PHD

Integrated process planning and scheduling for common prismatic parts in a 5-axis CNC environment

Guo, Yanwu

Award date:
2006

Awarding institution:
University of Bath

Integrated Process Planning and Scheduling for common prismatic parts in a 5-axis CNC environment

Yanwu Guo

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Mechanical Engineering

2006

Abstract

In modern manufacturing, effective CAD/CAM integration, especially for 5-axis machining, has not been fully implemented and current manufacturing systems still lack flexibility. In this thesis, attempts have been made toward developing an integrated strategy for product design and its downstream manufacturing processes, including scheduling for establishing an adaptive and flexible system.

To generate detailed operation instructions for transforming an engineering design into a final part, a Computer Aided Process Planning (CAPP) system (an essential linkage between CAD and CAM), has been developed for common prismatic components in a 5-axis CNC machining environment. Furthermore, an adaptive Integrated Process Planning and Scheduling system (IPPS) has been developed to generate an optimised schedule by optimising both the process planning and scheduling simultaneously.

The four major modules that form the prototype CAPP system, namely the *Feature information input module*, *Operation selection module*, *Cutting conditions calculation module* and *Operation sequencing module*, have been designed and implemented. The feature technology, heuristic rules and evolutionary algorithms have been used to enable these modules to work effectively and efficiently and a case study has been conducted to verify the ability of the prototype system.

Furthermore, an independent operation sequencing module for 3-axis machining and an independent IPPS module have been discussed and implemented. The representations of process plans and schedules have been given and the performance criteria to evaluate the generated process plans and schedules have been discussed. To provide an optimised solution to the process planning and IPPS problems, a modern evolutionary algorithm, the Particle Swarm Optimisation (PSO) algorithm, has been employed and modified. Through case studies, a comparison has been made between the result of the modified PSO algorithm and previous published results using the Genetic Algorithm (GA) and the Simulated Annealing (SA) algorithm respectively, and for these cases the PSO algorithm has been shown to outperform both the GA and SA in the majority of applications by consideration of the computation efficiency, optimisationability and robustness.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Dr. Geraint Owen and Professor Tony Mileham (University of Bath) for their constant help, encouragement, patience, comments and for their great guidance throughout the research, to make this research work possible. It has been my greatest pleasure and honor to be their student and work with them.

I would like to thank Dr. Weidong Li for his selfless guidance and help in my research work during his stay at University of Bath. I also would like to thank Dr. Adrian Bowyer, my internal examiner for his guidance.

I wish to thank all the staffs, students and friends (Jafar, Qian, Lian, et. al) I have worked with in the Innovative Manufacturing Research Centre, who share much of their talents, knowledge and research experience with me. It is really a great pleasure to work in this environment. Thank you all!

I also wish to thank the IMRC and the Department of Mechanical Engineering at the University of Bath, which funded this research. I also wish to thank all the staffs in the manufacturing laboratory for their help.

A special gratitude should be expressed to my parents and Bo Jin. Throughout my study in the University of Bath, they provided me with the firmest support.

Table of Contents

Notations

CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
FMS	Flexible Manufacturing Systems
CIM	Computer Integrated Manufacturing
AI	Artificial Intelligence
IPPS	Integrated Process Planning and Scheduling
TAD	Tool Approach Direction
GA	Genetic Algorithm
SA	Simulated-Annealing algorithm
CNC	Computerized Numeric Control
PSO	Particle Swarm Optimisation
GT	Group technology
ANN	artificial neural networks
STEP	Standard for the Exchange of Product
AP	Application Protocols
B-rep	Boundary representation
CSG	Constructive Solid Geometry
DSG	Destructive Solid Geometry
EAD	External Access Directions
OOPS	Object Oriented Programming Strategy
DFM	Design For Manufacturing
TS	Tabu Search
TSP	Travelling Salesman Problem
STAD	Single Tool Axis Direction
MTAD	Multiple Tool Axis Direction
OOP	Object Oriented Programming
OPT	Operation Type
HSS	High Speed Steel
R_a	Roughness average
TMC	Total Machine Cost of the plan

TTC	Total Tool Cost of the plan
TSC	Total Set-up Cost of the plan
TMCC	Total Machine Change Cost of the plan
TTCC	Total Tool Change Cost of the plan
APC	Additional Penalty Cost of violating constraints in the plan
TC	Total Cost of the plan
TMT	Total Machining Time of the plan
TMCT	Total Machine Change Time of the plan
TTCT	Total Tool Change Time of the plan
APT	Additional Penalty Time of violating constraints in the plan
TPT	Total Processing Time of the plan
TR	Totally Randomly
FR	First Randomly
TS	Totally Set
FS	First Set

List of Figures

List of Tables

Chapter 1 Introduction

Modern manufacturing faces several challenges such as stiff global competition, low volume, large variety production, the requirement of high productivity and product quality, shorter lead times from design to manufacturing and rapidly changing customer requirements (Patil and Pande 2002, Maturana et al. 1999). These challenges have acted as a driving force for the application of new technologies in industries. In respect to hardware, particularly as a development in the aeronautic and automobile industries, 5-axis NC machines have become widely used in machining of complex geometry surfaces such as turbine blades, impellers, propellers, 3D cams, moulds and dies. With two more degrees of freedom than traditional 3-axis machines, 5-axis machining offers many advantages over 3-axis machining, including better tool accessibility, low setup cost, and easier to machine complex surfaces (Mahbubur et al. 1997, Lo 1999, Ho and Hwang 2003). The software and hardware for many manufacturing methodologies have been developed such as Computer Aided Design (CAD), Computer Aided Manufacturing (CAM), Flexible Manufacturing Systems (FMS) and Computer Integrated Manufacturing (CIM). CIM aims to integrate the highly fragmented manufacturing operations in an enterprise in order to utilise the resource and information more effectively. A lot of research has addressed CAD/CAM integration (the heart of CIM) in last several decades.

However, effective CAD/CAM integration, especially for 5-axis machining, has not been implemented and current manufacturing systems still lack flexibility. In order to overcome these drawbacks, an integrated strategy for product design and its downstream manufacturing processes, including scheduling for establishing an adaptive and flexible system, is imperative. In this thesis, attempts have been made towards this direction. A Computer Aided Process Planning (CAPP) system, an essential linkage between CAD and CAM, has been developed for 5-axis machining and an adaptive integrated CAPP and scheduling system has been implemented.

In this chapter, the background of research is presented in section 1.1. Then the overall aims of the research are described in section 1.2. Finally section 1.3 gives an organisational outline of the thesis.

1.1 Background

1.1.1 Computer Aided Process Planning (CAPP)

As described above, the focus of CIM is on information as the crucial element linking all facets of the manufacturing enterprise. While the geometry information is created in CAD, the manufacturing information is concerned with the production planning and plant operation (Kang et al. 2003). Information cannot be effectively transferred from CAD to CAM without a Computer-Aided Process Planning (CAPP) system as a linkage.

Process Planning, as defined by Chang and Wysk (1985), is the act of preparing detailed operation instructions to transform an engineering design to a final part. It involves determining the most appropriate manufacturing processes and the order in which they should be performed to produce a given part or product specified by design engineering. In general, a process plan contains routes, processes, process parameters, machines, set-ups, tools required for production of parts and the tool path. The process plan must be developed within the limitations imposed by available processing equipment and productive capacity of the factory (Groover

2002). Although the process-planning functions may be different according to different industries, they involve several or all of the following activities (Chang 1990, Lee D.H. et al. 2001):

- Selection of machining operations;
- Sequencing of machining operations;
- Selection of cutting tools;
- Selection of machine tools;
- Determining setup requirements;
- Calculations of cutting parameters;
- Tool path planning and generation of NC part programs;
- Design of jigs and fixtures

The use of computer techniques to automate the tasks of process planning – Computer Aided Process Planning (CAPP), is taken to support process planners in the planning process and assist in taking decisions. As a key technology for CAD/CAM integration, CAPP strongly influences the cost of production and the quality of a product. The greater the degree of automation of a CAPP system, the shorter the time from design to manufacturing, and the better the quality of the final product owing to the elimination of human error (Yip-Hoi and Dutta, 1996, Groover 2002).

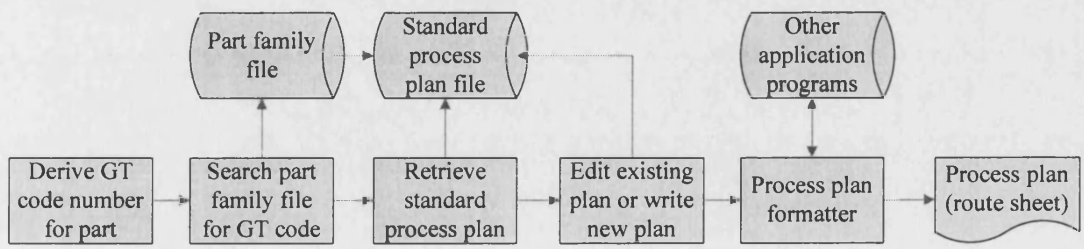


Figure 1.1 Operation of a variant CAPP system (Groover 2002)

The developed CAPP systems can be generally classified into two categories, namely, variant systems and generative systems. Variant systems use the classification and coding of Group Technology (GT) to select a baseline process plan for a part family. The process plan for a new part is created through retrieving the plan of a similar part that has been developed and stored in a database, and then modifying it as necessary (Bhaskara Reddy et al. 1999, Groover 2002). The variant systems may cut down process planning time dramatically, especially for similar components, so are currently dominant in industry. However, they are deficient in planning the processes of new products with many new features or structures. Figure1.1 shows the operations of a variant CAPP system.

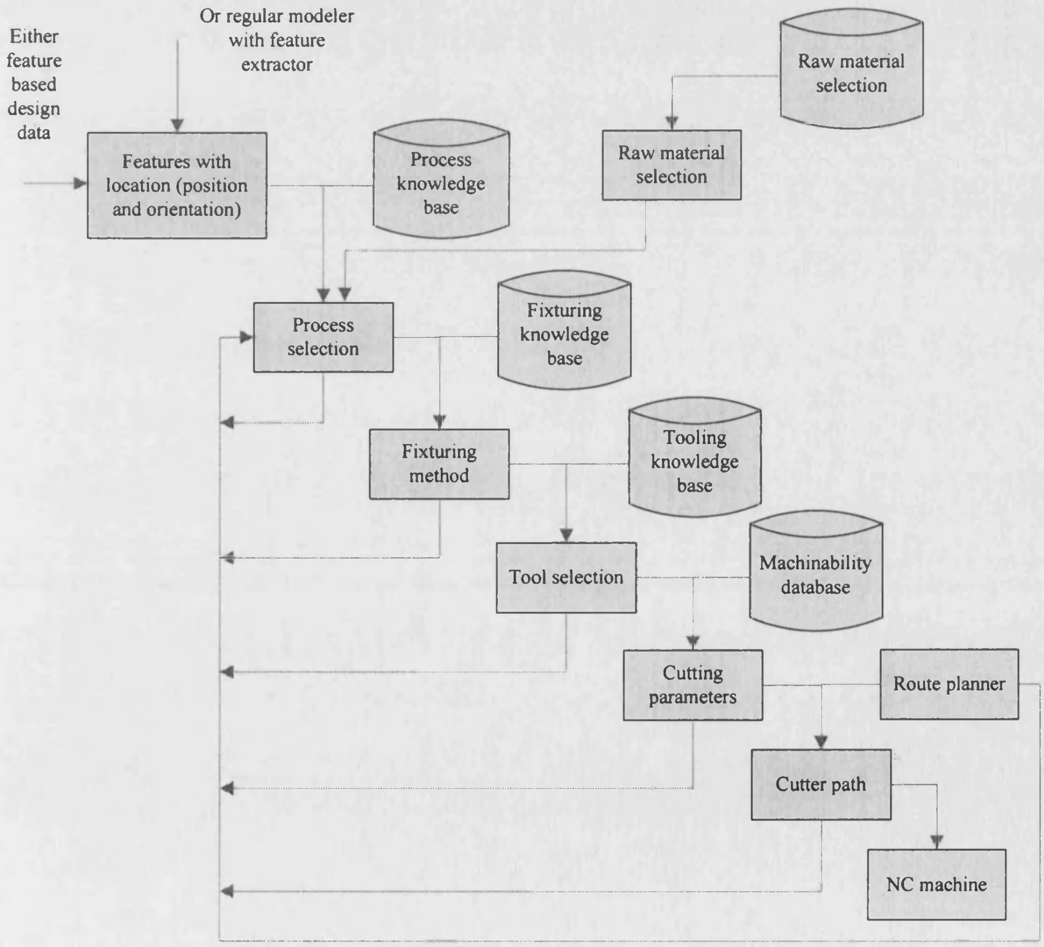


Figure 1.2 Automated process planning - generative technique (Rembold 1993)

Generative systems are based on heuristic reasoning and Artificial Intelligence (AI) technologies and can generate a new process plan for a part from scratch by applying intelligent decision rules to the part, based on the specific manufacturing conditions of companies. In comparison to the variant method, the generative method needs less human intervention and new parts can be planned as easily as existing components. Although it is time-consuming, costly and error-prone to acquire the expert knowledge and the decision rules, with keener global competition and the requirement of delivering new products more efficiently, the development of generative systems is imperative to facilitate process planning with higher flexibility and adaptability. As figure 1.2 shows, a generative process planning system comprises three main components:

1. Knowledge base: the technical knowledge of manufacturing and the logic used by successful process planners to make decisions on various aspects of process planning must be captured and coded into a computer program.
2. Computer-compatible part description: Part description forms a major part of the information needed for process planning. The description contains all the pertinent data needed to plan the process sequence. Two possible descriptions are (1) the geometric model of the part developed on a CAD system during product design, or (2) a group technology code number of the part defining its features in significant detail.
3. Inference engine: a generative CAPP system requires the capability to apply the planning logic and process knowledge contained in the knowledge base to a given part description. The CAPP system applies its knowledge base to solve a specific problem of planning the process for a new part. This problem-solving procedure is referred to as the "inference engine" in the terminology of expert

systems. By using its knowledge base and inference engine, the CAPP system synthesizes a new process plan for each new part presented to it (Groover 2002).

In the last two decades, many technologies have been introduced into CAPP systems, such as feature technology, Artificial Intelligence (AI) to improve the performance of CAPP systems and the integration of CAPP and other manufacturing activities especially scheduling has also been addressed.

1.1.2 Feature technology

The support of integrated product design and manufacturing entails two requirements: (1) product design representation and reasoning capability from various product life-cycle considerations, such as manufacturability analysis, and (2) manufacturing process design capability to plan efficient and flexible manufacturing by exploiting the product information provided by these product design representations (Kim et al. 2001). It has been universally recognised that the geometric model of a part designed using conventional CAD systems is not sufficient for process planning or other reasoning and planning purposes (Mantyla et al. 1996). Feature technology is an emerging tool for this purpose. Historically, the concept of a feature originated in the process planning of machined parts. It therefore follows that linking CAD to CAPP for machined products using features has become the focus of numerous research efforts in recent years (Wong and Wong 1995).

As figure 1.3 shows, the features refer to the design features and machining features (Fu et al. 2003). The viewpoints of a part are different for a designer and a process planner. For the process planner, the feature is commonly viewed as a machining feature (manufacturing feature). For example, a slot could be seen as a general slot milled by a milling machine; a hole can be considered as a drilled or bored hole. For the designer, design features are expressed in geometric terms. However machining features can express explicitly the methods of production while implying the geometry and function of the features.

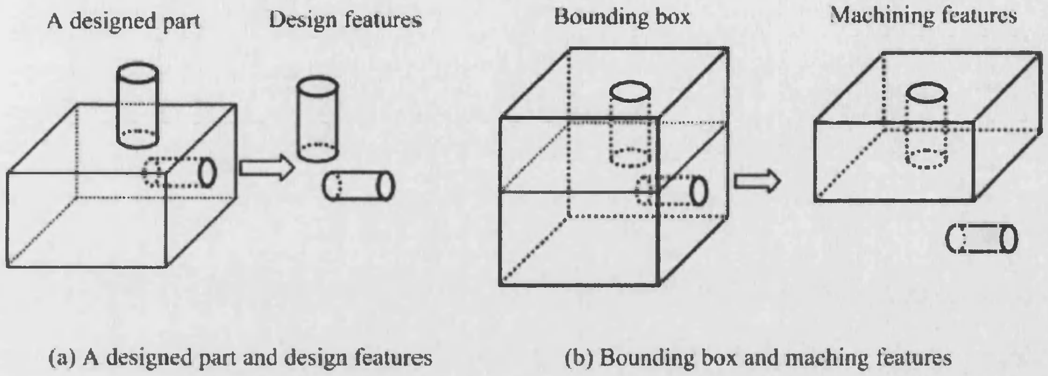


Figure 1.3 Design Feature and Machining feature (Fu et al. 2003)

Feature recognition and design by features are the two major approaches to create feature models (Bronsvoort and Jansen 1993). Feature recognition makes direct use of geometric models and generates application-specific feature models using various recognition rule sets regarding the application. A principal advantage of feature recognition is the possibility of using conventional CAD systems directly. However, there are problems with feature recognition such as feature interactions hindering its practical applications. With a design by features approach, the designer specifies a design model using a set of design features defined in a feature-based model system (Lee and Kim 1999). In contrast to feature recognition, design by features can capture the design and manufacturing information during the design stage. A feature has its specific geometry and must be associated with some feature attributes including dimensions, tolerance, etc. Also, features should carry information regarding process planning, manufacturing, and inspection. In other words, the designer can choose the manufacturing processes whilst working on the design. It reduces remarkably the amount of work for recognising features. Now more and more commercial CAD software support design by features. For example, Unigrphics NX2 enables users to model a part with holes, pockets, slots and boss features, etc.

1.1.3 Integration of process planning and scheduling (IPPS)

In manufacturing, both process planning and scheduling functions are responsible for the efficient allocation and utilisation of resources. As described above, process planning is used to plan manufacturing resources and operations for a part to ensure the application of good manufacturing practice and maintain the consistency of the desired functional specifications of the part during its manufacture. Scheduling is used to determine the most appropriate moment to execute each operation for the launched production orders, taking into account the due date of these orders, a minimum workshop inventory, a maximum resource utilisation, etc., in order to obtain high productivity in the workshop (Li and McMahon 2006, Kempenaers and Pinte 1996, Aldakhilallah and Ramesh 1999, Yang et al. 2001, Wong et al. 2006). Traditionally, in the batch working industry, process planning and workshop scheduling are done separately and sequentially, where the process plan is determined before the actual scheduling with no regard for the scheduling objectives. The process planning system first generates a reasonable process plan for each part including applicable manufacturing resources (machines and tools), set-up plans and a feasible operation sequence of the part. During the process, it usually assumes that all the applicable manufacturing resources are available for this part. The scheduling system then specifies the schedule of manufacturing resources on each operation (job) of the parts according to the importance of jobs, availability of resources and time constraints.

However, this simple sequential approach ignores the relationship between scheduling and process planning. If a process plan is prepared offline without due consideration of the actual shop floor status, it may become infeasible due to changes or constraints in the manufacturing environment and heavily unbalanced resource assignments. Also due to the different objectives of these two systems, it is difficult to produce a satisfactory result in simple sequential executions of the two systems. And because the process plan for each part is generated independently without consideration of other parts, when these generated process plans for different parts are taken forward for scheduling, they may not be schedulable to meet the requirement due to time and resources constraints. The merit of integrated process planning and scheduling (IPPS) is to increase production feasibility and optimality by combining both the process planning and scheduling problems (Huang et al. 1995).

1.1.4 Optimisation of CAPP and IPPS

1.1.4.1. Optimisation of CAPP

In developing computer-aided process planning (CAPP) systems, one of the major difficulties is the selection of suitable setup plans and machining resources, and sequencing the machining operations so that the least machining cost of the part can be obtained (Qiao et al. 2000, Lee D.H. et al. 2001). Traditional CAPP approaches aim mainly at generating a single feasible plan for a given part. However, the introduction of new manufacturing technologies, (e.g. design for manufacturing

(DFM) and integration of process planning and job shop scheduling (IPPS)), to support DFM, the best process plan for a given part in a designated machining environment must be generated and fed back to the designer for evaluation. To support dynamic scheduling, a CAPP system must be able to generate plans with alternative routes and sequences to suit the variable status of the shop floor (Ma, et al. 2000).

As one of the most important tasks and also a bottleneck task in developing a CAPP system, the operations sequence generation problem can usually be modelled as a large-scale and combination optimisation problem with constraints. The complexity of solutions to the problem is highly dependent upon the shape and the number of features of a machined part. For instance, the process planning practice has shown that, for rotational parts, the topology relationships among most of the features comply with an explicit machining order of "from the left end to the right end' or vice versa and "from internal to external'. It can greatly reduce the number of operation combinations so as to generate a smaller search space, leading to various types of heuristics for operations sequence generation with less difficulty (Du and Huang 1990, Usher and Bowden 1996). However, for a prismatic part, this problem is intractable, with difficulties in the following aspects (Li et al. 2002):

(1) The geometric relationships between features in a prismatic part are complicated, and the explicit heuristic rules for sequencing the operations corresponding to the features are deficient.
(2) Each feature might have several candidate Tool Approach Directions (TADs), cutting machines and tools to machine it. The decision processes of selecting machining resources and set-up plans as well as sequencing operations are

sometimes contradicting. The evaluation criteria coming from some aspects, such as minimum usage of expensive machines and tools, minimum number of setups, minimum number of machine and tool changes, and achieving good manufacturing practice, are also conflicting in certain cases. To carry out the different decision processes considering the evaluation criteria simultaneously is imperative to a globally optimised solution. However, it is usually difficult for some reasoning approaches.

(3) For a part, there usually exist several alternative process plans that can achieve the predetermined optimisation objective. To generate and provide the alternative optimal plans can help process planners make a reasonable decision according to the workshop environment and fixture conditions. However, in the existing published approaches, few contributions have been made towards this direction.

It is necessary to develop an optimisation approach for the machining operations sequencing problem in CAPP considering the above factors in order to improve CAPP system's performance and adaptability.

1.1.4.2. Optimisation of IPPS

As described in the previous sections, for a process planning system, the decision of which machine tool to select is usually made based on the objective of achieving the correct quality, the minimal manufacturing cost and ensuring good manufacturability. In this process, all the resources are assumed to be available. But in a real job shop, not all the generated process plans for a group of parts are schedulable according to the time and resource feasibility. In the traditional way to overcome this, it is necessary to iteratively re-invoke the process planning system to produce alternative plans for further trials until an acceptable scheduling solution is obtained. However, the above iterative process brings forth two serious problems in practical applications (Li and McMahon 2006).

(1) Firstly, it is quite tedious and time-consuming to search for a feasible solution to meet the requirements of process planning and scheduling simultaneously and an overall optimised target is even more difficult to achieve. Meanwhile, the value of a process plan can be severely discounted since the assumption that all resources are available during the process planning stage might not be fully valid in the scheduling stage. For instance, the generated process plans sometimes cause some machines to be overloaded, further to create bottlenecks whilst the capabilities of other machines are not fully utilised.

(2) Secondly, a job/batch shop is usually in dynamic adjustment due to the non-availability and maintenance of resources, or the arrival and release of new jobs. Such a dynamic shop floor brings challenges for the process planning system to accommodate the changes efficiently, and a new round of searching and compromise of the process planning and scheduling needs to be carried out again in their vast solution spaces.

So it is necessary to develop a closer integration of the process planning and scheduling systems to achieve an overall optimisation.

In the last decade, a number of research workers have addressed these two areas of optimisation using different Artificial Intelligence (AI) technologies such as Genetic Algorithms (GA), Simulated-Annealing algorithms (SA) and so on.

1.1.5 Five-Axis machining

Developments in the aeronautic and automobile industries brought new technological challenges, related to the growing complexity of the products and the new geometries modelled in CAD systems. These more complex geometries impose new challenging manufacturing situations for the development of new machining technology, namely 5-axis machining (Baptista and Antune Simoes 2000).

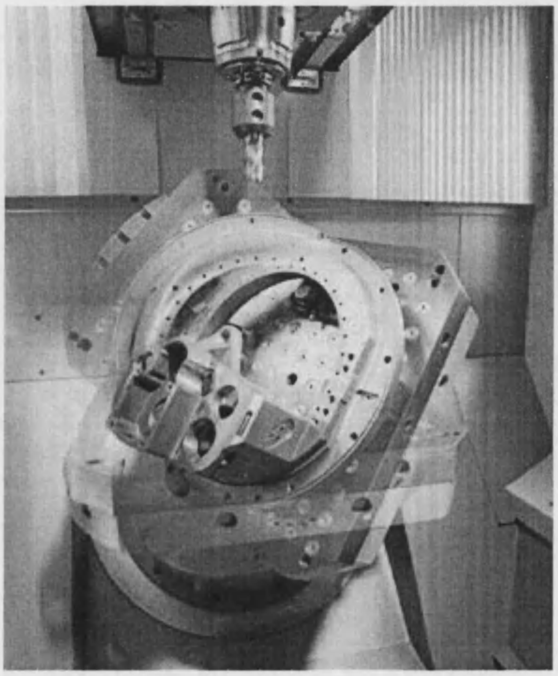


Figure 1.4 Worktable of 5-axis CNC machining centre

The number of axes of a machine tool normally refers to the number of degrees of freedom or the number of independent controllable motions on the machine slides. The ISO axes nomenclature recommends the use of a right-handed coordinate system, with the tool axis corresponding to the Z-axis. A three-axis milling machine has three linear slides X, Y and Z which can be positioned anywhere within the travel limit of each slide. The tool axis direction stays fixed during machining. This limits the flexibility of the tool orientation relative to the workpiece and results in a number of different set ups. To increase the flexibility in possible tool workpiece orientations, without the need of re-setup, more degrees of freedom must be added. Five degrees of freedom are the minimum required to obtain maximum flexibility in tool workpiece orientation, this means that the tool and workpiece can be oriented relative to each other under any angle. For a conventional three linear axes machine this can be achieved by providing 2 extra rotational slides (Bohez 2002). A work table of a 5-axis machining centre is shown in figure 1.4.

Since 5-axis machines have two more degrees of freedom than traditional 3-axis machines, 5-axis machining offers many advantages over 3-axis machining, including better tool accessibility, faster material removal rates, low setup cost, and improved surface finish (Mahbubur et al. 1997, Lo 1999, Ho and Hwang 2003). It allows parts to be machined with geometry that would have been very difficult, if impossible on conventional 3-axis machines. Furthermore, it allows parts with more straightforward geometry to be machined with significantly fewer set-ups. 5-Axis CNC machines are widely used in machining of sculptured surfaces such as turbine blades, impellers, propellers, 3D cams, moulds and dies. Some examples are shown in figure 1.5.

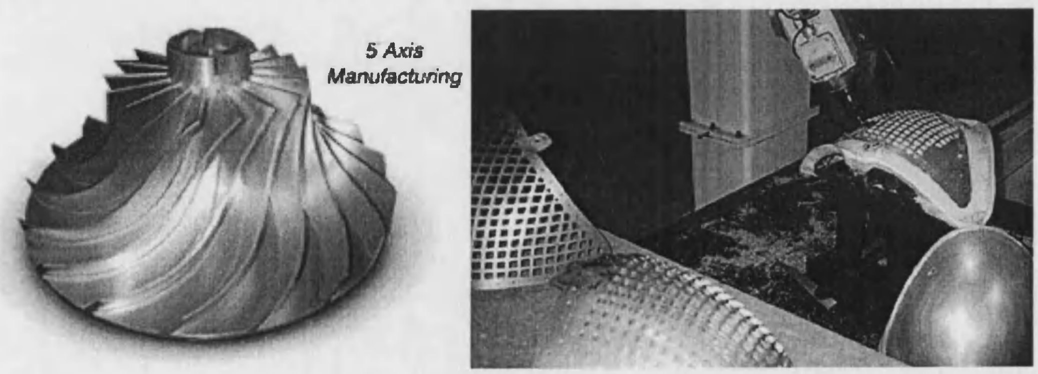


Figure 1.5 Products machined by 5-axis machine

Internationally, as the price of these machines has fallen over recent years, 5-axis machining centres have been used for machining normal prismatic parts. However, conventional 3-axis CAPP work has not been fully exploited in a 5-axis environment. Current research in 5-axis is focused on automatic tool path generation and in particular the activity of deciding the distribution of cutter locations to fulfil the requirements of machining high quality complex surfaces. There still remains a significant lack of knowledge in how to optimise the manufacturing process for conventional prismatic parts in a 5-axis environment.

1.2 Overall aims

The overall aims of this research are to develop a generative CAPP system for common prismatic components in the 5-axis CNC machining environment and develop an IPPS module to optimise the process planning and scheduling simultaneously.

The detailed aims and objectives will be discussed in chapter 3.

1.3 Organisation of the Thesis

The thesis is organized in 9 chapters as follows:

In Chapter 1, the background related to this research is introduced and overall aims of the research are given.

In Chapter 2, a review of the related research is discussed. Three topics that are relevant to this research – feature technology, optimisation of operation sequence and of Integration of process planning and scheduling, are reviewed.

Chapter 3 will give the aims and objectives and the methodology of this research is also given.

Chapter 4 presents the information philosophy of the system and gives the proposed system structure.

In Chapter 5, the feature-based model input to the system is first introduced. Meanwhile, feature representation using object oriented programming strategy is presented. Then the methods of selecting machining operations including tools, TADs, parameters are presented. Finally the calculation of machining time and output of machining operations are discussed.

In Chapter 6, a Particle Swarm Optimisation (PSO) algorithm is developed to optimise the process of machining operations sequencing using combined evaluation criteria, which include machining costs, cutting tool cost, machine changes, tool changes, and the number of setups. In this approach, some preliminary precedence constraints between features and operations are defined and manipulated. In this chapter, the difference between 3-axis and 5-axis process planning optimisation is discussed and PSO algorithms for both 3-axis and 5-axis are presented.

Chapter 7 presents a PSO algorithm for optimising the Integrated Process Planning and Scheduling (IPPS) problem. The problem is first defined, then the representations for process plans and schedules are given and different criteria of performance are discussed.

In Chapter 8, the implementations of the independent operation sequencing module and IPPS module are first presented and the corresponding case studies are discussed. Then the developed prototype CAPP system for 5-axis machining and its implementation are described. A case study is used to illustrate the capabilities and adaptabilities of the developed approaches.

Finally, Chapter 9 gives the conclusions and contributions of this research. The limitations of the developed systems are presented and Suggestions for future work are outlined.

Chapter2 Review of Literature

2.1 Introduction

As described in Chapter 1, process planning is concerned with the preparation of the procedure sheets that contain the processing steps by which the product should be manufactured. Since the concept of Computer Aided Process Planning (CAPP) was first conceived in the 1960's (Niebel 1965 and Schenk 1966), it has been continually developed. The impetus for the interest comes from two sources: firstly, industry in an attempt to increase productivity (Wang and Chang 1987), secondly, the fast development of computer software and hardware (Wang and Wysk 1988).

In the initial CAPP systems, the approach was to find optimum machining parameters and cutting conditions. Then, the approach evolved into report generation and documentation retrieval. In the latter case, Group technology (GT) was used to help locate similar parts, thus becoming process plans. It was not until ten years had elapsed that some kind of generative approach was developed. Although the introduction of AI and expert systems boosted both the interest in the problem and the capability of the systems, the results are still far from desirable.

In this chapter, some of the relevant existing CAPP systems for prismatic components are reviewed and discussed. Feature technology, an important element in process planning, is also examined. Then machining operations sequencing optimisation is reviewed and Integrated Process Planning and Scheduling systems are finally discussed.

2.2 Overview of CAPP approaches

There are two main approaches for developing a CAPP system: Variant and Generative. The variant technique uses the classification and coding of parts to initiate the process planning activity (Rembold et al. 1993). When a plan is to be generated for the production of a new product, a standard plan for a similar product is retrieved and modified for the new product. The plan may be a parameterised model of the part, and the user just enters the parameters of the part needed to be described. This approach is generally useful in cases where there are a lot of similarities between products. Typical examples include CAM-I'S CAPP, MIPLAN and MULTICAPP, which are described below.

CAM-I'S CAPP: is an acronym for "CAM-I's Automated Process Planning system" developed by Mc Donnell Douglas Automation Company (McAuto) under a contract from CAM-I (Link 1976). It is probably the first and also the most widely used of all process planning systems. CAPP is a database management system written in ANSI standard FORTRAN. It was developed primarily as a research tool to demonstrate the feasibility of computer assisted process planning, with logic based on group technology methods to classify and code parts. In CAPP, a structure is provided for a database, retrieval logic, and interactive editing capability. The coding scheme for part classification and output format are added by the user. A 36-digit maximum alphanumeric code is allowed. A coded scheme tailored to the user application is usually appropriate.

MIPLAN and MULTICAPP: Both MIPLAN (Schaffer 1980) and MULTICAPP were developed in conjunction with OIR (Organization for Industrial Research Inc.). They are both variant systems that use the MICLASS coding system for part description. They are data retrieval systems which retrieve process plans based on part code, part number, family matrix, and code range. By inputting a part code, parts with a similar code (user-defined similarity) are retrieved. The process plan for each part is then displayed and edited by the user. They are similar to the CAM-I CAPP system with MICLASS embedded as part of the system.

Since the 1990s, most research has focused on generative CAPP. But there are still a lot of variant CAPP and hybrid CAPP (variant and generative) being applied in industry. For example, IAI-CAPP proposed by Chang et al. (2000) combines variant and generative CAPP and is capable of generating plans that are either similar to existing workpieces or new plans. In IAI-CAPP, fuzzy logic (FL) and artificial neural networks (ANN) are integrated to perform the dynamic recognition and adaptive-learning tasks of the workpieces and process plans. Also, it adopts the idea of the important (critical) feature concept for evaluating the suitability of existing process plans for incoming product designs.

The Generative approach, on the other hand, does not use any stored standard plan. When a plan is generated, the system uses information about a part's geometry, machining or assembly data, machines (including robots) and their parameters, as well as process planning rules.

As figure 1.2 in Chapter 1 shows, the design data is transmitted to the planners by a modelling system which captures the design features, functions and general designer intentions for the product. This information along with process knowledge and raw material data are used to perform the process selection. The remainder of the processes are very close to the variant technique transactions. The major difference is that the CAD system data plays a major role in the generation of new plans and therefore the part description no longer needs to be done by codes which access pre-stored routes and plans. These can now be generated to fit the part geometry and manufacturing context of the geometry. Several generative process planning systems have been developed such as TIPPS (Chang 1985), BEPPS-NC (Zhang and Mileham 1991), BEPPS-GSCAPP (Rustom and Mileham 1992), GF-CAPP (Gonzalez and Rosado 2003) and PSG-CAPP (Sadaiah et al. 2002), and these are described below.

TIPPS is an acronym for "Totally Integrated Process Planning System" developed by Chang and Wysk at Virginia Polytechnic Institute and State University (Chang and Wysk 1985). In a sense TIPPS is a new generation of APPAS and CADCAM. It integrates CAD and generative process planning into a unified system employing AI and decision tree approaches. The system uses a special language called Process Knowledge Information (PKI) to describe the procedural knowledge and a CAD boundary representation as part data input. Using the terminal's cursor the user indicates the surfaces to be machined in order to determine manufacturing processes, sequence, machining parameters and time estimation.

BEPPS-NC is a generative process planning system for rotational parts developed at Bath University (Zhang 1991, Zhang and Mileham 1991, Zhang and Mileham 1989). It uses a 2D wire frame product model as an input, typical of various CAD systems in the format of DXF (Drawing Interchange File). The system is mainly composed of CAD interpreter, Process planner, NC code generator and BEPPS-NC viewer.

BEPPS-GSCAPP is a generative process planning system for prismatic parts developed at Bath University (Rustom 1992, Rustom and Mileham 1992, Rustom and Mileham 1990, Rustom and Mileham 1989). It is aimed at parts being produced on conventional machine tools in a batch manufacturing environment. The system is of particular interest since it was chosen as an application example to indicate this research work's potential for direct interfacing with a CAPP package. The BEPPS-GSCAPP consists of four main options: User's help, Process Planning, Decision logic modification and Database file modification. The options three and four enable the user to modify the decision logic files and database files when they need to be updated.

GF-CAPP system proposed by Gonzalez and Rosado (2003) generates the process plans. These constitute all of the alternatives for the required sequences and provide good flexibility in a standardized CAPP system that is claimed to be generally applicable in industry. These alternatives explicitly include feasible alternatives for machines and, as a consequence, alternative processes for operations.

PSG-CAPP proposed by Sadaiah et al. (2002) can be divided into three modules: the first module is concerned with feature extraction, the second and third modules deal with planning the set-up, machine selection, cutting tool selection, cutting parameter selection and generation of the process plan sheet. PSG-CAPP is claimed to have the ability to extract the majority of features from the CAD model and in generating process plans for prismatic components.

Compared with generative systems, variant systems are currently more mature and dominant in industry, but they are deficient in planning the processes of new products with many new features or structures. With keener global competition and the requirement of delivering new products more efficiently, the development of generative systems is imperative to facilitate process planning with higher flexibility and adaptability.

In recent years, researchers have focused their efforts on the following areas to improve the overall performance of generative CAPP systems:

(1) Applying new concepts or technology to the more general issue of automated process planning such as object-oriented systems (Chep et al. 1999, Zhang et al. 1999), feature-based systems (Patil and Pande 2002, Case and Harun 2000), agent-based systems (Sun et al. 2001, Wang et al. 2003, Gu et al. 1997, Maturana et al. 1999).

(2) Optimising specific aspects of CAPP systems mainly in operation sequencing. These include genetic algorithms (Qiao et al. 2000, Bhaskara Reddy et al. 1999), simulated annealing-based optimisation algorithms (Ma et al. 2000, Lee D.H. et al. 2001), Fuzzy Petri net algorithms (Wu et al. 2002) and some hybrid approaches (Li et al. 2002).

(3) In the area of integrating process planning with other production activities, a lot of research has been aimed at developing Integrated Process Planning and Scheduling (IPPS) systems, and includes work from Li and McMahon (2006), Yan et al. (2003), Zhang and Yan (2005), Morad and Zalzala (1999) and Kim et al. (2003).

The following sections will discuss current research in feature technology, optimisation methods along with some examples of CAPP systems and IPPS systems.

