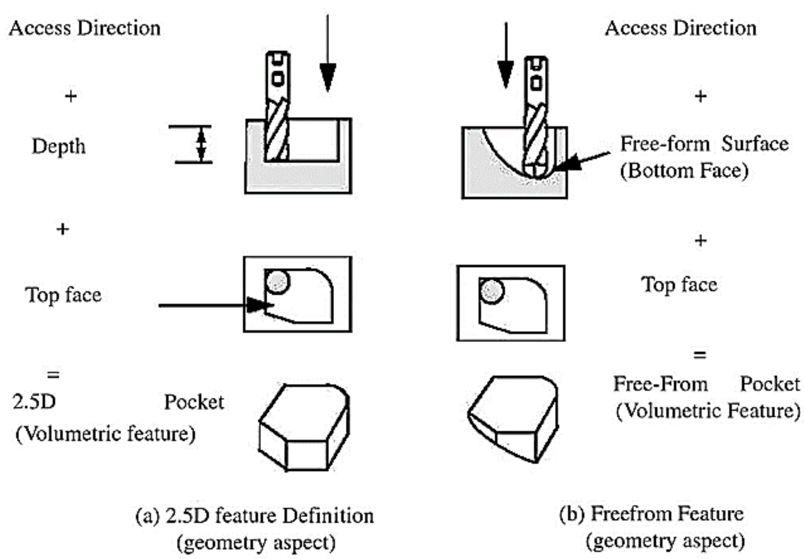


Figure 2.4 Free Form and 2.5D Features [16]

Intersecting features are the ones which merge with one another across their boundaries to from composite features. These are often complicated to detect and classify accordingly on the basis of the original feature's simple shapes without intersection which combine to make composite features during machining operations. Because of this, a certain precedence relationship is formed between simple features. Lee et al [17] proposed a method to address intersecting features and facilitate AFR in such case. The underlying concept used in the method is a combination of breadth first graph search and topological sorting. Orthographic projection is the concept on which AFR is based upon in this case.

Computer Aided Process Planning (CAPP) is another related area in which AFR is of very real significance. In order to link CAPP with CAD/CAM systems, Rameshbabu et al [18] formulated a hybrid technique for AFR. They used combination of face adjacency graph and volume decomposition on STEP AP-203 based CAD part file. Their methodology is capable of recognizing features which intersect boundaries of other features. They then classify the identified features on the basis of operational precedence. Then the machine setup is generated keeping in view of the constraints applicable. One notable drawback of the proposed method is that it assumes that only one machine will be used to make all the features and cannot take into account operations on multiple machines for the same part.

Garcia et al [19] takes a different approach to the integration of CAD/CAM and CAPP systems. They developed a methodology to create preprocess plans to a Holonic Manufacturing System (HMS). The preprocess plans provide the information early in the planning stage about the capabilities of the machines required to produce the desired part. Their AFR system is capable of providing type, shape, material and tolerance of the features. This method offers flexibility promised by HMS as well as a very efficient way of implementing the said flexibility. To accomplish this, a new file format was developed called Pro-FMAextended which rectifies the shortcoming (Lack of feature-based models) of the conventional and modern data models such as VRML and STEP.

Miao at al [20] realized the issue that there is very little to nonexistent communication between CAD packages and their CAM/NC (Numerical Control) modules. These are not feature based model which could be used downstream in applications like process planning. They divided automatic process planning into sub activities like AFR and process planning based on the features extracted. STEP file format is used to export the model from CAD software to AFR software. Then a process plan is generated incorporating selection of tools, sequence of operations and setup planning on the basis of important attributes of features like location, shape and TAD (Tool Approach Direction). Then the process plan is exported into a commercial CAM system to generate the tool path.

2.2 Process Planning

As the part evolves, the machine and tool requirements change. RPP is the way to deal with those changes. This in turn causes the previously selected process plan to change. To combat this challenge, ElMaraghy[3] developed a mapping machine which had two way functionality to exchange data between machine tools and product features.

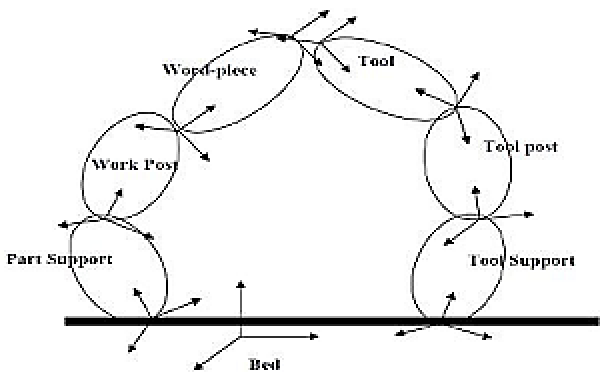


Figure 2.5 Basic structure of machine[21]

A new approach to represent basic structure of machine (Figure 2.5) was developed by Elmaraghy [3]. This format resembled a kinematic chain which captures multiple important factors; motion of axes of tool (order, number, type) and the possibility of different feature manufacturing. The clustering in this approach is based on precedence relations between operations and logical constraints. Afterwards optimization with application of constraints provides feasible process plans from which best is picked.

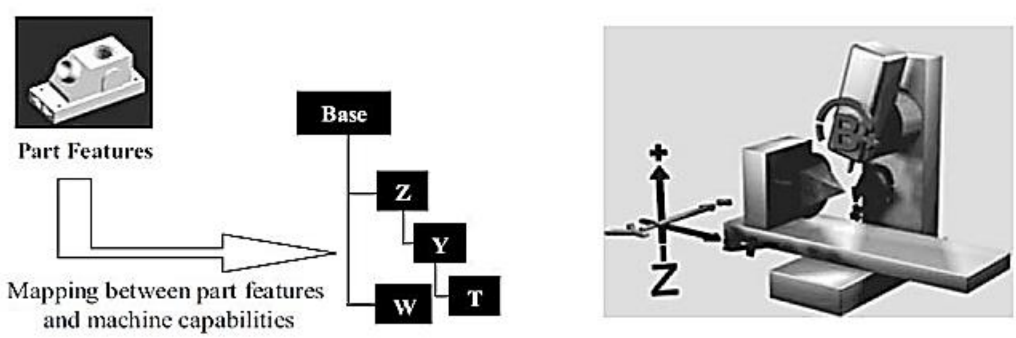


Figure 2.6 Two way mapping and target machine[3]

To implement re-planning, a two-way scheme (Figure 2.6 above) of mapping between capabilities of the machine and part features was developed by Elmaraghy.

Shabaka et al [22] proposed an approach to generate the most suitable machine by implementing the concept of coevolution. This approach required the following inputs; TAD, precedence graph, tolerance datum constraints, dimensions of part and technological constraints (Figure 2.7 below). The main criterion for generation of machine configurations is TAD in this approach. The approach was applied to a standard part as follows.

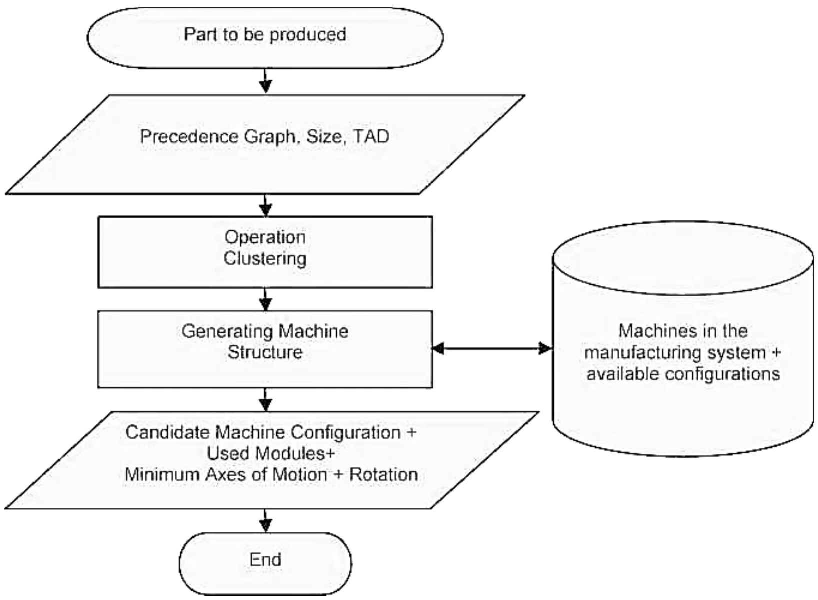
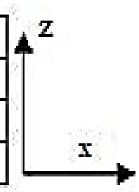


Figure 2.7 Algorithm for generation of machine configuration[22]

Table 2.1 TAD

Op.	Tool Approach Directions (TADs)					
	+x	-x	+y	-y	+z	-z
1	0	0	0	0	1	1
2	0	0	0	0	0	1



In Shabaka's approach, for the part ANC 090 (Figure 2.8 below), minimum required machine capabilities are generated. This is a generic approach; thus it can be used to generate minimum machine capabilities for a whole family of parts in a RMT. One of the limitation of this approach is the necessity to deal with machines which are fixed.

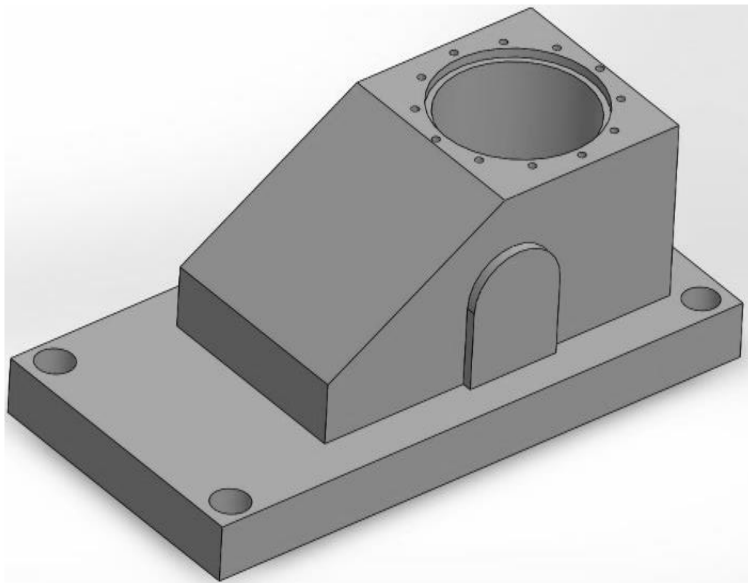


Figure 2.8 Part ANC 090[23]

2.2.1 Process Plan Generation

There are multiple constraints to be considered in the real world when generating prospective process plans. Baqai[1] implemented a methodology to generate process plans while simultaneously tackling the following types of constraints.

- Datum Constraints
- Precedence Constraints
- Technological Constraints
- Geometrical Constraints

Datum constraints are the ones which arise because of holding orientation of a part while it is being machined. Datum is used as a reference.

Precedence constraints dictate which operations come at what stage in the machining process. This is to provide a flow of operations in advance.

Technological constraints which specific operations/features are to be machine before or after certain others.

Geometrical constraints define if a feature created during machining can be used as a reference datum. Those features and hence operations are subjected to geometrical constraints.

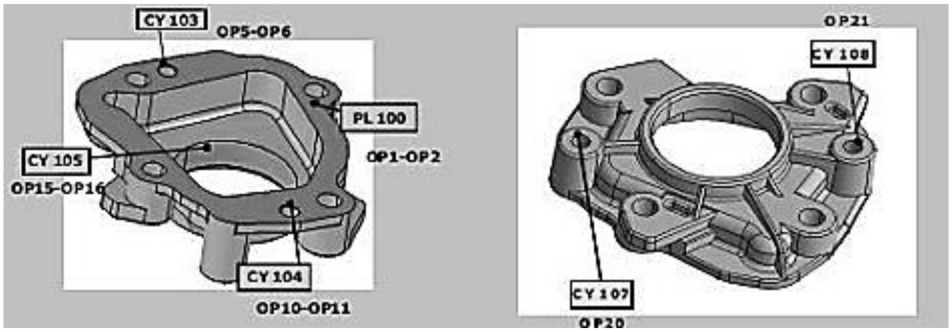


Figure 2.9 Standard part CAI[1]

Baqai[1] used a standard part CAI (Figure 2.9) to generate a precedence chart. Precedence chart shows which operations are interdependent on others. If an operation is dependent on another, there are two possibilities; either it comes before or it comes after the other operation. Precedence chart shows these relationships for every operation (Table 2.2 below).

Table 2.2 Precedence chart - CAI[22]

Operation List

OP5 follows OP1

Features

		PL100	PL100	CY103	CY103	CY104	CY104	CY105	CY105	CY107	CY108
		1	2	5	6	10	11	15	16	20	21
PL100	1	0	1	1	1	1	1	0	0	0	0
PL100	2	-1	0	1	1	1	1	0	0	0	0
CY103	5	-1	-1	0	1	0	0	0	0	0	0
CY103	6	-1	-1	-1	0	0	0	0	0	0	0
CY104	10	-1	-1	0	0	0	1	0	0	0	0
CY104	11	-1	-1	0	0	-1	0	0	0	0	0
CY105	15	0	0	0	0	0	0	0	1	0	0
CY105	16	0	0	0	0	0	0	-1	0	0	0
CY107	20	0	0	0	0	0	0	0	0	0	0
CY108	21	0	0	0	0	0	0	0	0	0	0

OP20 no interaction with OP1

OP1 precedes OP11

After the generation of precedence graph, Baqai[1] grouped the operations on the basis of following

- Similarity of feature TAD
- Rank of operation
- Similarity of type of operation
- Similarity in direction of spindle

An algorithm was applied to group the operations on the basis of the above mention criterion and the following result was obtained (Figure 2.10 below).

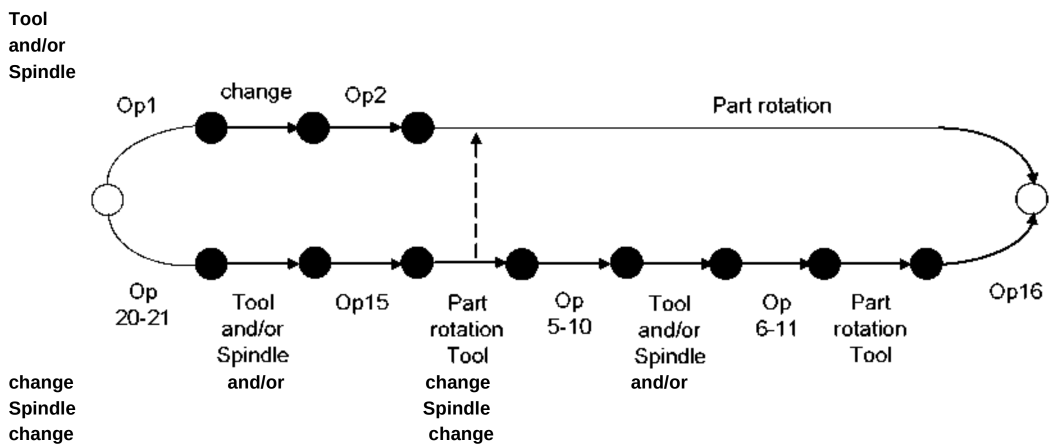


Figure 2.10 Generated process plan[1]

The algorithms and approaches discussed in this section were dealing with the aspect of process planning. As in most realistic scenarios, there can coexist multiple feasible process plans which can be used to machine the part. There exists a need to optimize the feasible process plans while remaining compliant to subjected constraints. The following section discusses the selected tool for optimization of all the feasible process plans.

2.3 Optimization

The stage of process plan generation provides a number of feasible process plans. Then comes the stage when a process plan suiting the needs of manufacturing as well as being efficient (Time and cost) must be selected. This stage is known as optimization and is basically the picking of the best in a bunch, based on some predefined criterion subject to some constraints.

2.3.1 Genetic Algorithm (Optimization Technique)

Genetic algorithm is a non-linear optimization algorithm (Figure 2.11 below) consisting of the following stages.

Initialization of population

The process plans generated which are deemed feasible are the initial population. Other parameters such as stopping criteria, size of population are also input at this stage.

Evaluation of Fitness

The optimization scheme of GA is based on feeding the data through a fitness function. This function provides a fitness score.

Selection

On the basis of fitness scores generated from the fitness function, the fittest individuals are selected. These individuals are the parents for the next generation as these possess the most desired genes (Characteristics). In the next generation, these selections are most likely to produce off springs that are even better than their parents.

Recombination/Crossover

The operator of crossover is one of the mechanisms GA uses to provide better chances of getting closer to the best solution in the next generation. In this operation, the genes (Qualities) are exchanged between parents at crossover points in accordance with one or more of many developed crossover mechanisms (Figure 2.12 below).

Mutation

This operator in GA is inspired by natural selection. Mutation is the source of new characteristics. New does not necessarily mean desired. If same pool of characteristics is chosen from generation over generation, then the results are bound to get stagnant. To ensure diversity, mutation introduces new genes (Characteristics) into the population. This increases the chances of getting to the most optimum solution (Figure 2.13 below).

Stopping Criteria

There are two criteria defined before the initiation of GA. These ensure that the algorithm does not continue running in a fruitless manner. First criterion; generation limit defines the total number generations the GA would run for. Second criterion is implemented to stop GA if the fitness value of the population does not appreciably increase over a certain number of generations. If either of these criteria is met, GA terminates.

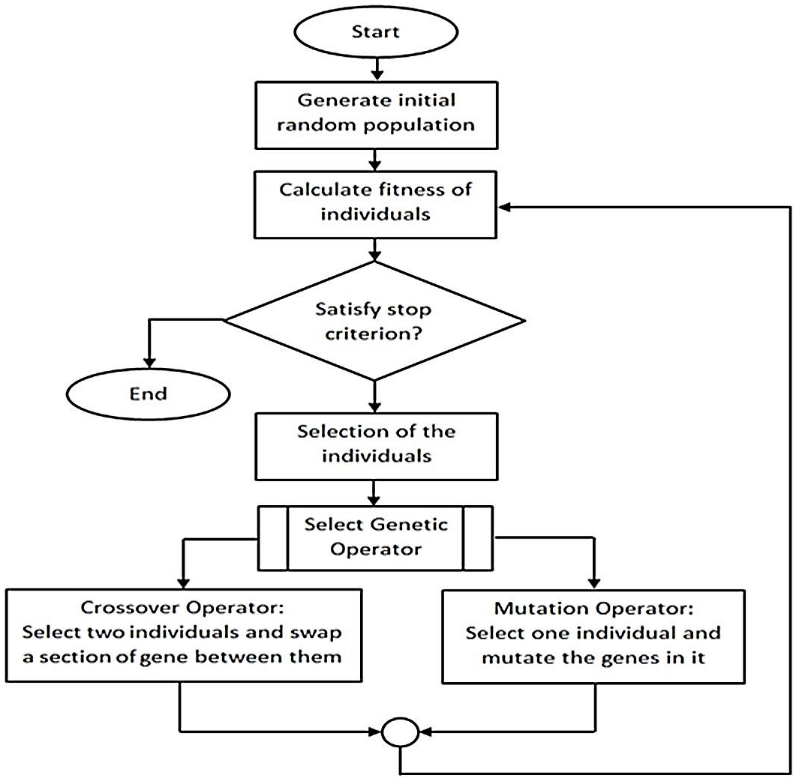
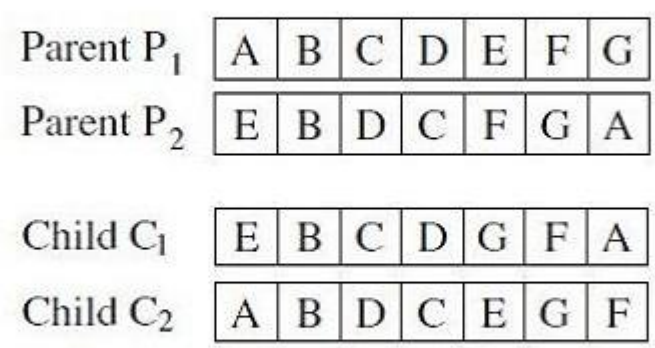


Figure 2.11 Genetic Algorithm Flow Chart



Parent P_1	A	B	C	D	E	F	G
Parent P_2	E	B	D	C	F	G	A
Child C_1	E	B	C	D	G	F	A
Child C_2	A	B	D	C	E	G	F

Figure 2.12 Crossover (uniform)

Child C_1	E	B	**F**	D	G	**C**	A
Child C_2	A	**E**	D	C	**B**	G	F

Figure 2.13 Mutation

2.4 Summary

I can be inferred after going through the literature review that the gaps identified at the end of Chapter 1 have not been addressed before is research work. The following approach will be undertaken to bridge the gaps found

A framework of automated optimization of process plans will be developed. This framework will have two main parts, Feature extraction and optimization of all the process plans possible for machining those features. The communication between the two parts will be via Excel® and a standard format for entering data into excel will be designed so that the process afterwards can be designed to be totally seamless.

The expected outcomes from the framework will be as follows

- The overall process behind the framework will be as streamlined as possible to reduce the dependence on end-user
- The GA tool used for optimization needs to be optimized to provide results in as few generations as possible to reduce cost of computation
- The crossover and mutation operators will be designed as such to completely remove the possibility of a non-constraint compliant process plan to be declared feasible
- A GUI to for the operator to make it easy to load the Excel® file and run the optimization application

Chapter 3

3 Proposed Methodology – The Framework

The framework will be implemented in stages as illustrated in Figure 3.1. Each stage will be explained in the following pages.

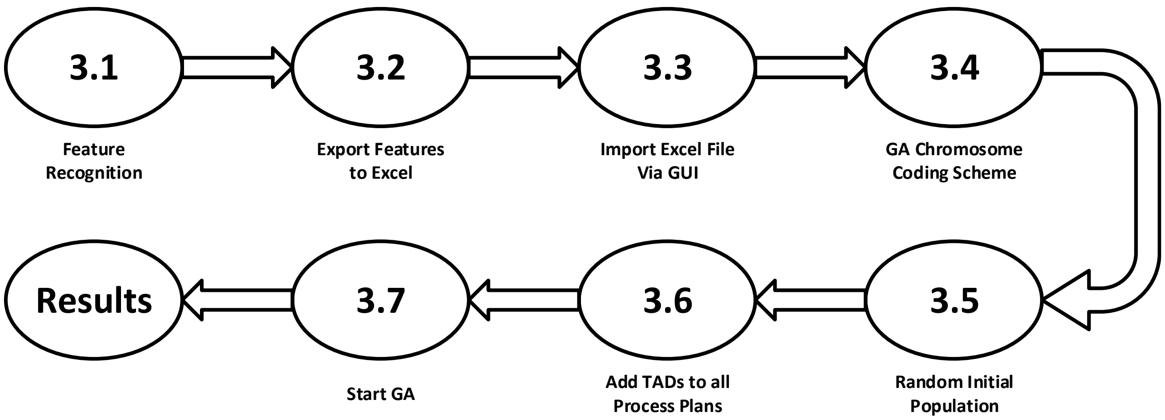


Figure 3.1 Proposed Process Flow

3.1 Feature Recognition in CAD

Feature recognition is carried out in commercial CAD software (SW) SOLIDWORKS™. The software accepts standard CAD formats used in the industry like STEP, STL to name a few. It also accepts proprietary formats from other CAD packages (Pro-Engineer®, CREO®, AutoCAD®) with limited success. As SW is primarily a CAD tool, it requires third party add-on to gain the ability to perform feature recognition. A commercially available software (CW) CAMWorks® was used in this regard. The CW add-on runs inside the SW environment (Figure 3.2 below) and integrates in a seamless manner.