2.3 Feature Technology

As introduced in chapter 1, there are two key issues with CAD/CAM integration: (1) product design representation and reasoning capability from various product life-cycle considerations, such as manufacturability analysis, and (2) the capability of manufacturing process design to plan efficient and flexible manufacturing systems by exploiting the product information provided by these product design representations (Kim et al. 2001). Features encapsulate the engineering significance of portions of the product geometry and, as such, are applicable in product design, product definition, and reasoning about the product, in a variety of applications such as manufacturing planning (Shah and Mäntylä, 1995). Thus, features have been used as a means of interfacing in computer integrated manufacturing (CIM) through computer aided process planning (CAPP), and feature technology has been considered an indispensable tool for integrating design and manufacturing processes. Feature recognition and design by features are the two major approaches to create feature models (Bronsvoort and Jansen 1993).

The features generally refer to the design features and machining features (Fu et al. 2003). Design features are expressed in geometric terms, while machining features express explicitly the methods of production while implying the geometry and function of the features. For the process planner, the feature is commonly viewed as a machining feature (manufacturing feature) and can be used directly to plan the operations. Standard for the Exchange of Product (STEP) Application Protocol (AP) 224 defines machining features as classes of shapes representing volumes to be removed from a part by machining (Step Tools Inc.).

The following sections will briefly review the main considerations in machining process planning, concentrating on part representation, feature definition, feature taxonomy and feature representation.

2.3.1 Part Representation

As discussed previously, part description, the major component of generative CAPP, should consist of shape, dimension, tolerance, materials and surface conditions. Traditionally, this information is contained in engineering drawings, but 3-D geometric modelling in computer aided design (CAD) has developed as an alternative part representation in recent years. There are two main geometric modelling approaches. Boundary representation (B-rep) describes parts by the faces that bound them, in turn bounded by edges and vertices (Chang 1990). In the case of the CSG representation, the algorithms make use of the CSG tree made up of voluminal primitives. Purely geometric representations are, however, limited in their ability to support process planning. One of the techniques used by Shpitalni and Fisher (1991) is to convert the CSG tree into a Destructive Solid Geometry (DSG) tree so as to get primitives describing the position of the material, according to material cutting operations. Nevertheless, because of the limitations of conventional geometric models, high-level part representations have emerged based on features, which are modelling entities that combine geometric and other attributes with information about engineering intent (Shah and Mantyla 1995, Case et al. 1994).

The feature-based product information model (for manufacturing planning) identifies the design geometry in terms of holes, slots, pockets, bosses, fillets, chamfers and other design elements that can be machined. Manufacturing planning requires additional information such as the knowledge of the characteristic shape and features producible by the various processes and the process capability in terms of dimensions, locations, tolerances and surface roughness. The information should also cover more specialized factors such as tool accessibility, fixturing possibilities and the ability to be inspected (Sharma and Gao 2002). Many methods exist for creating feature models in a geometric-modelling context, for example Deneux et al. (1994) have listed three methods of workpiece feature representations:

(1) With the help of the manual definition of features according to a catalogue generally suited to the CAD system, possibly improved by the user.

(2) With the help of a direct design using design features stemmed from an expert system.

(3) With the help of automatic recognition of features from a solid model.

In the first method, the CAD model translates interactively in terms of geometric features. In the second method, the designer directly uses design features that allow generic forms of specific geometries with technical functionalities to be combined. In the case of design features, Gao and Case (1993) include the geometry, roughness tolerances, geometric relationships and material specifications, e.g. type of material, hardness and strength. In the third method, feature recognition is based on algorithms that extract geometric features from the CAD system database. The other information, that was previously proposed as comments (ASCII symbols), e.g. unusual tolerance or material, is presented as informative objects that can be more easily extracted from the CAD model.

In these three methods, the features used, extracted and recognised are not manufacturing features, as they do not take production processes into account. The first two methods of feature representation facilitate the inclusion of the activities of engineering design. The third method facilitates the inclusion of the manufacturing planner's activities that consist of describing the tasks to be undertaken without defining the real details. The method based on primary feature recognition from a

CAD model allows surfaces that cannot be split during the definition of detailed machining process, taking production means into account, to be defined. These surfaces are set according to three kinds of criteria: geometric, topological or tolerance and technological. These sets of surfaces have been called machining features. This solution allows the workpiece to be represented in terms of machining features. Thus, the methods of machining process design can be applied to each manufacturing feature independently.

2.3.2 Feature definition and Taxonomies

1. Feature definition

There are many feature definitions. One of the initial feature definitions is that proposed by Shah and Rogers (1988): A feature is a set of information related to the description of a part. And as described in the previous section, manufacturing features are what process planners are concerned with. To make manufacturing retrieval easier, four feature definitions have been specified by Chep and Tricarico (1999): form features, precision features, technological features and manufacturing features.

A form feature is a set of faces of the representation of the workpiece boundaries. These faces have topological links of concavity that form cavities in the product. For each form feature, it is possible to define a set of edges that are the limits of the form feature: boundary edges. The definition of the form feature is only based on topology.

A precision feature is a set of boundary faces which have tolerancing links. According to Jong et al. (1992), a precision feature can be divided into tolerance features and surface roughness features. Tolerance features define the tolerance deviations of nominal forms and measurements. Three sorts of tolerance features have been defined by Shah:

(1) Tolerance features that are coupled to one parameter of the form feature, e.g. the tolerance of the diameter of a hole.

(2) Tolerance features that affect the relationship between the geometric elements which are used to define a form feature, e.g. parallelism tolerance.

(3) Tolerance features coupled to relationships between form features.

With surface roughness features, surface processing can be linked to surfaces.

A technological feature is a set of boundary faces which have technological links. Technological features define material information of the workpiece, e.g. composition, physical and mechanical properties: heat treatments to be applied to the workpiece, and also surface treatments (thermal, thermo-chemical, mechanical).

Form features, precision features and technological features are called primary features. A manufacturing feature is a set of primary features. All form, precision and technological features that define a manufacturing feature are linked by constraints that can be of two types:

(1) Topological links that are links between different form features, characterized by a constraint.

(2) The existence of common boundaries (tolerancing or technological links) between different primary features is characterized by a constraint (Chep and Tricarico 1999).

2. Feature taxonomies

Instead of specifying all the geometrical and topological information that defines a feature for every separate feature type, it is possible to group features with common properties into classes. These can then be further divided into sub-classes to form a tree structure, or hierarchy. These classification structures are commonly called feature taxonomies and since they are of a hierarchical nature, the properties of a class can be inherited by its sub-classes.

There are two major benefits of using feature taxonomies:

(1) By allowing large amounts of varied features to be classified into coherent groupings, it helps in the recall of previously defined features, their subsequent editing, and the design of new features. The hierarchical description of features also allows simple features to be combined into more complicated features, and hence the final model (Case and Acar 1989).

(2) Feature taxonomies can provide a framework for the parametric generation of geometry at the design stage. Without a rigorous taxonomy, it is difficult to produce analytic and predictable algorithms for the complex task of process planning (Requicha and Vanderbrande 1988).

Several different taxonomies have been developed by researchers. In Gindy (1989)'s feature taxonomy, features are characterized by the number of orthogonal directions from which the feature volume might be approached. These are known as External Access Directions (EADs), and all features will have between 0 and 6 EADs. The three external access directions for a through slot are shown on figure2.1. Further classification on the basis of the type of profile (open or closed) and whether or not the feature volume penetrates through the component gives the nine basic feature classes (bosses, pockets, holes, non-through slots, through slots, notches, steps, real faces and imaginary faces).

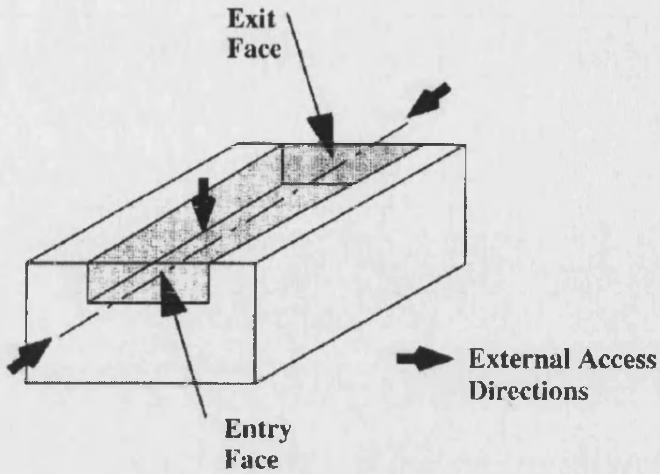


Figure 2.1 The topology and feature definition of a through slot (Case and Harun 2000)

In Patil and Pande's system (2002), features are primarily classified into two types:

(1) Gross Features. These resemble raw stock for CNC machining from which various feature shapes are machined out, which includes Rectangular and Contoured features.

(2) Local features. These represent families of features having varying geometries but the same topological characteristics (connectivity). Based on the manner in which features appear on the faces, the local features are further classified as follows:

- Face-based features. Holes, pockets, feature patterns-arrays of holes.
- Edge-based features. Slots
- Corner-based features. Steps.

2.3.3 Feature representation using Object Oriented Programming Strategy (OOPS)

After the feature taxonomies have been classified, the OOPS can be used in CAPP systems for the representation and hierarchical organization of feature data and associated process message services.

In IFPP developed by Patil and Pande (2002), OOPS is used to represent the features information. Figure 2.2 shows the information stored in the object oriented paradigm for a slot as an example feature. The feature object stores the location and identification attributes of the feature. It provides polymorphic user interfaces to input data for these attributes. The local object deals with the dimensional attributes of the feature (e.g. length, width, and the height for the rectangular slot). It also records the depth status (blind/through) of the feature. Information, such as tolerances and surface roughness, is stored as process data by this object. Polymorphic methods are used to validate the feature based on design and manufacturability issues.

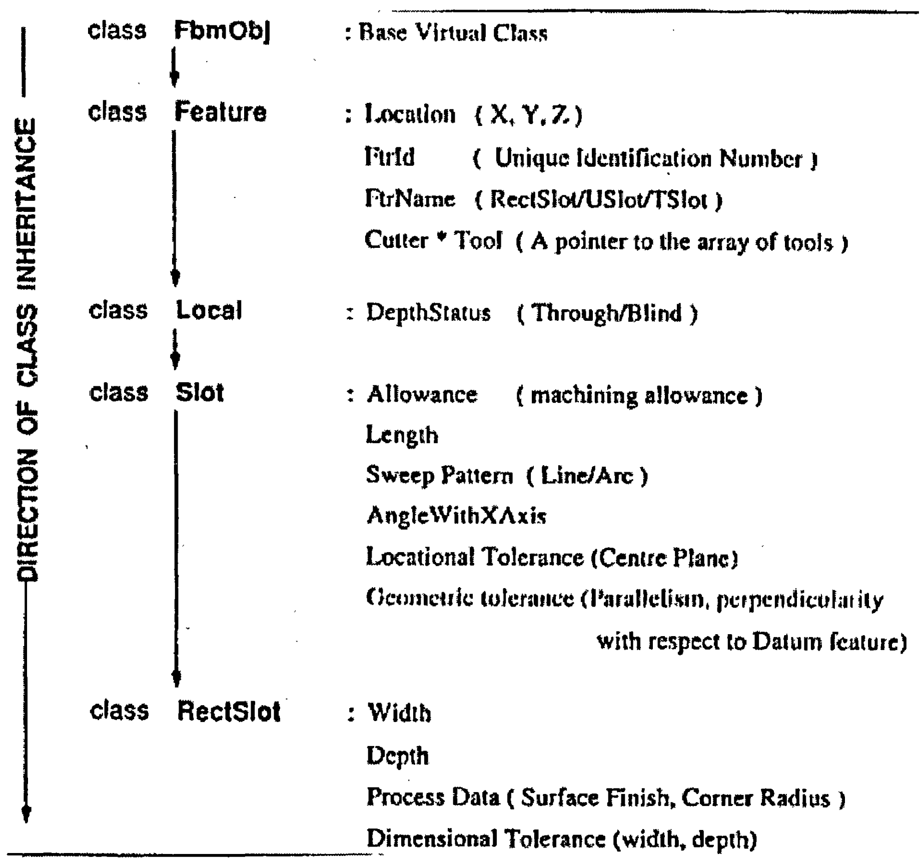


Figure 2.2 Typical information of a slot in OOPS (Patil and Pande 2002)

The main advantages of the object oriented design strategy is it provides the capability to organize and represent the feature information for easy message processing and offers the flexibility to modify the definition of an object, its structure, message and linkup without affecting the rest of the system configuration.

2.3.4 Standard for the Exchange of Product (STEP)

Preparation of product (feature) data for computer aided manufacturing planning is a difficult task, as features are domain dependent. This implies that the same design needs to be expressed in different feature-based descriptions to satisfy different downstream applications (Sharma and Gao 2002). There have been attempts to extract data directly (feature-based design) or indirectly (feature recognition) from the CAD database. In order to develop a procedure for consistent, unambiguous data abstraction from a generic data structure, the foundation or the standard on which the procedure is based is very crucial. Among many data exchange formats developed, Drawing Transfer File (DXF), Initial Graphics Exchange Standard (IGES) and STEP model data are the most widely used formats. In contrast to DXF and IGES, STEP (standing for standard for the exchange of product model data) is officially titled ISO 10303, and is aimed at defining a standard file that includes all information necessary to describe a product from design to production. It supports multiple application domains, for instance, mechanical engineering, electronics, architecture (Owen 1993). The following are some reasons for using STEP:

- STEP is a standard that can grow. It is based on a language (EXPRESS) and can be extended to any industry. A standard that grows will not be outdated as soon as it is published.
- The EXPRESS language describes constraints as well as data structure. Formal correctness rules will prevent conflicting interpretations. STEP CASE tools such as ST-Developer use these descriptions to create more robust, maintainable systems.
- STEP is international, and was developed by users, not vendors. User-driven standards are results-oriented, while vendor-driven standards are technology-oriented. STEP has, and will continue to, survive changes in technology and can be used for long-term archiving of product data.

In the STEP Application Protocols, Application Protocol 224 (AP224), the mechanical part definition for process planning using machining features, contains all of the information needed to manufacture the required part, including materials, part geometry, dimensions and tolerances, applicable notes and specifications, and administrative information.

Many CAD software packages now support the AP224 format drawing file and a new standard namely ISO14649, recognised informally as STEP-NC, is being developed which represents a data model for Computer Numerical Controllers. It may integrate CAD, CAM and CNC more easily in the future.

2.4 Operation Sequence Optimisation

2.4.1 Introduction

In developing computer-aided process planning (CAPP) systems, the determination of the operations sequence is one of the most important tasks and also a bottleneck task in the process (Qiao et al. 2000). Traditional CAPP approaches mainly aim at generating a single feasible plan for a given part. However, with the introduction of new manufacturing technologies, e.g. design for manufacturing (DFM) and the integration of process planning and job shop scheduling, the best process plan for a given part in a designated machining environment must be generated and fed back to the designer for evaluation. To support dynamic scheduling, a CAPP system must be able to generate plans with alternative routes and sequences to suit the variable status of the shop floor (Ma et al. 2000).

The operation sequencing problem can be defined as the problem of determining the sequence of operations required to produce a part with the objective of minimising the sum of machine, setup and tool change costs, while satisfying the precedence constraints among operations (Lee D.H. et al. 2001). The operations sequence generation problem can usually be modelled as a large-scale and combination optimisation problem with constraints. The complexity of solutions to the problem is highly dependent upon the shape and the number of features of a machined part. For instance, the process planning practice has shown that, for rotational parts, the topology relationships among most of the features comply with an explicit machining order of "from the left end to the right end' or vice versa and "from internal to external'. This can greatly reduce the number of operation combinations so as to generate a smaller search space, leading to various types of heuristics for operations sequence generation with less difficulty (Du and Huang 1990, Usher and Bowden 1996). However, for a prismatic part, this problem is intractable, with difficulties in the following areas (Li et al. 2002).

(1) The geometric relationships between features in a prismatic part are complicated, and the explicit heuristic rules for sequencing the operations corresponding to the features are deficient.

(2) Each feature might have several candidate tool approach directions (TADs), cutting machines and tools. The decision processes of selecting machining resources and set-up plans as well as sequencing operations are sometimes contradictory. The evaluation criteria coming from some aspects, such as minimum usage of expensive machines and tools, minimum number of setups, minimum number of machine and tool changes, and achieving good manufacturing practice, are also conflicting in certain cases. To carry out the different decision processes considering the evaluation criteria simultaneously is imperative to ensure a globally optimised solution. However, it is usually difficult for some reasoning approaches.

(3) In a part, there usually exist several alternative process plans that can achieve the predetermined optimisation objective. To generate and provide the alternative plans can help process planners make a reasonable decision according to the workshop

environment and fixture conditions. However, in the existing approaches, few contributions have been made to how this should be achieved.

Whereas tremendous efforts have been made in developing heuristic approaches to operation sequence generation for prismatic parts (Karinthi et al. 1992, Zhang et al. 1994), the sequencing problem is far from being solved (Qiao et al. 2000).

2.4.2 Optimisation Methods

Although a lot of CAPP systems have been reported in the literature, only a few have considered the optimisation of the operations sequence or the generating of alternative sequences.

In the knowledge-based reasoning approach, Chang (1990) and Chang et al. (1998) developed the QTC system, in which machining operations with the same TAD are aggregated as a setup. The sequence of the machining operations and setups is reasoned according to the precedence constraints, which stem from geometric interactions between operations, location tolerance requirements, reference or datum requirements, and good manufacturing practices. An optimum sequence is selected from several feasible sequences based on the minimum number of setups. Chu and Gadh (1996) expanded Chang's aggregation concept by clustering the operations that are machined with the same cutting tool into a setup so as to reduce the number of tool changes. In the APSS system reported by Wong and Siu (1995), the operations sequencing algorithm consists of three consecutive algorithms, viz., the transformation, refinement, and linearization algorithms. The transformation algorithm works on the geometric and technological information of a part and generates preliminary precedence constraints between features according to the "surface priority" and "process capabilities" knowledge bases. A tree structure is created to represent the necessary precedence of the operations. In the refinement algorithm, the details of the operations in the generated tree are enhanced and refined using the "refinement" knowledge base. For example, for a general drilling operation, the central drilling or pilot drilling operations are determined and specified. In the linearization algorithm, the tree structure is linearized into the final required operation sequence.

In the research by Kruth and Detand (1992), a generic Petri-net is used to represent a parametric feature and its related operations. After being evaluated using manufacturing knowledge bases, such as general machine data, machine axes data, and manufacturing capability data, the separated Petri-nets for compound features or features with identical TADs are first joined together. The same procedure is then applied to the features located in the different TADs, and a large Petri-net is finally formed, in which all valid alternatives to machine the part are described.

Lin and Wang (1993) presented integer-programming models for selecting and sequencing operations and tools for process plans with the objective of minimizing tool changeovers and solving them with commercial integer-programming software. Irani et al. (1995) proposed a graph-manipulation approach for operations sequencing. The Hamiltonian Path (HP) analogy for a process plan was developed and the Latin Multiplication Method (LMM) for constrained enumeration of all the feasible HPs was implemented. The optimal process plan is an HP that corresponds to the least number of set-up disruptions required from start to finish to process each feature once and only once. Lee et al. (2001) suggested two branch-and-fathoming algorithms to obtain optimal and near-optimal solutions for operation-sequencing problems with the objective of minimizing the sum of machine, set-up and tool change costs. They considered the precedence constraints and suggested systematic procedures to remove unfeasible and unpromising solutions, respectively. Kim et al. (2001) proposed a feature recognition based method to generate machining precedence relations systematically, based on the geometric information of the part. Tolouei-Rad (2003) proposed an efficient algorithm for automatic machining sequence planning in 2.5D milling operations, which generates feasible machining sequences based on the bilateral precedence between machining operations and results in minimized tool changes.

Conventional local search techniques have also been applied to various operation-sequencing problems. However, they still have some deficiencies: (1) since they are based on heuristic inferencing and reasoning, the search is not global and optimum plans might be lost during the reasoning processes; (2) in a complicated machining environment, the reasoning efficiency is low; and (3) the alternative operation sequences generated by some methods are feasible but not the optimal process plans (Li et al. 2002).

To resolve these problems, Evolutionary algorithms which are capable of searching globally in the whole search space have been applied to operation sequencing optimisation. In last decade, a lot of research has investigated the use of Genetic algorithms (GA) and Simulated Annealing (SA) algorithms in process planning.

1. Simulated Annealing algorithm (SA)

The SA algorithm is derived from the Boltzmann statistical mechanics. Since a SA occasionally chooses points uphill from its current point, it can escape from a local minimum and more effectively search the function space to find the global minimum. Thus, SA is often well-suited for solving constrained non-linear optimisation problems in a global search strategy. Brown and Cagan (1997) used the generative SA algorithm to search for the optimal process plan for rotational parts. Chen et al. (1998) used SA to solve the set-up sequence problem. However, these developed SA algorithms focus their search space in a rather limited domain or space.

Ma et al. (2000) proposed a simulated annealing-based optimisation for the operation selection and sequencing problem with the objective of minimising the sum of operation processing costs and change costs. The SA-based search algorithm can be generally described as follows.

Step 1: Randomly generate a feasible plan (OpM1; OpM2; . . . ; OpMn), called the current-plan.

Step 2: Start from the initial temperature T=T0, while not reaching the final temperature T_{lowest}.

{

Step 2.1: Make a random change to the current-plan, let temp-plan be the plan after the change.

Step 2.2: Check to make sure that temp-plan is valid. Otherwise, go back to step 2.1.

Step 2.3: Calculate the costs of current-plan (E1) and temp-plan (E2).

If E2 < E1

Let temp-plan be current-plan;

Else

Randomly generate X (0< X < 1);

If X < $e^{(E1-E2)/T}$

Let temp-plan be current-plan;

Else

Let current-plan remain unchanged;

End if

End if.

Step 2.4: Repeat steps 2.1-2.3 until a criterion is satisfied.

Step 2.5: Reduce the temperature to a new T.

}

Several important issues to be considered when applying a SA include:

(1) Representation schemes of solutions;

(2) Definition of the cost evaluation function;

(3) Definition of the neighbourhood mechanism for the generation of temporary solution;

(4) Design of a cooling schedule. The parameters in the cooling schedule are namely: an initial temperature, a temperature update rule, the number of iterations to be performed at each temperature step and a stopping criterion for the search.

2. Genetic Algorithm (GA)

Global search techniques like genetic algorithms (GA) for operation sequencing have been applied in Zhang et al. (1997), Qiao et al. (2000), BhaskaraReddy et al. (1999). The GA makes an analogy with the process of natural evolution by combining the `survival of the fittest' among solution strings with structured, yet randomized, information exchange and creates offspring having desirable characteristics. GAs require a method for representing an operations sequence as a string whose elements define a list of machining operations by considering some or all of the manufacturing constraints. GAs generate the optimal or near optimal result by following these steps:

(1) To generate the initial population (composed of strings called chromosomes, namely, operation sequences);

(2) To select chromosomes according to some reproduction strategies;

(3) To apply crossover and mutation operations.

These steps are repeated until an aspiration criterion is reached.

Several important issues to be considered when applying a GA to an application problem include:

(1) The representation of the parameters of the problem under study as chromosomes. There are two common representation methods for numerical optimisation problems: binary string representation and integer/real number representation. The representation method to use is determined by the ease of modelling the problem itself as well as the performances of the algorithm in terms of accuracy and computation time;

(2) A suitable fitness evaluation function to assess the quality of output is a mathematical equation. Where this method cannot be used, a rule-based procedure can be constructed;

(3) In the designed chromosomes, there are usually some precedence constraints. The crossover and mutation operations employed in a GA might cause the precedence constraints to be destroyed. The method to handle constraints and conduct search in feasible space is a major difficulty in applying GAs;

(4) The selection of a suitable procedure for each genetic operator for improving the efficiency and quality of the search is another issue. For example, in the selection operator, there are mainly two alternative procedures: proportional selection ("roulette wheel") and ranking-based selection. In the crossover operator, some

common alternative strategies include one-point crossover, two-point crossover and cycle crossover.

The choice or design of the control parameters in GAs depend on the problem and the representation schemes employed. Important parameters include the population size, and the crossover and mutation rates. These parameters should be designed for a general condition for the problem instead of being specific for a certain case study.

3. Problems with current optimisation approaches

Besides GA and SA algorithms, a Tabu Search (TS) algorithm has also been introduced in Lee et al. (2001) and Li et al. (2004). Different from SA, it defines a set of moves that are tabu to avoid cycling in the solutions. All these algorithms have been developed in the last decade and have made significant improvement in solving operation sequencing optimisation problems. However, there still remains potential for further improvements. These can be concluded as following:

(1) The representation of process plans (Operation-Tool-TAD) is still not complete as they do not include sufficient information especially for planning 5-axis machining. In 3-axis machining, a TAD indicates a determined set-up, but in 5-axis machining, the TAD of an operation can be achieved from 5 possible set-ups. This increases the difficulty of operation sequencing and set-up selection.

(2) Precedence constraints between operations need to be considered thoroughly and carefully so as to keep the solutions feasible. Different constraint handling mechanisms should be selected in terms of different characteristics of the algorithms.

(3) The performance evaluation criteria of a process plan need to be handled carefully, different criteria should be selected accordingly for different

objectives. For example, the process plan that can achieve the minimal machining time will often not be the process plan that has the minimal machining cost;

(4) Current algorithms are still not efficient. GA performs very well in the early optimising stage but later it is easy to be trapped into local optima so that it is not able to find an optimal solution especially for complex problems. SA converges fast and can find an optimal solution for most problems, but for a very complex problem, its probability of finding optimal solution is very low.

To improve overall performance, a more comprehensive representation scheme for the process plan needs to be developed, a more reasonable constraint handling mechanism needs to be developed and it is necessary to adopt a more agile, effective and efficient optimisation algorithm.

4. Particle Swarm Optimisation algorithm (PSO)

Particle Swarm Optimization (PSO) is a modern evolutionary computation technique based on a population mechanism. The PSO algorithm was inspired by the social behaviour of bird flocking and fish schooling (Kennedy and Eberhart 1995). Three aspects will be considered simultaneously when an individual fish or bird (particle) makes a decision about where to move: (1) its current moving direction (velocity) according to the inertia of the movement, (2) the best position that it has achieved so far, and (3) the best position that its neighbour particles have achieved so far. In the algorithm, the particles form a swarm and each particle can be used to represent a potential solution of a problem. In each iteration, the position and velocity of a particle can be adjusted by the following formulae that take the above three considerations into account. After a number of iterations, the whole swarm will converge at an optimised position in the search space.

$$V_i^{t+1} = w * V_i^t + c_1 * Rand() * (P_i^t - X_i^t) + c_2 * Rand() * (P_g^t - X_i^t) \quad \text{(Eq 2. 1)}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad \text{(Eq 2. 2)}$$

$$X_i = (X_{i1}, X_{i2}, ..., X_{iN}) \quad \text{(Eq 2. 3)}$$

$$V_i = (V_{i1}, V_{i2}, ..., V_{iN}) \quad \text{(Eq 2. 4)}$$

Here, i is the index number of particles in the swarm; t is the iteration number; V and X are the velocity vector and the position vector of a particle respectively. For an N-dimensional problem, V and X can be represented by N particle dimensions as equations 2.3 and 2.4 show. P_i is the local best position that the ith particle has achieved so far; P_g is the global best position that all the particles have achieved so far; w is the inertia weight to adjust the tendency to facilitate global exploration (smaller w) and the tendency to facilitate local exploration to fine-tune the current search area (larger w); $Rand()$ returns a random number in [0,1]; c_1 and c_2 are two constant numbers to balance the effect of P_i and P_g.

The PSO algorithm was initially developed for continuous optimisation problems. Recently, there has been successful research focused on discrete problems such as the Travelling Salesman Problem (TSP) (Wang et al. 2003, Pang et al. 2004 and Onwubolu and Clerc 2004) and the scheduling problem (Jerald et al. 2005).

However, the current PSO algorithm has not been applied to resolve the operation sequencing optimisation problems. Besides the common difficulties mentioned above, there are two major reasons due to the following characteristics of the PSO:

(1) Due to the inherent mathematical operators, suitable schemes to represent process plans by particles and how to determine the sequence of each process plan (particle) needs to be developed. And it is difficult for the current PSO algorithm to consider the different arrangements of machines, tools and TADs for each operation, and therefore the particle is unable to fully explore the whole search space.

(2) The PSO algorithm also suffers the drawback of becoming trapped in a local optimum. So it is necessary to develop new operators besides its mathematical operators to help it to escape from the local optimum.

2.5 Integrated Process Planning and Scheduling (IPPS)

As discussed in chapter 1, in a complex manufacturing situation, it is ideal to integrate the planning and scheduling more closely to achieve a global optimum in manufacturing, and increase the flexibility and responsiveness of the system.

In the past decade, there have been several attempts to address the integration of process planning and scheduling. Tan and Khoshnevis (2000) presented a review of the research in the process planning and scheduling area and discussed the extent of the applicability of the various approaches. More recent work can be generally classified into two categories: the enumerative approach and the simultaneous approach.

In the enumerative approach, all of the possible alternative process plans for each part are first generated. A schedule is then determined by choosing a suitable process plan of each part from their alterative sets according to the current resource constraints of a job shop and scheduling performance criteria (Tonshoff et al. 1992, Zhang and Mallur 1994; Zijm 1995, Sormaz and Khoshnevis, 2003, Kumar and

Rajotia 2003). Various strategies have been developed to exhaustively identify the possible alternative process plans based on multiple candidate manufacturing processes, set-up plans and manufacturing resources. In the FLEXPLAN system (Tonshoff et al. 1992), a Petri-net has been used to model and analyze the flexibility of process planning, and an AND/OR graph has been developed to represent the generated alternative plans. Based on the process plans, a strategy to pursue the minimum process time has been used to select the most suitable plan for each part from the scheduling point of view. The IPPM (Integrated Process Plan Model) system is another example with this approach (Zhang and Mallur 1994). A decision matrix has been first developed to represent and store all of the possible process plans generated using different set-ups and machine tools. In the matrix, the fuzzy logic technique has been incorporated to represent the imprecise information in the selection of the set-ups and machine tools. A scheduler then chooses a suitable process plan based on the shortest processing time principle.

Computer-Integrated Process Planning and Scheduling (CIPPS) developed by Aldakhilallah and Ramesh (1999) consists of four specific modules for automated feature recognition, the determination of minimal cover sets of all features of a product, the determination of an efficient and feasible process plan, and the generation of an efficient and feasible cyclic production schedule, respectively. Sormaz and Khoshnevis (2003) summarized a methodology for generation of alternative process plans in the integrated manufacturing environment consisting of four steps: selection of alternative machining processes, clustering and sequencing of machining processes, and generation of a process plan network.

However, the common drawbacks of the above research work are: (1) it is quite time-consuming to randomly identify all possible alternative process plans for complex parts. (2) Cyclic scheduling with alternative process plans to refine and achieve an optimal result is tedious and not efficient. Through a number of experimental computations, Usher (2003) concluded that the advantage gained by increasing the number of alternative process plans for a scheduling system to choose from diminishes rapidly when the number of the plans reaches a certain level.

The simultaneous approach is more effective and efficient in integrating the two functions. In this approach, the process planning and scheduling are both in dynamic adjustment until specific performance criteria can be satisfied. To facilitate the process, intelligent evolutionary algorithms, such as GA, SA, and heuristic rules, have been employed to generate optimised solutions to satisfy the constraints and objectives of process planning and scheduling simultaneously. A bi-criterion hierarchical approach is proposed in Brandimarte and Calderini's work (1995). The process planning module produces good process plans with low operation costs first. Then if a schedule generated based on the process plans is not satisfied using the makespan criteria, a heuristic procedure is invoked to reallocate some critical operations to alternative machines. In Zhang et al.'s work (2003), a facilitator is used to coordinate communications and interactions between the process planning module and scheduling module until both objectives of process planning and scheduling are satisfied. Through this guided refining process, the satisfactory solution can be achieved more efficiently than the enumerative approach.

To further enhance the algorithms performance, some unified optimisation models and algorithms have been developed (Morad and Zalzala, 1999, Kim et al. 2003;

Zhang and Yan, 2005, Moon and Seo 2005). Morad and Zalzala (1999) developed a GA-based integration scheme, in which process plans were represented as chromosomes, and crossover and mutations operations were used to explore the alternative process plans to achieve different objectives including the minimum makespan, set-up cost, or tardiness. Kim et al. (2003) developed a single optimisation model to integrate the process planning and scheduling. In this work, three rules, which are operation flexibility, sequencing flexibility and processing flexibility, have been employed to generate multiple process plans. From these multiple plans, a symbiotic GA has been used to search for an optimised process plan that satisfies scheduling objectives, such as the minimum makespan or the mean flow time. Zhang and Yan (2005) developed an optimisation model to combine the considerations from process planning and scheduling, such as the production cost, the tardiness time, the set-up cost, and the early finish time. Based on these, an improved hybrid GA-based approach was designed to optimise planning and scheduling simultaneously. Moon and Seo (2005) proposed a mathematical model to formulate this integration problem and used a GA-based algorithm to determine the optimal schedule of machine assignments and operations sequences to achieve minimised makespan. A SA-based algorithm developed by Li and McMahon (2006) is used to optimise the process planning and scheduling simultaneously with a combined model.

From the review of current research, the following issues are still outstanding:

(1) First, in process planning and scheduling, different criteria are used to address specific practical cases. For instance, from the process planning perspective, the lowest manufacturing cost is usually a desired target, while the scheduling

usually needs to look for the most balanced utilisation of machines, the minimum number of tardy jobs, the shortest makespan, etc. To meet the various requirements in practical situations, further improvement is required on the optimisation algorithm to make it more adaptive to accommodate diverse objectives for users to choose from.

(2) Second, both process planning and scheduling are *NP*-hard (Non-deterministic Polynomial) combinatorial optimisation problems. There are two major difficulties in IPPS Compared to optimisation of operation sequencing for a single part, (1) the search space of IPPS is much bigger than that of operation sequencing; (2) the optimisation problem becomes more complex as the number of parts increase, and which also needs to consider complex manufacturing constraints, such as operation precedence constraints and manufacturing resource constraints. All of these will increase the computation time dramatically.

(3) Third, current developed systems do not consider that dynamic changes of the shop floor's situation, such as routine machine maintenance, machine break down and new orders arrivals, are able to be inserted into the current schedule to meet the deadlines. Any occurrence of these situations will probably make the current schedule infeasible and result in the need to replan the whole schedule. The process of replanning is more complex and time consuming due to new operation precedence constraints and manufacturing resource constraints.

The above issues need to be considered when building the IPPS model. And to improve system performance, the optimisation algorithm also needs to be more adaptive and efficient by adopting more intelligent heuristics and search strategies.

2.6 Summary

In this chapter, the variant and generative approaches of CAPP, feature technology, operation sequencing, IPPS and related optimisation methods have been reviewed. As can be seen, the focus of developing CAPP systems has changed to applying new concepts or technology to current CAPP systems, optimising specific aspects of CAPP such as operation sequencing and integration of CAPP with scheduling and production planning etc. Much research has been carried out in these areas. However the practical implementations of these methodologies are still far from satisfactory. One of the primary reasons is that these problems involve complex decision-making processes, and current algorithms do not cover the whole range of these processes. It is necessary to adopt a more steady, adaptive and efficient algorithm to optimise the operation sequencing and IPPS problems. Furthermore, most of the research is based on 3-axis machining. It is known that with two more degrees of freedom, 5-axis CNC machining reduces the number of setups, but it also increase the difficulty of decision making in process planning. For every feature, the Tool Approach Direction (TAD) can be achieved by different setups due to the ability of the workpiece's rotational movements. And there maybe more than one TAD for specific features such as steps and holes, which not only increase the complexity of decision making but also lead to more possible sequences to produce a part, which enlarge the search space considerably.

Therefore, it is necessary to create a flexible CAPP system for 5-axis CNC machining and develop an adaptive and efficient algorithm for optimising the operation sequencing and IPPS problems. This is the major objective of this thesis.

Chapter 3 Aims, Objectives and Methodology

This chapter presents the specific aims and objectives of the project, describes the options available to research this type of problem and why particular methods have been considered applicable to this research project.

3.1 Aims of the Research

The main aims of the research are:

a) To develop a CAPP system in the 5-axis CNC machining environment.

b) To develop a method to integrate CAPP and scheduling efficiently and implement it for a job shop manufacturing environment.

The main part of this research focuses on developing a CAPP system for the 5-axis CNC machining environment. As discussed in chapter 1, the use of 5-axis machining centres has grown such that they are now used for machining normal prismatic parts but the conventional 3-axis CAPP work has not been fully exploited in 5-axis. Current research in 5-axis is focused on automatic tool path generation and in particular the activity of deciding the distribution of cutter locations to fulfil the requirements of machining complex surface shape. There still remains a significant lack of knowledge in how to expand 3-axis CAPP methods into 5-axis and optimise the manufacturing process for conventional prismatic parts.

To support a job shop manufacturing environment that is characterized by the make-to-order operation and the demands of small volumes with a large variety, it is beneficial to integrate CAPP and scheduling systems to optimise the whole process simultaneously to meet the demands of customers quickly and reduce the manufacturing cost.

3.2 Objectives of the Research

To achieve the aims of the research the following objectives are identified:

- To specify a 5-axis CAPP system.
- To develop a 5-axis CAPP system that is based on the input of a feature-based model and which includes
    - Routings which specify operations, operation sequences, work centres, standards and tooling.
    - Optimised process plans which typically provide more detailed, step-by-step work instructions including dimensions related to individual operations, machining parameters and set-up instructions.
- In developing the CAPP system, the objectives will be to:
    - Define the feature-based model to represent the part including all the dimensions, tolerances and roughness etc. as the input of the system.
    - Determine the feasible operations based on the input.
    - Determine machining parameters for generated operations and calculate the approximate machining time.
    - Build the appropriate representation for the process plan.
    - Optimise the operation sequence to achieve the minimal cost/time.
- To develop an Integrated Process Planning and Scheduling (IPPS) module that can optimise process plans and scheduling simultaneously to achieve an overall optimal objective.

3.3 Methodology of the Research

This sub-chapter describes some of the available research methods and why particular methods have been considered applicable to this research project.

3.3.1 Overview of Research Methods

A lot of research of methodology has been done in last few decades (Clarke 1972, Rose 1982, Trafford 2001, Chatting 2001, Mebrahtu 2005). Research is a scholarly or scientific investigation or enquiry that requires thorough study so as to present findings in a detailed and accurate manner (University of Bath, Mechanical Engineering course in research methods – ME50173, 2006). Doing research has two elements:

- Empirical knowledge: acquiring data, observations, facts, cases, etc.
- Theoretical knowledge: laws, principles, models, concepts, etc.

Generally, a research process follows either a deductive or inductive approach. The deductive approach first finds a theory (or proposal) and is then tested with data. This is more appropriate to most engineering types of research. The inductive approach gathers data and then thinks of a theory. The inductive approach is more suited to social sciences and humanities research.

There are a number of research classifications in the literature. Clarke (1972) and later, Howard & Peters (1990) classify forms of research as pure basic, basic objective, evaluative, applied and action. Alternatively, Philips and Pugh (1987) argued that the classification of pure and applied research is too simplistic and preferred to classify research as exploratory, testing out and problem solving research. However, within each of these classifications certain methodological problems have to be considered and resolved. These concern how to aggregate different clusters of independent data, the relative importance of analysing data gathered at different levels, and the wider issue of sampling frames for data collection (Bryman, 1989).

Trafford (2001) writes in appreciation of Kuhnian notion of paradigms which explain and produce significant shifts in understanding. Kuhn (1962A) suggested that scientific paradigms are examples of actual scientific practice, examples which include law, theory, application and instrumentation together ... to provide models from which spring particular coherent traditions of scientific research. According to Kuhn, paradigms are also the source of the methods, problem field, and standards of solution accepted by any mature scientific field at any given time (Kuhn, 1962B).

Burrell and Morgan (1979) present four assumptions about the nature of research: ontology, epistemology, human nature, methodological. This approach is more inclined towards research in social studies and includes the idea that hypotheses could be expressed which try to capture theoretical explanations of practice – by researchers who have incorporated these assumptions about how their research has been designed.

Rose (1982) produced a model which is also represented by Traford (2001) that shows how the key components of research are systematically related to one another by linking theory and evidence. He developed an ABCDE model as shown in Figure 3.1.

A. Theory: an explanatory statement about the phenomena.

B. Theoretical proposition: specific propositions to be investigated in the study.

C. Operationalisation: decisions made on how to carry out empirical work; technique of data collection; sampling; concepts and indicators, variables; units.

D. Field work: collecting data, practical problems of implementing stage C decisions.

E. Results: data analysis leads to findings; interpretation feed back to C, B, A.

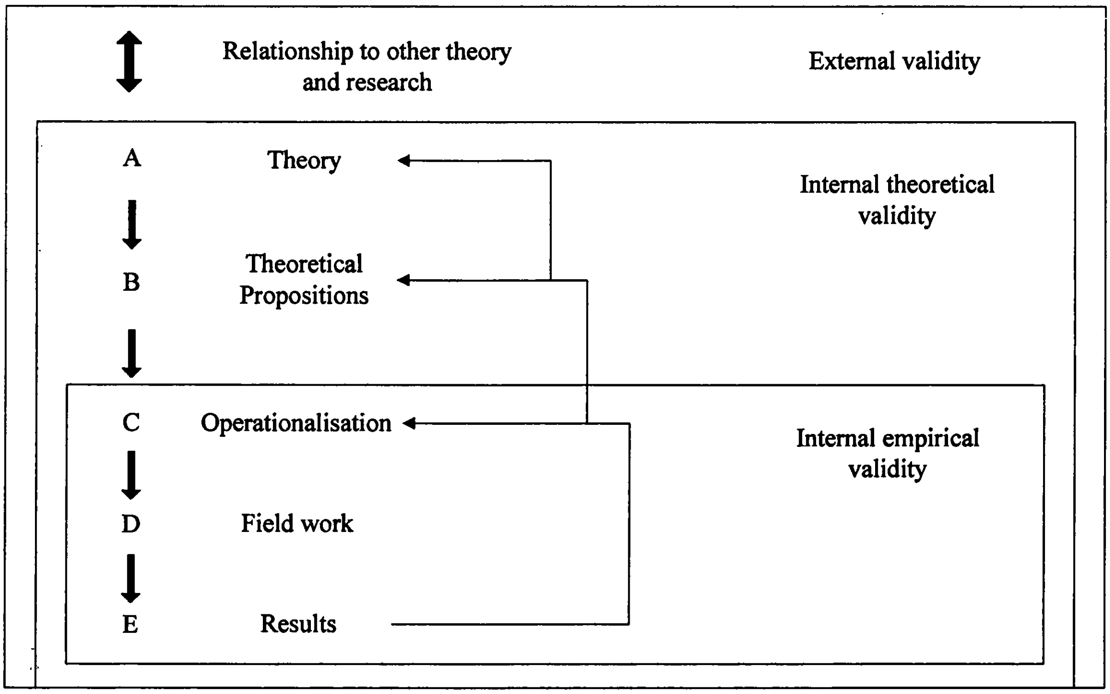


Figure 3.1 Rose's ABCDE model and distinction between three kinds of validity in Research (Trafford, 2001)

The model illustrates how researchers have justified progressing through each stage from theory to results. Rose developed the route further by indicating that by tracing back through the E-C-B-A route the validity of the research process can be evaluated. The significance of the model is that the C point in each model is central both to the developmental process as well as to the evaluative processes. The research assumptions from A to C relate to conceptual issues, whilst those from C to E relate to operational issues.

An approach described by Walton & Gaffney (1991) specifies a research cycle that comprises the following stages:

(1) Identification of a study topic

(2) Operationalisation of a hypothesis

(3) Selection of an observation sample

(4) Selection of a research method, gathering of data and generation of findings

(5) Derivation and dissemination of the implications for theory and practice

3.3.2 Research methods in Engineering

Although research methods in science and in engineering have plenty in common, according to the University of Bath, Mechanical Engineering course in Research methods – ME50173 (2006), they have some conceptual differences that include:

- Engineering incorporates science but also rules of thumb
- Engineering is "know how" not "knowing that"
- Engineering seeks safety but science seeks truth
- Engineering tries to avoid being refuted yet science tries to refute

Blockley & Henderson (University of Bath, Mechanical Engineering course in Research methods - ME50173 2006) describe engineering processes as having the following steps

- Encounter a problem
- Propose a solution
- Assess the consequences
- Decide how to embody the solution
- Embody it
- Test it

- Learn how dependable it was

As an extension of Blockley & Henderson's description of engineering processes, for a specific piece of engineering research, the slides of ME50173 (2006) show the following steps;

- Fix the basic area of the work
- Find out what is already known (review of previous work)
- Identify the problem or gap exactly (problem definition and hypothesis generation)
- Develop a precise objective
- Perhaps propose and build a trial artefact
- Collect data on its performance
- Analyse the data
- Draw conclusions
- Disseminate findings.

3.3.3 Methods for this research Project

This research project, as described in sections 3.1 and 3.2, deals with an applied manufacturing problem and thus falls into the engineering research category. The background and broad description of the research have been explained in Chapter 1.

Literature Review

Literature surveys include primary (such as archival journals, theses and dissertations), secondary (review journals, monographs and textbooks) and tertiary (indices, catalogues, encyclopaedias, bibliographies) literature (Chatting 2001, Mebrahtu 2005). The literature review conducted in this research included a current review from the academic and industrial points of view.

As shown in chapter 2, literature on CAPP system approaches, feature technology, operation sequence optimisation, integrated process planning and scheduling systems and optimisation methods were thoroughly surveyed and those relevant to the research reviewed. The literature review has been used extensively to aid the understanding of current research activities of CAPP systems and technologies, to avoid repeating research, developing aims and objectives to aid in the construction of hypotheses.

Problem definition and hypotheses

A broad definition of a problem is posed in the introduction section and summarised at the start of this chapter. Following an extensive literature review, the gaps in the CAPP systems for the 5-axis CNC and integrated process planning and scheduling environments were clearly identified and hypotheses generated. The hypotheses can be summarised as:

1) The difference between the development of 3-axis CAPP and that of 5-axis CAPP systems can be identified and 5-axis CAPP methods can be derived from 3-axis CAPP with consideration of characteristics of 5-axis CNC machining.
2) Feature-based model can be used to represent the part efficiently and Object Oriented Programming (OOP) can help to organise and represent the feature information for easy message processing and make the system extensible which is easy for post-development.
3) A new optimisation algorithm, i.e. Particle Swarm Optimisation (PSO) algorithm can be used to optimise the operation sequencing to improve the performance of CAPP system.

4) There is a need to build a model to resolve the optimisation of an IPPS system and it is necessary to apply a feasible algorithm to optimise it.

Developing precise objectives

The aims and objectives of the research have been described in section 3.1 and 3.2.

Proposing and building a prototype system

To achieve the aims and objectives and realise the hypotheses, a prototype CAPP system structure is first proposed and then implemented. Also the IPPS model is developed and implemented.

The proposed CAPP system comprises of four main modules: part information input module, operation selection module, machining parameter determination module and operation sequencing module. The details of the proposed system and development of these modules are shown in chapter 4. The representation of the IPPS model and optimisation method is also proposed and then implemented, which is independent to the proposed CAPP system.

Data collection and analysis

This research focuses on building a 5-axis CAPP system with a generative approach. As discussed in chapter 1, knowledge base, part representation and inference engine are three main components of generative CAPP system. The system is required to make decisions on various aspects of process planning in terms of the technical knowledge of manufacturing and the planning logic contained in the knowledge base to a given part description. The knowledge needs to be collected from different resources such as experiences which have been summarised by other researchers or process planners, machinery's handbooks, British Standards, tooling companies' catalogues and textbooks etc. Most of them are empirical and have been proved by real production. After the knowledge is organised, it can be stored in a database and coded into rules to help make decisions.

Case Studies

One of the most significant methods of research used to examine the industrial application of product development tools and methods is that of case studies (Chatting 2001, Mebrahtu 2005). The use of which, has been extensively documented by Yin (1993, 1994) and Johnson and Johnson (1997). In the context of this research, case studies are mainly used to test the validity of the system. It is imperative to ensure the validity of process plans generated by the CAPP system. In every step of the process of generating process plans, the data needs to be valid and feasible. Otherwise the rules and knowledge database need to be checked and modified to make sure output is correct. Also by using some well-known case studies by other researchers, the performance of the system can be identified. For example, by comparing performances of the PSO, GA and SA algorithms, the benefits of different optimisation algorithms can be identified and the parameters of the PSO algorithm can be determined and refined; which enhances the performance of the whole system.

Conclusion and dissemination of findings

Analysis and findings of experimental data from the case studies helped to conclude that the research substantiated the hypotheses. The findings are disseminated as a thesis, publications in conferences and journals.

Chapter 4 Proposed System

4.1 Introduction

The proposed system, in line with the aims and objectives, has an integrated generative 5-axis CAPP system and an independent IPPS module. The CAPP system is intended to generate an optimised process plan based on the input of a prismatic component from a CAD model. Note that optimised has been used to describe a best or towards optimal solution throughout this thesis. With the concentration on the operation selection, and operation sequencing optimisation for 5-axis CNC machining, the system does not include a feature recognition module which is replaced by a manually machining feature input module. The development of the feature recognition system is felt to be outside the scope of this thesis and represents an area for future study (see chapter 9). The IPPS module is designed to generate an optimised schedule including detailed process plans for a group of components in job workshop environment.

4.2 5-axis CAPP system

As described in chapter 1, when given a part description, a generative CAPP system uses an inference engine to generate process plans by applying the planning logic and process knowledge contained in the knowledge base. So a generative CAPP system is made up of functional modules and a knowledge database.

1. Proposed functional modules

As discussed in chapter 1 and chapter 2, there are several major activities in process planning, which also form the major functional modules of a generative CAPP system:

i) **Feature information input module** - A part is described using commonly used machining features together with their technological attributes, such as tolerances and surface finishes. Geometric tolerances have not been considered in this research. As discussed above, for simplification, the system does not as yet generate the features automatically from the part drawing designed using Unigraphics instead. It is required to input the feature information manually. In this research, 5 basic features are used, i.e. Face, Hole, Pocket, Slot and Step. For each feature, the user is asked to input not only their shape specifications but also their quality specifications. The detailed feature classification, representation and information needed for the next stage of process planning are presented in chapter 5.

ii) **Operation selection module** – This module aims to determine one or several operations required for each feature. This includes the selection of applicable machines, cutting tools, and tool approach directions (TAD's) based on the feature geometry and available machining resource. In this module, the system uses knowledge-based heuristic rules to generate alternative machines, cutting tools and TADs for a specific operation.

iii) **Cutting conditions calculation module** – When the operations including applicable machines, tools and TADs are generated, the cutting speed, spindle speed and feed rate can be selected and calculated. Then the machining time for each operation is estimated.

iv) **Operation sequencing module** – This module comprises of two parts: precedence constraints determination and operation sequencing optimisation.

The former will determine the precedence relationships between all the operations generated in module (ii). And the latter aims to determine the optimal sequence of all operations so that the precedence relationships among all the operations are maintained and the total machining time is minimised. In this module, a population-based Particle Swarm Optimisation (PSO) algorithm is developed to achieve this objective. Finally the optimal process plan is output.

2. Knowledge database

As a core part of a CAPP system, the knowledge database not only maintains machining resources, including the information of machines, tools and so on, but also maintains the machining technology knowledge which is used to help the function modules make decisions in generating the process plans. Also, the temporary information generated during the course of process planning needs to be maintained in the knowledge base.

A suitable database needs to be designed and requires a large amount of previous work including analysis, formalisation and representation of various manufacturing parameters and constraints, expert knowledge and experiences. To implement the above functional modules, the following databases have been designed:

1) Machining feature database: consists of the feature information for different parts, including feature type, feature dimension, tolerance, and surface roughness and so on. The feature information of a part is input into the system manually through a user interface and will be transferred to the operation selection module. The detailed design of the database is described in chapter 5.

2) Machining resource database: This includes two major resources. A) Machine data: consisting of all the data about the 5 axis CNC machine such as its capability, assigned tooling, coolant, spindle/axis, and the range of workpiece speeds and so on. Note that all process plans in this research have been generated for a DMU 50 eVolution 5-axis CNC. All this data can be used for TAD selection, machinability checking, set-up planning and machining parameter calculation. B) Tool data: composed of available tool types, dimensions, tool conditions and so on. This can be used for selecting suitable tools for each operation based on the tool capability.
3) Machining technology knowledge base: including the machining process capability knowledge (shape producing capability, dimension, tolerance and surface properties capabilities), process constraints knowledge (geometric constraints and technological constraints) and process economics (machining time, machining cost, tool cost, tool change time et al.). The operation selection, tool selection, machining parameters calculation and precedence constraints determination are all based on this knowledge base.

There are two schemes to represent knowledge in this research:

1) **Database scheme**: A Relational database is the most popular tool to store the factual knowledge due to its convenient management. The factual knowledge can be updated and edited easily, which does not require revising the programming of the system. It is difficult and time consuming to revise the code of systems especially when the software has been packaged. So most knowledge is stored in relational databases, in this research, it is a Microsoft Access database.

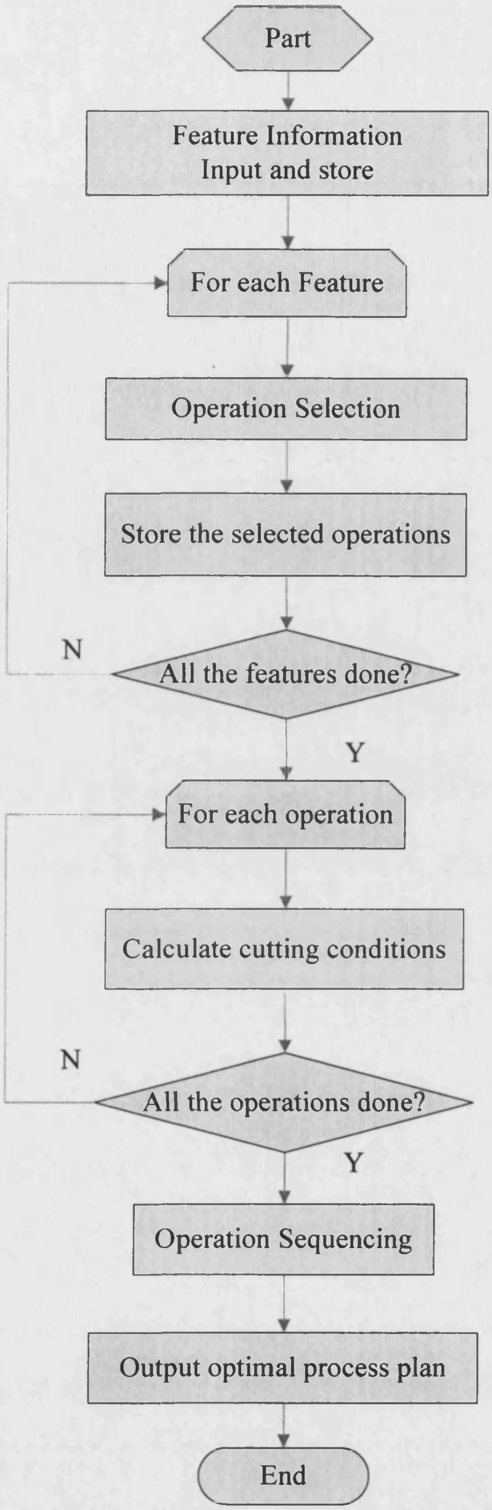


Figure 4.1 Information flow of the proposed system

2) **Production rule scheme**: This is designed to control the decision making procedure in operation selection, tool selection, TAD selection, machining

parameter calculation and operation sequencing base on the knowledge stored in the database. Also for the knowledge that can not be represented or stored in the database, is represented by rules and coded into system program.

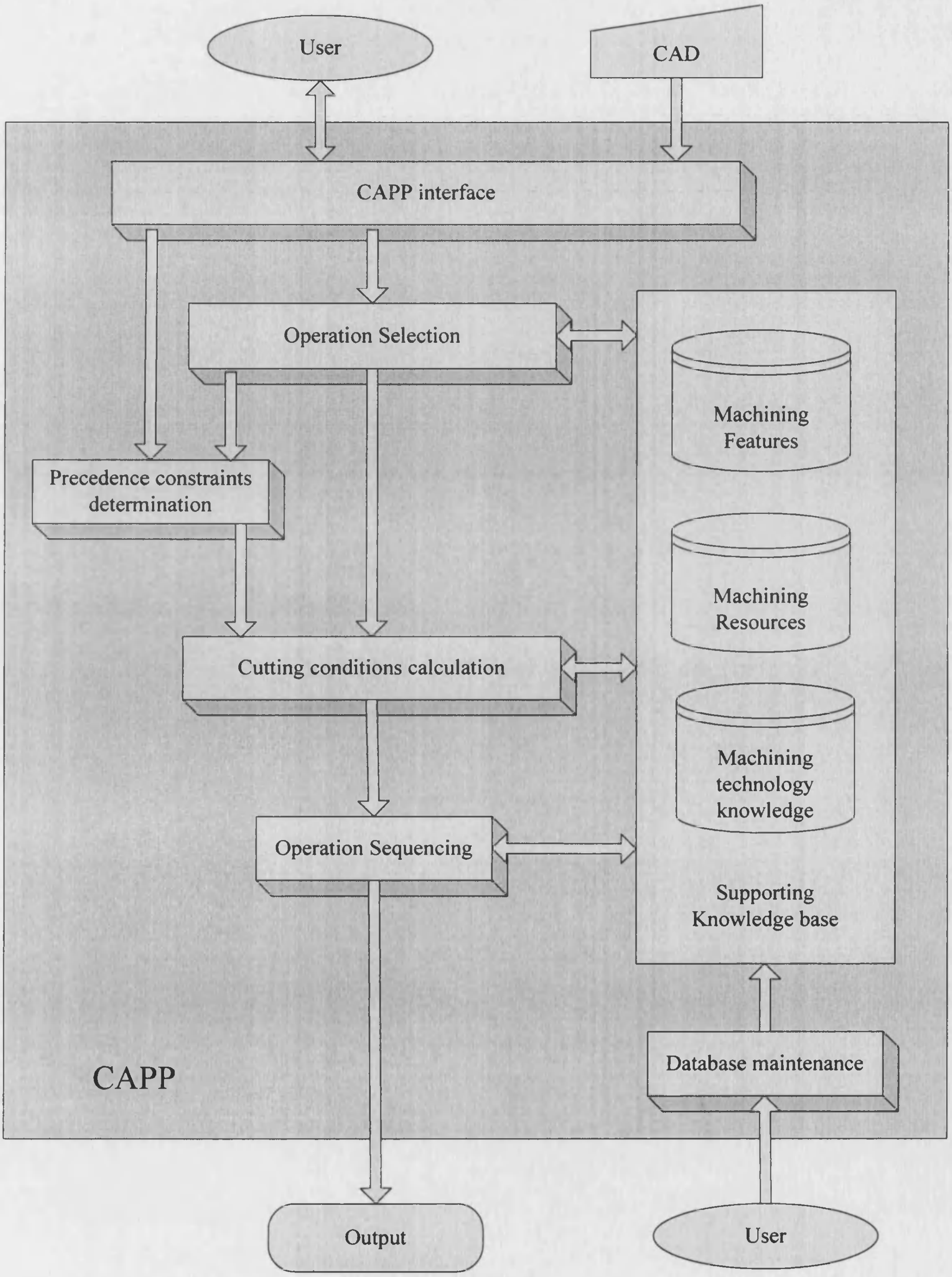


Figure 4.2 General structure of proposed CAPP system

The general information flow of the proposed system is shown in figure 4.1 and the structure of the proposed CAPP system is shown in figure 4.2. It can be seen that the procedure to generate a process plan for a part can be summarised in the following steps:

1) Feature information input. The features information of a part is input manually to the system and stored in a database through a user-computer interface.
2) Operation selection. In this stage, operations are selected by the operation selection module for each feature until all the features have been processed. The operation selection includes tools and TADs selection for each operation. All the applicable alternative tools and TADs are selected in this process. The output of this module is an operation list that contains all the operations information and is stored in a database.
3) Cutting conditions calculation. For each operation in the operation list generated by the operation selection module, the cutting speed, spindle speed, feed rate and machining time are calculated, and stored.
4) Determine the precedence relationship between all the operations according to the precedence constraints.
5) Operation sequencing. All the operations information and related machining parameters are represented in a representation scheme and the PSO algorithm is used to select the optimised resources among the alternatives for each operation and achieve an optimised operation sequence which can satisfy the precedence constraints between operations and minimise the total machining time or manufacturing cost.
6) Finally the optimised process plan is output to the user.

4.3 Integrated Process Planning & Scheduling

This module is an independent module to the 5-axis CAPP system and aims to achieve the overall optimisation of process planning and scheduling. But it can be added into the CAPP system in future to achieve an integrated manufacturing environment.

As discussed in chapter 2, the simultaneous approach is chosen to realise the integration of process planning and scheduling. In this approach, the following two issues need to be considered:

(1) Representations of process planning and scheduling. A process plan for a part can be represented by a series of machining operations, applicable resources for the operations, set-up plans, operation sequence, etc. Here a set-up can be generally defined as a group of operations that are manufactured on a single machine with the same fixture. While the scheduling task is to assign the parts and their machining operations to specific machines to be executed in different time slots, aiming at good shop floor performance. Here time and available resources (machines, tools etc.) are the key factors. Therefore an integrated scheme needs to be developed to represent the problem of IPPS which not only includes the operations, resource and sequences information for all the parts, but also includes the start time, machining time and finish time of every operation and the availability of machines and tools at a specific time.

(2) Performance criteria. To optimise an IPPS problem, suitable criteria need to be determined so as to judge the performance of the system and lead the optimisation algorithm in right direction. Here criteria are considered

including the manufacturing cost, the minimal tardiness, the makespan, and the balanced level of the machine utilisation.

Based on the representation of the IPPS problem and performance criteria, a Particle Swarm Optimisation (PSO) algorithm has been developed to make the optimisation.

4.4 Summary

In this chapter, a generative CAPP system for 5-axis CNC machining is proposed. The functions of modules in the CAPP system are described and the general information flow and structure of the system are given. The approaches and issues of implementing an IPPS module are also presented. Based on these, the detailed implementation of different modules of the CAPP system and IPPS module are extended and extensively investigated in ensuing chapters.

Chapter 5 Feature Based Operation Selection

As one of the most important activities of process planning, operation selection receives the geometric and technological information of the part and generates the possible operations that could be utilised to machine the part. In order to generate the operations effectively and achieve the optimised final process plans, the following issues need to be considered carefully:

1) **Part data representation**: The data needed to define a part in order to establish its process plan basically consists of all the information indicating its shape and therefore its geometry. It also includes information indicating the quality requirements of the part. This generally places restrictions on the geometry with respect to dimensions, shapes, positions and surface roughness. From a manufacturing point of view, however, the format of this information generated from CAD systems is not adequate. So a feature based model is adopted to represent the part data here.
2) **Ability to generate alternative operations for each feature**: Existing heuristic reasoning methods used to generate the process plan normally can not achieve the global optimal process plan because its search is limited in the local search space. To utilise a global optimisation method, such as PSO, to generate the optimal process plans, it is necessary to generate all the alternative operations which include alternative machines, tools and TADs at the operation selection stage.

This chapter presents the feature based modeller to represent the part data first, and then discusses the operation selection for different feature types based on the model in detail. Finally the selection of cutting conditions and machining time estimation are given.

5.1 Input of feature based part data

Preparation of product (feature) data for computer aided manufacturing planning is a difficult task, as features are domain dependent. This implies that the same design needs to be expressed in different feature-based descriptions to satisfy different downstream applications (Sharma and Gao 2002). There have been attempts to extract data directly (feature-based design) or indirectly (feature recognition) from the CAD database. In order to develop a procedure for consistent, unambiguous data abstraction from a generic data structure, the foundation or the standard on which the procedure is based is very crucial.

As described in section 1.1.2 of chapter 1, for the designer, design features are expressed in geometric terms. However for the process planner, the feature is commonly viewed as a machining feature (manufacturing feature). For example, a slot could be seen as a general slot milled by a milling machine; a hole can be considered as a drilled or bored hole. Machining features can express explicitly the methods of production while implying the geometry and function of the features. The objective of this module is to build a feature based modeller which can use machining features to represent a part data so that these machining features can be used directly for downstream activities of process planning such as operation selection. For simplification, the system developed in this research does not generate the features automatically from the part drawing designed by Unigraphics, it is required to input the feature information manually in this process.

To achieve this objective, the following issues have been considered:

1) **Feature taxonomy**: Instead of specifying all the geometrical and topological information that defines a feature for every separate feature type, it is possible to group features with common properties into classes. These can then be further divided into sub-classes to form a tree structure, or hierarchy. Since they are of a hierarchical nature, the properties of a class can be inherited by its sub-classes. Although the number of possible features and feature classes is not finite, it may be possible to categorise feature classes into families that are relatively independent of the intended application domain of the features (Shah and Mäntylä, 1995).
2) **Feature representation**: How to represent a feature and what information needs to be included are two major problems of feature representation in this research. An object oriented design strategy has been adopted to provide the capability to organise and represent the feature information for easy message processing and offer the flexibility to modify the definition of an object, its structure, message and linkup without affecting the rest of the system configuration. The machining feature information is determined based on the STEP AP224 standard (STEP 1999).

5.1.1 Feature Taxonomy

There are a number of feature classification schemes. Among them are those based on geometrical properties of the features, such as the work of Gindy (1989). Others are based on machining methods associated with features that include rotational features created by machining operations on a turning machine, and prismatic features created by machining operations on a milling machine or a three-axis machining centre (Tseng and Joshi, 1998). There are also those based on the number of possible tool approach directions that can be used to machine the part: STAD (single tool axis direction) and MTAD (multiple tool axis direction) (Chu and Gadh, 1996). These classification schemes have advantages in certain respects, but major problems (e.g. non-standard and incompleteness) hinder their practical applications in integrated environments for design and manufacturing.

STEP is introduced to define a standard file that includes all information necessary to describe a product from design to production. An ISO STEP application protocol specifies the manufacturing information and process plans using manufacturing features to machine discrete mechanical parts, and it supports multiple application domains, for instance, mechanical engineering, electronics, architecture (Owen 1993). Some researchers have developed feature recognition methods based on STEP. For example, Bhandarkar and Nagi (2000) developed a Boundary-representation (B-rep) based feature extraction system that takes a STEP file as input and produces a form-feature STEP file; and Han et al. (2001) proposed a geometric reasoning feature recognition kernel using STEP as input and output formats. In ISO 10303 STEP-AP224 (Mechanical product definition for process planning using form features), machining features are defined as a type of manufacturing feature that identifies a volume of material to be removed to obtain the final geometry from the initial stock (STEP, 1999). Sixteen machining feature classes are defined (Boss, Pocket, Hole, Slot, Protrusion, Rounded_end, Outer_round, Step, Planar_face, Revolved_feature, Spherical_cap, General_outside_profile, Thread, Marking, Knurl, General_volume_remove). These machining features contain all of the information needed to manufacture the required part, including materials, part geometry, dimensions and tolerances, applicable notes and specifications, and administrative information.

In order to simplify the algorithm generation and development time, the research work presented in this thesis has been carried out using a restricted set of 5 features. These features are namely: planar face, pocket, slot, hole and step. A hierarchical classification (see table 5.1) is proposed based on the following principles.

1) A machining feature is defined as a geometrical entity, which is related to a group of particular machining processes and can be mapped to a suitable machining method.
2) The feature classification and its validity are based on a multi-viewpoint considering manufacturing requirements with topological information.
3) If a set of features have similar geometric and topological characteristics and can be machined with similar processes, they are called a feature class. A sub-class is regarded as an instance of its main class.
4) The classification is hierarchical, where a subclass inherits common properties from a higher class. This reduces the number of properties that have to be independently specified for each new feature.
5) The TAD (Tool Approach Direction) to machine a feature needs to be considered. For example, Step and Through Hole features can be machined by tools from two directions. This can be classified into Single TAD (STAD) and Multiple TAD (MTAD) features.
6) The feature definition in the ISO STEP AP224 standard is considered as a guideline for industrial use.

Table 5.1 Feature Classification

Feature Name	Sub Classification	TAD Type
Face		STAD
Hole	Through Hole	MTAD
	Blind Hole	STAD
Pocket	Open Pocket	STAD
	Close Pocket	STAD
Slot	Through Slot	STAD
	Blind Slot	STAD
Step	Open Step	MTAD

5.1.2 Feature representation

Based on the above classification, an object oriented design strategy has been used to represent the hierarchical organisation of feature data. Before using an Object Oriented Programming (OOP) language (C++ is used here) to represent the feature classes, it is necessary to decide the detailed information each feature class includes (in C++ class point of view, this information is called the member variable of a feature class and can be used directly or indirectly by downstream application). The member variables of a feature class are defined based on the requirements for process planning. A standard feature class can be explicitly defined with four kinds of member variables: identifier, dimensions, location and Technological specification.

1) **Identifier**: A number of basic terms understandable to both the designers and the system, namely, feature name, feature ID and feature class type. If it is a sub class feature, the feature sub class type is also required. With the feature class type and sub class type, the geometric type of feature is defined.

2) **Dimensions**: The dimensions including length, width, depth, diameter (for hole) and corner radius etc. are used to further specify the geometric information of the feature.
3) **Location**: is used to identify the spatial relationship between a feature and the stock. To determine the location of a feature, the original point of the feature needs to be determined. The selections of the original points for each feature type are different. The original points of Face, Slot, Step and Pocket features are defined as the point with the lowest coordinates in the X, Y and Z directions. While the original point of Hole feature is defined as the centre point of the hole with the highest coordinates in this hole's centre line direction. Based on these definitions, the directions of feature's length, width and depth is determined as the directions of feature extension from the original point.
4) **Technological specification**: Variables of tolerances and surface roughness are attached to each feature class to represent the technological requirement of features. Operation selection and tools selection will be based on this information.

A set of constraints need to be checked at the feature input stage to ensure the feature validity. There are three types of constraints: geometric and topological, machining, and interacting constraints which are described in the following:

- Geometric and topological constraints. Usually, these constraints appear as a standard range for specifying the size limits, which can be calculated using mathematical equations based on the shape parameters, class, and position and orientation of the feature. For example, the dimension of a hole cannot be larger than the size of the stock on which it is being placed; the depth of a blind hole must be restricted to be less than the size of the stock where the hole is to be added, otherwise the blind hole would become a through-hole.

- Machining constraints. It is possible that some features have valid geometric shapes and topology but still are invalid features because of their non-machinability. Different from other constraints, machining constraints mainly depend on the machining attributes of features and the specific workshop environment that features will be manufactured in (e.g. machine tools can be available). For instance, long and thin holes may be regarded as invalid if no machining methods are available for their manufacturing. At the design stage, the check for machining constraints is limited to constraints that can be defined by algebraic expressions, e.g. the ratio of height to radius. Other machining attributes (e.g. tolerance and accuracy) are examined at the process planning stage, i.e. during selection of machining operations.
- Interacting constraints. Geometrical, topological and machining constraints are insufficient to fully retain feature validity when feature interactions occur. As known, feature interactions can cause serious constraint violations of valid feature instances. Therefore, the constraints for feature interactions must be defined, such as the dependent properties between parent and child features. An example is shown in Figure 5.1 where pocket B is added based on pocket A and becomes a child feature of pocket A. Due to this interacting constraint, pocket B will be invalid if pocket A is deleted (Ding 2003).

Considering all the above information, table 5.2 shows the common member variables for all the five types of feature. Figures 5.2-5.6 illustrate the different information needed for these five features.