3.1.1 Importing the Part into CAD environment

The part to be used is imported into the CAD environment (ANC090[1] in this case). The CW package is initialized and “Extract Machinable Features” is selected. It takes a few moments for the CW package to detect the features in the part in accordance with the complexity of the part. Once the detection is finished, CW shows the features detected in a list (Figure 3.3 below)

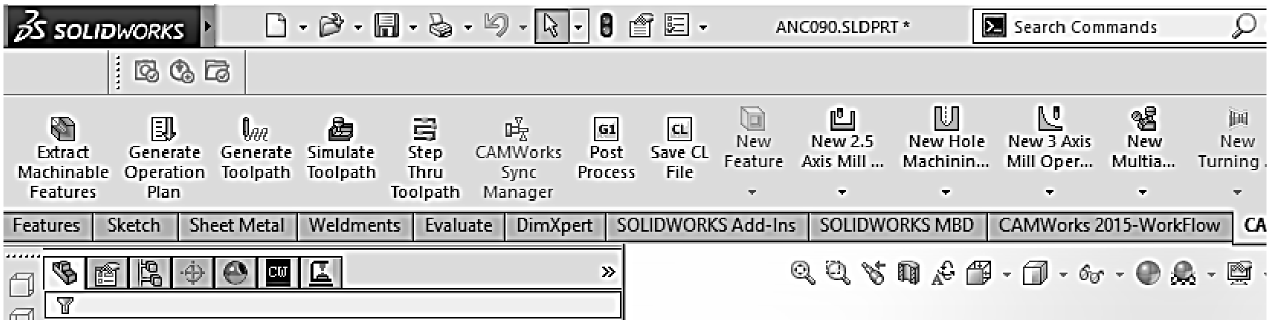


Figure 3.2 CAMWorks® Inside SOLIDWORKSTM Environment

3.1.2 Recommended Tools for each operation

The CW Software also suggests tools to be used for machining each feature (Figure 3.4 below). These tools are selected from a database of standard machining tools built into the CW software. In case custom tools are to be used, CW also supports addition of custom tools to the library to be used exclusively wherever applicable.

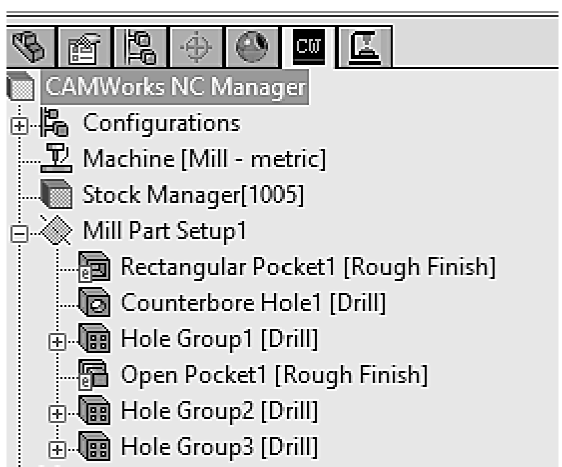


Figure 3.3 List of Recognized Features

3.2 Manual Export of Features & Tool Data to Excel

The user has to manually enter the features data into an excel file in a predefined format. This data includes

- Features
- Number of operations per feature
- Relationship between features (Table 3.1 below). The list of types of relationships is not exhaustive
- Tools data (Table 3.3 below)
- TAD data (Table 3.5 below)
- Machine data (Table 3.2 below)

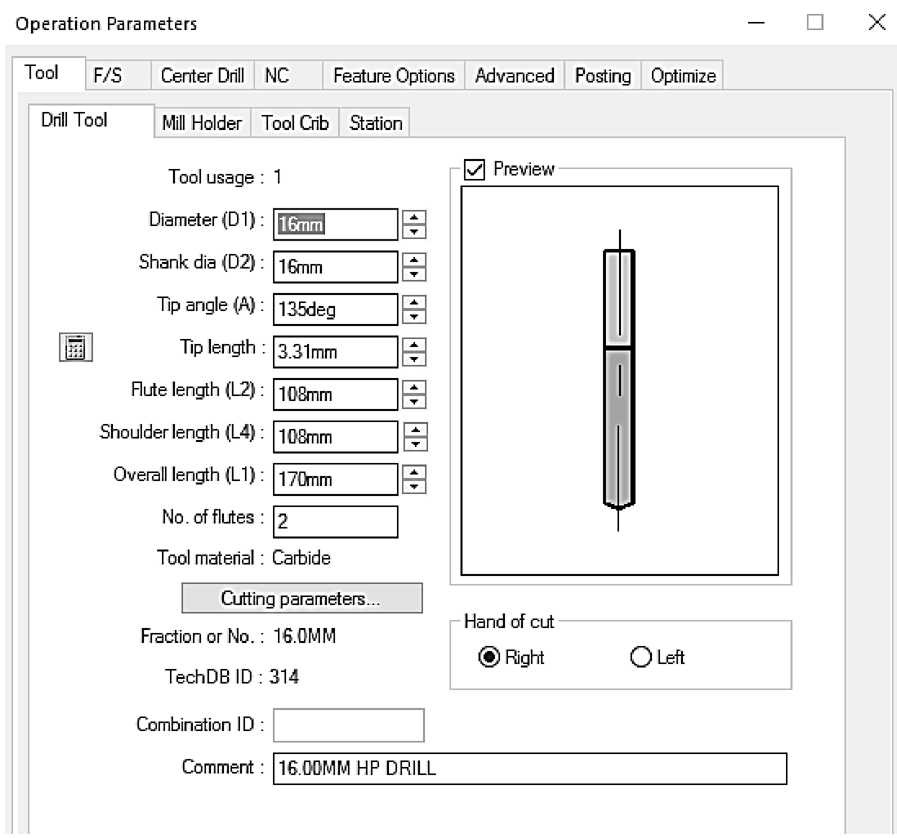


Figure 3.4 Suggested Tool for Drilling Operation

Table 3.1 Topological Interaction Table

	A	B	C	D	E	F	G	H	I	J	K	L
1	No. Of Ops	Name	PL100	PL101	PL102	PL103	CY100	CY101	CY102	CY103	CY104	CY105
2	2	PL100	0	0	0	0	0	0	0	0	4	4
3	2	PL101	0	0	0	0	4	4	4	4	0	0
4	2	PL102	0	0	0	0	0	0	0	0	0	0
5	2	PL103	0	0	0	0	0	0	0	0	0	0
6	2	CY100	0	11	0	0	0	0	0	0	0	0
7	2	CY101	0	11	0	0	0	0	0	0	0	0
8	2	CY102	0	11	0	0	0	0	0	0	0	0
9	2	CY103	0	11	0	0	0	0	0	0	0	0
10	2	CY104	11	0	0	0	0	0	0	0	0	0
11	2	CY105	11	0	0	0	0	0	0	0	0	0
12												
13				0 = No Interaction								
14				1 = Starts on								
15				2 = Starts in								
16				3 = Starts Coaxial								
17				4 = Pierce On								
18				5 = Pierce In								
19				6 = Pierce Coaxial								
20				7 = Cut Through								
21				8 = Tangent								
22				9 = Secant With (for holes)								
23				10 = Not possible at the same time								
24				11 = Secant with (for planes with holes)								

3.3 Import the Excel File into the Optimization Algorithm

A GUI has been designed to aid the user in importing the data file into the MATLAB® program (Figure 3.5 below). The GUI has been designed to be as simple as possible, providing only the most relevant controls to avoid overwhelming the operator. Everything leading after the press of "Execute!" button is fully automated and generic.

The GUI allows the operator to assign maximum number of generations allowed for the optimization algorithm. The size of initial population is also user selectable. The weights (importance) assigned to minimize part rotation and tool change can also be configured here. A built in safety interlock has been designed which checks if the values assigned to weights are feasible or not (Figure 3.5 below). The GUI contains the default values of variables which can be modified if desired.

Table 3.2 Machine Data

Machine Type	**TADs**					
	Neg X	**Pos X**	**Neg Y**	**Pos Y**	**Neg Z**	**Pos Z**
5 Axis	1	1	1	1	1	0

Table 3.3 Tools Data

Type	Name		Tool ID
Plane	PL100	1	1
		2	2
Plane	PL101	3	1
		4	2
Plane	PL102	5	3
		6	4
Plane	PL103	7	3
		8	4
~~Hole~~	~~CY100~~	9	1
		10	2
Hole	CY101	11	1
		12	2
Hole	CY102	13	1
		14	2
Hole	CY103	15	1
		16	2
Hole	CY104	17	5
		18	6
Hole	CY105	19	7
		20	8

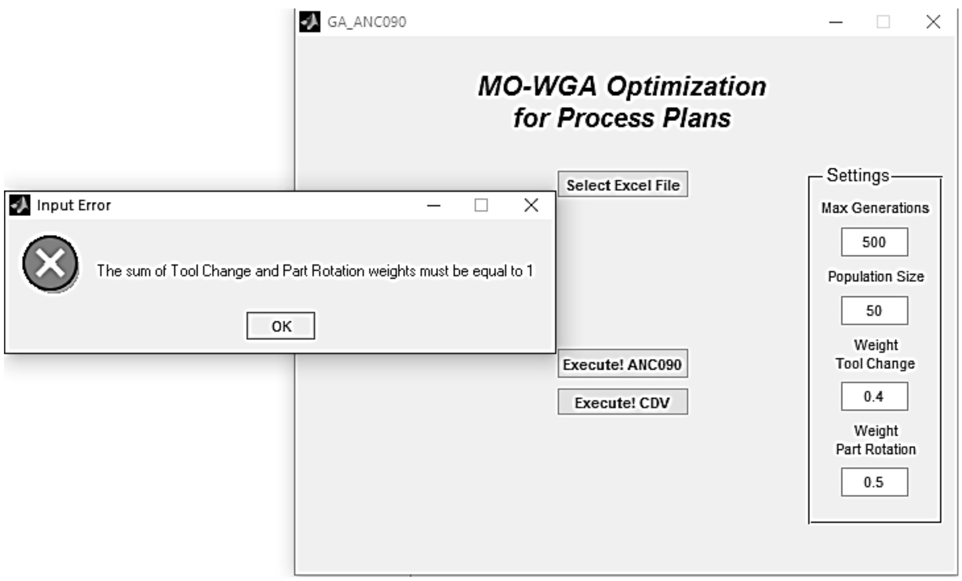


Figure 3.5 GUI for the Optimization Program

3.4 Coding Scheme for Individual Process Plans

The coding scheme used for representing a single PP is shown in Table 3.4 below. The columns in the PP are divided equally into operational sequence and TADs (Highlighted grey). The operational sequence part is depicting the sequence of individual operations in the PP. The sequence varies from one PP to another. In contrast, the sequence of TADs remains always the same in the orange region. The sequence of operations in the grey region remains same regardless of the position of the operation in the yellow part of the PP.

Table 3.4 Process Plan Coding Scheme

Process Plan																											
Operations Sequence														TAD assigned to each operation													
														OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14
1	2	3	4	5	12	7	9	6	8	13	10	11	14	4	5	4	4	5	5	6	5	5	6	5	4	4	4

3.5 Selection of Random Initial Population

The size of initial population is selected by the operator via the GUI (Figure 3.5 above). The Initial population is generated randomly from the pool of all possible PPs. The selection PPs in the initial population have to abide by the precedence constraints set forth by the interaction of features with each other. Certain operations need to be performed before others. There can be multiple sets of precedence constraints for each part. All the constraints have to be followed for a PP to be declared feasible. The constraints are checked from the precedence matrix (Annex A & B) which is generated automatically from the topological interaction table (Table 3.1 above).

In this case, the size of initial population is set at 50 APPs. The algorithm is designed to generate 50,000 APPs at once. All of the APPs are checked for precedence constraint compliance. When the required number (50) of APPs have been generated, next step is started.

3.6 Addition of TADs to the APPs in Initial Population

Every feature and subsequently operation has a set of possible TADs using one of which the feature can be machined. The list of TADs for each feature is read from the input data file (Table 3.5 below). The list specifies the possible TADs for each operation by assigning “1” to the possible axes and “0” to the axes not available. A TAD of ‘-Z’ means that the feature is open to be accessed from ‘+Z’ axis and the tool can approach via ‘-Z’ translation.

During the initial population generation, only the yellow part of the PP (Table 3.4 above) is generated. For each operation, a random TAD is picked from the possible TADs and assigned in the orange part of the PP (Table 3.4 above). This process is repeated for each operation and all of the PPs in the initial population. This method ensures that all of the TADs assigned to operations are in accordance with the provided data file and do not violate any constraints.