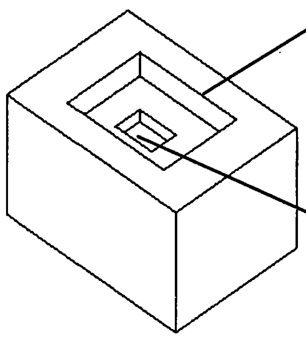


Figure 5.1 Example of interacting features (Ding 2003)

Table 5.2 Common member variables of features

Common member variables for all the features:		
Data types	Variables	Descriptions
String	PartName	The name of the part
Long	FeatureID	The id of feature
String	FeatureName	The name of feature comprised of feature type and id
String	FeatureType	The type of feature, e.g. Face, Slot, Step, Pocket and Hole
String	FeatureSubType	The sub type of feature class, for example, rectangular slot
float	DatumX	The X coordinate value of original point of the feature
float	DatumY	The Y coordinate value of original point of the feature
float	DatumZ	The Z coordinate value of original point of the feature

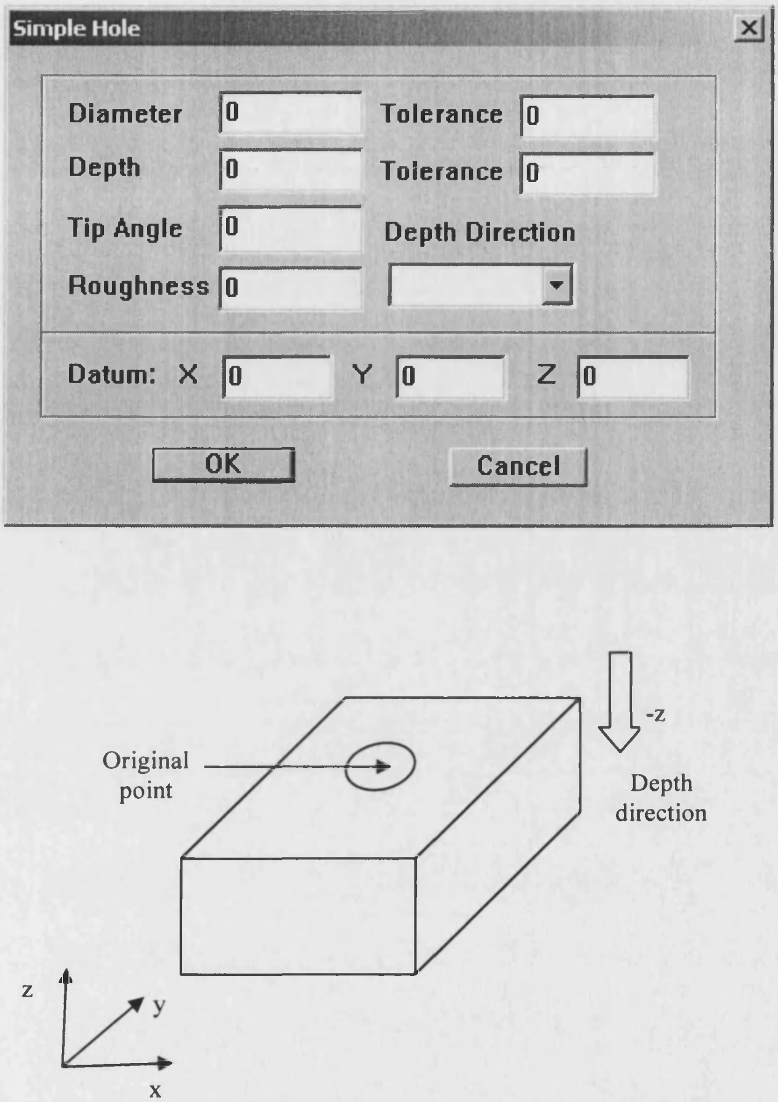


Figure 5.2 Interface of hole feature information input

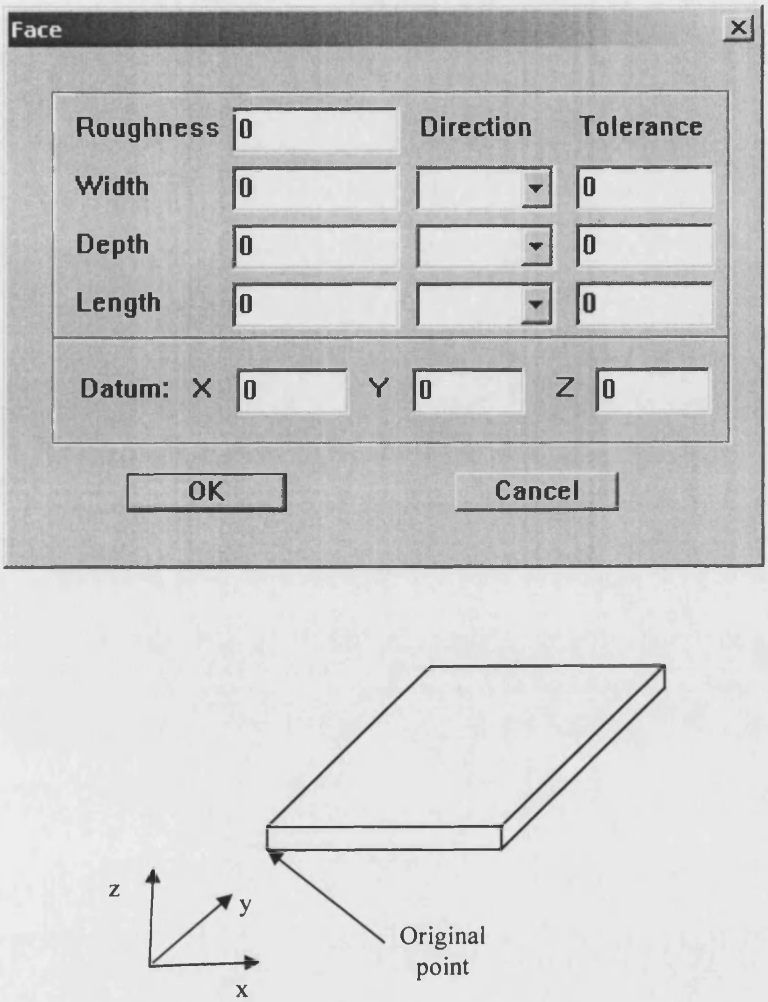
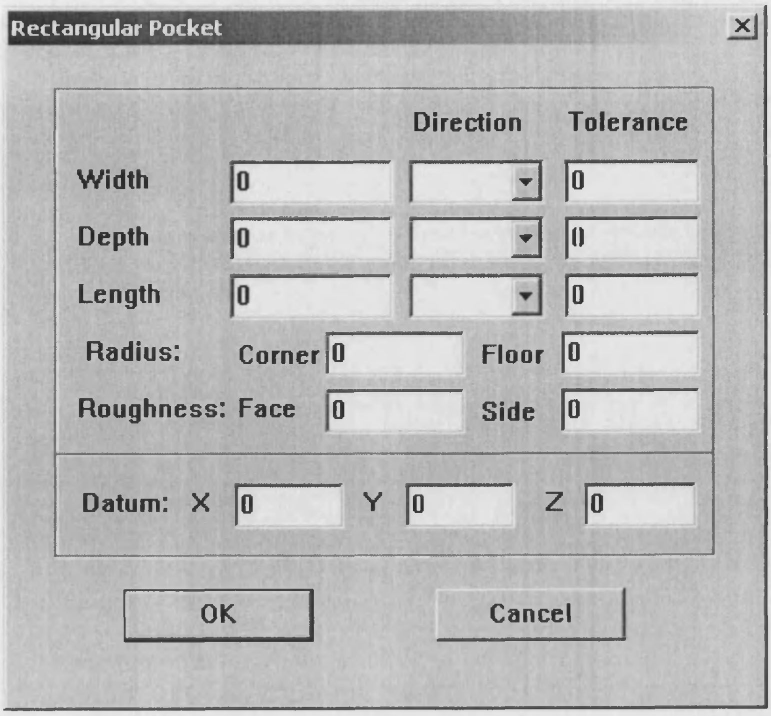


Figure 5.3 Interface of face feature information input



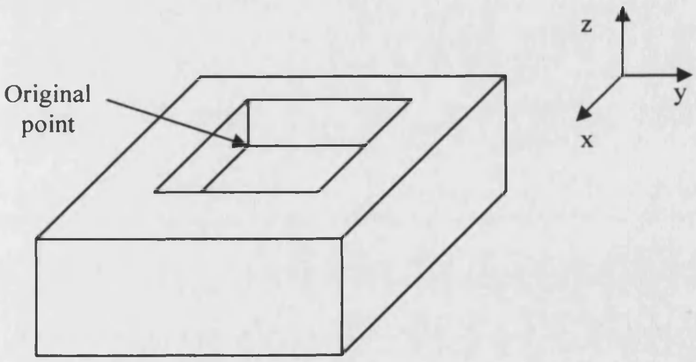


Figure 5.4 Interface of pocket feature information input

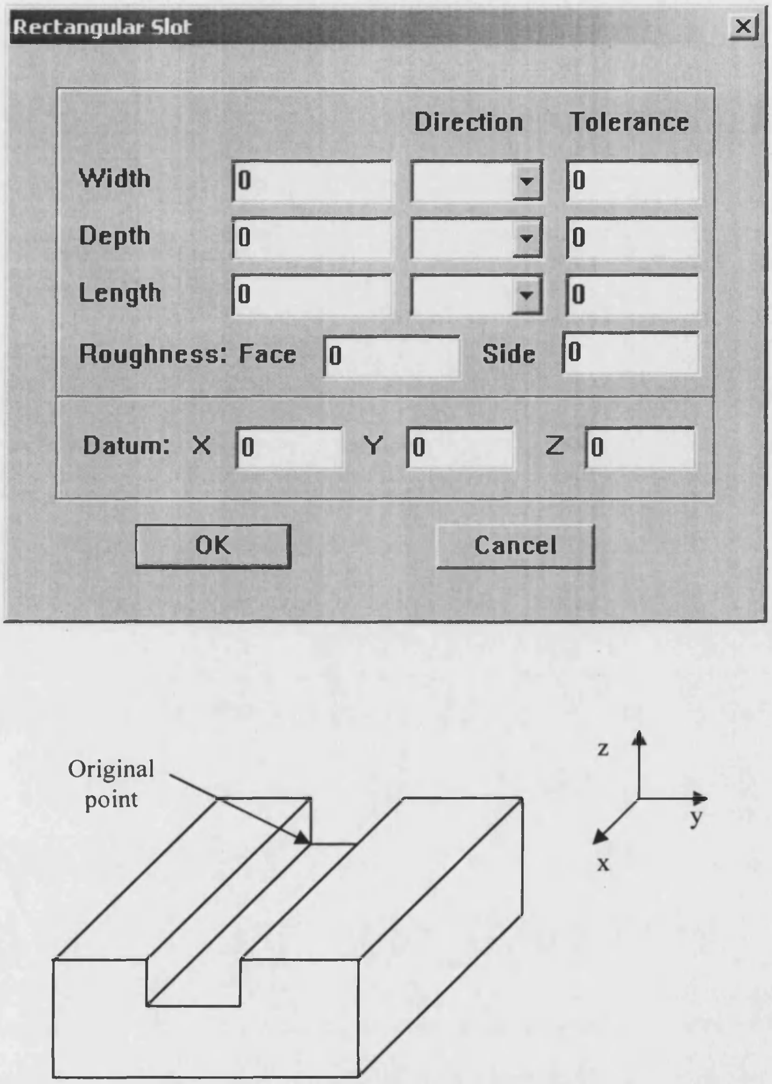


Figure 5.5 Interface of slot feature information input

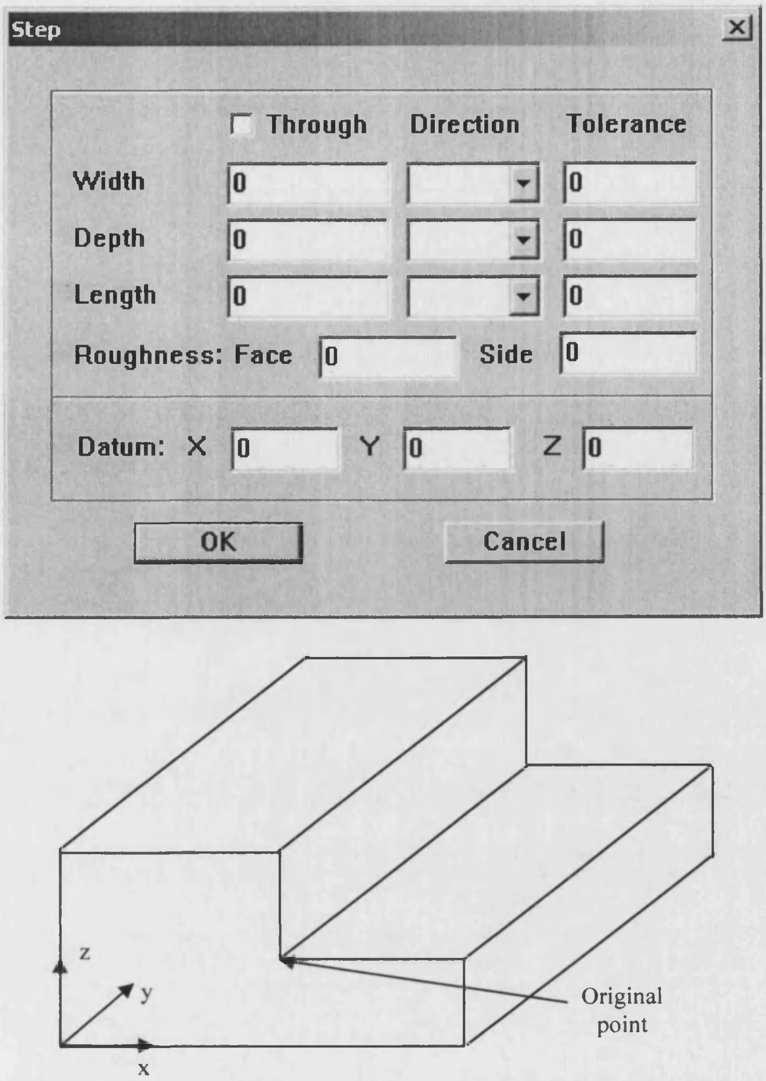


Figure 5.6 Interface of step feature information input

After all the feature information is input into the system, it is automatically stored into a database. The design of the database will be described in chapter 8.

5.2 Selection of machining operations based on feature type

After the features of a component are created, it becomes possible to identify the operations that are executed to machine these features. This process can be achieved by determining the operations for each feature one by one. To achieve this mapping process, the following steps need to be executed:

1) Extraction of feature information from the feature-based model. As described in the previous section, an object-oriented feature-based model is used in this research, which defines a component in terms of its features. From a process planning perspective, a feature can be made from raw material by one or more operations. In order to choose the operations suitable for meeting a feature's specification, the following information in the feature-based model should be extracted and considered:
   - Feature class
   - Nominal dimensions
   - Dimension tolerances
   - Surface roughness (Ra) (Note that geometric tolerances have not been included in this research)
2) Find the operation types (OPT) that can achieve the attributes (shape-Feature class, dimensions, tolerances, and surface roughness) of the feature. An OPT refers to an operation without any attachment of tool (T), and tool approach direction (TAD), e.g., drilling and milling. When operation type is determined, it is necessary to be subdivided according to their capability, into rough, semi-finishing and finishing operations that form an operation-set. How to divide the operations depends on the dimension tolerance and surface roughness which rough, semi-finishing and finishing operations can achieve. This will be discussed in detail in the latter sections.
3) Tool selection. For each OPT, find all the possible tools with which the operation can be executed. In this research, the 5-axis CNC machine centre is used as the only machine tool and the effort is concentrated on cutting tool selection. Generally, the selection of a proper cutting tool depends on the machining

operations necessary for processing the feature with regard to its size and finishing requirements. From the tools available on the market, a selection of different tool types and sizes have been selected and stored in the system's cutting tool database file. They have been selected to cover the needs of the component features to be processed and the machine tool used. In general, tool information supplied by Sandvik and British Standards have been identified for use in the system. The system utilises both high speed steel (HSS) and carbide cutting tools.

The selection of the appropriate cutting tool type and size for machining the features used in this system is influenced by four main constraints: (1) Machining Process Constraint, (2) Feature Dimensions Constraint, (3) Machinability Constraint and (4) Economic Constraint. These constraints are used for the selection of the proper cutting tool and are summarised in the following stages:

Stage 1: Retrieve the machining operation and feature information and check the possible cutting tool types that can be used. As indicated earlier the machine tool is already selected, therefore, the search for the appropriate cutting tool type and size is concentrated on cutting tools which are related to the 5-axis CNC machine. More than one machine tool might be selected for the same operation depending on the feature types, sizes and finishing requirements. Here a "tool preference criteria" is used to select the applicable tool type. For each feature, every applicable tool type is given a coefficient to represent the degree of preference. The higher the coefficient, the higher the performance the tool of this type can machine the corresponding feature. Table 5.2 shows the Tool type recommendation for each feature in this system.

Table 5.3 Features and Cutting Tool Type Recommendation

ToolType	Face	Step	Slot	Hole	Closed Pocket	Sideopen Pocket
Drill				3		
EndMill	1.3	1.3	1.4		1.5	1
FaceMill	1.5	1.3				
Reamer				2		
SlotMill			1.5		1.1	

Stage 2: Check the feature dimensions to retrieve the cutting width in order to search for the applicable sized cutting tool from the tool list in the selected tool types indicated in stage 1. Tool size for all of the flat-feature operations could be bigger than the feature width except for slots and pockets. Therefore, this factor is taken into account at this stage to ensure the selection of the proper tool size especially for slots, pockets and holes features. In this stage, the range of tool size is determined for the operation.

Stage 3: Search for the applicable cutting tools in the cutting tools database in terms of selected tool types and tool size which falls in the tool size range determined in stage 2.

Stage 4: Eliminate the less cost efficient cutting tools. There might be more than one cutting tool selected and some tools may not be efficient. For example, to machine a 50mm wide face feature, all the tools whose diameter is less than 75mm may be selected including 75mm, 50mm and 20mm diameter tools.

Obviously, using a 20mm diameter tool to machine the face will probably cost three times more than a 75mm tool. In this case, the 20mm diameter should be eliminated from the selection list. Also, with too many tools in the selection list, for some tools, there maybe more than one TAD, and this will generate many more alternatives to execute the operation so as to make optimisation of process plans more difficult (There is possibility that this method may eliminate the optimum tools, for example, a less efficient tool might be used for machining other features so as to save the tool change time and save the total machining time. In this prototype system, this possibility is reduced because the tool change time is much less than the machining time for a feature in a 5-axis CNC machine centre.).

To select the most efficient tools for the operation, one of the methods is to estimate the machining cost using this tool and delete those tools that have high machining cost. This estimate requires details of materials, tools, work holding and details of the operations used to manufacture the component and is influenced by the cutting conditions selected. However, the data required to make these calculations is not readily available during the tool selection stage. It can be argued that machining cost is simply a factor of machining time which is influenced mainly by the choice of tools used (if the machine used has been determined). Rather than attempt to calculate machining costs per se, it is far more practical, in tool selection terms, to attempt to calculate machining time. If costs are required then machining times can be multiplied by a value of machining cost per unit time (Maropolous et al. 2000).

But this process is still tedious and time consuming due to the large number of possible machining time calculations for several tools and the comparison of them. Actually, it is well known that the most important factors influencing the cutting conditions (cutting speed, feed) are the tool material, tool type and tool size. Component material, tool material and tool type can be used to select the suitable cutting speed and feed and tool size can determine the tool path length. So, for simplification, the less efficient cutting tools can be eliminated from the tool list with consideration of tool material, tool type and tool size. As indicated previously, different tool types have been given a coefficient to represent their degree of preference. The carbide insert tools can achieve more cutting speed and feed than HSS tools. The bigger the tool size, the shorter the tool path length and the more efficient the tool is. In this system, for simplification, the tools selected after the evaluation stage are limited to 2 tools for each operation.

4) Determination of all the feasible TADs for each cutting tool selected. A process plan for a part consists of operation types, applicable candidate machining resources, set-up plans, machining parameters, operation sequence, etc. A set-up can be defined as a group of operations that are machined on a single machine with the same fixture. In a 3-axis machining environment, a set-up is a group of features with the same TAD machined on a 3-axis machine. However, in 5-axis, different TADs can be achieved by the two extra degrees of freedom movements using the same fixture. Therefore, two different TADs do not necessarily mean two different set-ups. The feature can be machined with the same fixture only if the TAD of the operation for this feature can be achieved by the rotary or swivel motion of the work table. For a 5-axis machine, one single set-up (same fixture) can achieve 5 different TADS. On the other hand, one single TAD can be

possibly achieved by 5 different set-ups. Although the TADs can not determine the set-up, it is necessary to determine the TADs at this stage so as to help the optimisation of the operation sequence and determine the final set-up to achieve the minimal machining cost or machining time.

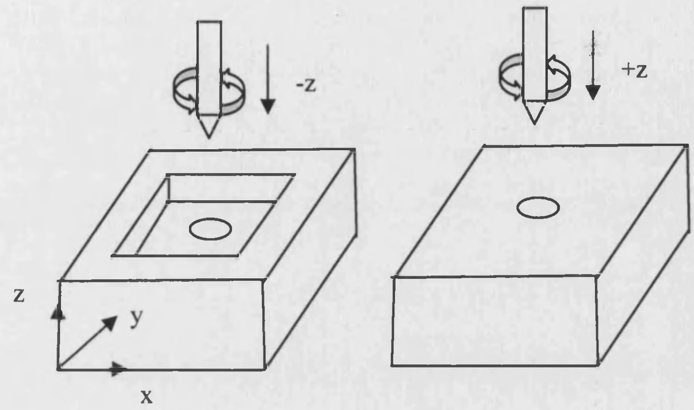


Figure 5.7 Through hole with two TADs

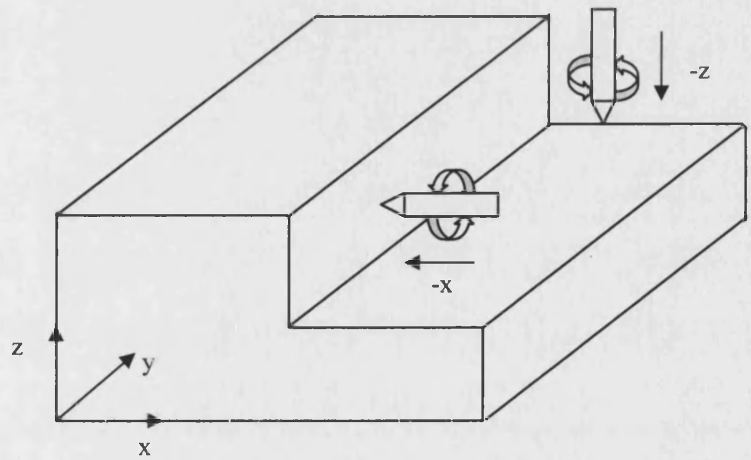


Figure 5.8 Through Step with two TADs

Two special features in this system can be machined in two TADs. As shown in figure 5.7, a through hole can be drilled in two directions. Of the 6 major TADs, the through step feature shown in figure 5.8 can be machined from both the –x and –z directions. A feasible TAD should satisfy the following conditions.

- Tool accessibility: if a cutting tool for machining a feature along one of its TADs is blocked by other features on the part, or the cutting tool cannot be positioned in the part to machine the feature along the TAD correctly, the TAD for the feature is considered to be inaccessible and invalid.
- Fixture: if there are no valid fixture elements for holding the part on the machine along one of its TADs, the feature cannot be fixtured and machined along the TAD and the TAD is unfeasible.
- Availability of cutters: if the volume of a feature along a TAD is beyond the scope of any cutting tool available, the feature cannot be machined along the TAD and the TAD is unfeasible.
- Tolerance and surface roughness requirements: an operation should not violate the tolerance and surface roughness requirements when a feature is machined along one of its TADs. Otherwise, the TAD is unfeasible. (Li et al. 2002)

5) Iterate step 1) to step 4) until all the operations for all of the features are generated. Here for one single feature, there may be one or more operations that are executed to machine it, i.e., rough milling, semi-finishing milling and finish milling a face feature. Each operation may also have several operation alternatives in terms of different tools and TADs. If there are two applicable tools and two applicable TADs for each tool, there will be four operation alternatives for this operation.

When all the operations are generated, they will be stored into the *OperationList* table in database.

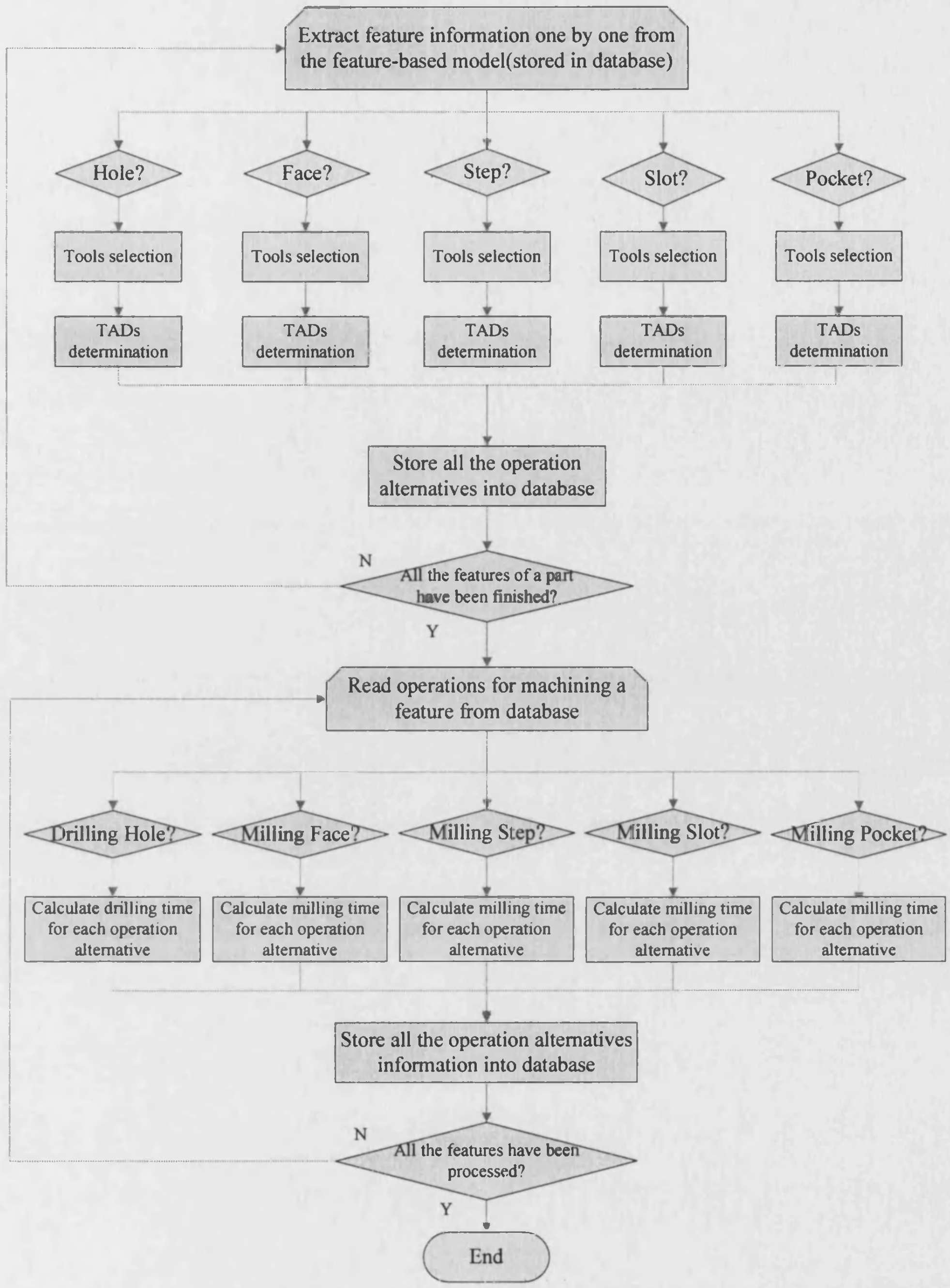


Figure 5.9 Work flow of operation selection

6) Cutting conditions estimation and machining time calculation. Selecting the cutting conditions for each operation alternative means selecting the cutting speed, spindle speed and feed for the corresponding tool and TAD. Based on these results, the machining time for each operation alternative can be calculated.

This step is repeated until the machining times for all the operation alternatives are calculated and stored.

The work flow of the above steps is illustrated in figure 5.9. As described previously, only the general operation selection method is discussed. The detailed operation selection and machining time calculation for different specific features will be presented in the following sections.

5.2.1 Operation Selection for Face feature

1. Operation types determination.

As described previously, when the feature information is extracted from the database and feature type is known, the first step is to determine the operations that can achieve the technological requirements of the feature. This process can be achieved by the following steps:

Step 1: Identify the surfaces where machining is required and check their dimensions (length, width and depth) and finishing requirements (tolerances and roughness).

Step 2: Check for finishing requirements. If either the dimensional tolerances or surface roughness can not be achieved by milling, the finishing operation of grinding is required. Because of the limitation in this research, only the 5-axis CNC machining is considered. So in this case, it is only required to leave the allowance for grinding by 0.25mm and reduce the total depth of cut by 0.25mm at the same time.

Step 3: Check for surface roughness. If the roughness Ra≤0.8μm, the operation of finishing milling is required. An allowance for finish milling is 0.2~0.5mm is made and the total depth of cut for the semi-finishing and rough milling operations is reduced by 0.2~0.5mm at the same time. If the roughness Ra≤3.2μm, the operation of semi-finishing milling is required. An allowance for semi-finishing milling is 0.5~1.5mm and the total depth of cut for rough milling is reduced by 0.5~1.5mm. If the roughness Ra>3.2μm, only roughing is required. By this method, the operations are divided whilst the allowances and depth of cut for operations can be calculated. All the operations required to machine the feature form an operation-set.

2. Cutting tool selection.

Tool selection involves the determination of the following (Chang 1990):

- Tool type — drill, face mill, end mill, reamer, etc.
- Tool material — HSS, carbide, etc.
- Tool geometry — helix angle, rack angle, etc.
- Tool dimension — overall length, flute length, diameter

The tool type is determined by the operation type and feature type. Table 5.2 shows the cutting tool type recommendation for different features. The tool material selection is based on the raw material stock and its hardness. Suggested tool material data has been taken from Machinery's Handbook (Oberg et al. 2004). If HSS tools can not machine the component according to its hardness, then Carbide insert cutters are selected. Otherwise, both HSS and Carbide insert tools are in the option list. Tool geometry is selected based on the feature geometry, raw material condition, and tool material. For simplification, this prototype system does not consider the tool geometry for the time being, suggested routes to achieve this are presented in Chapter 9– Future Work.

The dimensions of the tools that are used to carry out the operation are extracted from the feature properties. For all operations, there are only three tool dimensions that are of interest at this stage of tool selection: overall length, flute length and tool diameter. Overall tool length is the length which ensures the spindle has a collision free movement. It is determined using the following procedure:

Step 1: Find the intersection of the intermediate workpiece boundary model with a cylinder which has the same diameter as the spindle, and align the axis of the cylinder with the feature.

Step 2: Do step 1 for the final part boundary model with the cylinder.

Step 3: The minimum overall tool length is the difference between the extreme point in the spindle approach direction from step 1 and the further point in the spindle approach direction from step 2.

The flute length is the actual height of the refined feature. The height is determined by the approach direction. The tool diameter, on the other hand, varies depending on the operation taking place and feature dimensions. Here a method by Maropoulos et al. (2000) is adopted to determine the tool diameter. For a face feature, the diameter of a tool used to carry out a facing operation is 1.35 times the width of the face feature (it is assumed that cutting takes place along the length of the feature and that the width is smaller than or equal to the length).

In the above process of tool selection, the tool type selection and tool diameter determination are the most important factors. When the tools have been selected, the overall length and flute length can be used to check the machinability of features using the selected tools.

3. TAD determination.

As shown in table 5.1 and table 5.3, *face* is a feature with a Single TAD (STAD), a facemill and endmill can be used to machine it. So the TAD for the corresponding tool is of the opposite direction of depth. For example, the direction of depth of a face feature is +z, then the TAD is the opposite direction of +z, i.e. –z.

The above three major steps to determine the operations for machining a face feature is illustrated in figure 5.10.

5.2.2 Operation Selection for Slot feature

Figure 5.11 shows the work flow for the operation selection of a through slot feature. It can be seen that the operations required to manufacture a through slot are similar to those required to create a face feature. The differences between these two procedures are the tool diameter determination and tool type selection.

For a through slot, slot mills and end mills have the higher tool selection priority. For slots, the width of the feature is the upper bound of the tool diameter. To allow good manufacturing practice, the maximum diameter of the tool is selected as the width of slot minus a margin of 1 mm. The non-through slots mainly need the diameter of their finishing operation tool to match the corner radius of the closed end of the slot (In some cases, it can be achieved by smaller tool with circular path). However, to avoid a situation where the tool is "sucked" into the component through inadvertent climb-milling, the radius of the tool should be about a millimetre less than that of the corner.

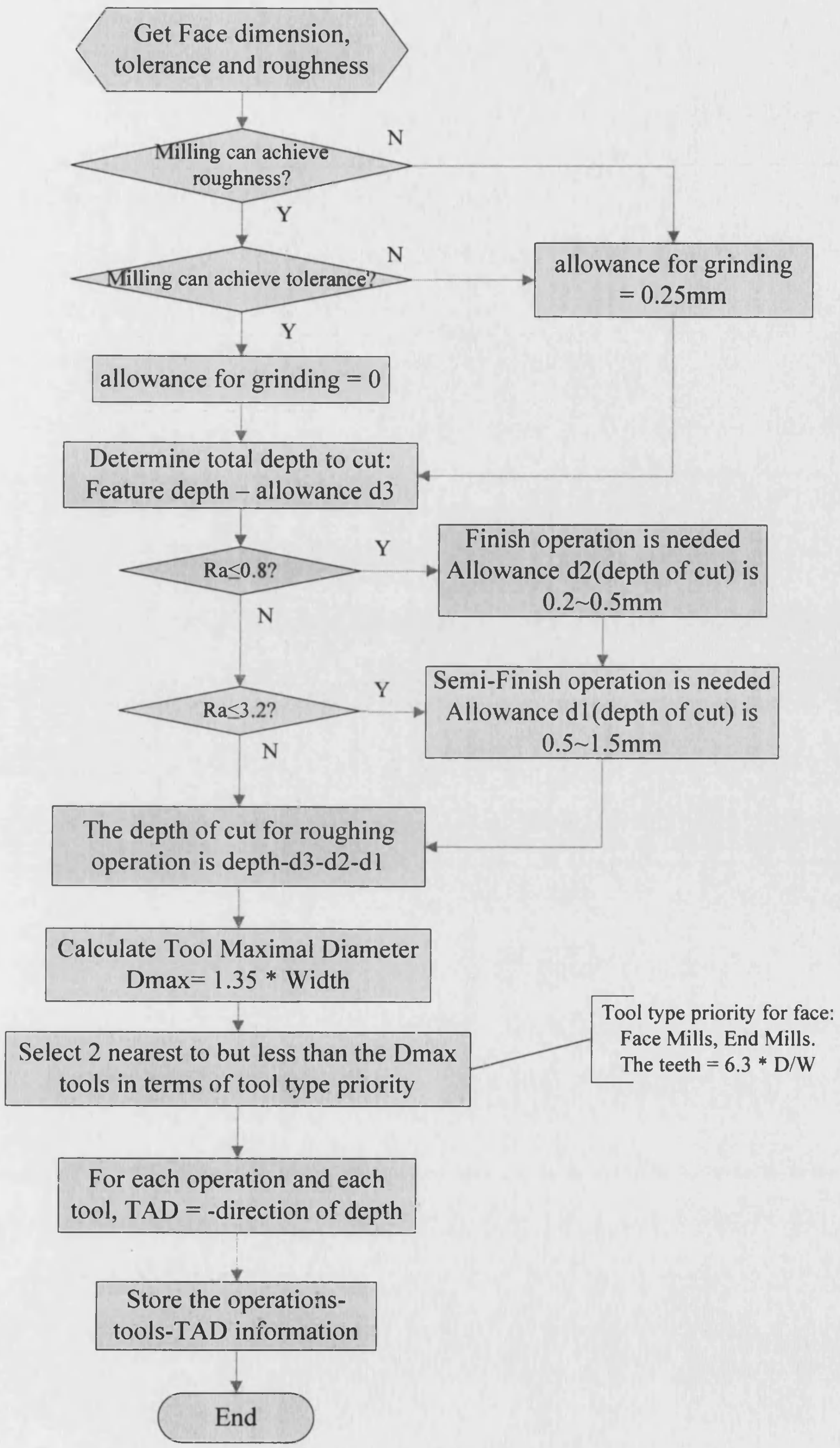


Figure 5.10 work flow of operation selection for face feature

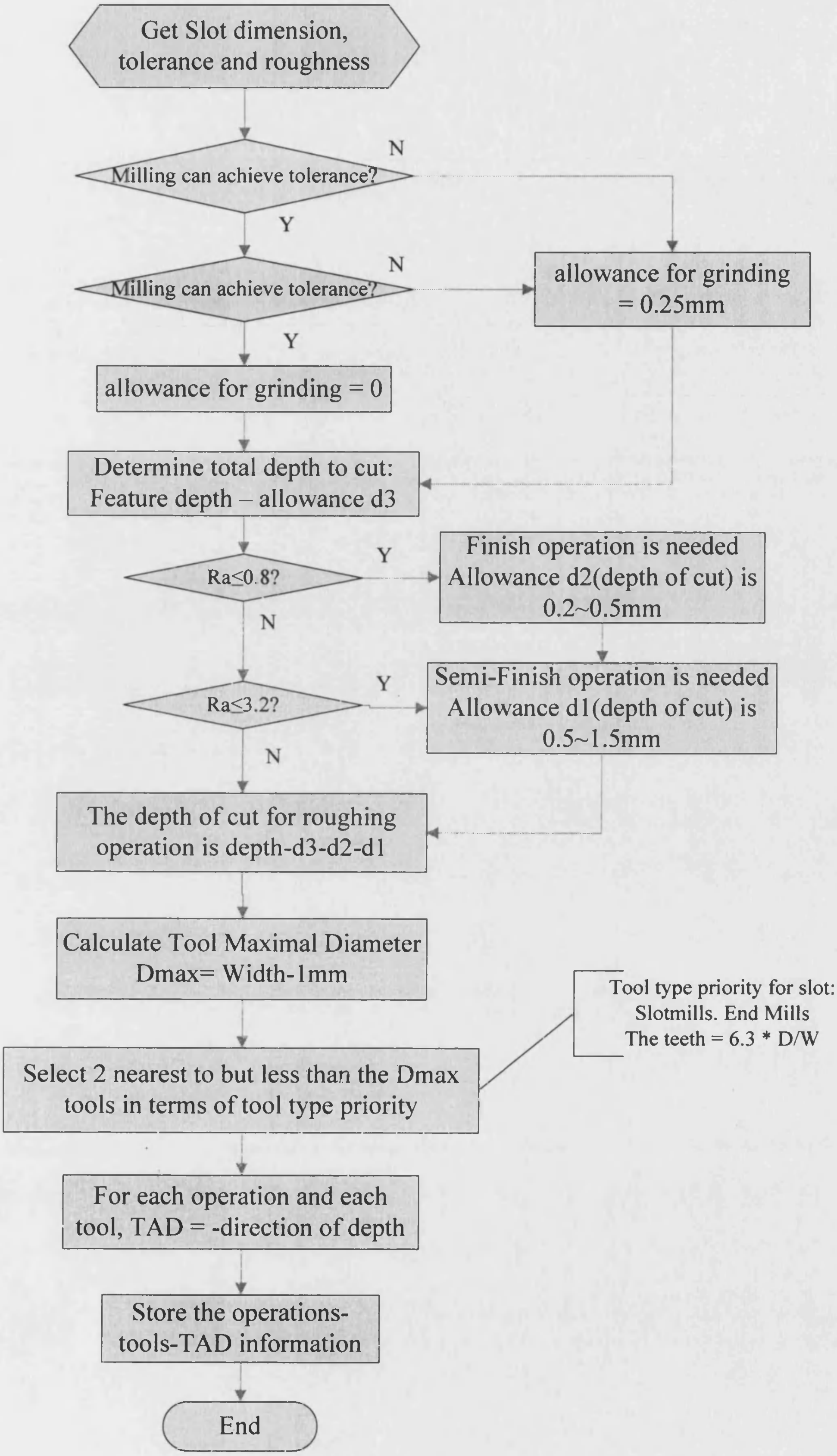


Figure 5.11 work flow of operation selection for through Slot feature

5.2.3 Operation Selection for Step feature

Step features can be machined from two TADs as figure 5.8 shows. For each of the TADs, the operation may be subdivided into roughing, semi-finishing and finishing operations. The depth of cut, tool diameters and cutting conditions for each TAD will be different unless the depth and the width of the step feature are equal. So it is necessary to consider these two TADs differently.

A definition of an operation-set is introduced here to represent the operations for one single TAD. So there are two alternative operation-sets for machining a step feature while only one operation-set for other features. For a through step, the operation subdivision, tool selection and tool diameter determination are similar to those required to machine a face feature. The only difference is that the first step of operation selection for a step is to divide the two operation-sets into the width and depth directions of the step feature. Then in the first operation-set, the TAD is in the opposite direction to the depth. The depth and width of the step in this operation-set are as same as the original depth and width of the step feature. In the second operation-set, the TAD is in the opposite direction to the width. Here the depth and width of the step in this operation-set are changed to the original width and depth of the step feature respectively. The work flow of operation selection for a through step feature is illustrated in figure 5.12.

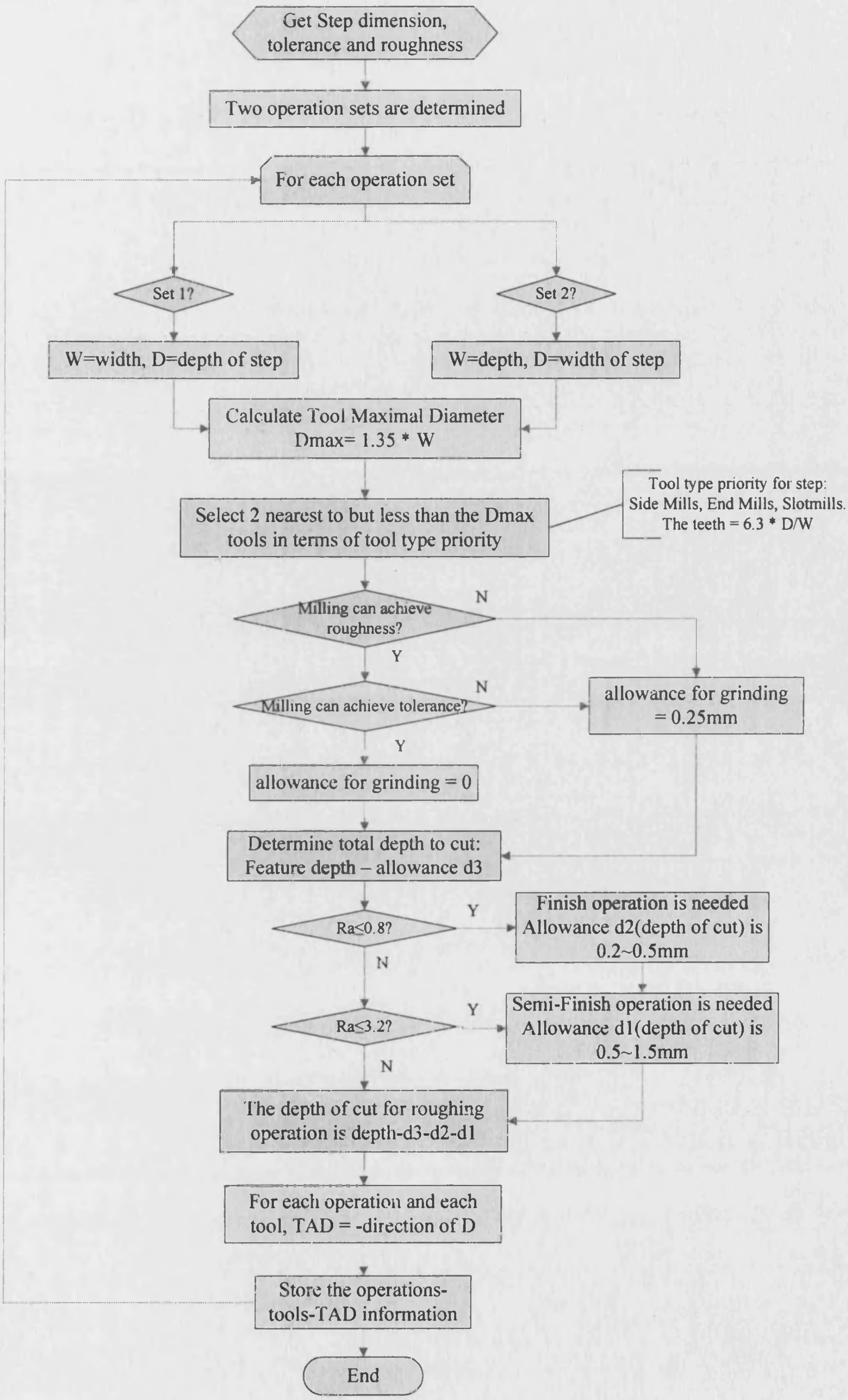


Figure 5.12 Work flow of operation selection for through step

5.2.4 Operation Selection for Pocket feature

Closed pockets are created in a similar manner to slots and steps, with the additional requirement of a drilling operation to allow the roughing tool access into the component. This, however, imposes a constraint in that the diameter of the milling tool is dependent on the diameter of the drilling tool used. A method of selection developed by Maropolos and Baker (2000) is used here. It is assumed that drills up to 50mm are readily available according to the range of drills supplied by the tooling manufacturers (Sandvik, Seco and Stellram). Thus, if a third of the width of the closed pocket is 50 mm or less, then an access hole with a diameter of a third of the pocket width should be drilled (minimum diameter is set to 20mm). A third of the width is used as the benchmark value to ensure that enough material remains for the subsequent milling operation. If the pocket width is larger than 150mm, then an access hole of 50 mm should be drilled. Except that the diameters of tools for the milling operation are constrained by the diameter of the drilled access hole. The milling operations that follow can be selected the same as for slots and steps.

Figure 5.13 shows the work flow of operation selection for a closed pocket feature.

5.2.5 Operation Selection for Hole feature

Compared to other features, the operation selection for hole features is easier due to the easy determination of the drill diameter which is precisely the same as the diameter of the feature. If the drilling operation can not achieve the dimension tolerances and roughness, reaming is required for the finishing operation. Here 0.4 mm is left on the diameter as the reaming allowance.

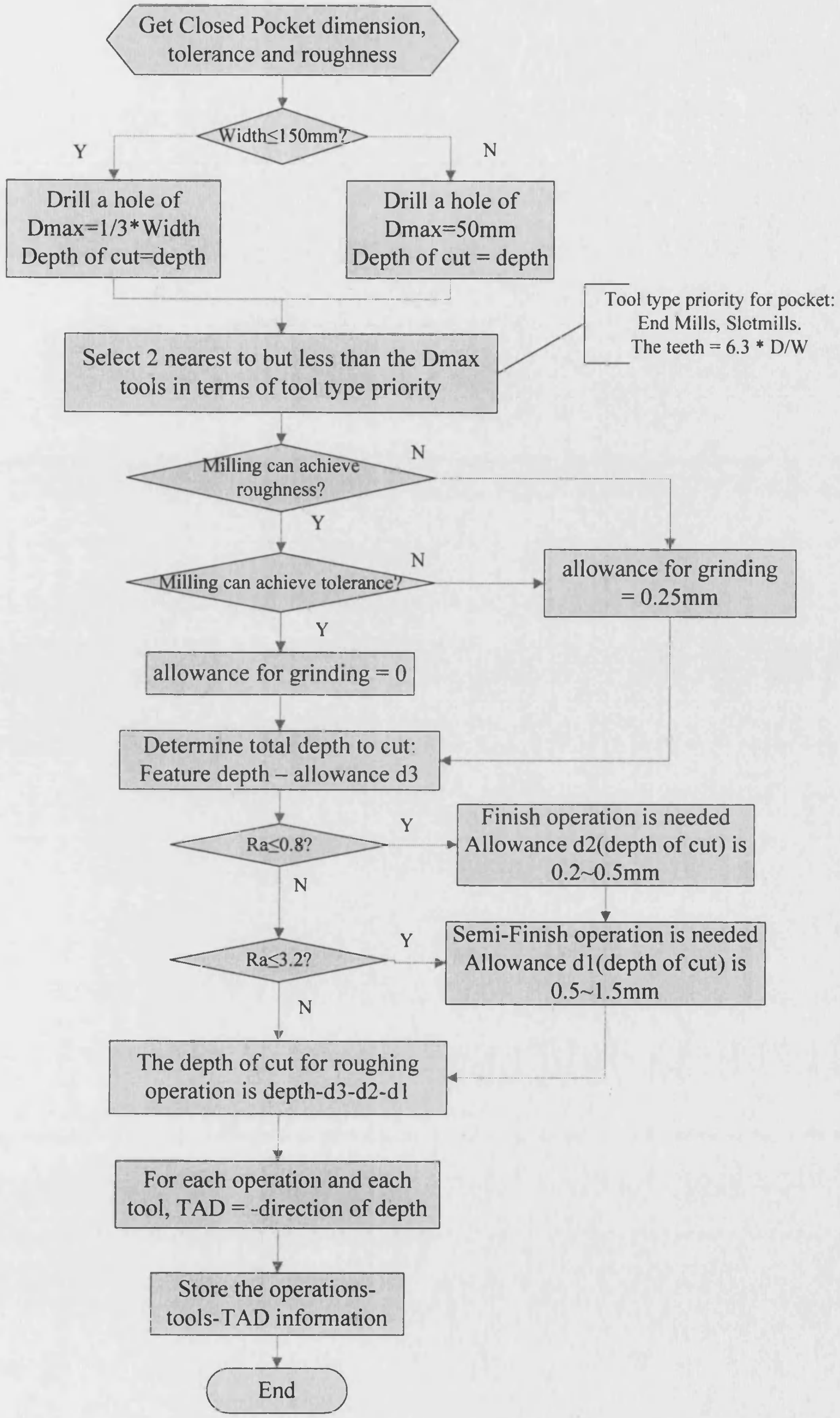


Figure 5.13 Work flow of operation selection for closed pocket feature

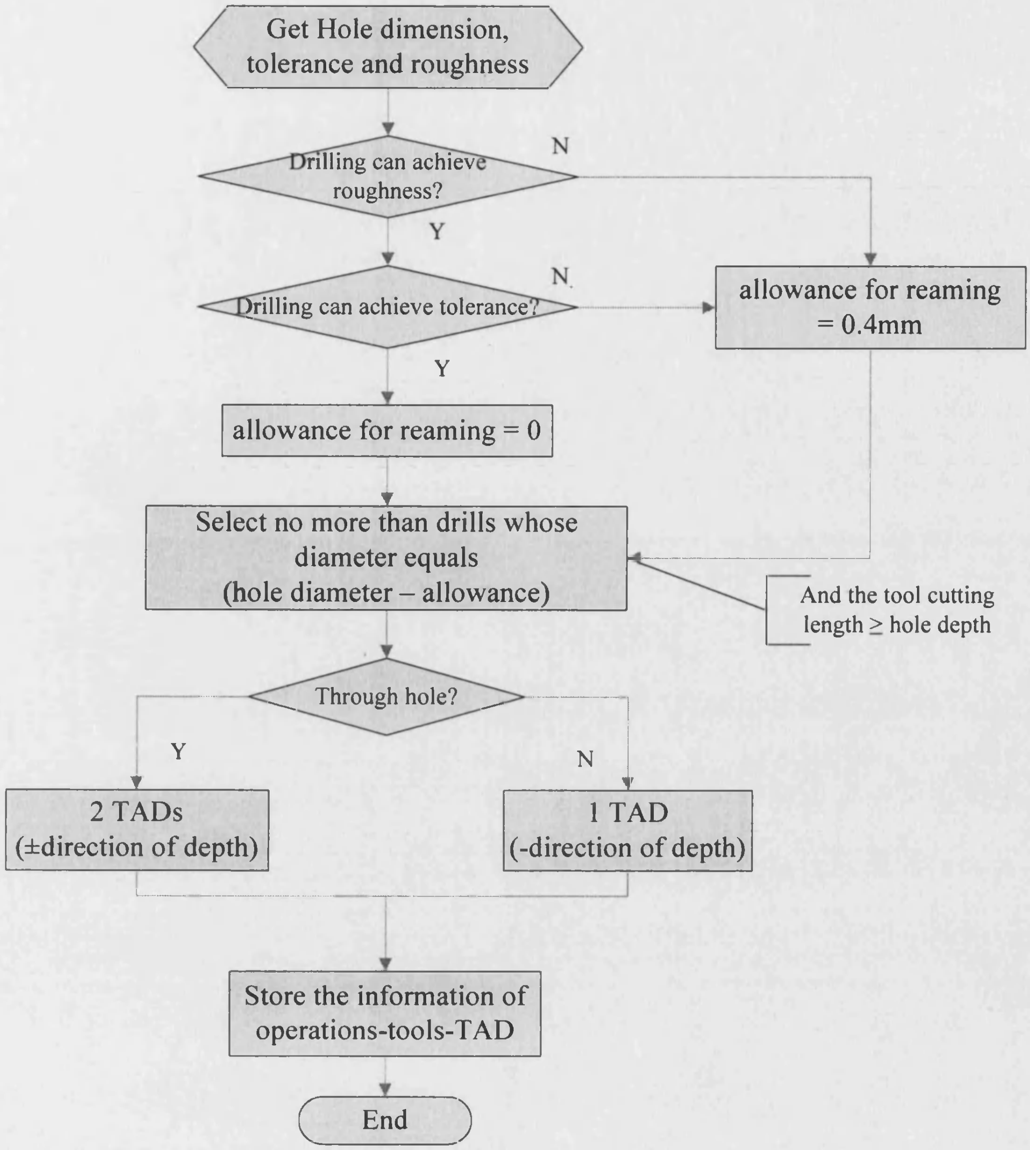


Figure 5.14 Work flow of operation selection for hole feature

Similar to the step feature, a through hole also has two possible TADs. However, it is not necessary to separate the data into two operation-sets to machine the feature. Unlike the step feature, both TADs for a through hole will be drilled with the same tool and same cutting conditions. So they are only considered as alternative operations with different TADs. The work flow of this selection procedure is shown in figure 5.14.

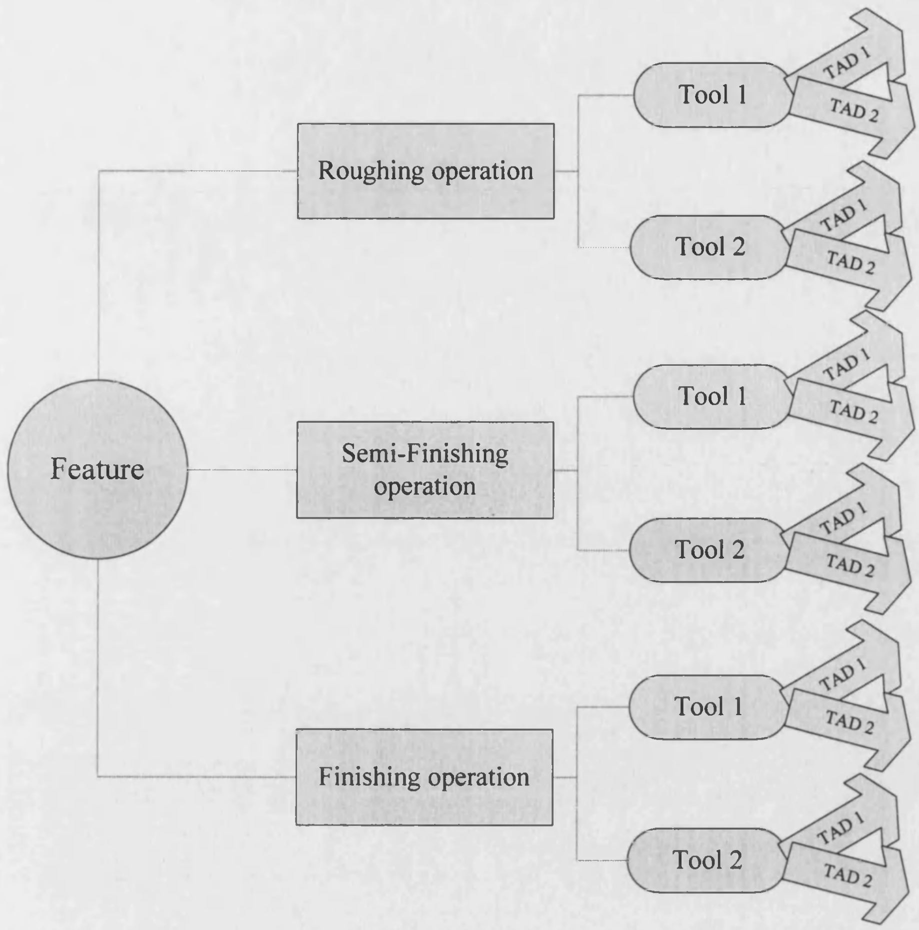


Figure 5.15 Operation alternatives

As discussed in the above sections, a number of common parameters have been determined through the procedure of operations selection. These were: (i) operation description (drilling, reaming, rough milling, semi-finish milling or finish milling), (ii) a list of applicable tools with their diameters, total length and flute length, (iii) Depth of cut for each operation and (iv) TADs for each operation. These parameters along with corresponding feature information make up alternative operations for machining a particular feature. For example, in figure 5.15, a feature needs roughing, semi-finishing and finishing operations. Each operation can be executed by two applicable tools and each tool can complete the operation in two TADs. Therefore, to machine this feature, three operations are required and each operation has 2x2 = 4 operation alternatives. All of this information is stored into the operation list table in the database for use in cutting condition parameters selection and machining time calculation.

5.3 Cutting condition estimation and machining time calculation

Once the machine tool and tooling have been selected for the part under consideration, there are only three other parameters remaining that can influence the success of the machining. These are the cutting speed, feed rate and depth of cut to be used for each operation (Ludema et al., 1987). To accurately determine the precise data for any machining operation can be difficult without knowledge of the exact practicalities involved. But it is possible to give a good estimate of the speeds and feeds involved based on raw material and these estimates are considered satisfactory for the optimisation process. Once these are calculated, the machining times can be calculated for each operation and the total machining time determined.

5.3.1 Cutting condition estimation

There are numerous factors that should be considered when setting all three of the above process parameters. These included:

Operating constraints such as manufacturing practice, the manufacturing process, machine tool characteristics and capability and available processing time as specified by production planning.

Operating requirements such as the workpiece material and geometry, the operation being performed and the tooling data.

Tool performance factors such as the tool material and geometry and the use of cutting fluids. (Scallan 2003)

As all of the above have been considered in detail for the selection of suitable tooling, they will not be considered again here. Only those factors that have a significant influence on the calculation of the process parameters will be considered in this chapter namely:

- The workpiece material and geometry
- The tool material and geometry

These two factors are to be considered when estimate the cutting conditions.

1. Surface cutting speed

The cutting speed for a machining operation refers to the speed at which the cutting edge of the tool passes over the surface of the workpiece. It is invariably also referred as the surface speed. It is always considered as the maximum relative speed between the tool and the workpiece and is usually quoted in metres per minute (m min^{-1}). The cutting speed V_c is subsequently used to calculate the time taken for the operation, that is, the machining time T.

Generally, cutting speeds for specific combinations of part and tool material are stated in ranges as given in Table 5.4 (Scallan 2003). In practice, the high end of the range will be for light finishing and the lower ends for roughing cuts. The ranges are suitable for average metal cutting conditions.

Table 5.4 Surface cutting speeds in metres per minute (Scallan 2003)

Part material	Surface cutting speed (m min^{-1})	
	HSS	Carbide
Low-carbon steels	20-110	60-230
Medium-carbon steels	20-80	45-210
Steel alloys (Ni-based)	20-80	60-170
Grey cast iron	20-50	60-210
Stainless steels	20-50	55-200
Chromium nickel	15-60	60-140
Aluminium	30-110	60-210
Aluminium alloys	60-370	60-910
Brass	50-110	90-305
Plastics	30-150	50-230

2. Spindle speed

Typically the cutting speeds are determined by using handbooks and reference material. From these the actual spindle speed to achieve the desired surface speed is then calculated.

The actual spindle speed to be set, which will maintain the required surface speed, depends on the diameter of the cutter (for milling and drilling). Therefore, if a small diameter and a large diameter have to be machined at the same surface speed, then the smaller diameter must rotate quicker. The equation presented to calculate the cutting speeds can be used to calculate the spindle speed by simple transposition as follows:

$$N = \frac{V_c \times 1000}{\pi D} \qquad \text{(Eq5. 1)}$$

Where N is the revolution of the cutter for milling/drilling, V_c the surface cutting speed (m min^{-1}) and D the diameter of the cutter for milling and drilling (mm).

3. Feed rates

The feed rate of a machining operation is defined as the speed at which the cutting tool penetrates the workpiece. This is usually stated in either millimetre per spindle revolution (mmrev^{-1}) or as millimetre per minute (mm min^{-1}).

The manufacturers of milling cutters state recommended feed rates in mmrev^{-1}(f_r) mm min^{-1}(f_m) or mm/tooth(f_t). For this research feed rates quoted in mm/tooth have been used and can be used to determine the mm rev^{-1} as follows:

$$f_r = f_t n \qquad \text{(Eq5. 2)}$$

Where n is the number of teeth on the cutter.

From this, the feed f_m in mm min^{-1} can be calculated as follows:

$$f_m = f_r N = f_t n N \qquad \text{(Eq5. 3)}$$

Some typical feed rates for milling are shown in table 5.5 for both HSS and carbide cutters.

Table 5.5 Typical feed rates for milling in millimetres per tooth (Scallan 2003)

Part material	Surface cutting speed (m min^{-1})			
	HSS		Carbide	
	Face mills	End mills and slot mills	Face mills	End mills and slot mills
Low-carbon steels	0.2-0.5	0.1-0.25	0.1-0.75	0.15-0.40
Medium-carbon steels	02.-0.5	0.1-0.25	0.1-0.75	0.15-0.40
Steel alloys (Ni-based)	0.2-0.8	0.15-0.4	0.3-1.2	0.2-0.5
Grey cast iron	0.15-0.65	0.075-0.3	0.15-0.75	0.075-0.4
Stainless steels	0.2-0.6	0.1-0.3	0.3-1.2	0.2-0.5
Chromium nickel	0.1-0.6	0.1-0.3	0.3-1.2	0.2-0.5
Aluminium	0.25-0.75	0.15-0.4	0.25-1.0	0.1-0.5
Aluminium alloys	0.25-0.75	0.15-0.4	0.25-1.0	0.1-0.5
Brass	0.25-0.5	0.1-0.25	0.25-0.65	0.1-0.4
Plastics	0.2-0.8	0.15-0.4	0.2-1.2	0.1-0.6

HSS drills are used extensively for producing smaller holes. Since small diameter drills are liable to break, the feed rate is related to drill size as shown in table 5.6 (Scallan 2003). For the production of larger drilled holes, carbide drills are preferred. The feed rates for these are similar to those for carbide endmill cutters (The Peck-drilling is not considered in this research at the moment). However, it should be noted that as the depth of the hole being drilled increases, the speeds and feeds should be reduced. Finally, the feed used will also depend on the surface roughness required.

Table 5.6 Typical feed rates for HSS and carbide drills (Scallan 2003)

Drill diameter (mm)	Drilling feed rate f_r (mm rev^{-1})	
	HSS	Carbide
2	0.05	0.15
4	0.10	0.15
6	0.12	0.15
8	0.15	0.18
10	0.18	0.25
12	0.21	0.25
14	0.24	0.28
16	0.26	0.32
18	0.28	0.32
20	0.30	0.32

The feed rate f_m (in mm min^{-1}) for drilling tools can be determined, using the feed f_r in mm rev^{-1}, from the equation:

$$f_m = f_r N \tag{Eq5. 4}$$

4. Depth of cut for milling

A general definition for depth of cut is that the depth of cut can be defined as the difference between the original surface and that being produced by the cutting tool. There are various factors that can affect the depth of cut. However, of these the most important are the tool and workpiece material and the tool geometry. General recommendations for depth of cut are given for a variety of processes in machinery's handbook (Oberg et al. 2004).

For milling, general guidelines for both face and slot milling recommend a cutting depth of 1-4 mm, while end milling depths should be around 1-2 mm (Schey, 1987). However, in general a maximum depth of cut half the cutter diameter, up to 8 mm, can be used (Kalpakjian, 1995). For simplification, 8mm, 6mm and 4mm are selected for rough milling, semi-finish milling and finish milling operations respectively in this research.

5.3.2 Calculation of machining time

All three process variables described above will affect the time taken for machining. In turn the machining time will determine the output for the components being machined and have a direct bearing on the cost of manufacture. In job and batch manufacturing where there tends to be a high variety of work, the development of such data is difficult and for many jobs the times have to be calculated in order to accurately estimate the production rate/output and the cost. To calculate the machining times the speed, feed and depth of cut outlined above are used together with the type of feature and its dimensions.

The estimation of machining time for drilling operations is the most straightforward and is performed using the following equation:

$$T_d = \frac{d_c}{f_m} \quad \text{(Eq5. 5)}$$

where T_d is the machining time for a drilling operation (min) and d_c the required total depth of cut (mm). The above assumes that the drilling operation is continuous, rather than, say, a pecking operation. It is also assumed that the surface roughness of the drilled hole is not an issue.

In face milling, slot milling and end milling where the axis of rotation of the cutter is perpendicular to the bottom surface being machined. The main cutting action is from the teeth on the periphery of the cutting tool, while the tool face provides a finishing action. The machining time for these milling types can be calculated as follows:

$$T_m = \frac{l_m}{f_m} n_{pdr} \qquad \text{(Eq5. 6)}$$

$$n_{pdr} = \left| \frac{d_c}{d_m} \right|^{+} \qquad \text{(Eq5. 7)}$$

where T_m is the machining time for a milling operation (min), l_m the length of tool path per pass for the milling operation (mm), f_m the feed rate (mm min^{-1}), n_{pdr} the number of passes for milling operations, $|x|^{+}$ the round up to the next integer number, d_c the required total depth of cut and d_m the depth of cut in terms of operation sub type. So it can be seen that all the parameters which are required to calculate machining times have been determined except l_m , the length of tool path per pass (mm).

The tool travel distance comprises the distance the tool moves in the tool approach direction and the length of the tool path in the local XY plane (the plane perpendicular to the approach direction). The operation for which tool travel is most easily determined is drilling. Cutting distance to drill a hole is shown in figure 5.16. The parameters required to determine l_m for the milling operation are the length, L, width, W and the corner radius, r, of the feature, as well as the diameter, D, of the cutting tool. Assumptions about how material is removed to create a feature have been used to develop the equations that define the l_m length. The method developed by Maropoulos et al. (2000) is used here to calculate the l_m value.

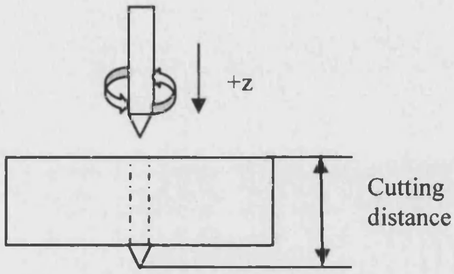
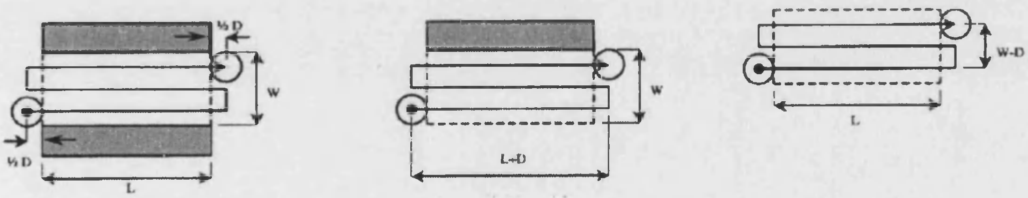


Figure 5.16 Cutting distance of a through hole

Figure 5.17 shows the zigzag operations required for milling through slots, through steps and faces and the l_m can be calculated using equations (Eq 5.8) and (Eq 5.9).



(a) Through slot (b) Through step (c) Face

Figure 5.17 Zigzag cutting operation for milling through slots, through step and face (Maropoulos et al. 2000)

$$n_{pw} = \left| \frac{W}{D} \right|^{+} \qquad \text{(Eq5. 8)}$$

$$l_m = n_{pw}(L + D) + W - D \qquad \text{(Eq5. 9)}$$

where n_{pw} is the number of passes required to cut the width of the feature and L, W, D the dimensions described above and shown in figure 5.17.

As shown in figure 5.18, for one end closed slots, l_m can be calculated in equation (Eq 5.10) and (Eq 5.11), where $|x|^{+}$ is the round up to the nearest even number.

$$l_m = L(2 + n_{pw}) + 2W - 2D \qquad \text{(Eq5. 10)}$$

$$n_{pw} = \left|\frac{W}{D}\right|^{+even} - 2 \qquad \text{(Eq5. 11)}$$

$$l_m = L(1 + n_{pw}) + 2W \qquad \text{(Eq5. 12)}$$

$$n_{pw} = \left|\frac{W}{D}\right|^{+even} - 1 \qquad \text{(Eq5. 13)}$$

For one end closed steps l_m can be calculated using equation (Eq 5.12) and (Eq 5.13).

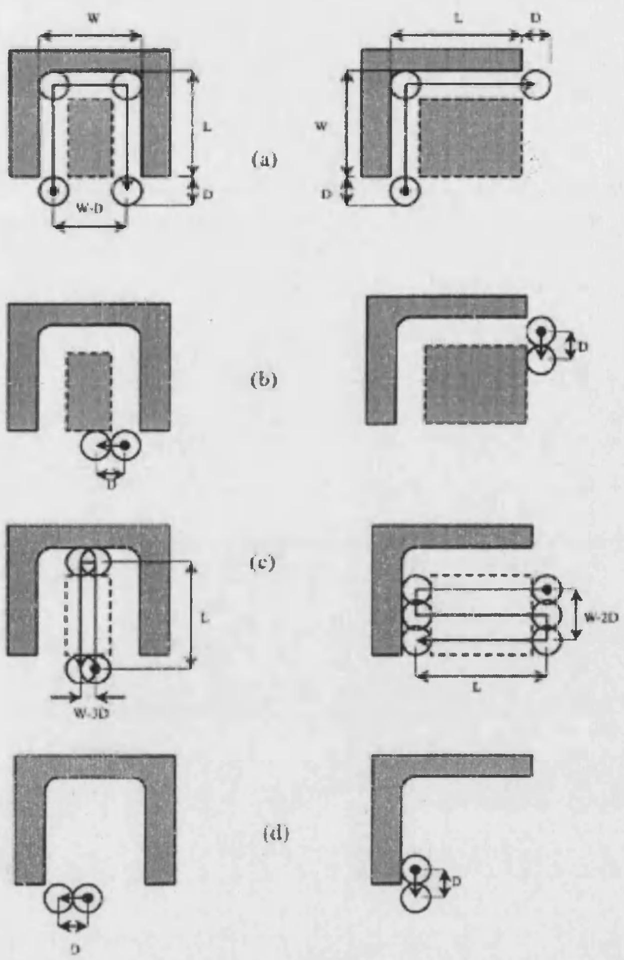


Figure 5.18 Cutter movement during milling of non-through slots and steps (Maropoulos et al. 2000)

The cutter movement during milling of a closed pocket is shown in figure 5.19 and the l_m can be calculated using equation (Eq 5.14) and (Eq 5.15).

$$l_m = L(2 + n_{pw}) + 4W - D(10 + n_{pw}) \quad \text{(Eq5. 14)}$$

$$n_{pw} = \left|\frac{W}{D}\right|^{+even} - 2 \quad \text{(Eq5. 15)}$$

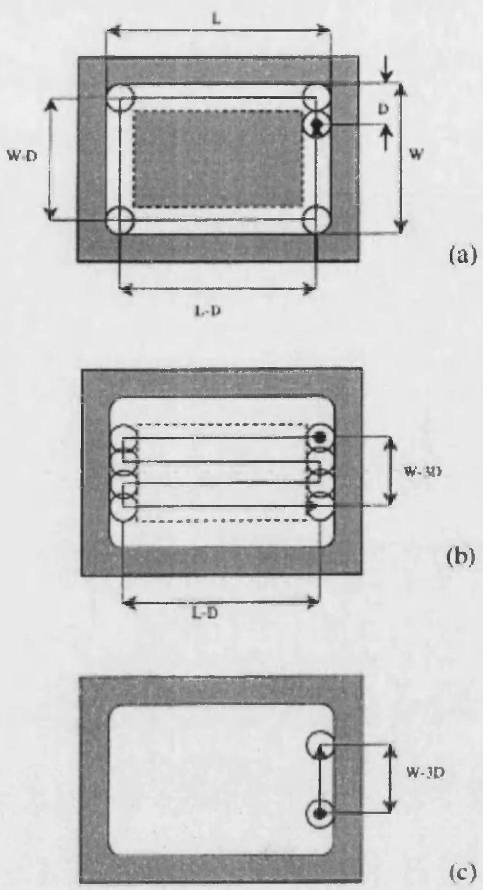


Figure 5.19 Cutter movement during milling of a closed pocket
(Maropoulos et al. 2000)

As described previously, several operation alternatives are generated in the operation selection module. Every operation alternative may require different tools to execute it, which results in different cutting conditions being chosen and may result in different machining time. Therefore it is necessary to select suitable cutting conditions and calculate the corresponding machining times for them. Figure 5.20 to figure 5.24 show the work flow of calculating machining times for different features.