Table 3.5 TAD Data

Type	Name		TAD					
			Neg X	Pos X	Neg Y	Pos Y	Neg Z	Pos Z
Plane	PL100	1	0	1	0	1	1	0
		2	0	1	0	1	1	0
Plane	PL101	3	0	0	0	0	1	0
		4	0	0	0	0	1	0
Plane	PL102	5	0	0	1	0	1	0
		6	0	0	1	0	1	0
Plane	PL103	7	0	0	0	1	1	0
		8	0	0	0	1	1	0
Hole	CY100	9	0	0	0	0	1	1
		10	0	0	0	0	1	1
Hole	CY101	11	0	0	0	0	1	1
		12	0	0	0	0	1	1
Hole	CY102	13	0	0	0	0	1	1
		14	0	0	0	0	1	1
Hole	CY103	15	0	0	0	0	1	1
		16	0	0	0	0	1	1
Hole	CY104	17	0	0	0	0	1	0
		18	0	0	0	0	1	0
Hole	CY105	19	0	0	0	0	1	0
		20	0	0	0	0	1	0

3.7 Initial Population fed to the GA

The initial population with the addition of TADs to each operation is passed onto the GA.

3.7.1 Objective – Minimize Fitness Function

The fitness function is a means of checking the viability of a PP in comparison with the other APPs. In this instance, the objective of the GA is to minimize the fitness function (Equation 1 below).

$$Obj_{fn} = \text{Minimize} \sum_{i=1}^{number\ of\ APPs} (W_{pr} \times Rot_{ip} + W_{tc} \times Chg_{it} + V_{if})$$

Equation 1 Fitness Function

W_{pr} = Weight assigned to part rotation

W_{tc} = Weight assigned to tool change

Rot_{ip} = Part rotation count for each APP

Chg_{it} = Tool change count for each APP

V_{if} = Violation factor for each APP

3.7.2 GA Operator – Crossover

Crossover is one of the two GA operators used to generate off springs for the next generation. Two parents are selected to generate two off springs which share some data from each of the two parents. This creates a possibility to achieve a better fitness value for the off springs as they share traits from each of the two parents. The base of the crossover technique was developed in 2013 [24]. This research builds up on the said process.

1. Two parents are selected from the available current generation population based on fitness (Figure 3.6 below).

Process Plan																											
Operations Sequence														TAD assigned to each operation													
														OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14
1	2	3	4	5	7	11	6	14	12	8	9	10	13	4	2	5	4	6	6	6	6	5	6	5	4	4	4
1	2	3	4	5	7	11	8	13	9	6	10	14	12	4	4	5	5	6	5	5	5	6	5	5	4	4	4

Figure 3.6 Selected Parents

2. The operational sequence portion of both parents is compared from first operation for same operations. The number of same operations is checked each time. These operations are excluded from the crossover. In the Figure 3.7 below, the selected operations for crossover are enclosed in the black rectangle.

3. The selected operations are crossed over into two off springs. The novel part of this approach is that the TADs related to the swapped operations are also swapped (Figure 3.8 below). This increases the chances of finding a tangibly fit offspring as compared to the parents. The crossover operation is carried out predefined number of times per generation. This parameter can be modified as required.

Process Plan																											
Operations Sequence														TAD assigned to each operation													
														OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14
1	2	3	4	5	7	11	6	14	12	8	9	10	13	4	2	5	4	6	6	6	6	5	6	5	4	4	4
1	2	3	4	5	7	11	8	13	9	6	10	14	12	4	4	5	5	6	5	5	5	6	5	5	4	4	4

Figure 3.7 Crossover area selected

Process Plan																											
Operations Sequence														TAD assigned to each operation													
														OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14
1	2	3	4	5	7	11	6	14	12	8	9	10	13	4	2	5	4	6	6	6	6	5	6	5	4	4	4
1	2	3	4	5	7	11	8	13	9	6	10	14	12	4	4	5	5	6	5	5	5	6	5	5	4	4	4
1	2	3	4	5	7	11	8	13	9	6	10	14	12	4	2	5	4	6	5	6	5	6	5	5	4	4	4
1	2	3	4	5	7	11	6	14	12	8	9	10	13	4	4	5	5	6	6	5	6	5	6	5	4	4	4

Figure 3.8 Crossover Completed

(Off springs below the parents)

3.7.3 GA Operator – Mutation

Mutation operator is also based upon the technique developed in 2013 [24].

1. Mutation rate is preselected in the algorithm. It dictates how often does mutation actually happen. In the current case, this has been set to 0.3 or 30% as the default value. This means that randomly 3 out of 10 times, mutation takes place. The mutation rate is a configurable parameter and can be adjusted as required.
2. Mutation sites are chosen at random each time the mutation function is called. The operations at these sites are swapped to ensure that there are no repetitions of operations in the overall operational sequence.

Process Plan																											
Operations Sequence														TAD assigned to each operation													
														OP1	OP2	OP3	OP4	OP5	OP6	OP7	OP8	OP9	OP10	OP11	OP12	OP13	OP14
1	2	3	4	5	12	7	9	6	8	13	10	11	14	4	5	4	4	5	5	6	5	5	6	5	4	4	4
1	2	3	4	5	12	7	8	6	9	13	10	11	14	4	5	4	4	5	5	6	6	5	5	5	4	4	4

Figure 3.9 Mutation Execution

3. After the mutation has been carried out, the PP is checked for precedence constraint compliance. If any constraint is found violated, the PP is assigned a violation factor of 100 which is added to the fitness value of the corresponding PP.

3.7.4 Termination Criteria

Termination of GA can be achieved via either one of the two criteria whichever is fulfilled first

1. Total number of generations reaches the limit set forth in the GUI. The default value is 500.
2. The generational fitness becomes stagnant and does not show a difference of 25 (default value) or more for at least 200 generations. It can be observed in (Figure 3.10 below) that the GA has terminated before 500 generations.

3.7.5 Results Displayed

The result of GA is displaced in the command window of MATLAB® software. The result contains the following information

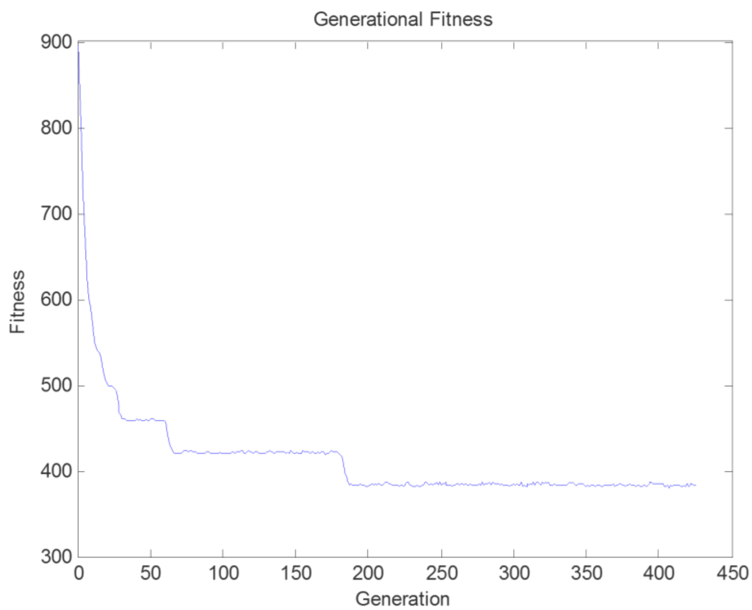


Figure 3.10 Generational FItness Chart

- Fitness / Generations chart (Displayed in a separate window) (Figure 3.10 above). This chart shows the sum of fitness of all individuals on a generation by generation basis. As it can be seen that in the initial generations, the decline in fitness values is exponential which changes the behavior near generation 35. There are improvements in generations 60 and 180 after which the graph becomes fairly straight. As the termination conditions dictate that if there is no appreciable change in combines fitness through 200 generations, the GA terminates at generation 425.

- Fitness value for the selected PP (Figure 3.11 below)
- Tool ID in the first row, corresponding TADs in the second row and selected PP in the third row(Figure 3.11 below)
- Number of part rotations(Figure 3.11 below)
  Each Part rotation denotes the fact that the process plan calls for a TAD that the machine cannot handles without changing the part orientation first
- Number of tool changes(Figure 3.11 below)

  The difference in tool ID between operations denotes different tools being used

```
Row 1 = Tool ID, Row 2  = TAD, Row 3 = Process Sequence

optima =

  Columns 1 through 11

     1     1     2     2     3     5     5     7     4     8     8
     2     5     2     4     5     5     5     5     5     4     4
     1     3     4     2     5     9     7    11     6    13    14

  Columns 12 through 14

     8     6     6
     4     5     5
    12     8    10


part_rotations =

     0


tool_changes =

     7
```

Figure 3.11 Results Displayed in Command Window

Chapter 4

4 Case Studies and Results

4.1 ANC 090 (Part) [23]

The part ANC 090 has been taken from previous works in the area of CAPP[23]. The part itself (drawing) and list of features is attached in Appendix A. The features and related operations information was imported into MATLAB® by selecting the Excel® data file in the GUI.

4.1.1 Data in Excel® file

1. TAD data (Table 4.1 below). This represents which operations are possible from a given TAD. The list specifies the possible TADs for each operation by assigning "1" to the possible axes and "0" to the axes not available. A TAD of '-Z' means that the feature is open to be accessed from '+Z' axis and the tool can approach via '-Z' translation.

Table 4.1 ANC 090 TAD Data

Type	Name	Op #	TAD					
			Neg X	Pos X	Neg Y	Pos Y	Neg	Pos Z
Plane	PL100	1	0	1	0	1	1	0
		2	0	1	0	1	1	0
Plane	PL101	3	0	0	0	0	1	0
		4	0	0	0	0	1	0
Plane	PL102	5	0	0	1	0	1	0
		6	0	0	1	0	1	0
Plane	PL103	7	0	0	0	1	1	0
		8	0	0	0	1	1	0
Hole	CY100	9	0	0	0	0	1	1
		10	0	0	0	0	1	1
Hole	CY101	11	0	0	0	0	1	1
		12	0	0	0	0	1	1
Hole	CY102	13	0	0	0	0	1	1
		14	0	0	0	0	1	1
Hole	CY103	15	0	0	0	0	1	1
		16	0	0	0	0	1	1
Hole	CY104	17	0	0	0	0	1	0
		18	0	0	0	0	1	0
Hole	CY105	19	0	0	0	0	1	0
		20	0	0	0	0	1	0

2. Tools data (Table 4.2 below). The tool ID represents a physical tool. The tool IDs are assigned to physical tools by the operator earlier. These are just a means of representing the tools during the optimization process. All the tool IDs will be converted to actual tools before manufacturing so that appropriate tools may be used.

Table 4.2 ANC 090 Tools Data

Type	Name	OP #	Tool ID
Plane	PL100	1	1
		2	2
Plane	PL101	3	1
		4	2
Plane	PL102	5	3
		6	4
Plane	PL103	7	3
		8	4
Hole	CY100	9	1
		10	2
Hole	CY101	11	1
		12	2
Hole	CY102	13	1
		14	2
Hole	CY103	15	1
		16	2
Hole	CY104	17	5
		18	6
Hole	CY105	19	7
		20	8

3. Machine data (Table 4.3 below). This denotes the capability of a machine to access the work piece from each TAD without reorientation. Usually this metric is represented by DoF. Here the alternate representation has been developed for ease of use of data during the optimization process. The alternate representation goes for actual tool approach directions possible for the machine while keeping the part orientation fixed. The case study considers a 5 DoF machine.

Table 4.3 ANC 090 Machine Capabilities

Machine Type	**TADs**					
	Neg X	**Pos X**	**Neg Y**	**Pos Y**	**Neg Z**	**Pos Z**
5 Axis	1	1	1	1	1	0

4. Topological interaction data (Table 4.4 below). This data defines the interaction of features with other features. This dictates which features and hence operations are required to be done before others. The first column contains information about how many operations exist on a per feature basis. The columns following 3 contain topological data. This data is converted into precedence matrix to extract individual constraints. The process of doing so has been excellently discussed by Baqai[1]. The precedence table represents in a binary way if there is any relationship between two operations. Two or more operations of the same feature also have precedence constraints.