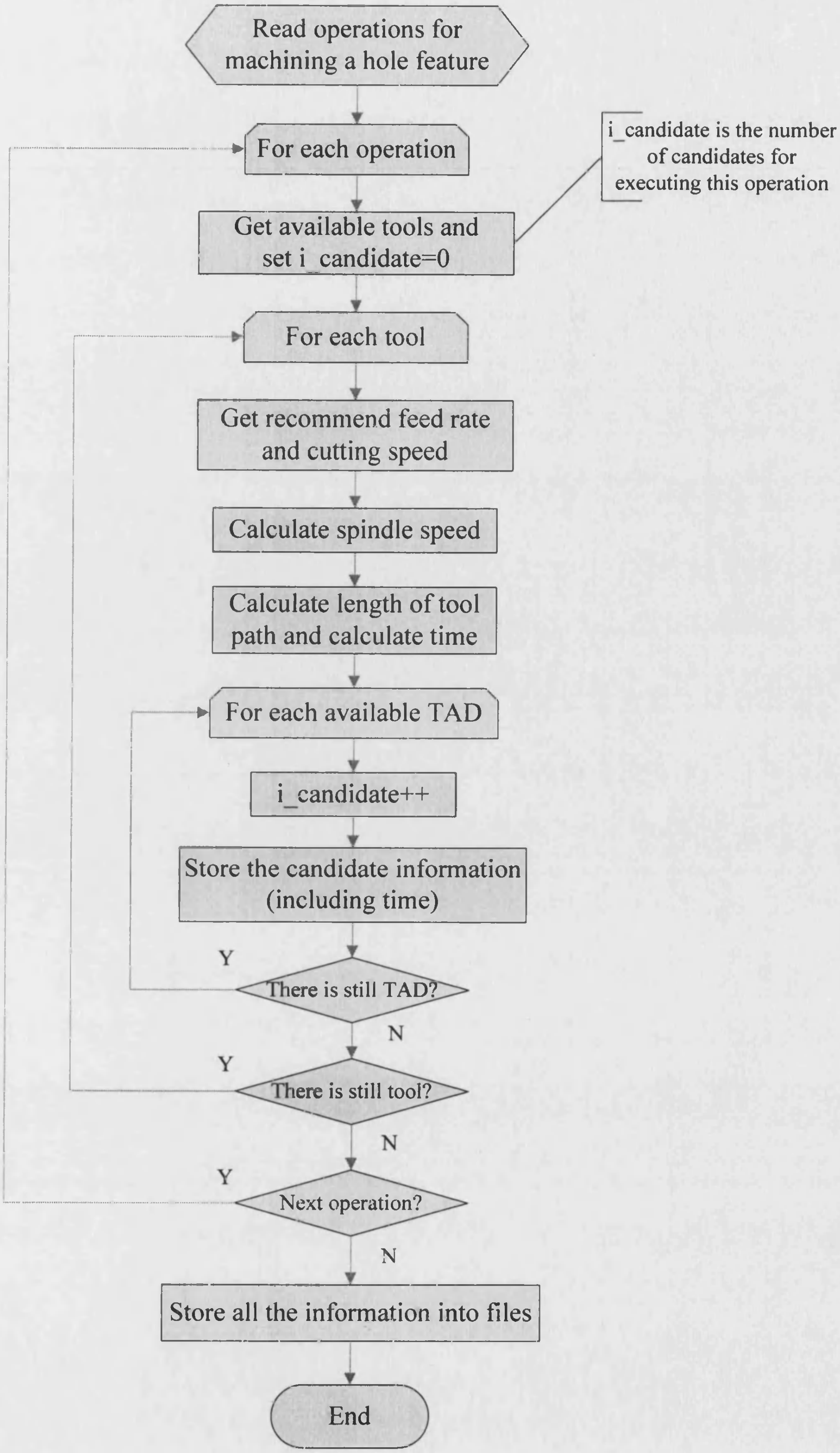


Figure 5.20 Work flow of machining time calculation for hole features

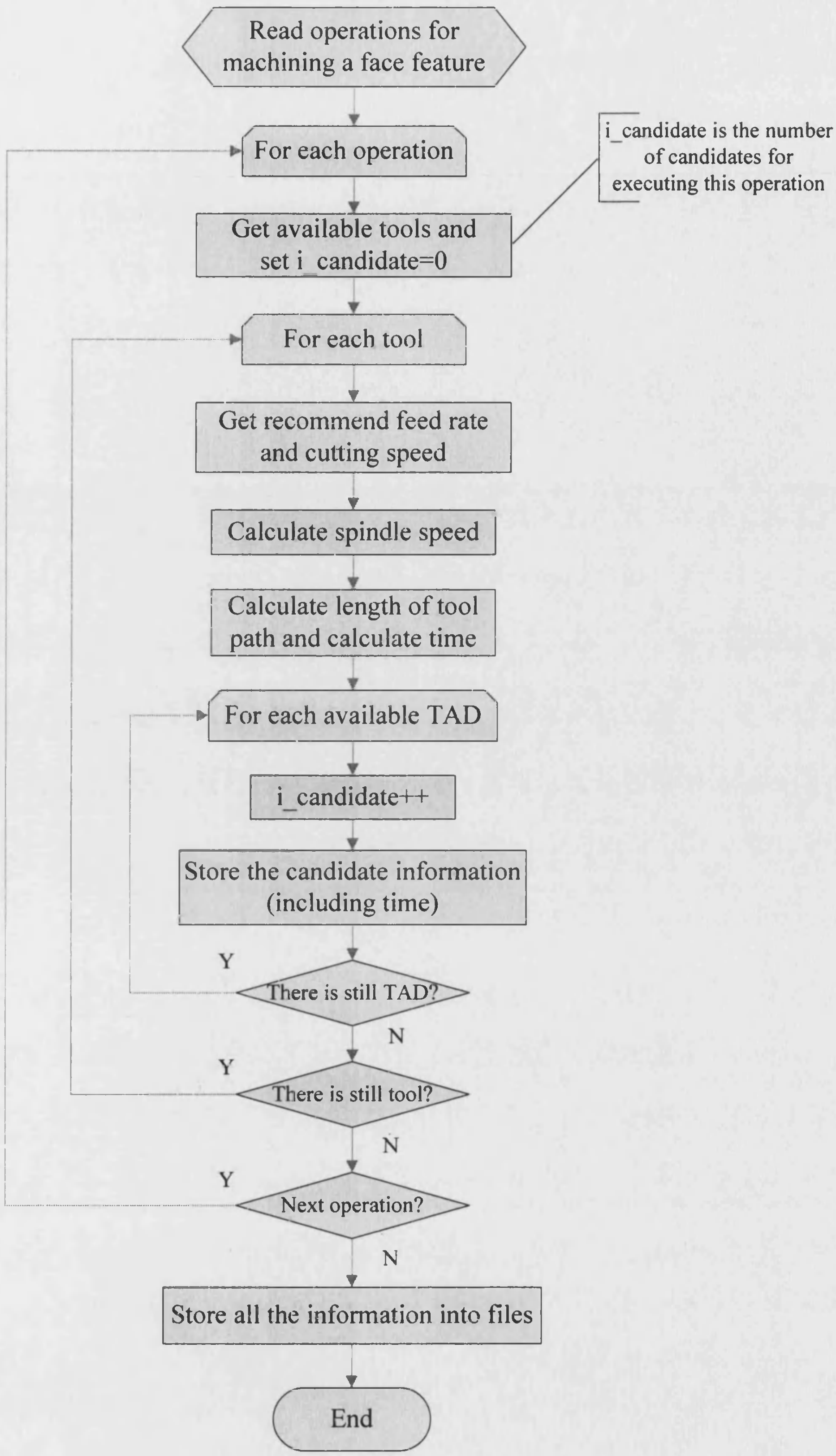


Figure 5.21 Work flow of machining time calculation for face features

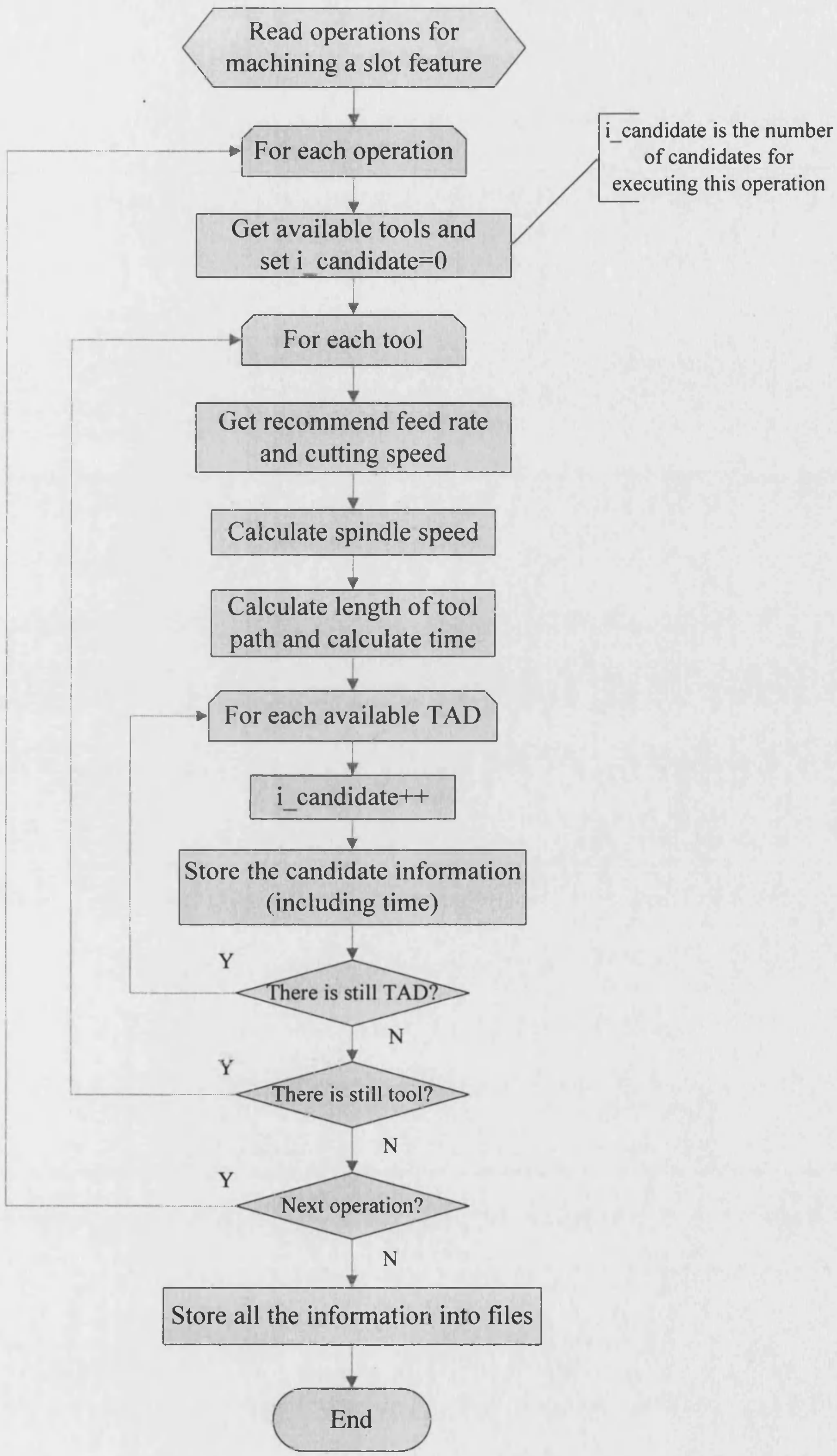


Figure 5.22 Work flow of machining time calculation for slot features

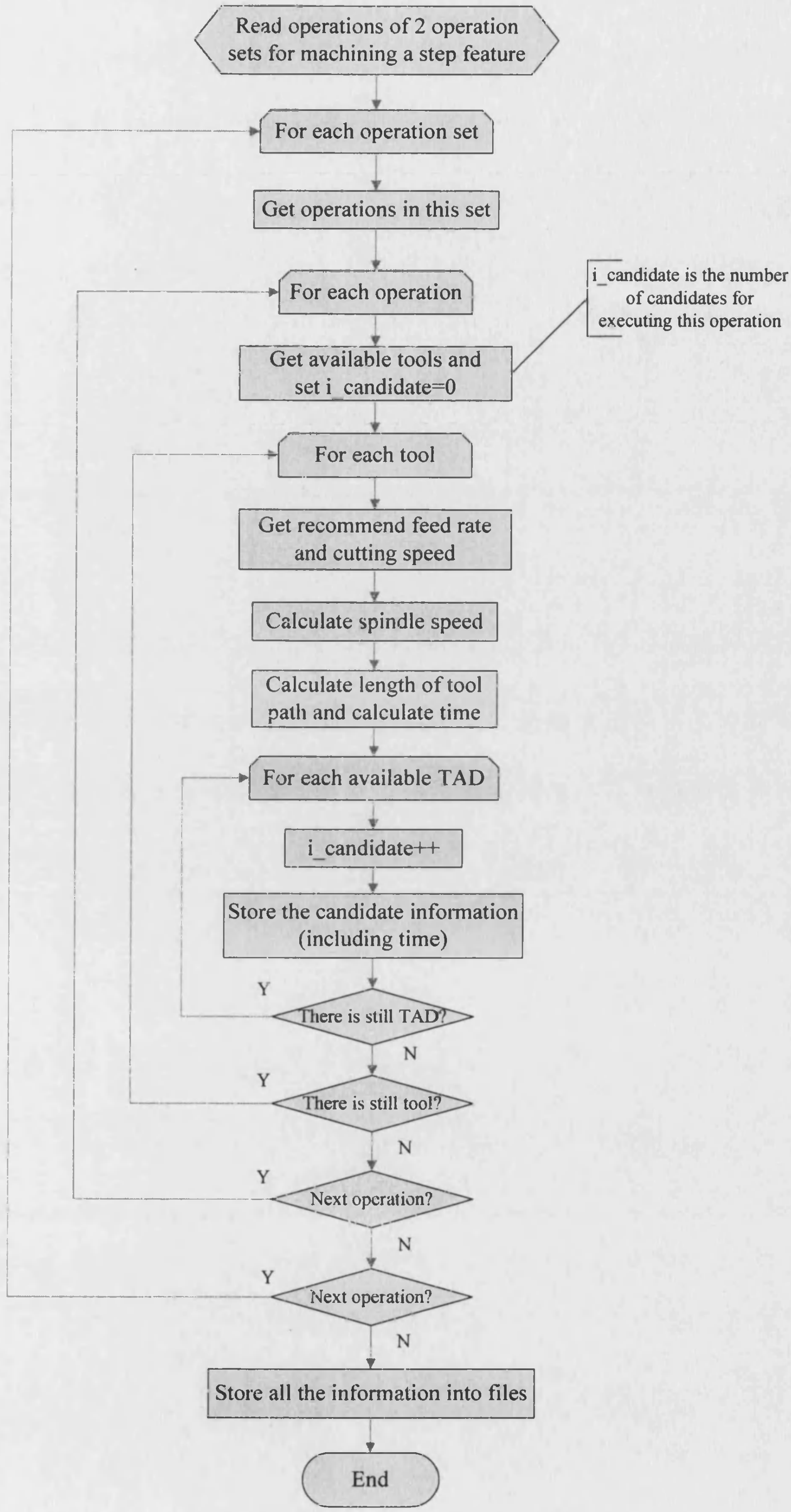


Figure 5.23 Work flow of machining time calculation for step features

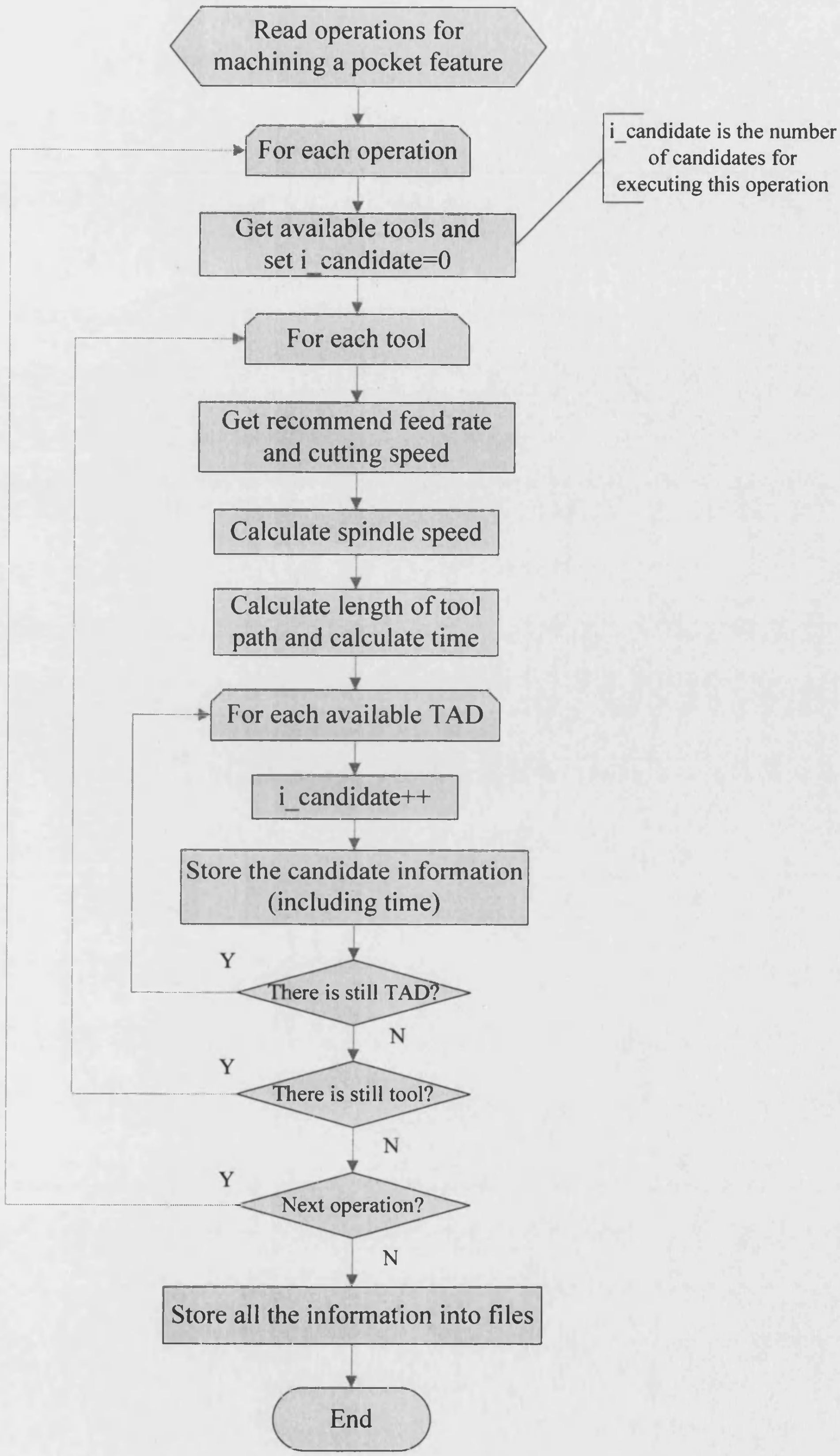


Figure 5.24 Work flow of machining time calculation for pocket features

5.4 Summary

This chapter first discussed the part information required for process planning, then presented the feature taxonomy and developed a feature-based modeller to input the features information of a part for subsequent process planning. After using an Object Oriented Programming Strategy to represent the feature, the system selects the operations for each feature. In this procedure, the operation subdivision, tools selection and TAD selection are described in detail. Finally, the cutting conditions estimation and machining time calculation for operations are presented and discussed.

Chapter 6 Operation Sequencing with Particle Swarm Optimisation algorithm

6.1 Introduction

Operations sequencing is one of the most important activities in generative process planning and it is used to determine the sequence of machining operations that are required to produce a part. In operations sequencing, it is necessary to apply good manufacturing practices and maintain the consistency of the desired functional specifications of a part. As discussed in section 2.4, a good sequence of operations can ensure low machining cost (affecting machine utilisation, setups, tool changes, etc.) and satisfy precedence constraints amongst the operations. However, for parts with complex structures and features, operations sequencing is well known as a complicated combinatorial decision problem. The major difficulties include: (1) the search space is usually very large, and many previously developed methods can not find optimised solutions effectively and efficiently, and (2) there are usually a number of precedence constraints in sequencing the operations due to manufacturing practice and rules, which make the search more difficult.

To address these issues, some optimisation approaches based on modern heuristics or evolutionary algorithms, such as the Genetic Algorithm (GA) (Bhaskara Reddy et al. 1999, Qiao et al. 2000, Yip-Hoi and Dutta 1996, Zhang et al. 1997, Ding et al. 2005), Simulated Annealing (SA) algorithm (Ma et al. 2000, Lee et al. 2001) and Tabu search algorithm (Lee et al. 2001, Li et al. 2004), have been developed in the last two decades and significant improvements have been achieved. However, there still remains potential for further improvement, as discussed in chapter 2 and as follows:

(1) Current representation of process plans (Operation-Tool-TAD) is still not complete and does not include sufficient information especially for 5-axis machining. In 3-axis machining, a TAD indicates a definite set-up, but in 5-axis machining, the TAD of an operation can be achieved from 5 possible set-ups. This increases the difficulty of operation sequencing and set-up selection.

(2) Precedence constraints between operations need to be considered thoroughly and carefully so as to keep the solutions feasible. Different constraint handling mechanisms should be selected in terms of different characteristics of the algorithms.

(3) The performance evaluation criteria of a process plan need to be handled carefully, different criteria should be selected accordingly for different objectives. For example, the process plan that can achieve the minimal machining time does not assure this process plan has the minimal machining cost;

(4) Current algorithms are still not efficient. GA's perform very well in the early optimising stage but later it is easy for them to be trapped into local optima and not find the optimised solution, especially for complex problems. SA converges quickly and can find optimal solution for problems that are not very complex, but as the complexity of the problems increases, its possibility of finding optimised solution reduces.

To improve overall performance, a more comprehensive representation scheme of process plans needs to be developed, a more reasonable constraint handling mechanism needs to be developed and it is necessary to adopt a more agile, effective and efficient optimisation algorithm.

In this chapter, a Particle Swarm Optimisation (PSO) approach has been developed to concurrently consider the processes of selecting machining resources, determining setup plans, and sequencing operations for a prismatic part as an optimisation procedure. The representation of the process plan and the evaluation criteria for 3-axis machining are first addressed. This is then extended to 5-axis machining. Finally the details of applying the PSO algorithm for operation sequencing are described.

6.2 Knowledge Representation of Process Plans for 3-axis machining

6.2.1 Introduction of PSO algorithm

As described in chapter 2, Particle Swarm Optimization (PSO) is a modern evolutionary computation technique based on a population mechanism. The PSO algorithm was inspired by the social behaviour of bird flocking and fish schooling (Kennedy and Eberhart 1995). Three aspects will be considered simultaneously when an individual fish or bird (particle) makes a decision about where to move: (1) its current moving direction (velocity) according to the inertia of the movement, (2) the best position that it has achieved so far, and (3) the best position that its neighbour particles have achieved so far. In the algorithm, the particles form a swarm and each particle can be used to represent a potential solution of a problem. In each iteration, the position and velocity of a particle can be adjusted by the following formulae that take the above three considerations into account. After a number of iterations, the whole swarm will converge at an optimised position in the search space.

$$V_i^{t+1} = w * V_i^t + c_1 * Rand() * (P_i^t - X_i^t) + c_2 * Rand() * (P_g^t - X_i^t) \quad \text{(Eq 6. 1)}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad \text{(Eq 6. 2)}$$

$$X_i = (X_{i1}, X_{i2}, ..., X_{iN}) \quad \text{(Eq 6. 3)}$$

$$V_i = (V_{i1}, V_{i2}, ..., V_{iN}) \quad \text{(Eq 6. 4)}$$

Here, i is the index number of particles in the swarm; t is the iteration number; V and X are the velocity vector and the position vector of a particle respectively. For an N-dimensional problem, V and X can be represented by N particle dimensions as Equations 6.3 and 6.4 show. P_i is the local best position that the ith particle has achieved so far; P_g is the global best position that all the particles have achieved so far; w is the inertia weight to adjust the tendency to facilitate global exploration (smaller w) and the tendency to facilitate local exploration to fine-tune the current search area (larger w); $Rand()$ returns a random number in [0,1]; c_1 and c_2 are two constant numbers to balance the effect of P_i and P_g.

6.2.2 Representation of the process plan

To conduct process planning, parts are represented by manufacturing features. Figure 6.1 shows a part composed of m features. Each feature can be manufactured by one or more machining operations (n operations in total for the part). Each operation can be executed by several alternative plans if different machines, cutting tools or set-up plans are chosen for this operation (Case and Harun 2000, Maropoulos et al. 2000, Carpenter and Maropoulos 2000). A set-up is usually defined as a group of operations that are machined on a specified machine with the same fixture. Here, a set-up is equivalently defined as a group of operations with the same Tool Approach Direction (TAD) machined on a machine. For example, in figure 6.2, a through hole with two TADs is considered to be related to two set-ups (Li et al. 2004). A process plan for a part consists of all the operations needed to machine the part and their relevant machines, cutting tools, TADs, and operation sequences. A good process plan of a part is built up based on two elements: (1) the optimised selection of the machine, cutting tool and TAD for each operation; and (2) the optimised sequence of the operations of the part. Hence, the developed algorithm needs to address these two aspects.

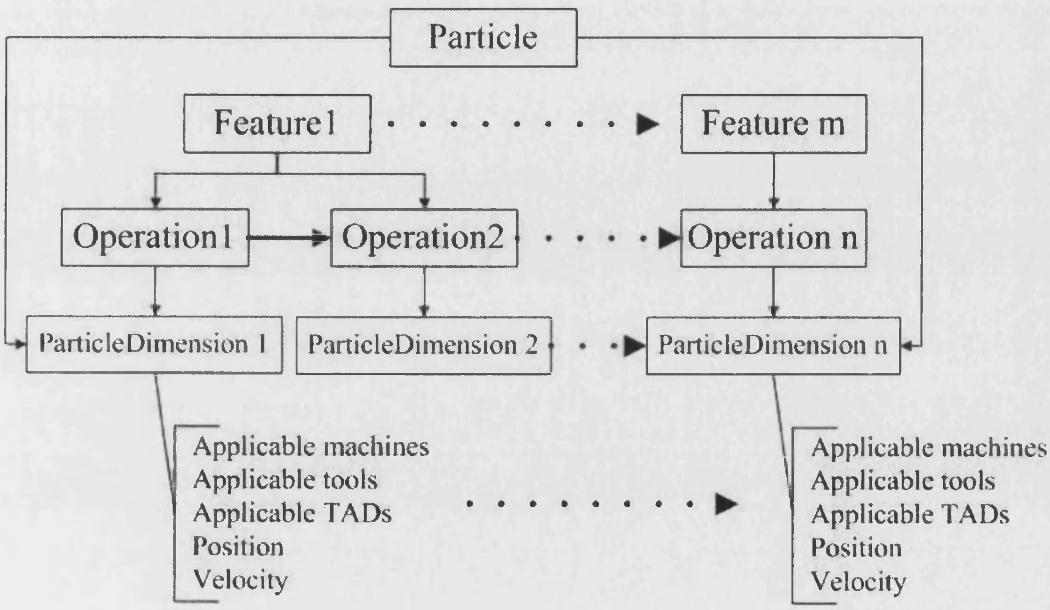


Figure 6.1 Representation of a process plan (particle).

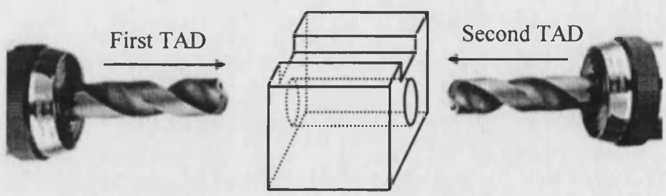


Figure 6.2 A through hole with two TADs.

To apply the PSO algorithm to the process plan optimisation problem, two issues have to be handled first:

(1) Encode a process plan to produce a particle. As shown in figure 6.1, each operation is modelled as a particle dimension that includes information on machines, cutting tools and TADs, and the details are listed in Table 6.1. Here a position variable and a velocity variable are used to represent the position and velocity of an operation, respectively. All the particle dimensions (operations) executed to make the part form a particle (a process plan). As shown in table 6.2, the array variable *Oper[n]* represents a process plan which consists of *n* ParticleDimensions (operations). A particle can be initialised in the following steps:

- All the operations are given an Operation_id from 1 to *n*.
- Machine_list, Tool_list and TAD_list applicable for each operation are specified, and a machine, tool and TAD are randomly selected from the three lists to execute the operation.
- A random position between [0, 1] and a random velocity between [-1, 1] are initialised for each ParticleDimension in the particle. The sequence of operations is determined by the relative values of their positions.

In table 6.2, an initialised particle with 5 ParticleDimensions is shown.

(2) Decode the particle to get a sequenced process plan. In each iteration, when all the ParticleDimensions in a particle have been updated, the operation sequence can be determined by the relative positions of the ParticleDimensions (Cagnina et al. 2004). For example, in table 6.3, the sequence of the particle dimensions will be (operation3, operation4, operation5, operation2, operation1) according to the descending order of their position values. By using a number of iterations to update the positions and velocities of the particle dimensions in each particle, an

optimised sequence (i.e., an optimised process plan) can be achieved after a number of iterations.

Table 6.1 Class definition of a particle dimension (an operation).

Class ParticleDimension: an operation		
Data types	Variables	Descriptions
Int	Operation_id	The id of the operation
int	Machine_id	The id of a machine to execute the operation
int	Tool_id	The id of a cutting tool to execute the operation
int	TAD_id	The id of a TAD to apply the operation
Int[ ]	Machine_list[ ]	The candidate machine list for executing the operation
Int[ ]	Tool_list[ ]	The candidate tool list for executing the operation
Int[ ]	TAD_list[ ]	The candidate TAD list for applying the operation
double	Position	The position value of the operation
double	Velocity	The velocity value of the operation

Table 6.2 Class definition of a particle (a process plan).

Class Particle: process plan		
Data types	Variables	Descriptions
ParticleDimension	Oper[n]	Define a process plan Oper[n] based on the above class-ParticleDimension. n is the number of operations in the plan
double	TMC	Total Machine Cost of the plan
Double	TTC	Total Tool Cost of the plan
Double	TSC	Total Set-up Cost of the plan
Double	TMCC	Total Machine Change Cost of the plan
double	TTCC	Total Tool Change Cost of the plan
Double	APC	Additional Penalty Cost of violating constraints in the plan
Double	TC	Total Cost of the plan
ParticleDimension	Pi[n]	Store the best plan that the particle has achieved so far

Table 6.3 An initialised particle.

Operation_id	1	2	3	4	5
Machine_id	2	3	3	2	1
Tool_id	5	4	6	3	4
TAD_id	3	3	-3	2	1
Position	0.1	0.2	0.5	0.4	0.3
Velocity	0.5	-0.3	-0.6	0.8	0.1
Relative Position (Sequence no.)	5	4	1	2	3

6.2.3 Evaluation criteria

Machining cost is typically used to measure the quality of a process plan quantitively. The machining cost of a plan is comprised of machine utilisation costs, tool utilisation costs, machine change costs, set-up change costs, tool change costs and additional penalty cost (Li et al. 2002, Li et al. 2004). The costs can be computed as below.

The Total Machine Cost (TMC). TMC is the total costs of the machines used in a process plan, and it can be computed as:

$$TMC = \sum_{i=1}^{n} (Oper[i].Machine\_id * MCI) \qquad \text{(Eq 6. 5)}$$

where MCI is the machine cost index for a machine.

The Total Tool Cost (TTC). TTC is the total cost of the cutting tools used in a process plan, and it can be computed as:

$$TTC = \sum_{i=1}^{n} (Oper[i].Tool\_id * TCI) \qquad \text{(Eq 6. 6)}$$

where TCI is the tool cost index for a tool.

Number of Set-up Changes (NSC), Number of Set-up (NS), and Total Set-up Cost (TSC). After the particle is decoded to a sequenced process plan, in a 3-axis machining environment, a set-up change between two consecutive operations in the sequence can be defined according to table 6.4 (set-up change in 5-axis machining will be discussed in a later section), and the NSC can be computed as:

$$NSC = \sum_{i=1}^{n-1} \Omega_2(\Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id), \\ \Omega_1(Oper[i].TAD\_id, Oper[i+1].TAD\_id)) \qquad \text{(Eq 6. 7)}$$

The corresponding NS can be computed as:

$$NS = 1 + NSC \qquad \text{(Eq 6. 8)}$$

The Set-up Cost (SC) is considered to be the same for each set-up. Hence,

$$TSC = \sum_{i=1}^{NS} SC \qquad \text{(Eq 6. 9)}$$

where $\Omega_1(X,Y) = \begin{cases} 1 & X \neq Y \\ 0 & X = Y \end{cases}$, $\Omega_2(X,Y) = \begin{cases} 0 & X = Y = 0 \\ 1 & otherwise \end{cases}$.

(4) Number of Machine Changes (NMC) and Total Machine Change Cost ($TMCC$).

$$NMC = \sum_{i=1}^{n-1} \Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id) \qquad \text{(Eq 6. 10)}$$

The Machine Change Cost (MCC) is considered to be the same for each machine change. Hence,

$$TMCC = \sum_{i=1}^{NMC} MCC \qquad \text{(Eq 6. 11)}$$

(5) Number of Tool Changes (NTC) and Total Tool Change Cost ($TTCC$). A tool change is defined in table 6.5. NTC is computed as:

$$NTC = \sum_{i=1}^{n-1} \Omega_2(\Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id), \Omega_1(Oper[i].Tool\_id, Oper[i+1].Tool\_id)) \qquad \text{(Eq 6. 12)}$$

Similarly, the Tool Change Cost (TCC) is considered to be the same for each tool change. Thus

$$TTCC = \sum_{i=1}^{NTC} TCC \qquad \text{(Eq 6. 13)}$$

(6) Number of Violating Constraints (NVC) and Additional Penalty Cost (APC). During the optimisation process, it is difficult to ensure that each particle obeys

the constraints. To solve this problem, a penalty method has been used to adjust an infeasible particle towards its feasible domain.

$$NVC = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Omega_3(Oper[i].Operation\_id, Oper[j].Operation\_id)) \qquad \text{(Eq 6. 14)}$$

A fixed Penalty Cost (PC) is applied to each violated constraint. Thus

$$APC = \sum_{i=2}^{NVC} PC \qquad \text{(Eq 6. 15)}$$

where

$$\Omega_3(X,Y) = \begin{cases} 1 & \text{The sequence of } X \text{ before } Y \text{ violates constraints} \\ 0 & \text{The sequence of } X \text{ before } Y \text{ is in accordance to constraints} \end{cases}$$

(7) The Total Cost (TC).

$$TC = TMC + TTC + TSC + TMCC + TTCC + APC \qquad \text{(Eq 6. 16)}$$

Table 6.4 Definition of when a setup change is required in 3-axis machining

Conditions of Machining Two Consecutive Operations	**A Setup 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

Table 6.5 Definition of when a tool change is required

Conditions Of Machining Two Consecutive Operations	**A 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

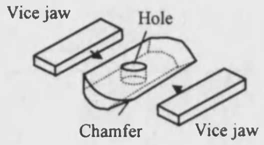
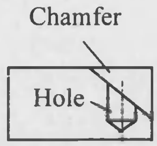
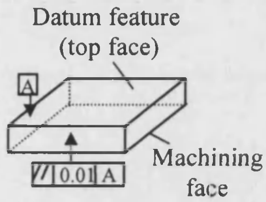
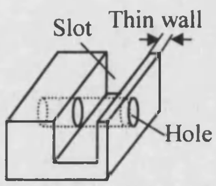
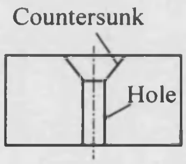
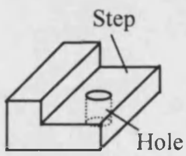
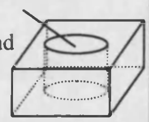
6.2.4 Precedence constraints

The geometric and manufacturing interactions (Faheem et al. 1998; Ong et al. 2001) between features as well as the technological requirements in a part can be considered to generate some preliminary precedence constraints between machining operations. The precedence constraints between operations are usually classified into seven types: (1) fixture interaction, (2) tool interaction, (3) datum interaction, (4) feature priority, (5) fixed order of machining operations, (6) thin-wall interaction and (7) material-removal interaction. (Li et al. 2004) The classifications, definitions and illustrative examples of precedence constraints are given in Tables 6.6 and 6.7. A feasible operation sequence must comply with the precedence constraints.

Table 6.6 Definitions and classifications of precedence constraints.

Constraints	Definitions
Fixture interactions	The clamping or supporting faces for machining a feature are destroyed by machining another feature earlier.
Tool interactions	The positioning faces required by a cutting tool to machine a feature are removed by the machining of another feature earlier.
Datum interaction	In order to locate a part for machining or inspection, some datum faces in the part are used as reference planes. A datum interaction occurs when machining a feature destroys the datum required for another feature.
Feature priorities	A feature should be machined before its associated features. Another case is that a feature should be machined first to provide entrance face for machining an interacting feature.
Fixed order of machining operations	This case includes some explicit precedence constraints, for example, turning-grooving-chamfering prior to thread cutting.
Thin-wall interactions	A thin-wall interaction occurs when the distance between features is very small and causes precedence constraints in machining.
Material-removal interactions	For two features with geometric interactions, if the different material removal sequences of features influence the cost or the quality of machining and cause precedence constraints between these features, a material-removal interaction occurs.

Table 6.7 Examples of precedence constraints.

Constraints	Examples	Explanations
Fixture interactions		The hole should be machined before the chamfer, otherwise it cannot be fixtured.
Tool interactions		In order to position a drilling tool correctly, the drilling of the hole should precede the machining of the chamfer.
Datum interaction		The top face (the datum feature) should be machined prior to the base face.
Thin-wall interactions		The good practice should be drilling the hole, then machining the slot to avoid the deformation of the thin wall.
Feature priorities		The countersunk is an associated feature and should be machined after the primary hole.
Material-removal interactions		The step should be machined prior to the hole for achieving high machining efficiency (milling is faster than drilling) and surface quality.
Fixed order of machining operations	Operations for a hole: (1) Drilling (2) Boring and (3) Reaming 	A typical sequence of machining a hole is drilling-boring and reaming.

6.3 The Modified PSO Algorithm

A traditional PSO algorithm can be applied to optimise the process plan in the following steps:

(1) Initialisation:

- Set the size of a swarm, e.g., the number of particles "*Swarm_Size*" and the max number of iterations "*Iter_Num*".
- To ensure the optimisation proceeds successfully, the initial populations of the swarm generated should be spread sufficiently over the search space to represent as wide a variety of solutions as possible. The method introduced in section 6.2.2 is used to initialise all the particles in the swarm. After the populations are initialised, it is required to decode every particle (process plan) in the swarm to get the operation sequence of each particle and then calculate the total cost (TC) of each particle according to equations 6.5-6.16.
- Set the local best $P_i[n]$ and the global best P_g with the lowest total cost TC.

(2) Iterate the following steps until *Iter_Num* is reached:

- For each particle in the swarm, and each ParticleDimension, (i.e., operation in particle), update ParticleDimension's velocity and position values according to equations 6.1 and 6.2, i.e., *Oper[1].Position, Oper[2].Position, …, Oper[n].Position.*
- Decode the particle into a sequenced process plan in terms of new position values and calculate the TC of the particle. Update the local best $P_i[n]$ and the global best P_g if a lower TC is achieved.

(3) Decode global best P_g to get the sequenced process plan.

However, the traditional PSO algorithm introduced above is still not effective in resolving the operation sequencing problem. There are two major reasons for this:

(1) Due to the inherent mathematical operators, it is difficult for the traditional PSO algorithm to consider the different arrangements of machines, tools and TADs

for each operation, and therefore the particle is unable to fully explore the whole search space.

(2) The traditional algorithm usually works well in finding solutions at the early stage of the search process (the optimisation result improves fast), but is less efficient during the final stage. Due to the loss of diversity in the population, the particles move quite slowly with low or even zero velocities. This make it is hard to reach the global best solution (Stacey et al. 2003). Therefore, and as with GA's, the whole swarm is prone to be trapped in a local optimum from which it is difficult to escape.

To solve these two problems and enhance the ability of the traditional PSO algorithm to find the global optimum, new operators, including mutation, crossover and shift, have been developed and incorporated in a new modified PSO algorithm. Meanwhile, considering the characteristics of the algorithm, the initial values of the particles and P_g (the global best position of all the particles in Eq 6.1) have been well manipulated.

The important modification details are described below.

(1) New operators in the algorithm

- Mutation. In this strategy, an operation is first randomly selected for a particle. From its candidate machining resources (Machine_list[], Tool_list[] and TAD_list[]), an alternative set (machine, tool, TAD) is then randomly chosen to replace the current machining resource in the operation. The probability of applying this strategy is defined as P_m.

- Crossover. Two particles in the swarm are chosen as Parent particles for a crossover operation. In the crossover, a cutting point is randomly determined, and each parent particle is separated as left and right parts of the cutting point. The positions and velocities of the left part of Parent 1 and the right part of Parent 2 are reorganised to form Child 1. The positions and velocities of the left part of Parent 2 and the right part of Parent 1 are reorganised to form Child 2. The probability of applying the crossover is defined as P_c.
- Shift. This operator is used to exchange the positions and velocities of two operations in a particle so as to change their relative positions in the particle. The probability of applying the shift is defined as P_s.

(2) Escape method for P_g

- Because the global best particle P_g influences every particle in the swarm, during the optimisation process, if the iteration number of obtaining the same best fitness is more than 10, then the mutation and shift operations are applied to P_g to try to escape from the local optima.

The workflow of the modified PSO algorithm is shown in figure 6.3.

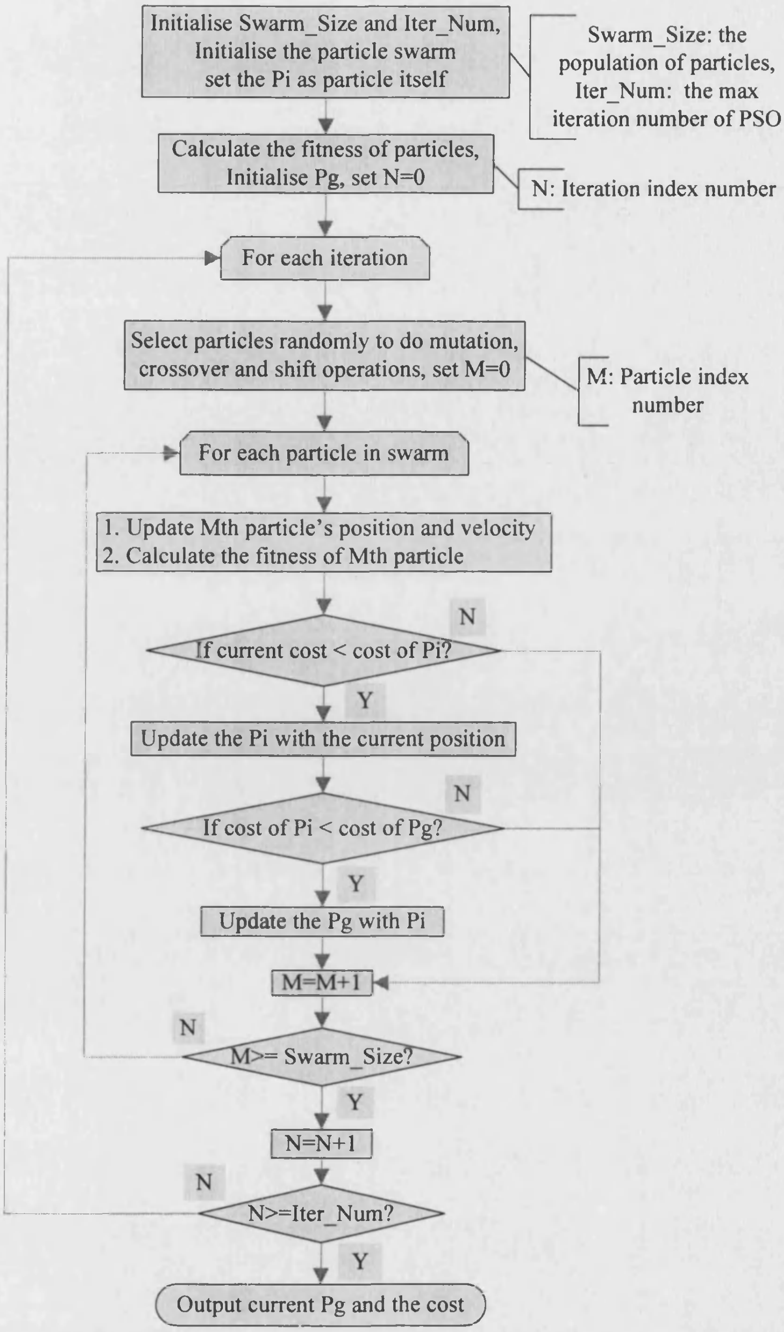


Figure 6.3 Workflow of the PSO algorithm.

6.4 PSO algorithm application in 5-axis machining

There are 3 major issues that need to be considered when optimising the process plans for 5-axis machining:

(1) Set-up plan determination. A set-up can be defined as a group of operations that are machined on a single machine with the same fixture. As discussed in chapter 5, in a 3-axis machining environment, a set-up is a group of features with the same TAD machined on a 3-axis machine, so different machines or different TADs in two consecutive operations mean a set-up change (shown in table 6.4). Here set-up is determined by the current TAD. However, in 5-axis, different TADs can be achieved by the two extra degrees of freedom movements with the same fixture (set-up). Therefore, two different TADs do not necessarily mean two different set-ups. The feature can be machined with the same fixture only if the TAD of the operation for this feature can be achieved by rotating or swivelling the work table. For a 5-axis machine, one single set-up (same fixture) can achieve 5 different TADS. On the other hand, one single TAD can be possibly achieved by 5 different set-ups (In theory, if the TAD of operation x is ZO (-3), the possible set-ups for this operation can be located in any of 5 directions except OZ (+3)). Therefore the representation of process plans for 5-axis machining and determination of set-up plans needs to be considered carefully.

(2) Different performance evaluation criteria. Due to the difficulty of determining the machining cost, it is easier to evaluate the performance of process plans by total machining time which has been calculated in the operation selection stage together with estimates of the set-up change time and the tool change time.

(3) Operation-set constraint. In chapter 5, an operation-set that may include rough milling/drilling, semi-finish milling/drilling or finish milling/drilling operations is introduced to represent the operations in one single TAD. So there are two alternative operation-sets for machining a step feature namely operation-set 0

and operation-set 1. Different tools and TADs are selected for the operations in these two operation-sets. Therefore if an operation in operation-set 0 is selected in the process plan, the remaining operations in the operation-set should be selected and the operations in the operation set 1 must not be selected. The final optimal feasible process plan must comply with this constraint.

To solve these problems, the representation of process plans in 5-axis machining need to be extended from that in 3-axis machining and set-up determination needs to be considered more flexibly.

1. Representation of process plans in 5-axis

The class definitions of particle dimension (operation) and particle (process plan consisting of all the operations) are extended as shown in table 6.8 and table 6.9. This added information enables the algorithm to comply with the operation set constraint and helps determine the set-up plans.

Table 6.8 Class definition of a particle dimension (an operation).

Class ParticleDimension: an operation		
Data types	Variables	Descriptions
Int	OperationSet_id	The id of the operation set
Int	Operation_id	The id of the operation in OperationSet_id
Int	Machine_id	The id of a machine to execute the operation
Int	Tool_id	The id of a cutting tool to execute the operation
Int	TAD_id	The id of a TAD to apply the operation
Int[ ]	Machine_list[ ]	The candidate machine list for executing the operation
Int[ ]	Tool_list[ ]	The candidate tool list for executing the operation
Int[ ]	TAD_list[ ]	The candidate TAD list for applying the operation
Int	Setup	The direction of set-up
double	Position	The position value of the operation
double	Velocity	The velocity value of the operation
double	Machine_time	The machining time for this operation

Table 6.9 Class definition of a particle (a process plan).

Class Particle: process plan		
Data types	Variables	Descriptions
ParticleDimension	Oper[n]	Define a process plan Oper[n] based on the above class-ParticleDimension. n is the number of operations in the plan
Int	Operationset	How many operation sets for each feature
Int	OperationsInSet	How many operations in the specific operation set
double	TMT	Total Machining Time of the plan
double	TMCT	Total Machine Change Time of the plan
double	TTCT	Total Tool Change Time of the plan
double	APT	Additional Penalty Time of violating constraints in the plan
double	TPT	Total Processing Time of the plan
ParticleDimension	Pi[n]	Store the best plan that the particle has achieved so far

2. Evaluation criteria

With the input from the operation selection module, the total processing time can be used to evaluate the process plan quantitively. The processing time for a part is comprised of machining times, machine change times, set-up times, tool change times and additional penalty time. The times can be computed as below.

Total Machining Time (TMT). TMC is the total machining time used for executing all the operations to machine the part, and it can be computed as:

$$TMT = \sum_{i=1}^{n}(Oper[i].Machine\_time) \qquad \text{(Eq 6. 17)}$$

Number of Set-up Changes (NSC), Number of Set-up (NS) and Total Set-up Time (TST). After a particle is decoded to a sequenced process plan, in the 5-axis machining environment, a set-up change between two consecutive operations in the sequence can be defined according to table 6.10 and the NSC can be computed as:

$$NSC = \sum_{i=1}^{n-1}\Omega_2(\Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id), \\ \Omega_3(Oper[i].TAD\_id, Oper[i+1].TAD\_id)) \qquad \text{(Eq 6. 18)}$$

The corresponding NS can be computed as:

$$NS = 1 + NSC \qquad \text{(Eq 6. 19)}$$

The Set-up Time (ST) is considered to be the same for each set-up. Hence,

$$TST = \sum_{i=1}^{NS} ST \qquad \text{(Eq 6. 20)}$$

where $\Omega_1(X,Y) = \{ \begin{matrix} 1 & X \neq Y \\ 0 & X = Y \end{matrix}$, $\Omega_2(X,Y) = \{ \begin{matrix} 0 & X = Y = 0 \\ 1 & otherwise \end{matrix}$,

$\Omega_3(X,Y) = \{ \begin{matrix} 1 & X = -Y \\ 0 & otherwise \end{matrix}$.

(4) Number of Machine Changes (NMC) and Total Machine Change Time ($TMCT$).

$$NMC = \sum_{i=1}^{n-1} \Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id) \qquad \text{(Eq 6. 21)}$$

The Machine Change Time (MCT) is considered to be the same for each machine change (in this research, only one machine is used). Hence,

$$TMCT = \sum_{i=1}^{NMC} MCT \qquad \text{(Eq 6. 22)}$$

(5) Number of Tool Changes (NTC) and Total Tool Change Time ($TTCT$). A tool change is defined in table 6.5. The NTC is computed as:

$$NTC = \sum_{i=1}^{n-1} \Omega_2(\Omega_1(Oper[i].Machine\_id, Oper[i+1].Machine\_id), \Omega_1(Oper[i].Tool\_id, Oper[i+1].Tool\_id)) \qquad \text{(Eq 6. 23)}$$

Similarly, the Tool Change Time (TCT) is considered to be the same for each tool change. Thus

$$TTCT = \sum_{i=1}^{NTC} TCT \qquad \text{(Eq 6. 24)}$$

(6) Number of Violating Constraints (NVC) and Additional Penalty Time (APT).

During the optimisation process, it is difficult to ensure that each particle obeys the constraints. To solve this problem, a penalty method has been used to adjust an infeasible particle towards its feasible domain.

$$NVC = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Omega_4 (Oper[i].Operation\_id, Oper[j].Operation\_id) \qquad \text{(Eq 6. 25)}$$

A fixed Penalty Time (PT) is applied to each violated constraint. Thus

$$APT = \sum_{i=2}^{NVC} PT \qquad \text{(Eq 6. 26)}$$

where

$$\Omega_4(X,Y) = \begin{cases} 1 & \text{The sequence of } X \text{ before } Y \text{ violates constraints} \\ 0 & \text{The sequence of } X \text{ before } Y \text{ is in accordance to constraints} \end{cases}$$

(7) The Total Processing Time (TPT).

$$TPT = TMT + TTCT + TST + APT \qquad \text{(Eq 6. 27)}$$

3. Determination of set-up plans

- The first set-up is important because if the first set-up has been decided, the following set-ups are the same as the first set-up until an operation can not be operated on this set-up. (Set-up change can be determined by checking if a machine is changed or the TAD of the next operation is opposite to the current set-up direction as shown in table 6.10). When the set-up has to be changed, then the next set-up is treated the same way as the first step.

- Because the set-up for the first operation can be determined as one of five set-up directions. It can be achieved by selecting the set-up randomly or deterministically. The following 4 methods can be used to determine different set-ups (shown in figure 6.4):

  ➢ Totally Randomly (TR): Randomly get set-up for every operation.

- First Randomly (FR): Randomly get set-up for first operation, and then if needed, change the set-up, the set-up is set as the next operation's TAD.
- Totally Set (TS): Set first operation's TAD as the set-up for the first operation, then if needed, change the set-up, the set-up is set as next operation's TAD.
- First Set (FS): Set first operation's TAD as the set-up for first operation, then if needed, change the set-up, the set-up is set randomly in one of 5 possible directions.

Table 6.10 Definition of when a setup change is required in 5-axis machining

Conditions of Machining Two Consecutive Operations	**A Setup Change**
Same TAD and same machine	No
Same TAD and different machines	Yes
TAD of second operation is NOT opposite to the current set-up and same machine	No
TAD of second operation is opposite to the current set-up and same machine	Yes
Different TADs and different machines	Yes

- In reality, there may be some directions that can not be used for the next set-up, so it is necessary to limit the possible setups. For example, if a component can only be setup in 3 directions (-3, 3, -1) then these will be ZO, OZ, XO.

These methods of set-up determination will be compared and illustrated through case studies in chapter 8.

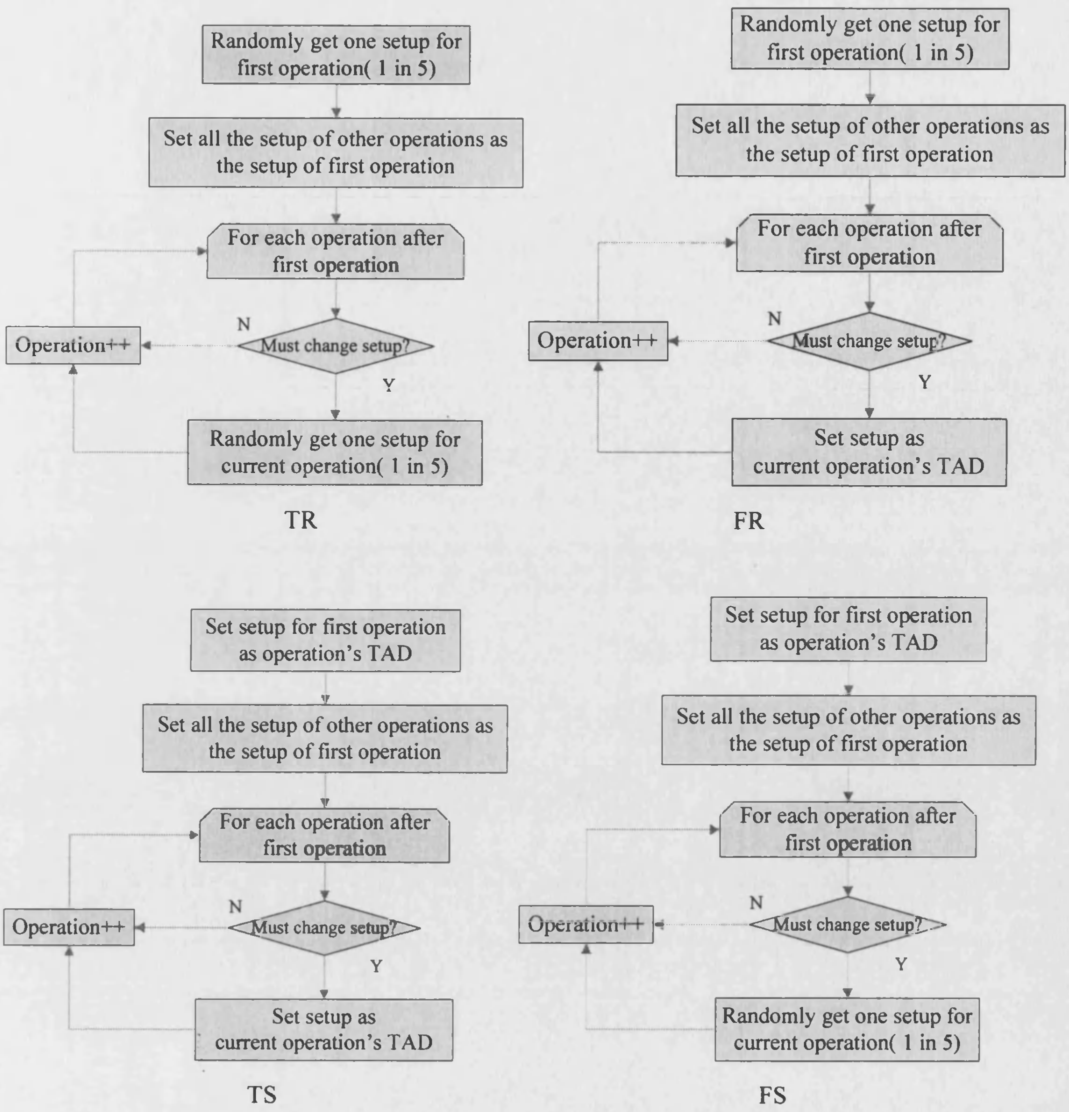


Figure 6.4 Work flow of set-up plan determination (TR, FR, TS, FS)

6.5 Summary

For a CAPP system in a dynamic workshop environment, the activities of selecting machining resources, determining set-up plans, and sequencing machining operations should be considered simultaneously so as to achieve the global lowest machining cost or lowest total processing time. Operation sequencing is one of the crucial tasks in process planning. However, it is an intractable process to identify an optimised operation sequence with minimal machining cost in a vast search space constrained by manufacturing conditions. In this chapter, the complicated operation sequencing process has been modelled as a combinatorial optimisation problem, and a modern evolutionary algorithm, i.e., the Particle Swarm Optimisation (PSO) algorithm, has been employed and modified to solve it effectively. Initial process plan solutions are formed and encoded into particles of the PSO algorithm. The particles "fly" intelligently in the search space to achieve the best sequence according to the optimisation strategies of the PSO algorithm. Meanwhile, to explore the search space comprehensively and to avoid being trapped into local optima, several new operators have been developed to improve the particles' movements to form a new modified PSO algorithm. The operation sequencing in 3-axis machining was first discussed and a evaluation criteria of machining cost applied. Then the differences between operation sequencing for 3-axis and that of 5-axis were given and the model and developed PSO approach extended to 5-axis machining. The determination of the PSO algorithm parameters and case studies for both 3-axis and 5-axis machining are described in Chapter 8.

Chapter 7 Integrated Process Planning and Scheduling (IPPS)

7.1 Introduction

The integration of process planning and scheduling is one of the most important functions to support flexible planning in a job shop manufacturing environment. Traditionally in the batch working industry, as described in chapter 1 and chapter 2, process planning and workshop scheduling are done separately and sequentially. Here the process plan is determined before the actual scheduling with no regard for the scheduling objectives. However, this simple sequential approach ignores the relationship between scheduling and process planning. The two functions are interrelated because both of them take part in the assignment of factory machines to production tasks (Moon and Seo 2005). If a process plan is prepared offline without due consideration of the actual shop floor status, it may become unfeasible due to changes or constraints in the manufacturing environment and heavily unbalanced resource assignments. Also due to the different objectives of these two systems, it is difficult to produce a satisfactory result in simple sequential executions of the two systems.

As discussed in chapter 2, the simultaneous approach has advantages over an enumerative approach. It is more effective and efficient to integrate the process planning and scheduling activities that are both in dynamic adjustment until specific performance criteria can be satisfied. Although a lot of effort has been made in this area, there are still several issues that need to be considered, such as performance criteria and objectives, constraints in the IPPS, algorithm efficiencies and replanning due to the dynamic change of the job shop floor (details in section 2.5 of chapter 2).

In this chapter, the problem of IPPS is first defined and a unified representation model developed to incorporate the two functions is described. Based on this model, a PSO-based approach has been developed to optimise the integration problem. Different performance criteria, such as makespan, total job tardiness and balanced level of machine utilisation have been defined in the optimisation approach to evaluate the performance of the schedule. The method to realise the replanning function is finally discussed.

7.2 Integrated Process Planning and Scheduling

7.2.1 Problem definition

Process planning and scheduling are both essential functional modules in product development and manufacturing. As presented in previous chapters, the major tasks in process planning include:

(1) Generating machining operations based on the features of a part to meet desired functional specifications and achieve good manufacturability,

(2) Identifying all the alternative applicable machining resources for the operations and calculating the machining times for all the alternative operations,

(3) Optimising the operation sequence to achieve the minimised manufacturing cost or manufacturing time, and

(4) Determining the set-up plan according to the optimised operation sequence and selected manufacturing resources.

Figure 6.1 (Representation of a process plan) shows an unordered process plan for a part which includes a series of machining operations, together with applicable manufacturing resources for the operations. Operations sequencing is used to determine a sequence by exploiting a sequence space derived from the combination of all the operations whilst obeying the precedence constraints among them. In this research, only the 3-axis machining is considered, so here a set-up is specified as a group of operations with the same Tool Approach Direction (TAD) (it means the same fixture) executed on the same machine.

It can be seen from chapter 6 that optimisation of the operation sequencing can generate the optimal process plan by selecting alternative manufacturing resources (machines, tools and TADs) and determining an optimised sequence to achieve the corresponding objectives. Figure 7.1 and figure 7.2 show the two kinds of flexibility in this procedure (Li and McMahon 2006): 1) processing flexibility refers to the possibility of performing an operation on alternative machines with alternative tools or TADs, 2) operation sequencing flexibility corresponds to the possibility of interchanging the sequence in which the operations are executed.

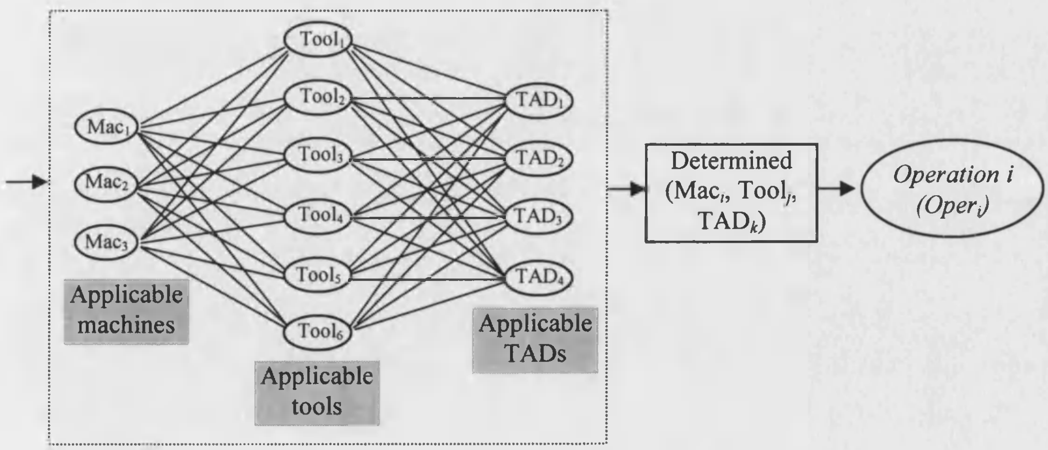


Figure 7.1 Example of processing flexibility (Li and McMahon 2006)

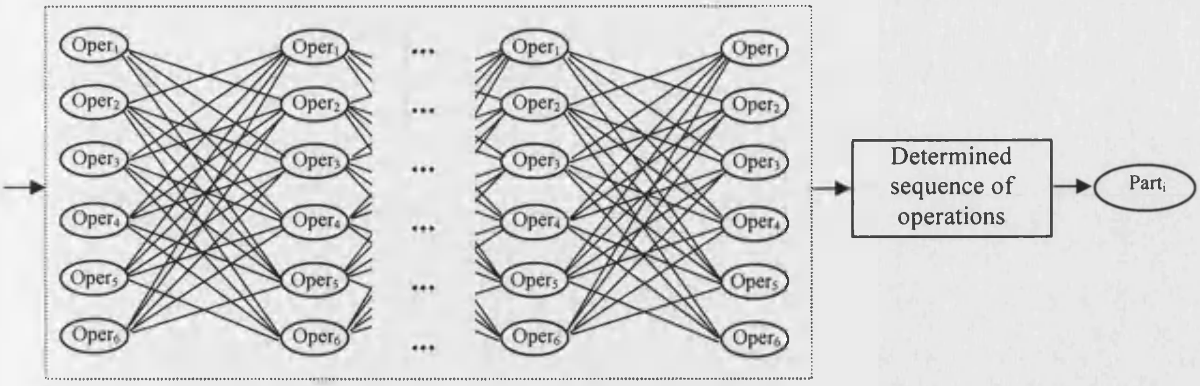


Figure 7.2 Example of operation sequencing flexibility (Li and McMahon 2006)

Based on the generated process plans of the parts, the scheduling task is to allocate the time for all the operations that are required to machine the parts to specific machines with the objectives of minimising makespan, balancing machine utilisation, minimising total tardiness, etc. Scheduling flexibility which is shown in figure 7.3 (Li et al. 2006) makes it possible to generate alternative schedules for the jobs by arranging the different sequences of parts to be machined.

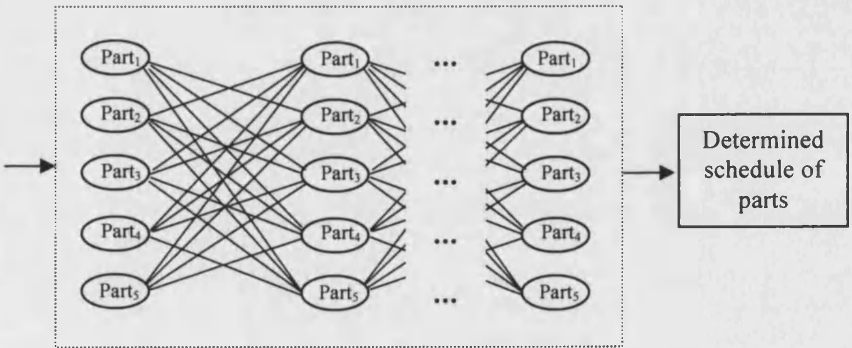


Figure 7.3 Example of scheduling flexibility (Li and McMahon 2006)

The objective of Integrated Process Planning and Scheduling (IPPS) in a job shop is to determine an optimal schedule with operation sequences for the jobs (Moon and Seo 2005). Therefore, the IPPS problem can be defined as: *given a set of n parts which are to be processed on m machines with operations including alternative manufacturing resources (machines, tools and TADs), select suitable manufacturing* *resources and sequence the operations so as to determine a schedule in which the precedence constraints among operations can be satisfied and the corresponding objectives can be achieved.*

Figure 7.4 is used to illustrate this problem. For instance, there are 3 parts that can be machined by 3, 2 and 3 operations on 3 machines respectively. For the different parts, there are precedence constraints among the operations to machine them (Part1: Oper1→Oper2→Oper3, Part2: Oper4→Oper5, Part3: Oper6→Oper7→Oper8). When all these 8 operations are sequenced (Oper1→Oper4→Oper2→Oper6→Oper3→Oper7→Oper8→Oper5 as shown in figure 7.4) and the manufacturing resources are specified (machine, tool and TADs), the schedule can be determined accordingly. The optimisation of IPPS is to optimise the operation sequence and selection of the manufacturing resources so as to achieve the optimal objectives (Makespan for instance in figure 7.4) whilst maintaining the schedule feasible with respect to the precedence constraints. It can be seen that the problem can be modelled as an extension of the operation sequencing optimisation problem (which relates to a single part described in chapter 6) into one of multi-parts with scheduling objectives. To achieve this, the representations of the process plans and schedule need to be extended and a related method of schedule determination based on the generated sequence and evaluation criteria needs to be considered.

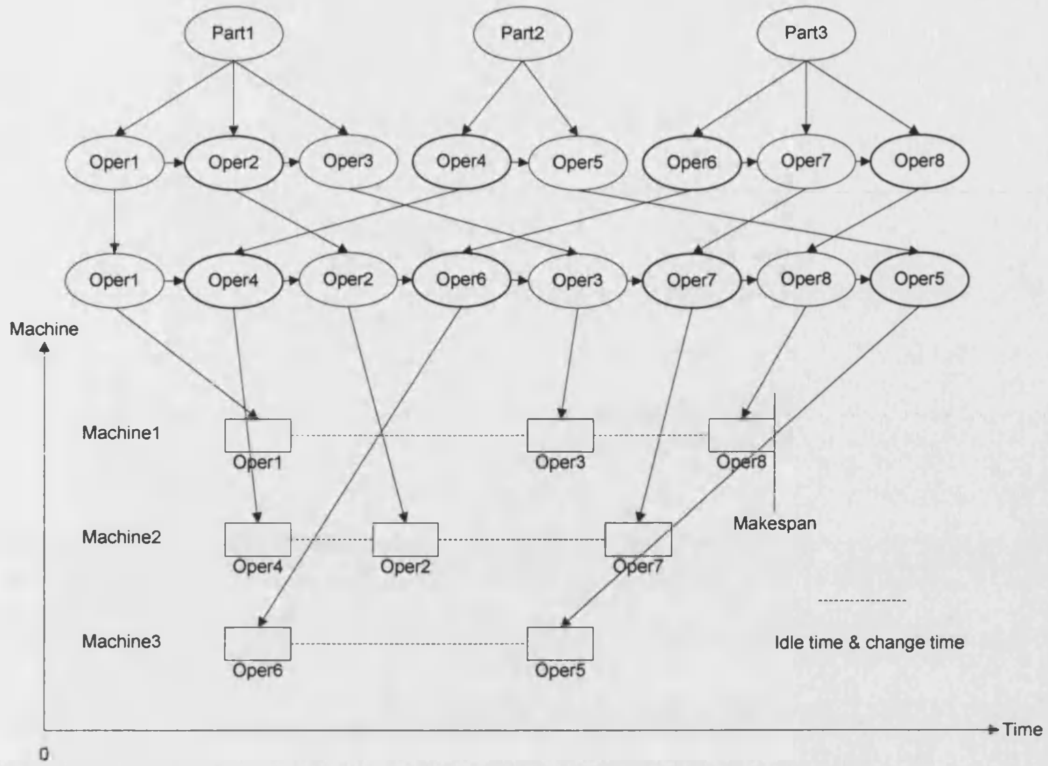


Figure 7.4 Illustration of IPPS problem

7.2.2 Representation of process plans and schedules

To apply the PSO algorithm to the optimisation of the IPPS problem, the representation scheme of process plans needs to be extended to contain more information for the consideration of scheduling functions. Table 7.1 and table 7.2 show the class definitions of *Operation* and *Process plan* respectively. Compared with table 6.1, several new variables including *Mac_time*, *Change_time*, *Machine_s_time* and *Machine_e_time* are added to record and track the time related to the execution of the operation so as to determine the time allocation on the machines.

Table 7.1 Class definition of a particle dimension (an operation).

Class ParticleDimension: an operation		
Data types	Variables	Descriptions
int	Operation_id	The id of the operation
Int	Part_id	The id of part to which the operation belongs
int	Machine_id	The id of a machine to execute the operation
int	Tool_id	The id of a cutting tool to execute the operation
int	TAD_id	The id of a TAD to apply the operation
int[ ]	Machine_list[ ]	The candidate machine list for executing the operation
int[ ]	Tool_list[ ]	The candidate tool list for executing the operation
int[ ]	TAD_list[ ]	The candidate TAD list for applying the operation
double	Mac_time	The machining time for this operation
double	Change_time	The change time required for this operation including tool change, set-up change and machine change
double	Machine_s_time	The start machining time of executing this operation
double	Machine_e_time	The end machining time of executing this operation
double	Position	The position value of the operation
double	Velocity	The velocity value of the operation

Table 7.2 Class definition of a particle (a process plan).

Class Particle: process plan		
Data types	Variables	Descriptions
ParticleDimension	Oper[n]	Define a process plan Oper[n] based on the above class-ParticleDimension. n is the number of operations in the plan
double	TC	Total Cost of the plan
double	APC	Additional Penalty Cost of violating constraints in the plan

To record the machine utilisation status and operations being executed on every machine at different time, a *machine* class is defined as shown in table 7.3. As discussed in section 7.2.1, when the sequence for all the operations is generated and the manufacturing resources are selected, the assignments of specific operations and machines are determined and therefore the schedule is obtained.

Table 7.3 Class definitions of a machine

Data types	Variables	Descriptions
int	Machine_id	The id of a machine
double	Available_time	The time when the machine is available (free) to execute operations
int	Current_oper_no	Record the current operation id
int	Num_operation	Record number of operations executed on this machine
int	Oper_no_list[]	Record all the executed operations on this machine
int	Oper_Part_no[]	Record all the corresponding part_id for Oper_no_list[]
double	Oper_s_time[]	Record the corresponding start time for Oper_no_list[]
double	Oper_p_time[]	Record the corresponding preparation time for Oper_no_list[]
double	Oper_e_time[]	Record the corresponding end time for Oper_no_list[]

7.2.3 Evaluation criteria

As described in section 7.2.1, the IPPS problem can be modeled as an extension of the operation sequencing optimisation problem relating to a single part into a multi-part with scheduling objectives. When the sequence of all the operations is generated and the manufacturing resources are specified, it is required to determine the schedule based on this information and calculate the makespan, total tardiness etc. to check if these meet the objective.

Assuming there are m machines available in a job shop and there are n operations required for machining p parts. *Operation i* is denoted as $Oper[i]$, *Machine j* is denoted as $Machine[j]$, *Part k* is denoted as $Part[k]$ and *Operation l* executed on *Machine j* is denoted as $Machine[j].Oper\_no\_list[l]$. There are five assumptions when determining the schedule:

1) Every machine is available at time 0 and as soon as the current operation is finished, the machine becomes available again.
2) All the operations have been sequenced as $(Oper[1], Oper[2], ... Oper[n])$.
3) The set-up time T_{set-up}, tool change time $T_{tool-change}$ and machine change time $T_{machine-change}$ are considered to be the same for each set-up, tool change and machine change respectively.
4) $T_{tool-change}$ is contained in T_{set-up} and T_{set-up} is contained in $T_{machine-change}$ ($T_{tool-change} < T_{set-up} < T_{machine-change}$). This means if more than one type of change occurs, only the bigger one is counted.

With these assumptions, the schedule can be determined by the following steps:

1) Initialisation:
   - Set the $Machine[j].Available\_time$=0, $j = 1,2,...m$.
   - Set the $Oper[i].Machine\_s\_time = 0$, $Oper[i].Machine\_e\_time = 0$, $Oper[i].Change\_time = 0$, $i = 1,2,...n$
2) Set the time and machine for the first operation $Oper[1]$:
   - Get the machine specified to execute the first operation $Machine[j] = Oper[1].Machine\_id$.
   - Get the part to which the first operation belongs $Part[k] = Oper[1].Part\_id$
   - Save the operation to the operation list $Machine[j].Oper\_no\_list[0] = 1$ and the corresponding part id $Machine[j].Oper\_part\_no[0] = Part[k]$.
   - For the first operation, it is required to set-up first, so the preparation time for the operation on this machine is $Machine[j].Oper\_p\_time[0] = T_{set-up}$.

- Get and store the start time for the operation on this machine $Machine[j].Oper\_s\_time[0] = Oper[1].Machine\_s\_time = 0 + Machine[j].Oper\_p\_time[0]$
- And end time for the operation on this machine $Machine[j].Oper\_e\_time[0] = Oper[1].Machine\_e\_time = Machine[j].Oper\_s\_time[0] + Oper[1].Mac\_time$.
- Then get the available time for this machine $Machine[j].Available\_time = Machine[j].Oper\_e\_time[0]$.

3) For $Oper[i]$, $i = 2,3,...n$, iterate the following steps

- Get the machine specified to execute the operation $Machine[j] = Oper[i].Machine\_id$.
- Get the part to which the operation belongs $Part[k] = Oper[i].Part\_id$
- Save the operation to the operation list $Machine[j].Oper\_no\_list[l] = Oper[i]$ and the corresponding part id $Machine[j].Oper\_part\_no[l] = Part[k]$.
- Check if it is required to change machine, set-up or tool. So the preparation time for the operation on this machine is $Machine[j].Oper\_p\_time[l] = Max(T_{machine\_change}, T_{set-up}, T_{tool-change})$, where $Max(T_{machine\_change}, T_{set-up}, T_{tool-change})$ means the longest time of three. The definition of a set-up change is shown in table 7.4.

Table 7.4 The definition of a set-up change on a specific machine.

Conditions of Machining Two Consecutive Operations	A Set-up Change
Same TAD and same part	No
Same TAD and different parts	Yes
Different TADs and same part	Yes
Different TADs and different parts	Yes

- Get and store the start time for the operation on this machine $Machine[j].Oper\_s\_time[l] = Oper[i].Machine\_s\_time = Latest(Machine[j].Available\_time, Part[k].Last\_oper\_time) + Machine[j].Oper\_p\_time[0]$, where $Part[k].Last\_oper\_time$ is the end time of the operation prior to the current operation for the $Part[k]$ which can be derived from $Machine[j].Oper\_no\_list[l]$ ($l = 1,2,...Machine[j].Num\_operation$, $j = 1,2,...m$). $Latest(Machine[j].Available\_time, Part[k].Last\_oper\_time)$ denotes the latest time of $Machine[j].Available\_time$ and $Part[k].Last\_oper\_time$.
- Get and store the end time for the operation on this machine $Machine[j].Oper\_e\_time[l] = Oper[i].Machine\_e\_time = Machine[j].Oper\_s\_time[l] + Oper[i].Mac\_time$.
- Then get the available time for this machine $Machine[j].Available\_time = Machine[j].Oper\_e\_time[l]$.

4) Calculate the Addition Penalty Time (APT) using the equations (25) and (26) in section 6.4.

When all the operations are processed in the above steps, the sub schedule for every machine is determined and the total schedule like figure 7.4 can be generated. With all the data and information obtained, three criteria of evaluating the schedules can be calculated as follows.

1) Makespan: $Makespan = \overset{m}{\underset{j=1}{Max}}(Machine[j].Available\_time)$.

2) Total job tardiness: The due date of a part is denoted as DD, and the completion moment of the part is denoted as CM. Hence,

$$Part\_Tardiness = \begin{cases} 0 & \text{if } DD \text{ is later than } CM \\ CM - DD & \text{Otherwise} \end{cases}$$

3) Balanced level of machine utilization: the Standard Deviation concept is introduced here to evaluate the balanced machine utilization (assuming there are m machines, and each machine has n operations).

$$Average\_Utilization = \frac{\sum_{i=1}^{n}(Operation[i].Mac\_T)}{n}, (j = 1,.., m)$$

$$\chi = \frac{\sum_{j=1}^{m}(Machine[j].Utilization)}{m}$$

$$Utilization\_Level = \sqrt{\sum_{j=1}^{m}(Machine[j].Utilization - \chi)^2}$$ (Li and McMahon 2006)

7.3 The PSO algorithm with replanning ability

The modified PSO algorithm developed in chapter 6 has been applied to the optimisation of the IPPS problem with only two changes:

1) The objectives have been changed from the minimized Total Machining Cost (*TMC*) and Total Processing Time (*TPT*) to the least Makespan and total job tardiness.

2) The schedule has been determined by the method discussed in section 7.2.3 based on the sequenced operations.

With consideration of the above two changes, the PSO algorithm can be used to optimise the IPPS problem before the jobs are processed. Current approaches do not consider that it is possible to make dynamic changes to the shop floor's situation, such as routine machine maintenance, machine breakdown and new orders insertion to the current schedule to meet the deadlines. Any occurrence of these situations will probably make the current schedule unfeasible and require the replaning of the whole schedule. In this research, two types of changes are considered, namely machine breakdown and new order arrivals. The following will discuss these two situations respectively.

1. Machine breaks down.

If a machine breaks down, it will not only affect the part being machined on it, but also make other operations that are supposed to be executed on this machine unfeasible. Suppose $Machine[j]$ breaks down at time T_b, and repairing the machine requires time T_p. The following assumptions are made:

- The replanning generates a schedule from the next available times for $Machine[j]$, $j = 1,2,...m$.
- The available time of the machine that breaks down $Machine[j].Available\_time = T_b + T_p$.
- The breaking down of $Machine[j]$ does not affect the current operations of other machines. If an operation $Oper[i]$ is being executed on

$Machine[k](k \neq j)$ when $Machine[j]$ breaks down, then the available time of the $Machine[k](k \neq j)$ can be computed as follows: $Machine[k].Available\_time = Oper[i].Machine\_e\_time$.

- If no job is being processed on $Machine[k](k \neq j)$ when $Machine[j]$ breaks down, then the available time of the $Machine[k](k \neq j)$ can be computed as: $Machine[k].Available\_time = T_b$.

- If there is a part being machined when the machine breaks down, it does not destroy the part and only the operation disturbed needs to be re-executed in one of two ways: a) to be machined on the current machine after it is repaired, and b) to be rescheduled to be executed on other machines.

- Only the operations that have not been executed and the operation being executed on the broken down machine need to be rescheduled from the machine available time obtained previously.

With the above assumptions, it can be seen in figure 7.5, when machine 2 breaks down at time T_b, the available times for three machines are T_1, T_2 and T_3 respectively.

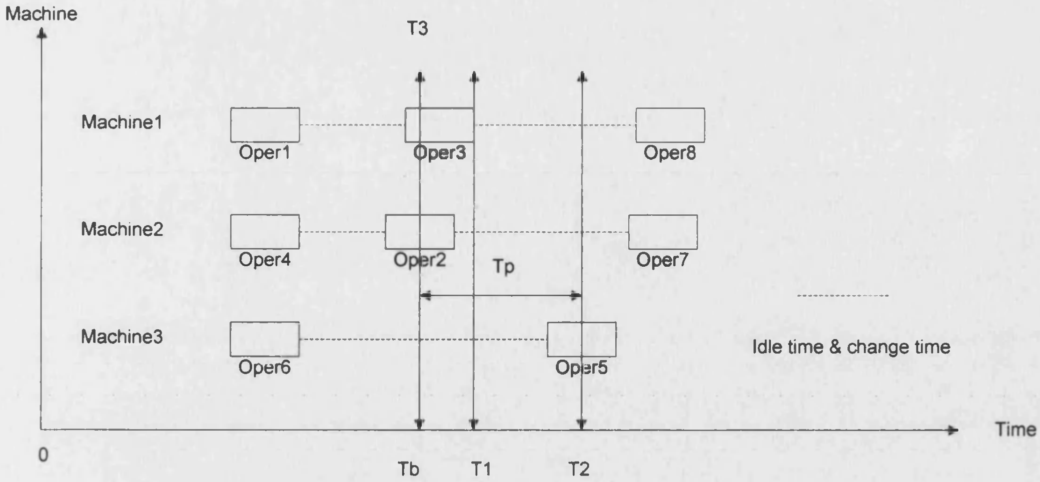


Figure 7.5 Determination of machines available times when machine2 breaks down

Therefore, with these assumptions, the replanning of the scheduling problem can be resolved by two changes applied to the PSO algorithm described in section 7.2:

1) Reduce the operations range to the operations that have not been executed.
2) Initialise the operations (particle dimensions) and machines with the new generated available time.

2. New order arrival.

Compared to machine breakdowns, the situation of the arrival of a new order is less complex. Suppose the new part arrives at time T_a. The following assumptions are made:

- The replanning generates a schedule from the next available times for $Machine[j]$, $j = 1,2,...m$.

- If an operation $Oper[i]$ is being executed on $Machine[j]$ when the new part arrives, then the available time of the $Machine[j]$ can be computed as follows: $Machine[j].Available\_time = Oper[i].Machine\_e\_time$.

- If no job is being processed on $Machine[j]$ when a new part arrives, then the available time of the $Machine[j]$ can be computed as: $Machine[j].Available\_time = T_a$.

- Only the operations that have not been executed and the new operations that are required to machine the new part need to be rescheduled from the machine available time obtained previously.

With the above assumptions, it can be seen in figure 7.6, when a new order arrives at time T_a, the available times for three machines are T_1, T_2 and T_3 respectively.