***Table 4.4 ANC 090 Topological Interaction* Data**

No. Of Ops	Name	PL100	PL101	PL102	PL103	CY100	CY101	CY102	CY103	CY104	CY105
2	PL100	0	0	0	0	0	0	0	0	4	4
2	PL101	0	0	0	0	4	4	4	4	0	0
2	PL102	0	0	0	0	0	0	0	0	0	0
2	PL103	0	0	0	0	0	0	0	0	0	0
2	CY100	0	11	0	0	0	0	0	0	0	0
2	CY101	0	11	0	0	0	0	0	0	0	0
2	CY102	0	11	0	0	0	0	0	0	0	0
2	CY103	0	11	0	0	0	0	0	0	0	0
2	CY104	11	0	0	0	0	0	0	0	0	0
2	CY105	11	0	0	0	0	0	0	0	0	0

4.1.2 Results

4.1.2.1 Selected process plan

1. Number of Part Rotations

   0

2. Number of Tool Changes

   9

3. Operational Sequence and TADs (Figure 4.1 below).

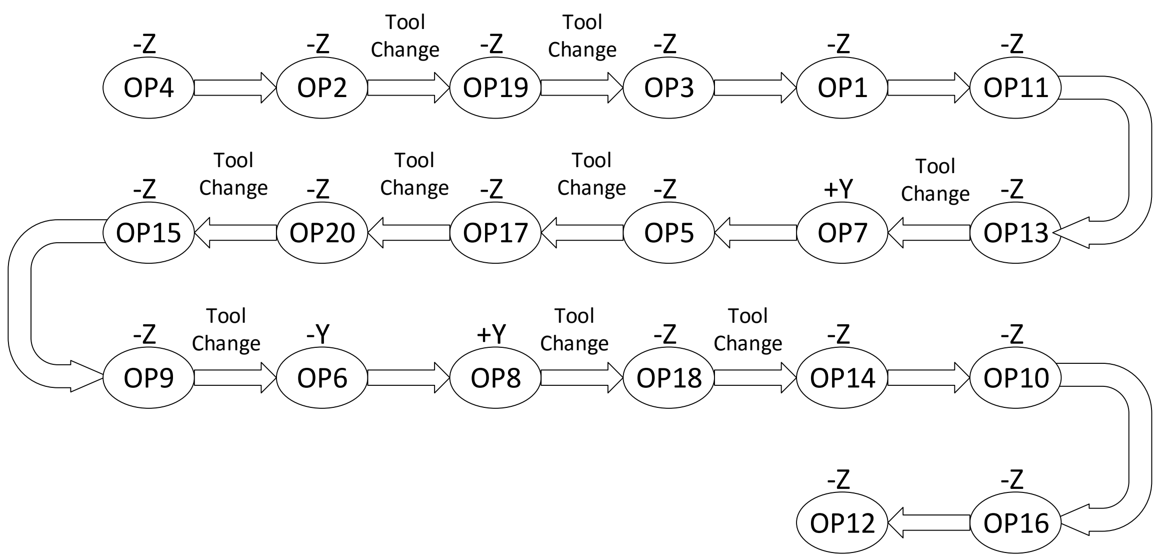


Figure 4.1 ANC 090 Selected PP

4.1.2.2 Generational Fitness

Upon termination of GA, a plot (Figure 4.2 below) is presented with generational fitness of APPs. Here it can be observed an exponential decay in fitness for the first 20 generations. The decay rate considerably slows down afterwards. More major drops in fitness can also be observed at generations 80, 335 and 340. Finally, the GA terminates at the 500th generation.

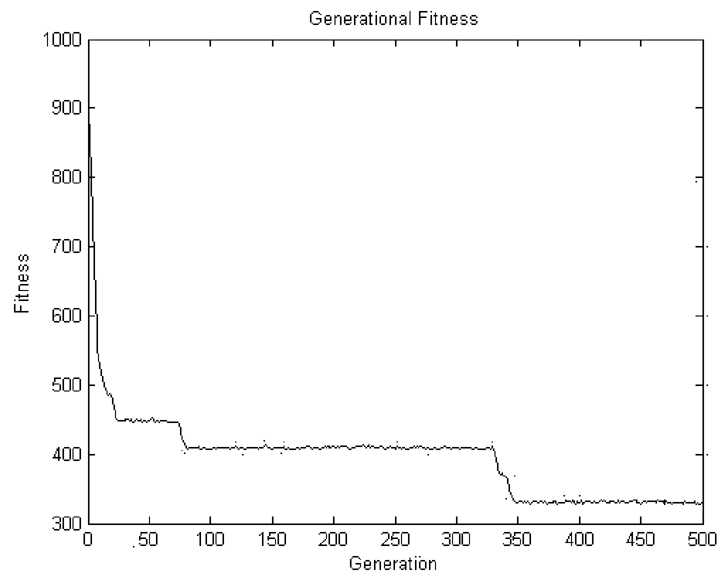


Figure 4.2 ANC 090 Generational Fitness Chart

4.2 CDV (Part) [1]

The part CDV has been taken from previous works in the area of CAPP [1]. The part itself (drawing) and list of features is attached in Appendix B The features and related operations information was imported into MATLAB® by selecting the Excel® data file in the GUI.

4.2.1 Data in Excel® file

1. TAD data (Table 4.5 below). This represents which operations are possible from a given TAD. The list specifies the possible TADs for each operation by assigning "1" to the possible axes and "0" to the axes not available. A TAD of '-Z' means that the feature is open to be accessed from '+Z' axis and the tool can approach via '-Z' translation.

2. Tools data (Table 4.6 below). The Tool ID represents a physical tool. The tool IDs are assigned to physical tools by the operator earlier. These are just a means of representing the tools during the optimization process. All the tool IDs will be converted to actual tools before manufacturing so that appropriate tools may be used.

3. Machine data (Table 4.3 above). This denotes the capability of a machine to access the work piece from each TAD without reorientation. Usually this metric is represented by DoF. Here the alternate representation has been developed for ease of use of data during the optimization

process. The alternate representation goes for actual tool approach directions possible for the machine while keeping the part orientation fixed. The case study considers a 5 DOF machine.

Table 4.5 CDV TAD Data

Type	Name	OP #	TAD					
			Neg X	Pos X	Neg Y	Pos Y	Neg Z	Pos Z
Plane	PL100	1	0	1	0	1	1	0
		2	0	1	0	1	1	0
Plane	PL101	3	0	0	0	1	1	0
		4	0	0	0	1	1	0
Hole	CY102	5	0	0	0	0	1	1
		6	0	0	0	0	1	1
Hole	CY103	7	0	0	0	0	1	1
		8	0	0	0	0	1	1
Hole	CY104	9	0	0	0	0	1	1
		10	0	0	0	0	1	1
Hole	FL106	11	0	0	0	0	1	0
Hole	FL108	12	0	0	0	1	0	0
Hole	FL109	13	0	0	0	1	0	0
Hole	FL110	14	0	0	0	1	0	0

4. Topological interaction data (Table 4.4 above). This data defines the interaction of features with other features. This dictates which features and hence operations are required to be done before others. The first column contains information about how many operations exist on a per feature basis. The columns following 3 contain topological data. This data is converted into precedence matrix to extract individual constraints. The process of doing so has been excellently discussed by Baqai[1]. The precedence table represents in a binary way if there is any relationship between two operations. Two or more operations of the same feature also have precedence constraints.

Table 4.6 CDV Tools Data

Type	Name	OP #	Tool ID
Plane	PL100	1	1
		2	2
Plane	PL101	3	1
		4	2
Hole	CY102	5	3
		6	4
Hole	CY103	7	5
		8	6
Hole	CY104	9	5
		10	6
Hole	FL106	11	7
Hole	FL108	12	8
Hole	FL109	13	8
Hole	FL110	14	8

Table 4.7 CDV Machine Capabilities

Machine Type	**TADs**					
	Neg X	**Pos X**	**Neg Y**	**Pos Y**	**Neg Z**	**Pos Z**
5 Axis	1	1	1	1	1	0

Table 4.8 CDV Topological Interaction Data

No. Of Ops	Name	PL100	PL101	CY102	CY103	CY104	FL106	FL108	FL109	FL110
2	PL100	0	0	0	4	4	4	0	0	0
2	PL101	0	0	0	0	0	0	4	4	4
2	CY102	0	0	0	0	0	0	0	0	0
2	CY103	11	0	0	0	0	0	0	0	0
2	CY104	11	0	0	0	0	0	0	0	0
1	FL106	11	0	0	0	0	0	0	0	0
1	FL108	0	11	0	0	0	0	0	0	0
1	FL109	0	11	0	0	0	0	0	0	0
1	FL110	0	11	0	0	0	0	0	0	0

4.2.2 Results

4.2.2.1 Selected process plan

1. Number of Part Rotations

   0

2. Number of Tool Changes

   7

3. Operational Sequence and TADs (Table 4.5 above).

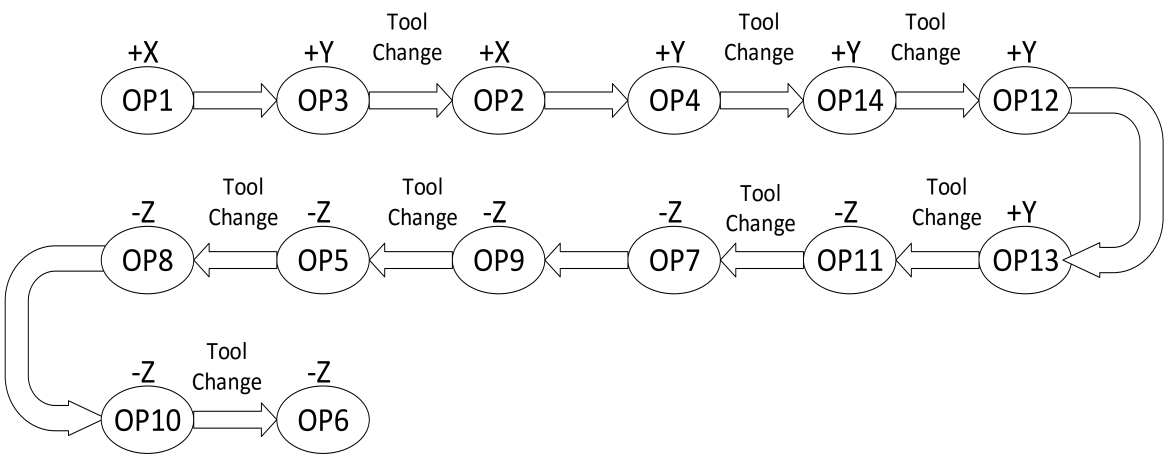


Figure 4.3 CDV Selected PP

4.2.2.2 Generational Fitness

Upon termination of GA, a plot (Figure 4.4 below) is presented with generational fitness of APPs. Here it can be observed an exponential decay in fitness for the first 15 generations. The decay rate considerably slows down afterwards. One more major drop in fitness can also be observed at generation 40. Finally, the GA terminates at the 455th generation as there wasn't tangible improvement in population fitness for 200 generations.

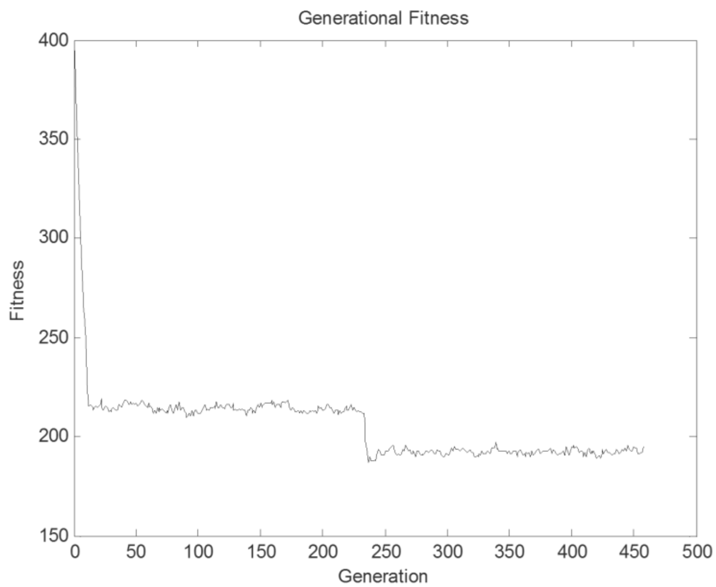


Figure 4.4 CDV Generational Fitness Chart

Chapter 5

5 Conclusion

The objective set forth during the initial phases of this work was to develop a framework for extracting features from a CAD part, feed those features to GA via Excel® for database keeping purposes and the ability to from a unified as well as a standard format. The framework developed has been tested to be fully generic and able to cope with change in part features and complexity over time and iterations.

The communication between CAD environment SOLIDOWORKS™ and Excel® presently does not exist due to technical limitations hence the user is required to manually enter the data between the two software environments. From here onwards, the process is fully automated in every sense. The system once given the input data file, requires no operator intervention at all.

Two novel developments were done during optimization of the GA. A new crossover method was developed as an extension to a previous work. This was done to include TADs in the crossover which leads to convergence faster and ensures diversity. Secondly, crossover operation points are not predetermined in the algorithm. These are decided on a pair by pair basis to achieve maximum diversity. Both of these developments still retain the premise of constraint compliance throughout the process.

In conclusion, the promise of framework envisioned above has been completely delivered.

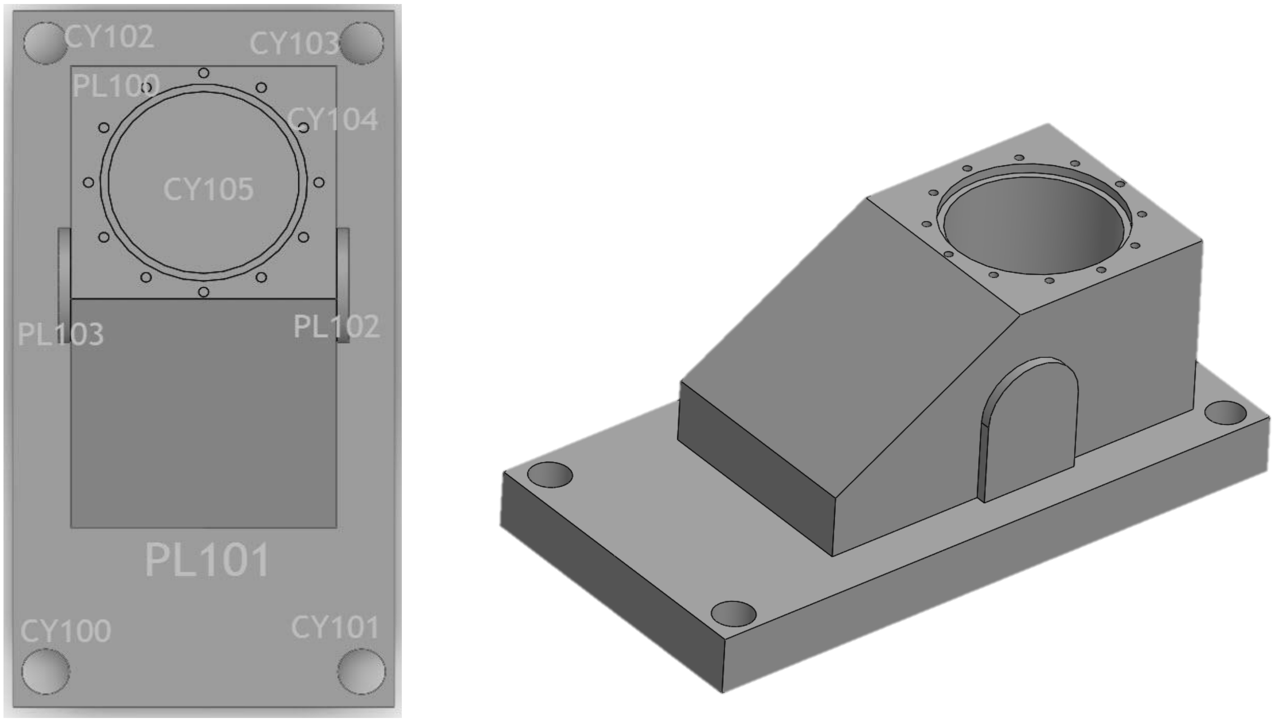
Chapter 6

6 Future Recommendations

In future, following areas of improvement can be considered to advance this work further

- Develop communication between CAD environment and Excel®
- Multiple and parallel setups could be considered
- Other machining factors such as operation time, spindle speed, depth of cut could be considered
- As each change in affinity for tool change and part rotation affects overall process cost, that could be considered for integration
- As each tool change and part rotation costs by increasing the overall process tolerance, these can be fed into the excel file as an additional objective

Appendix A – ANC 090 Part Details

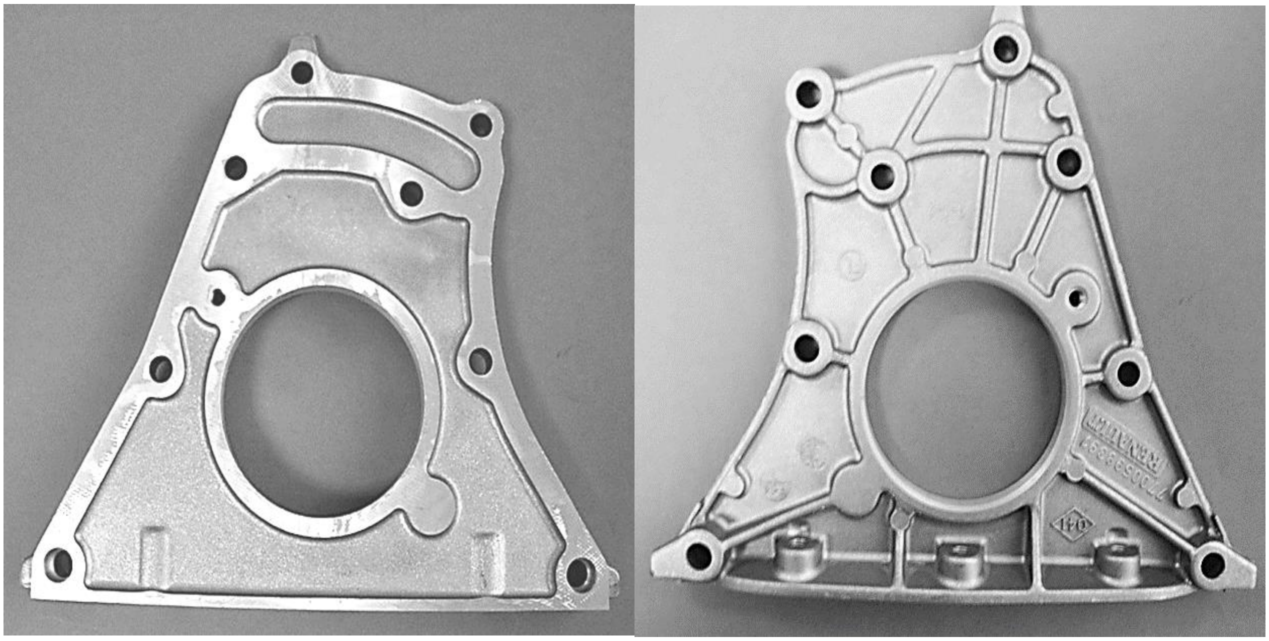


ANC 090

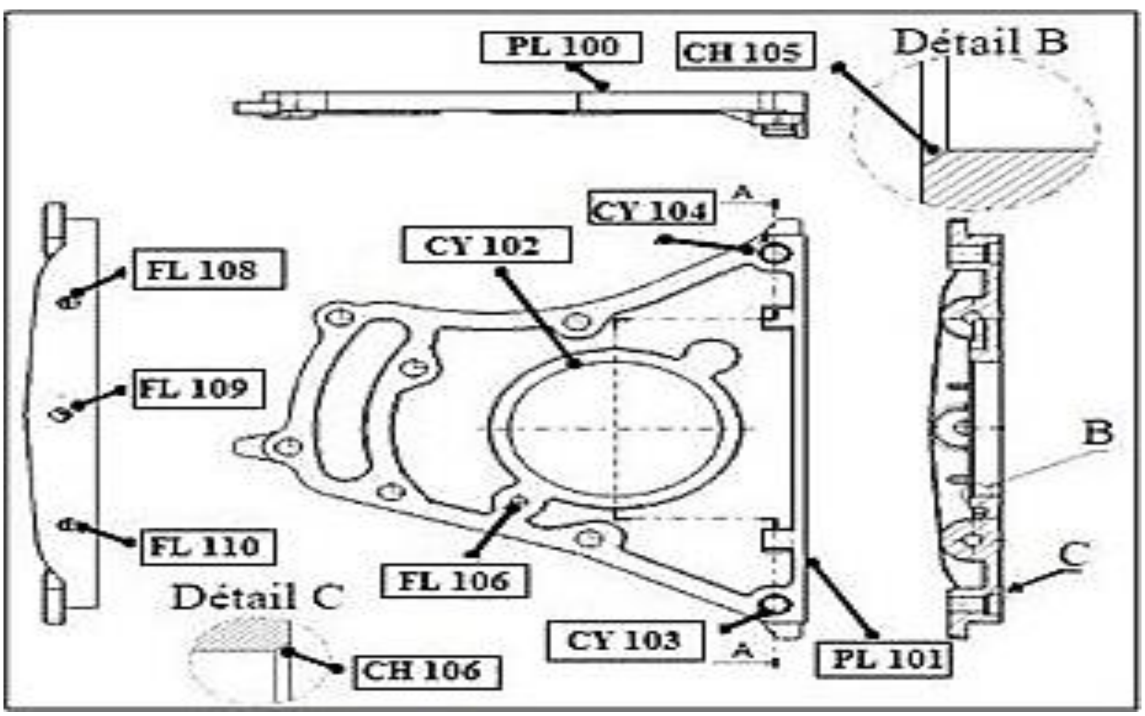
		PL100		PL101		PL102		PL103		CY100		CY101		CY102		CY103		CY104		CY105	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PL100	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
	2	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
PL101	3	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
	4	0	0	-1	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
PL102	5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PL103	7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0
CY100	9	0	0	-1	-1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
	10	0	0	-1	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0
CY101	11	0	0	-1	-1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	12	0	0	-1	-1	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0
CY102	13	0	0	-1	-1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
	14	0	0	-1	-1	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0
CY103	15	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	16	0	0	-1	-1	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0
CY104	17	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
	18	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0
CY105	19	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	20	-1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0

ANC 090 Precedence Matrix

Appendix B – CDV Part Details



CDV



CDV Features

C3:P16		PL100		PL101		CY102		CY103		CY104		FL106	FL108	FL109	FL110
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
PL100	1	**0**	1	0	0	0	0	1	1	1	1	1	0	0	0
	2	-1	**0**	0	0	0	0	1	1	1	1	1	0	0	0
PL101	3	0	0	**0**	1	0	0	0	0	0	0	0	1	1	1
	4	0	0	-1	**0**	0	0	0	0	0	0	0	1	1	1
CY102	5	0	0	0	0	**0**	1	0	0	0	0	0	0	0	0
	6	0	0	0	0	-1	**0**	0	0	0	0	0	0	0	0
CY103	7	-1	-1	0	0	0	0	**0**	1	0	0	0	0	0	0
	8	-1	-1	0	0	0	0	-1	**0**	0	0	0	0	0	0
CY104	9	-1	-1	0	0	0	0	0	0	**0**	1	0	0	0	0
	10	-1	-1	0	0	0	0	0	0	-1	**0**	0	0	0	0
FL106	11	-1	-1	0	0	0	0	0	0	0	0	**0**	0	0	0
FL108	12	0	0	-1	-1	0	0	0	0	0	0	0	**0**	0	0
FL109	13	0	0	-1	-1	0	0	0	0	0	0	0	0	**0**	0
FL110	14	0	0	-1	-1	0	0	0	0	0	0	0	0	0	**0**

CDV Precedence Matrix

Appendix C – Code

Main Code

```
h = findobj('Tag','Gui1');

 if ~isempty(h)
   % get handles and other user-defined data associated to GUI
   g1data = guidata(h);
   pop_size = str2double(get(g1data.edit4,'String'));
   gens = str2double(get(g1data.edit3,'String'));
   w_tc = str2double(get(g1data.edit5,'String'));
   w_pr = str2double(get(g1data.edit6,'String'));
   file = get(g1data.text8,'String');
 else isempty(h)
   clc;
   clear all;
   close all;
   pop_size = 50;
   gens = 500;
   w_tc = 0.5;
   w_pr = 0.5;
   [file,path] = uigetfile('*.xlsx; *.xlsm; *.xls');
 end

get_data;

%%%%%%%Generate Precedence Matrix%%%%%%%%%
curr_op = 1; curropi = 1; curropj = 1;
prec = zeros(n_ops,n_ops);
nfeat = numel(topo);
nfeat = sqrt(nfeat);
for a=1:nfeat
   max=curr_op-1+index(a);
   curr=curr_op;
   for i=curr: max
      for j=1:n_ops
         if i ~= j && j <= max && j >=curr
            if i < max && j > i
               prec (j,i) = -1;
            end
             if i > curr && j < i
               prec (j,i) = 1;
            end
```

```
        end
      end
      curr_op = curr_op+1;
    end
  end

  for b = 1: nfeat
    maxi = curropi - 1+index(b);
    curri = curropi;
    for i = curri : maxi
      curropj = 1;
      for c = 1: nfeat
        maxj = curropj - 1+ index(c);
        currj = curropj;
        for j = currj : maxj
          if (i ~= j && prec(j,i) == 0)
            if topo (c,b) == 11
              prec(j,i) = -1;
            end
            if topo (c,b) == 4
              prec(j,i) = 1;
            end
          end
          curropj = curropj + 1;
        end
      end
      curropi = curropi + 1;
    end
  end



  pp_ftns(pop_size,2,gens)=0;
  gen_ftns(gens,2)=0;
  n_copairs = ceil(0.14*pop_size);
  TADs = n_ops;
  n_mut = ceil(0.08*pop_size);
  r_mut = 0.3;
  v_factor = 100;
  lgen_ftns = 0;
  carry_over =pop_size-(n_copairs*4)-n_mut;
  bit=0;
  mut_space(n_mut*gens,n_ops*2)=0;


  tool_change = zeros(pop_size,2,gens);
  part_rot = zeros(pop_size,2,gens);
```

```
dof_req = zeros(pop_size,2,gens);
pen_ftns = zeros(pop_size,1,gens+1);
pop = zeros(pop_size,n_ops*2,gens);

ppgen; %Generate Initial Population

pop(1:pop_size,1:n_ops,1) = s_space(1:1:pop_size,1:n_ops);



%%%%%%%%%%%%%%%%%%Convert Op TAD to Axes%%%%%%%%%%%
tad_no(n_ops) = 0;
for ind1 =1:n_ops
    cnt1=1;
    for ind2 = 1:6
        if tad_data(ind1,ind2)==1
            tad_no(ind1) = tad_no(ind1) + 1;
            tad_detail(ind1,cnt1)=ind2;
            cnt1=cnt1+1;
        end
    end
end

cgen_pop = pop(:,:,1);
for ind1=1:pop_size
  for ind2=n_ops+1:(2*n_ops)
      temp1=randi(tad_no(ind2-n_ops),1);
     cgen_pop(ind1,ind2)=tad_detail(ind2-n_ops,temp1);
   end
end
pop(:,:,1) = cgen_pop;


%%%%%%%%Convert Machine TAD to Axes%%%%%%%%%%%
cnt1=1;
for ind = 1:6
        if m_data(ind)==1
            m_detail(cnt1)=ind;
            cnt1=cnt1+1;
        end

end

GA_ops_T;
```

get_data.m

```
n_feat = xlsread(file,'Part_Info','B2');
n_ops = xlsread(file,'Part_Info','B3');


st = [5,3]; %Starting Point of data
xax = 6;  %Number of columns
yax = n_ops; %Number of Operations
exc2mat;
tad_data = xlsread(file,'Features_Data',range);

st = [5,3];
xax = 1;
yax = n_ops;
exc2mat;
tool_data = xlsread(file,'Tools_Data',range);

st = [2,3];
xax = 6;
yax = 1;
exc2mat;
m_data = xlsread(file,'Machine_Data',range);

st = [3,2];
xax = n_feat;
yax = n_feat;
exc2mat;
topo = xlsread(file,'Toplogical_Interaction',range);

st = [1,2];
xax = 1;
yax = n_feat;
exc2mat;
index = xlsread(file,'Toplogical_Interaction',range);
```

exc2mat.m

```
asc = 64;
xdim = asc+xax+st(1)-1;
ydim = yax+st(2)-1;
divs = floor((xax+st(1)-1)/26);

if xax+st(1)-1 <=26
   rang = sprintf('%s%d',char(xdim),ydim);
else
   rang = sprintf('%s%s%d', char(asc+divs),char(xdim-(26*divs)),ydim);
```

```
end
%Converting Numbers of Alphanumeric values for excel data read
range = sprintf('%s%d:%s', char(asc+st(1)), st(2), rang);
```

ppgen.m

```
cond =1;
cnt=1;
dummy=zeros(1,n_ops);

ind_seqmat=1;

for i=1:(n_ops-1)
    initj=i+1;
    for j=initj:n_ops
        if prec(i,j)==1
            seqmat(ind_seqmat,1)=i;
            seqmat(ind_seqmat,2)=j;
            ind_seqmat=ind_seqmat+1;
        elseif prec(i,j)==-1
            seqmat(ind_seqmat,1)=j;
            seqmat(ind_seqmat,2)=i;
            ind_seqmat=ind_seqmat+1;
        end
    end
end

n_cons=(numel(seqmat)/2);


while cnt<=pop_size+(n_mut*gens)

 for f=1:50000
    popi(f,:)=randperm(n_ops);
end
popu = unique(popi,'rows');
test = popu;
n_pp=(numel(test)/n_ops);

clearvars feas;
clearvars ppi;
clearvars count;
clearvars pop;
count=0;

feas(n_pp)=0;
```

```
check1=0;
check2=0;
ppi=1;

for i=1:n_pp%Alternate Process plan loop
   flag=0';
   for j=1:n_cons%contraints loop
      op1=seqmat(j,1);
      op2=seqmat(j,2);
      for k=1:n_ops%process plan in consideration
         if test(i,k)==op1
            i_op1=k;
            check1=1;
         elseif test(i,k)==op2
            i_op2=k;
            check2=1;
         end
         if check1==1 && check2==1
            check1=0;
            check2=0;
            if i_op1>i_op2
               flag=1;
               %disp('Precendence broken');
               feas(i)=0;
               break;
            else
               flag=0;
            end
         end
      end
      if flag==1
         break;
      end
   end
   if flag==0
      feas(i)=1;
      count(ppi)=i;
      ppi=ppi+1;
   end
end
feasi=sort(feas,'descend');

if ppi>1
   e=numel(count);
   g=e+cnt-1;

   for d=cnt:g
```

```
        dummy(d,:)=test(count(d-cnt+1),:);
    end
    dummy = unique(dummy,'rows');
    clc;
    fprintf('Required PPs --> %d \n',(pop_size+(n_mut*gens)));
    dumy = numel(dummy)/n_ops;
    fprintf('Generated PPs --> %d \n',dumy);
    cnt = dumy+1;
end

end
s_space  = dummy(1:pop_size+(n_mut*gens),1:n_ops);
```

GA_ops_T.m

```
clearvars pp_ftns;
clearvars gen_ftns;
clearvars  tool_change;
clearvars  part_rot;
clearvars  cgen_temp;

tool_change = zeros(pop_size,2,gens);
part_rot = zeros(pop_size,2,gens);
pp_ftns=zeros(pop_size,2,gens);
gen_ftns=zeros(gens,1);
bit=0;
mut_chk = randperm(10);
mut_sel_i = randperm(10,r_mut*10);
mut_sel = mut_sel_i;
mut_cnt = 0;
gen_cnt = 0;
ftns_ctoff = 50;

for c_gen=1:gens
    if c_gen>1
        pop(:,:,c_gen) = cgen_temp;
        cgen_pop = cgen_temp;
        lgen_ftns = gen_ftns(c_gen-1,2);
    end
    cgen_temp = zeros(pop_size,n_ops*2);
    %%%%%%%%%%%%%%%%%%Tool Change
Computation%%%%%%%%%%%%%%%%%%%%
    for ind=1:pop_size
        for ind1=1:n_ops-1
            if tool_data(cgen_pop(ind,ind1)) ~= tool_data(cgen_pop(ind,ind1+1))
                tool_change(ind,1,c_gen)=ind;
                tool_change(ind,2,c_gen) = tool_change(ind,2,c_gen) +1;
```

```
        end
      end
      tool_change(ind,1,c_gen)=ind;
   end
   cgen_tchange = tool_change(:,:,c_gen);

%%%%%%%%Part Rotation Computation%%%%%%%%%%
   check1 = 0;
   check2 = 0;
   for ind=1:pop_size
      for ind1=1:n_ops-1
         if cgen_pop(ind,(cgen_pop(ind,ind1)+n_ops),1) ~=
cgen_pop(ind,(cgen_pop(ind,ind1+1)+n_ops),1)
            check1 = 0;
            check2 = 0;
            for ind2=1:numel(m_detail)
               if cgen_pop(ind,(cgen_pop(ind,ind1)+n_ops),1) == m_detail(ind2) && check1
== 0
                  check1=1;
               end
               if cgen_pop(ind,(cgen_pop(ind,ind1+1)+n_ops),1) == m_detail(ind2) &&
check2 == 0
                  check2=1;
               end
            end
            if check1 ==0 || check2 ==0
               part_rot(ind,2,c_gen) = part_rot(ind,2,c_gen) +1;
            end
            part_rot(ind,1,c_gen)=ind;
         end
      end
      part_rot(ind,1,c_gen)=ind;
   end
   cgen_prot = part_rot(:,:,c_gen);



%%%%%%%%%%Fitness Computation for PPs%%%%%%%%%%%%%%%
   for ind = 1:pop_size
      pp_ftns(ind,2,c_gen) = w_pr*cgen_prot(ind,2) + w_tc*cgen_tchange(ind,2) +
pen_ftns(ind,1,c_gen);
      pp_ftns(ind,1,c_gen) = ind;
   end
   pp_ftns(:,:,c_gen) = sortrows(pp_ftns(:,:,c_gen),2);

%%%%%%%%%%Fitness Computation for generation%%%%%%%%%%%%%%%
   temp1= sum(cgen_prot);
```

```
temp2= sum(cgen_tchange);

gen_ftns(c_gen,1)=c_gen;
gen_ftns(c_gen,2) = w_pr*temp1(2) +  w_tc*temp2(2);

if lgen_ftns-gen_ftns(c_gen,2) < ftns_ctoff && lgen_ftns-gen_ftns(c_gen,2) > 0
   gen_cnt = gen_cnt+1;
elseif lgen_ftns-gen_ftns(c_gen,2) > ftns_ctoff
   gen_cnt = 0;
end

if gen_cnt == 200
   break;
end

for ind =1:n_copairs
   cgen_temp(ind,:) = cgen_pop(pp_ftns(ind,1,c_gen),:);
   cgen_temp(ind+n_copairs,:) = cgen_pop(pp_ftns(ind+n_copairs,1,c_gen),:);
  %%%%%%%%%%CrossOver Routine%%%%%%%%%%%%%%%%%%%%%%
   co_gen;
   cgen_temp(ind+2*n_copairs,:) = off2;
   cgen_temp(ind+3*n_copairs,:) = off1;
end

for ind=1:carry_over
   cgen_temp((4*n_copairs)+ind,:) = cgen_pop(pp_ftns((2*n_copairs)+ind),:);
end

for ind=1:n_mut
   mut_chk1 = 0;

   if mut_cnt == 10;
      mut_chk = randperm(10);
      mut_sel = mut_sel_i;
      mut_cnt = 0;
   end

   mut_cnt = mut_cnt+1;
   mut_ind = 0;

   for ind5 =1: numel(mut_sel)
      if mut_sel(ind5) == mut_chk(mut_cnt)
         %%%%%%%%%%%Mutation Routine%%%%%%%%%%%%%
         mut;
         cgen_temp((n_copairs*4)+carry_over+ind,:) = muted;
         mut_sel(ind5) = [];
         mut_ind=1;
```

```
            end
            if mut_ind ==1
                mut_ind = 0;
                break;
             end
        end

        if mut_chk1 == 0
             cgen_temp((n_copairs*4)+carry_over+ind,:) =
cgen_pop(pp_ftns((2*n_copairs)+carry_over+ind),:);
        end

    end


end
gen_ftnss = gen_ftns(1:c_gen-1,:);
Sort_PP_ftns
figure
plot(gen_ftnss(:,1),gen_ftnss(:,2));
title('Generational Fitness');
xlabel('Generation') % x-axis label
ylabel('Fitness') % y-axis label
```

co_gen.m

```
par1 = cgen_pop(pp_ftns(ind,1,c_gen),1:n_ops);
par2 = cgen_pop(pp_ftns(ind+n_copairs,1,c_gen),1:n_ops);

rem = 0;
for ind5 = 1: n_ops
    if par1(ind5) == par2(ind5)
        rem = rem+1;
    else
        break;
    end
end

n_swap = n_ops-rem-1;
sim = zeros(2,n_swap);
swap = zeros(2,n_swap);


tpar1 = cgen_pop(pp_ftns(ind,1,c_gen),n_ops+1:n_ops*2);
tpar2 = cgen_pop(pp_ftns(ind+n_copairs,1,c_gen),n_ops+1:n_ops*2);
```

```
off1 = zeros(1,n_ops+TADs);
off2 = zeros(1,n_ops+TADs);
tad1 = zeros(n_swap);
tad2 = zeros(n_swap);

off1(n_ops+1:n_ops+TADs) = tpar1;
off2(n_ops+1:n_ops+TADs) = tpar2;


rem1=rem+n_swap+1;
swap(1,:) = par1(rem+1:rem+n_swap);
swap(2,:) = par2(rem+1:rem+n_swap);


off1(1:rem)= par2(1:rem);
off1(rem1:n_ops)= par2(rem1:n_ops);

off2(1:rem)= par1(1:rem);
off2(rem1:n_ops)= par1(rem1:n_ops);

sim(1,:) = ismember (swap(1,:),off1(1:n_ops));
sim(2,:) = ismember (swap(2,:),off2(1:n_ops));

%%%%%%%%Check similarities between the parents%%%%%%%%%%%
for var1 = 1:2
   for var2 = 1:n_swap
      if sim(var1,var2)==1;
         swap(var1,var2)=0;
      end
   end
end


dum = ismember (par2(rem+1:rem+n_swap),swap(1,:));
tad1 = swap(1,:);
tad2 = swap(2,:);

%%%%%%%%%% check which Operations already exist%%%%%%%%%%%
for var1 = 1:n_swap
   if dum(var1)==0
      off1(rem+var1) = par2(rem+var1);
   end
end
cntr=1;
for var1 = 1:n_swap
   if off1(rem+var1)==0
      for var2 = 1:n_swap
```

```
            if swap(1,var2)~=0
                off1(rem+var1) = swap(1,var2);
                ref1(cntr) =  swap(1,var2);
                cntr = cntr+1;
                swap(1,var2) = 0;
                break;
            end
        end
    end
end

dum = ismember (par1(rem+1:rem+n_swap),swap(2,:));
%%%%%%%%%%Swap Operations%%%%%%%%%%%
for var1 = 1:n_swap
    if dum(var1)==0
        off2(rem+var1) = par1(rem+var1);
    end
end
cntr=1;
for var1 = 1:n_swap
    if off2(rem+var1)==0
        for var2 = 1:n_swap
            if swap(2,var2)~=0
                off2(rem+var1) = swap(2,var2);
                ref2(cntr) =  swap(2,var2);
                cntr = cntr+1;
                swap(2,var2) = 0;
                break;
            end
        end
    end
end

for var1 = 1:numel(ref1)
        off1(n_ops+ref1(var1)) = tpar2(ref1(var1));
end

for var1 = 1:numel(ref2)
        off2(n_ops+ref2(var1)) = tpar1(ref2(var1));
end
```

mut.m

```
mut_chk1 = 1;
muted = zeros(1,n_ops);
in_var = cgen_pop(pp_ftns((2*n_copairs)+carry_over+ind),:);
muted = in_var;
```

```
cond =1;

ran1 = randi(n_ops,1);
sta = 1;
%%%%%%%%Generate Unique Mutation Locations%%%%%%%%
while sta==1
   ran2 = randi(n_ops,1);
   if ran2 ~= ran1
      sta=0;
   end
end

muted(ran1) = in_var(ran2);
muted(ran2) = in_var(ran1);
muted(ran1+n_ops) = in_var(ran2+n_ops);
muted(ran2+n_ops) = in_var(ran1+n_ops);
test = muted;

ind_seqmat=1;
%%%%%%%%%%Mutation %%%%%%%%%%%%%
for i=1:(n_ops-1)
   initj=i+1;
   for j=initj:n_ops
      if prec(i,j)==1
         seqmat(ind_seqmat,1)=i;
         seqmat(ind_seqmat,2)=j;
         ind_seqmat=ind_seqmat+1;
      elseif prec(i,j)==-1
         seqmat(ind_seqmat,1)=j;
         seqmat(ind_seqmat,2)=i;
         ind_seqmat=ind_seqmat+1;
      end
   end
end


n_pp =1;
n_cons=(numel(seqmat)/2);
check1=0;
check2=0;

%%%%%%%% Check for constraint compliance after mutation%%%%%%%%

for i=1:n_pp%Alternate Process plan loop
   flag=0';
   for j=1:n_cons%contraints loop
      op1=seqmat(j,1);
```

```
        op2=seqmat(j,2);
        for k=1:n_ops%process plan in consideration
            if test(i,k)==op1
                i_op1=k;
                check1=1;
            elseif test(i,k)==op2
                i_op2=k;
                check2=1;
            end
            if check1==1 && check2==1
                check1=0;
                check2=0;
                if i_op1>i_op2
                    flag=1;
                    %disp('Precendence broken');
                    pen_ftns((n_copairs*4)+carry_over+ind,1,c_gen+1) = v_factor;
                    break;
                else
                    flag=0;
                end
            end
        end
        if flag==1
            break;
        end
    end
end
```

Sort_PP_ftns.m

```
 optima = zeros (2,n_ops);
min=pp_ftns(1,2,1);
gen_id=0;
pp_id=0;
%%%%%%%%%%Finding the most fit PP%%%%%%%%%%%%%
for ind=1:c_gen
    for ind1=1:pop_size
        if pp_ftns(ind1,2,ind)<min
            min=pp_ftns(ind1,2,ind);
            gen_id=ind;
            pp_id=pp_ftns(ind1,1,ind);
        end
    end
end
min
```

```
optima(1,:) = pop(pp_id,1:n_ops,gen_id);

for ind=1:n_ops
   optima(2,ind) = pop(pp_id,(optima(1,ind))+n_ops,gen_id);
end
optima
part_rotations = part_rot(pp_id,2,gen_id)
tool_changes = tool_change(pp_id,2,gen_id)
```

Chapter 7

References

1. Baqai, A., *Co-conception des processus d'usinage et des configurations cinematiques d'un systeme de production reconfigurable.* 2010.
2. Chaube, A., L. Benyoucef, and M.K. Tiwari, *An adapted NSGA-2 algorithm based dynamic process plan generation for a reconfigurable manufacturing system.* Journal of Intelligent Manufacturing, 2012. **23**(4): p. 1141-1155.
3. Elmaraghy, H.A., *Changeable and reconfigurable manufacturing systems.* 2008.
4. M. R. Henderson and D.C. Amderson, *Computer Recognition and Extraction of Form Features: A CAD/CAM Link.* Computers in Industry, 1984. **5**: p. 329-339.
5. S. Joshi and T.C. Chang, *Graph-based heuristics for recognition of machined features from a 3D solid model.* Computer Aided Design, 1988. **20**(2): p. 58-66.
6. S. Prabhakar and M.R. Henderson, *Automatic form-feature recognition using neural-network-based techniques on boundary representations of solid models.* Computer Aided Design, 1992. **24**(7): p. 381-393.
7. Sakurai, H., *Volume decomposition and feature recognition: Part 1 -polyhedral objects.* Computer Aided Design, 1995. **27**(11): p. 833-843.
8. M. P. Bhandarkar and R. Nagi, *STEP-based feature extraction from STEP geometry for Agile Manufacturing.* Computers in Industry, 2000. **41**: p. 3-24.
9. J. H. Han, M. Kang, and H. Choi, *STEP-based feature recognition for manufacturing cost optimization.* Computer-Aided Design, 2001. **33**: p. 671-686.
10. N. Ozturk and F. Ozturk, *Neural network based non-standard feature recognition to integrate CAD and CAM.* Computers in Industry, 2001. **45**: p. 123-1335.
11. J. H. Han, et al., *Manufacturing feature recognition toward integration with process planning.* IEEE Transactions on Systems, Man, and Cybernetics—part b: Cybernetics, 2001. **31**(3).
12. N. Ahmad and A.F.M.A. Haque, *Manufacturing feature recognition of parts using DXF files.* 4th International Conference on Mechanical Engineering, 2001: p. 111-116.
13. Z. Huang and D. Yip-Hoi, *High-level feature recognition using feature relationship graph.* Computer-Aided Design, 2002. **34**: p. 561-582.
14. Bojan Babic, Nenad Nesic, and Z. Miljkovic, *A review of automated feature recognition with rule-based pattern recognition.* Computers in Industry, 2008. **59**(4): p. 321-337.
15. Y. Woo and H. Sakurai, *Recognition of maximal features by volume decomposition.* Computer-Aided Design, 2002. **34**: p. 195-207.
16. V. Sundararajan and P.K. Wright, *Volumetric feature recognition for machining components with freeform surfaces.* Computer-Aided Design, 2004. **36**: p. 11-25.
17. H. C. Lee, W. C. Jhee, and H.S. Park, *Generative CAPP through projective feature recognition.* Computers & Industrial Engineering, 2007. **53**: p. 241-246.
18. V. Rameshbabu and M.S. Shunmugam, *Hybrid feature recognition method for setup planning from STEP AP-203.* Robotics and Computer-Integrated Manufacturing, 2009. **25**: p. 393-408.
19. F. Garcia, et al., *Process Planning Based on Feature Recognition Method.* Assembly and Manufacturing (ISAM), 2011. **11**.

20. H. K. Miao, N. Sridharan, and J.J. Shah, *CAD-CAM integration using machining features.* Computer integrated manufacturing, 2002. **15**(4): p. 296-318.
21. Hasan, S.M. and A. Baqai, *An Approach for the Selection of Process Plans Based on Part Family Changes*, in *Advances in Sustainable and Competitive Manufacturing Systems: 23rd International Conference on Flexible Automation & Intelligent Manufacturing*, A. Azevedo, Editor. 2013, Springer International Publishing: Heidelberg. p. 65-77.
22. A.I Shabaka and Hoda.A.ElMaraghy, *Generation of machine configurations based on product features.* International Journal of Computer Integrated Manufacturing, 2007. **20**(4): p. 355-369.
23. Yorem Koren and M. Shpitalni, *Design of reconfigurable manufacturing systems.* Manufacturing Systems, 2011.
24. Zahid, T. and A.A. Baqai, *Multi-Criteria Optimization of Process Plans for Reconfigirable Manufacturing Systems: An Evolutionary Approach.* 2013(56192): p. V02BT02A045.

CERTIFICATE OF COMPLETENESS

It is hereby certified that the dissertation submitted by NS Ausama Nawaz, Reg No. **NUST201464466MCEME35114F**, Titled: ***An integrated framework for feature recognition and co-evaluation of process planning through Kinematics Configurations for flexible and changeable manufacturing systems*** has been checked/reviewed and its contents are complete in all respects.

Supervisor's Name: **Dr. Sajid Ullah Butt**

Signature: ______________________

Date: ______________________