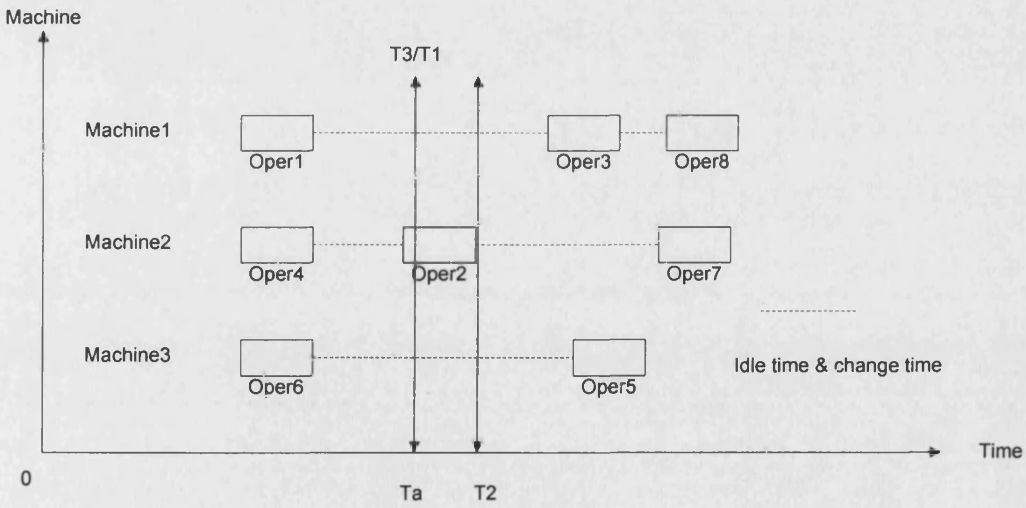


Figure 7.6 Determination of available times for machines when new order arrives at T_a

Therefore, with these assumptions, the replanning of the scheduling problem can be resolved by two changes applied to the PSO algorithm described in section 7.2:

1) Increase the operations range, including the operations of old parts that have not been executed and the operations that are required to machine the new part.
2) Initialise the operations (particle dimensions) and machines with the new generated available time.

Method to improve the efficiency of the algorithm in replanning:

It is required to reduce the computation time for generating a new schedule quickly when encountering the above situations. But the process of replanning will take more time, especially when adding new orders as these increase the search space and there is a need to keep the schedule feasible with consideration of more precedence constraints. As presented above, the critical step to replan the schedule is the initialisation of the particle (including all the operations need to be scheduled). Furthermore the old schedule generated before the situation occured was feasible and optimised whilst complying with all the precedence constraints. For efficiency, it is better to minimise changes to the existing plan as some allocated resources may well already be in place, e.g. tools and materials taken to machines in advance. Therefore the strategy has been to use the old schedule with some modifications as a new particle:

- For situations of machine breakdown, it is possible to initialise a particle by three steps: a) deleting the operations that have been executed in the old schedule, b) keeping the *velocity* and *position* values to keep the sequence among the operations, and c) changing the corresponding available time for the machines.

- For situations of new orders arriving, it is possible to initialise a particle by the following steps: a) deleting the operations that have been executed in the old schedule, b) keeping the *velocity* and *position* values to keep the sequence among the operations in the old schedule, c) adding the operations that are required to machine the new part to the end of old schedule, d) initializing the new added operations by selecting alternative manufacturing resources and set the *position* and *velocity* values, and d) changing the corresponding available time for the machines.

With this method, the optimised sequence in the old schedule is mostly kept and this saves a large amount of computation, and hence reduces the time for replanning the schedule. The case studies for solving the IPPS problem with the PSO algorithm will be given in chapter 8.

7.4 Summary

In this chapter, the problem of IPPS was firstly defined and modelled as an extension of the operation sequencing optimisation problem (which relates to a single part described in chapter 6) into a multi-part with scheduling objectives. The method to determine the schedule from the generated operation sequence was presented and different performance criteria, such as makespan, total job tardiness and balanced level of machine utilisation were defined in the optimisation approach to evaluate the performance of the schedule. Based on this model, a PSO-based approach was developed to determine the optimised results from the complex search space effectively and efficiently. Finally the situations of machines breaking down and new order arrival were discussed and methods to replan the schedule under these circumstances presented.

Chapter 8 System implementation and case studies

8.1 Introduction

The proposed system, its information flow and the functional modules have been discussed in chapter 4. The feature information input module and approaches of operation selection for 5 specific features have been proposed in chapter 5. To achieve the objectives of minimised machining cost or machining time, the optimisation of operation sequencing problems with a PSO algorithm for 3-axis machining and 5-axis machining environments were developed in chapter 6. The problem of Integrated Process Planning and Scheduling (IPPS) was defined and the optimisation of it with a PSO algorithm was discussed in chapter7.

In this chapter, implementation of the above functions will be described and case studies will be used to verify these functions. Firstly the implementation and case studies of operation sequencing for 3-axis and IPPS will be given respectively as independent modules and then finally the whole 5-axis CAPP system implementation and case study is illustrated.

8.2 Implementation and case studies for operation sequencing in 3-axis machining environment

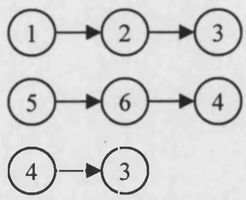
8.2.1 Hardware and software requirement

In this research, the CAPP prototype system and independent modules have been implemented on a Pentium IV PC with 1 Gb DDR Memory and Windows XP operating system unless stated specifically.

The operation sequencing module for 3-axis machining carries out the task of optimising the operation sequence to generate the optimal or near optimal process plans. The process plan representation and the PSO algorithm discussed in chapter 6 have been developed using Java 1.5 SDK. Three sample parts are used to verify and compare the efficiency of the algorithm. The operations and applicable manufacturing resources form the alternative operations and are placed in a file "operation.dat". The corresponding machine cost and tool cost are saved in a file "cost.dat". There are two ways to represent the precedence constraints between all the operations in a part. One is a Precedence Relationships Matrix (PRM) (Zhang et al. 1997, Li et al. 2002) and another is a Precedence-directed Graph (PG) (Lee et al. 2001) as figure 8.1 shows. The PG is easier to read but the PRM is easier to utilise in programs, so in this module, PRM is adopted to represent the Precedence constraints and all the precedence constraints in a part are saved in a file "sequence.dat". As figure 8.1 shows, (op1, op2) equalling 1 means op1 must be executed before op2, otherwise a violation of precedence constraint occurs and the operation sequence generated is not feasible.

	op1	op2	op3	op4	op5	op6
op1		1	1			
op2			1			
op3						
op4			1			
op5				1		1
op6				1		

(a) PRM



(b) PG

Figure 8.1 Two representations of Precedence Relationships

8.2.2 Sample parts

Three parts are used here as examples. The first part (Part 1 shown in figure 8.2) consists of 11 manufacturing features. These features can be machined with 14 operations (n = 14). The second part (Part 2 shown in figure 8.3) used by Zhang et al. (1997) consists of 14 manufacturing features and 14 operations (n = 14). The third part (Part 3 shown in figure 8.4) used by Shah et al. (1995) and Li et al. (2004) with more complex features and constraints consists of 14 manufacturing features and 20 machining operations (n = 20). The relevant information of machining resources, features, operations, and precedence constraints for each part are given respectively in tables 8.1-8.3 (Part 1), tables 8.4-8.6 (Part 2) and tables 8.7-8.9 (Part 3).

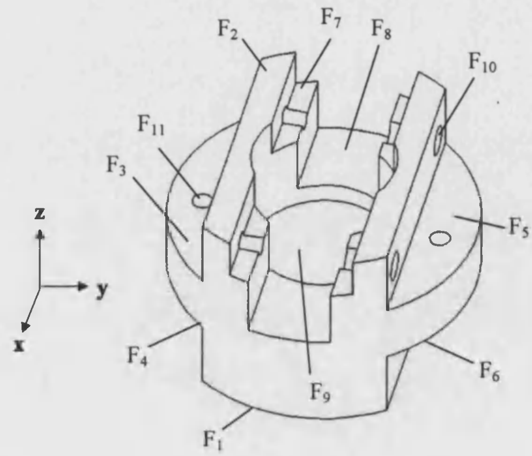


Figure 8.2 A sample part with 11 features – Part 1.

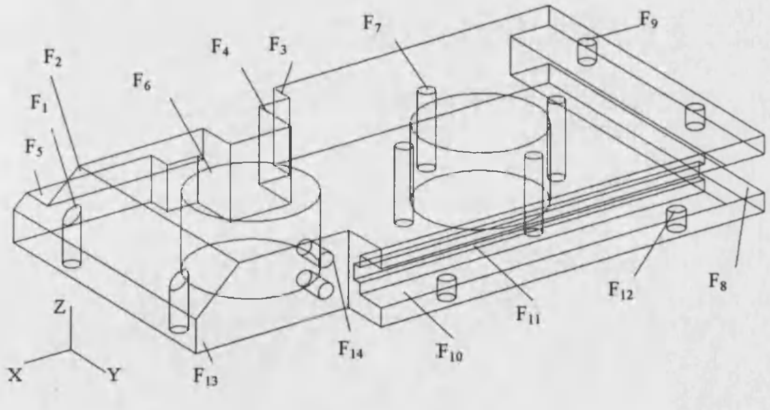


Figure 8.3 A sample part with 14 features – Part 2.

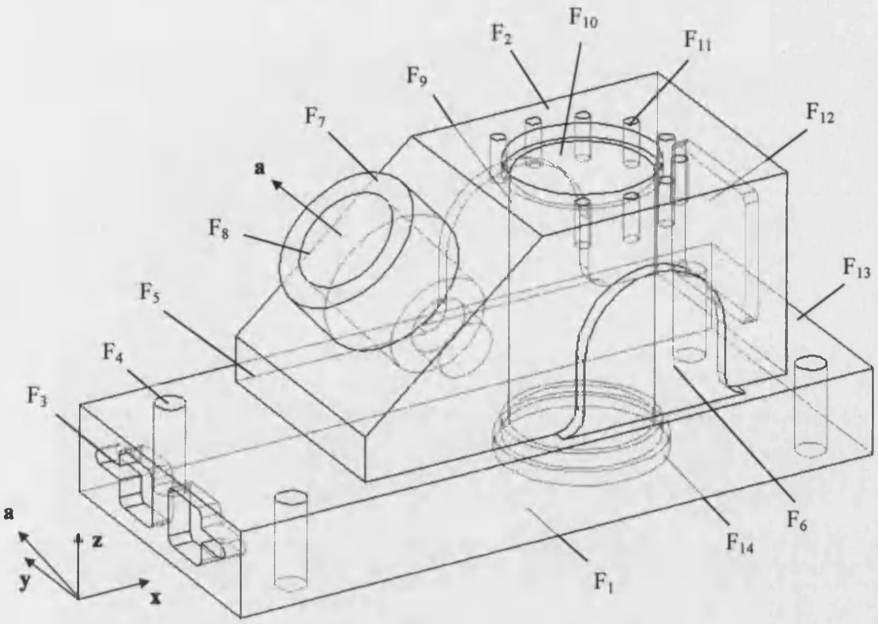


Figure 8.4 ANC 101 sample part with 14 features – Part 3.

Table 8.1 Available machining resources and costs in a workshop environment for Part 1.

Machines

No.	Types	*MCI*
M_1	Drill press	10
M_2	3-axis vertical milling machine I	40
M_3	3-axis vertical milling machine II	40
M_4	CNC 3-axis vertical milling machine	100
M_5	Boring machine	60

Tools

No.	***Types***	***TCI***
C_1	Drill 1	7
C_2	Drill 2	5
C_3	Drill 3	3
C_4	Drill 4	8
C_5	Tapping tool	7
C_6	Milling cutter 1	10
C_7	Milling cutter 2	15
C_8	Milling cutter 3	30
C_9	Reamer	15
C_{10}	Boring tool	20
C_{11}	Slot cutter	15
C_{12}	Chamfer tool	15

MCC = 160, SC = 120, TCC = 20, PC = 650

Table 8.2 The features, operations and candidate machining information for Part 1.

Features	Feature Descriptions	Operations (Oper_id)	TAD Candidates	Machine Candidates	Tool Candidates
F_1	A planar surface	Milling ($Oper_1$)	+z	M_2, M_3, M_4	C_6, C_7, C_8
F_2	A planar surface	Milling ($Oper_2$)	-z	M_2, M_3, M_4	C_6, C_7, C_8
F_3	A step	Milling ($Oper_3$)	+x, -x, +y, -z	M_2, M_3, M_4	C_6, C_7, C_8
F_4	A step	Milling ($Oper_4$)	+x, -x, +y, +z	M_1, M_2, M_3, M_4	C_2
F_5	A step	Milling ($Oper_5$)	+x, -x, -y, -z	M_2, M_3, M_4	C_6, C_7, C_8
F_6	A step	Milling ($Oper_6$)	+x, -x, -y, +z	M_2, M_3, M_4	C_7, C_8
F_7	A slot	Milling ($Oper_7$)	+x, -x, -z	M_2, M_3, M_4	C_7, C_8, C_{11}
F_8	A slot	Milling ($Oper_8$)	+x, -x, -z	M_2, M_3, M_4	C_6, C_7, C_8, C_{11}
F_9	A hole	Drilling ($Oper_9$) Reaming ($Oper_{10}$) Boring ($Oper_{11}$)	+z, -z	M_1, M_2, M_3, M_4 M_2, M_3, M_4 M_2, M_3, M_4, M_5	C_2, C_3, C_4 C_9 C_{10}
F_{10}	Four holes arranged in a replicated feature	Drilling ($Oper_{12}$) Tapping ($Oper_{13}$)	+y, -y	M_1, M_2, M_3, M_4 M_2, M_3, M_4	C_1 C_5
F_{11}	Two holes arranged in a replicated feature	Drilling ($Oper_{14}$)	+z, -z	M_1, M_2, M_3, M_4	C_9

Table 8.3 The precedence constraints for Part 1.

Constraints	Descriptions
Datum interactions	$Oper_1$ is the first operation. $Oper_2$ should be prior to $Oper_3$-$Oper_{14}$. $Oper_9$, $Oper_{10}$ and $Oper_{11}$ should be prior to $Oper_7$ and $Oper_8$.
Material removal interactions	$Oper_3$-$Oper_6$ should be prior to $Oper_{12}$-$Oper_{14}$.
Fixed order	$Oper_9$-$Oper_{10}$-$Oper_{11}$. $Oper_{12}$ should be prior to $Oper_{13}$.

Table 8.4 Available machining resources and costs in a workshop environment for Part 2.

Machines

No.	Types	*MCI*
M_1	Drill press	10
M_2	Milling machine	35
M_3	Three-axis vertical milling machine	60

Tools

No.	*Types*	*TCI*
C_1	Drill 1	3
C_2	Drill 2	3
C_3	Reamer	8
C_4	Boring tool	15
C_5	Milling cutter 1	10
C_6	Milling cutter 2	15
C_7	Slot cutter	10
C_8	Chamfer tool	10

$MCC = 160$, $SC = 120$, $TCC = 20$, $PC = 650$

Table 8.5 The features, operations and candidate machining information for Part 2.

Features	Feature Descriptions	Operations (Oper_id)	TAD Candidates	Machine Candidates	Tool Candidates
F_1	Two holes as a replicated feature	Drilling ($Oper_1$)	+z, -z	M_1, M_2, M_3	C_1
F_2	A chamfer	Milling ($Oper_2$)	-x, +y, -y, -z	M_2, M_3	C_8
F_3	A slot	Milling ($Oper_3$)	+y	M_2, M_3	C_5, C_6
F_4	A slot	Milling ($Oper_4$)	+y	M_2	C_5, C_6
F_5	A step	Milling ($Oper_5$)	+y, -z	M_2, M_3	C_5, C_6
F_6	Two holes as a replicated feature	Drilling ($Oper_6$)	+z, -z	M_1, M_2, M_3	C_2
F_7	Four holes as a replicated feature	Drilling ($Oper_7$)	+z, -z	M_1, M_2, M_3	C_1
F_8	A slot	Milling ($Oper_8$)	+x	M_2, M_3	C_5, C_6
F_9	Two holes as a replicated feature	Drilling ($Oper_9$)	-z	M_1, M_2, M_3	C_1
F_{10}	A slot	Milling ($Oper_{10}$)	-y	M_2, M_3	C_5, C_6
F_{11}	A slot	Milling ($Oper_{11}$)	-y	M_2, M_3	C_5, C_7
F_{12}	Two holes as a replicate feature	Drilling ($Oper_{12}$)	+z, -z	M_1, M_2, M_3	C_1
F_{13}	A step	Milling ($Oper_{13}$)	-x, -y	M_2, M_3	C_5, C_6
F_{14}	Two holes as replicate feature	Drilling ($Oper_{14}$)	-y	M_1, M_2, M_3	C_1

Table 8.6 The precedence constraints for Part 2.

Constraints	Descriptions
Tool interactions	$Oper_1$ should be prior to $Oper_2$
Datum interactions	$Oper_6$ should be prior to $Oper_7$. $Oper_{10}$ should be prior to $Oper_{11}$. $Oper_{13}$ should be prior to $Oper_{14}$.
Thin-wall interactions	$Oper_9$ should be prior to $Oper_8$. $Oper_{12}$ should be prior to $Oper_{10}$.
Material removal interactions	$Oper_8$ should be prior to $Oper_9$. $Oper_{10}$ should be prior to $Oper_{12}$. $Oper_{13}$ should be prior to $Oper_{14}$. $Oper_3$ should be prior to $Oper_4$.

Table 8.7 The information of available machines and cutting tools for Part 3.

Machines

No.	Types	*MCI*
M_1	Drilling press	10
M_2	3-axis vertical milling machine	40
M_3	CNC 3-axis vertical milling machine	100
M_4	Boring machine	60

Cutting Tools

No.	Types	*TCI*
C_1	Drill 1	7
C_2	Drill 2	5
C_3	Drill 3	3
C_4	Drill 4	8
C_5	Tapping tool	7
C_6	Mill 1	10
C_7	Mill 2	15
C_8	Mill 3	30
C_9	Ream	15
C_{10}	Boring tool	20

$MCC = 160$, $SC = 120$, $TCC = 20$, $PC = 650$

Table 8.8 The features and operations information of Part 3.

Features	Feature Descriptions	Operations (Oper_id)	TAD Candidates	Machine Candidates	Tool Candidates
F_1	A planar surface	Milling ($Oper_1$)	+z	M_2, M_3	C_6, C_7, C_8
F_2	A planar surface	Milling ($Oper_2$)	-z	M_2, M_3	C_6, C_7, C_8
F_3	Two pockets arranged as a replicated feature	Milling ($Oper_3$)	+x	M_2, M_3	C_6, C_7, C_8
F_4	Four holes arranged as a replicated feature	Drilling ($Oper_4$)	+z, -z	M_1, M_2, M_3	C_2
F_5	A step	Milling ($Oper_5$)	+x, -z	M_2, M_3	C_6, C_7
F_6	A protrusion (rib)	Milling ($Oper_6$)	+y, -z	M_2, M_3	C_7, C_8
F_7	A boss	Milling ($Oper_7$)	-a	M_2, M_3	C_7, C_8
F_8	A compound hole	Drilling ($Oper_8$) Reaming ($Oper_9$) Boring ($Oper_{10}$)	-a	M_1, M_2, M_3 M_1, M_2, M_3 M_3, M_4	C_2, C_3, C_4 C_9 C_{10}
F_9	A protrusion (rib)	Milling ($Oper_{11}$)	-y, -z	M_2, M_3	C_7, C_8
F_{10}	A compound hole	Drilling ($Oper_{12}$) Reaming ($Oper_{13}$) Boring ($Oper_{14}$)	-z	M_1, M_2, M_3 M_1, M_2, M_3 M_3, M_4	C_2, C_3, C_4 C_9 C_{10}
F_{11}	Nine holes arranged in a replicated feature	Drilling ($Oper_{15}$) Tapping ($Oper_{16}$)	-z	M_1, M_2, M_3 M_1, M_2, M_3	C_1 C_5
F_{12}	A pocket	Milling ($Oper_{17}$)	-x	M_2, M_3	C_7, C_8
F_{13}	A step	Milling ($Oper_{18}$)	-x, -z	M_2, M_3	C_6, C_7
F_{14}	A compound hole	Reaming ($Oper_{19}$) Boring ($Oper_{20}$)	+z	M_1, M_2, M_3 M_3, M_4	C_9 C_{10}

Table 8.9 Precedence constraints between machining operations for Part 3.

Constraints	Descriptions
Datum interactions	$Oper_1$ is the first operation. $Oper_5$ is prior to $Oper_4$ and $Oper_7$. $Oper_6$ is prior to $Oper_{12}$, $Oper_{13}$ and $Oper_{14}$. $Oper_7$ is prior to $Oper_8$, $Oper_9$ and $Oper_{10}$. $Oper_{11}$ is prior to $Oper_{12}$, $Oper_{13}$ and $Oper_{14}$. $Oper_{12}$, $Oper_{13}$ and $Oper_{14}$ are prior to $Oper_{15}$, $Oper_{16}$, $Oper_{19}$ and $Oper_{20}$.
Material removal interactions	$Oper_2$ is prior to $Oper_{12}$, $Oper_{13}$, $Oper_{14}$, $Oper_{15}$ and $Oper_{16}$. $Oper_{18}$ is prior to $Oper_4$ and $Oper_{17}$.
Fixed order	$Oper_8$ is prior to $Oper_9$ and $Oper_{10}$, and $Oper_9$ is prior to $Oper_{10}$. $Oper_{12}$ is prior to $Oper_{13}$ and $Oper_{14}$, $Oper_{13}$ is prior to $Oper_{14}$. $Oper_{15}$ is prior to $Oper_{16}$. $Oper_{19}$ is prior to $Oper_{20}$.

8.2.3 Determination of parameters

The main parameters of the PSO algorithm can be classified into three categories:

- Swarm characteristics: when the complexity of the part, features, operations and precedence constraints increases, the number of particles in the swarm (*Swarm_Size*) and Iteration number (*Iter_Num*) are set increasingly to enable the swarm to explore the search space further. After many trials, *Swarm_Size* has been set as 5000 and *Iter_Num* as 200 which are large enough to get favourable results for all of these three parts.
- Adjustment of global search and local search: c_1 and c_2 in formulas (Eq6.1) are both set as 2.0 to balance the velocity tendency to local best (P_i) and global best (P_g) (Clerc 1999). Two strategies to adjust the local exploration and global exploration have been tested: A) the inertia weight of particles, w is set to descend incrementally to facilitate the local exploration of the space at the beginning stage and a global exploration at the later stage; B) w is set to ascend incrementally to facilitate the global exploration of the space at the beginning stage and a local exploration at the later stage. Figure 8.5 shows the comparisons of these two strategies. Through trials on different parts, strategy A is considered to be better than strategy B. When w is set to be from 1.4 to 0.6, the algorithm can get favourable results.
- Probabilities that help the swarm escape from local optima: many trials have been done to test the optimised values of three probabilities of mutation, crossover and shift operators. The following values (P_m = 0.65, P_c = 0.2, and P_s = 0.3) have been shown to yield good performance of the PSO algorithm.

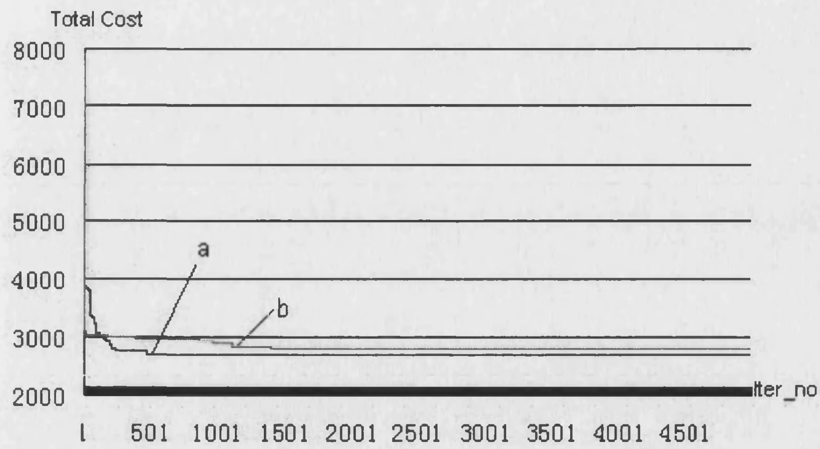


Figure 8.5 Comparison of two strategies to adjust global exploration and local exploration based on Part 3
(a: w is set to descend incrementally; b: w is set to ascend incrementally).

8.2.4 Computation results and comparison with other algorithms

Experiments have been done for all the three parts and table 8.10 shows a final plan for Part 3. The sequence of operations is determined by relative positions of operations and the total cost is 2535.0. Figure 8.6 shows the optimised fitness value of P_g for Part 3. It shows clearly that the algorithm converges very well and the adopted crossover operator improves the performance of the traditional PSO significantly.

Computation experience is usually used to verify and compare the efficiency of algorithms. The GA and SA algorithms developed by Li et al. (2002, 2004) have been used to compare their performance with this developed PSO algorithm. The experiments are based on 5000 iterations for each algorithm. The population of the GA is the same as that of the PSO algorithm. As shown in figure 8.7, at the initial optimisation stage, the GA optimises faster than the SA and the PSO (this is shown by a more rapid fall in figure 8.7). However, at the middle and late stages, the GA converges while the SA and the PSO continue to decline to give better results. From table 8.11, it can be observed that the SA and PSO algorithms outperform the GA in all the experiments of all three parts and both the SA and PSO can achieve results that are nearer the optimum. Comparing the characteristics of the SA and PSO, the PSO is more robust as the parameters of the SA have been found to be more sensitive to optimisation problems and more difficult to control (Li et al. 2002, 2004). Comparing the work flows of the GA (Li et al. 2002) and PSO (in chapter 6), it can be seen that the PSO needs to adjust the particle dimensions' by updating the velocities and positions of them due to its intrinsic mechanism so that it needs more computation than the GA. As population based algorithms, the PSO and the GA take more time to complete the 5000 iterations than the SA (90, 50 and 30 seconds respectively for part 3).

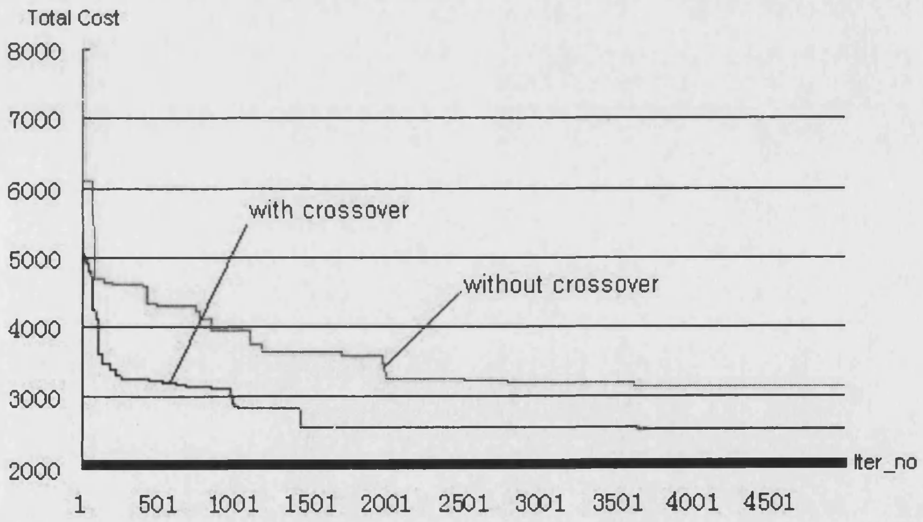


Figure 8.6 PSO with and without crossover (Part 3).

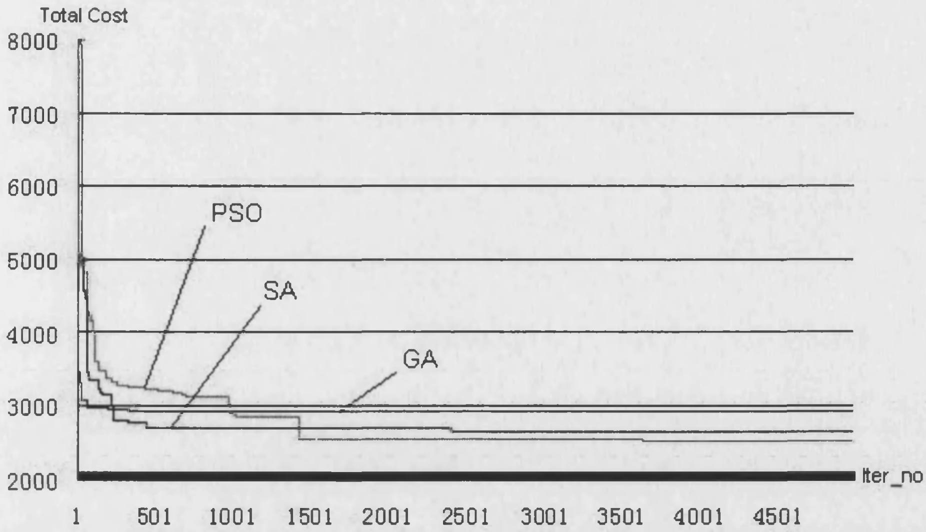


Figure 8.7 Comparisons of PSO, GA and SA for Part 3.

Table 8.10 A sample plan for Part 3 (total cost *Tc* = 2535.0).

Operation No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Machine	2	2	2	1	2	2	2	2	2	4	2	2	2	4	1	1	2	2	2	4
Tool	7	7	7	2	7	7	7	3	9	10	7	3	9	10	1	5	7	7	9	10
TAD	+z	-z	+x	-z	+x	-z	-a	-a	-a	-a	-z	-z	-z	-z	-z	-z	-x	-x	+z	+z
Relative Position (Sequence no.)	1	9	2	20	3	11	4	5	6	15	10	12	13	16	18	19	8	7	14	17

Table 8.11 The comparisons of GA, SA and PSO.

	Part 1		Part 2		Part 3	
Algorithm	Best cost achieved	Mean cost of 10 trials	Best cost achieved	Mean cost of 10 trials	Best cost achieved	Mean cost of 10 trials
GA	1381.0	1459.4	1228.0	1340.0	2667.0	2796.0
SA	1421.0	1447.4	1088.0	1122.0	2535.0	2668.5
PSO	1361.0	1430.0	1068.0	1103.0	2535.0	2680.5

8.3 Case studies for IPPS module

Two experiments are used here to verify the efficiency of the PSO algorithm for the IPPS problem. The first experiment is used to compare the efficiencies of the PSO, GA and SA algorithms. The second experiment is used to verify the replanning ability of the PSO algorithm under machine breakdown and new order arrival conditions. For simplification, the parameters determined in section 8.2 are used in the PSO algorithm for experiments in this section.

8.3.1 Experiment 1

The example parts and manufacturing resources from Li and McMahon (2006) are used here to verify and compare the efficiencies of the PSO, GA and SA approaches. The resources of a specific job shop are defined in Table 8.12.

Table 8.12 The resource of a job shop – machines and tools (Li et al. 2006).

Machines

Types	**No.**	**Cost ($)**
Drilling press	M_1	10
3-axis vertical milling machine I	M_2	40
3-axis vertical milling machine II	M_3	40
CNC 3-axis vertical milling machine	M_4	100
Boring machine	M_5	60

Cutting Tools

Types	**No.**	**Cost ($)**
Drill 1	C_1	7
Drill 2	C_2	5
Drill 3	C_3	3
Drill 4	C_4	8
Tapping tool	C_5	7
Mill 1	C_6	10
Mill 2	C_7	15
Mill 3	C_8	30
Reaming tool	C_9	15
Boring tool	C_{10}	20
Slot cutter	C_{11}	15
Chamfer tool	C_{12}	15

Setup_Index = 120.0 (s), MC_Index = 140.0 (s), TC_Index = 20.0 (s)

Two groups of parts are used for the experiment.

Group 1:

The first group consists of three parts, which are as the same as the parts used in section 8.2. These are taken from the works of Shah and Mantyla (1995) and Zhang et al. (1997). The relevant technical specifications of the three parts are defined in Tables 8.13, 8.14 and 8.15 (Here it is different to the information in section 8.2 due to the change of manufacturing resources).

Table 8.13 The technical specifications for Part 1.

Features	*Operations*	*TAD Candidates*	*Machine Candidates*	*Tool Candidates*	*Machining Time for Each Candidate Machine (seconds)*
F_1	Milling ($Oper_1$)	+z	M_2, M_3, M_4	C_6, C_7, C_8	40, 40, 30
F_2	Milling ($Oper_2$)	-z	M_2, M_3, M_4	C_6, C_7, C_8	40, 40, 30
F_3	Milling ($Oper_3$)	+x	M_2, M_3, M_4	C_6, C_7, C_8	20, 20, 15
F_4	Drilling ($Oper_4$)	+z, -z	M_1, M_2, M_3, M_4	C_2	12, 10, 10, 7.5
F_5	Milling ($Oper_5$)	+x, -z	M_2, M_3, M_4	C_6, C_7	35, 35, 26.25
F_6	Milling ($Oper_6$)	+y, -z	M_2, M_3, M_4	C_7, C_8	15, 15, 11.25
F_7	Milling ($Oper_7$)	-a	M_2, M_3, M_4	C_7, C_8	30, 30, 22.5
F_8	Drilling ($Oper_8$) Reaming ($Oper_9$) Boring ($Oper_{10}$)	-a	M_1, M_2, M_3, M_4 M_2, M_3, M_4 M_2, M_3, M_4, M_5	C_2, C_3, C_4 C_9 C_{10}	21.6, 18, 18, 13.5 10, 10, 7.5 10, 10, 7.5, 12
F_9	Milling ($Oper_{11}$)	-y, -z	M_2, M_3, M_4	C_7, C_8	15, 15, 11.25
F_{10}	Drilling ($Oper_{12}$) Reaming ($Oper_{13}$) Boring ($Oper_{14}$)	-z	M_1, M_2, M_3, M_4 M_2, M_3, M_4 M_2, M_3, M_4, M_5	C_2, C_3, C_4 C_9 C_{10}	48, 40, 40, 30 25, 25, 18.75 25, 25, 18.75, 30
F_{11}	Drilling ($Oper_{15}$) Tapping ($Oper_{16}$)	-z	M_1, M_2, M_3, M_4 M_2, M_3, M_4	C_1 C_5	26.4, 22, 22, 16.5 20, 20, 15
F_{12}	Milling ($Oper_{17}$)	-x	M_2, M_3, M_4	C_7, C_8	16, 16, 12
F_{13}	Milling ($Oper_{18}$)	-x, -z	M_2, M_3, M_4	C_6, C_7	35, 35, 26.25
F_{14}	Reaming ($Oper_{19}$) Boring ($Oper_{20}$)	+z	M_2, M_3, M_4 M_2, M_3, M_4, M_5	C_9 C_{10}	12, 12, 9 12, 12, 9, 14.4

Table 8.14 The technical specifications for Part 2.

Features	***Operations***	***TAD Candidates***	***Machine Candidates***	***Tool Candidates***	***Machining Time for Each Candidate Machine (seconds)***
F_1	Drilling ($Oper_1$)	+z, -z	M_1, M_2, M_3, M_4	C_1	12, 10, 10, 7.5
F_2	Milling ($Oper_2$)	-x, +y, -y, -z	M_2, M_3, M_4	C_{12}	20, 20, 15
F_3	Milling ($Oper_3$)	+y	M_2, M_3, M_4	C_5, C_6, C_{11}	18, 18, 13.5
F_4	Milling ($Oper_4$)	+y	M_2, M_3, M_4	C_6, C_7, C_8	16, 16, 12
F_5	Milling ($Oper_5$)	+y, -z	M_2, M_3, M_4	C_6, C_7, C_8	15, 15, 11.25
F_6	Drilling ($Oper_6$)	+z, -z	M_1, M_2, M_3, M_4	C_2	30, 25, 25, 18.75
	Reaming ($Oper_7$)	+z, -z	M_2, M_3, M_4	C_9	25, 25, 18.75
F_7	Drilling ($Oper_8$)	+z, -z	M_1, M_2, M_3, M_4	C_1	14.4, 12, 12, 9
F_8	Milling ($Oper_9$)	+x	M_2, M_3, M_4	C_6, C_7, C_8	15, 15, 11.25
F_9	Drilling ($Oper_{10}$)	-z	M_1, M_2, M_3, M_4	C_1	9.6, 8, 8, 6
F_{10}	Milling ($Oper_{11}$)	-y	M_2, M_3, M_4	C_6, C_7, C_8	10, 10, 7.5
F_{11}	Milling ($Oper_{12}$)	-y	M_2, M_3, M_4	C_6, C_7, C_8	10, 10, 7.5
F_{12}	Drilling ($Oper_{13}$)	+z, -z	M_1, M_2, M_3, M_4	C_1	9.6, 8, 8, 6
F_{13}	Milling ($Oper_{14}$)	-x, -y	M_2, M_3, M_4	C_6, C_7, C_8	16, 16, 12
F_{14}	Drilling ($Oper_{15}$)	-y	M_1, M_2, M_3, M_4	C_1	9.6, 8, 8, 6
F_{15}	Milling($Oper_{16}$)	+x, -x, +y, -y, +z	M_1, M_2, M_3, M_4	C_6, C_7, C_8	36, 30, 30, 22.5

Table 8.15 The technical specifications for Part 3.

Features	***Operations***	***TAD Candidates***	***Machine Candidates***	***Tool Candidates***	***Machining Time for Each Candidate Machine (seconds)***
F_1	Milling ($Oper_1$)	+z	M_2, M_3, M_4	C_6, C_7, C_8	20, 20, 15
F_2	Milling ($Oper_2$)	-z	M_2, M_3, M_4	C_6, C_7, C_8	20, 20, 15
F_3	Milling ($Oper_3$)	+x, -x, +y, -z	M_2, M_3, M_4	C_6, C_7, C_8	15, 15, 11.25
F_4	Milling ($Oper_4$)	+x, -x, +y, +z	M_1, M_2, M_3, M_4	C_2	15, 15, 11.25, 18
F_5	Milling ($Oper_5$)	+x, -x, -y, -z	M_2, M_3, M_4	C_6, C_7, C_8	15, 15, 11.25
F_6	Milling ($Oper_6$)	+x, -x, -y, +z	M_2, M_3, M_4	C_7, C_8	15, 15, 11.25
F_7	Milling ($Oper_7$)	+x, -x, -z	M_2, M_3, M_4	C_7, C_8, C_{11}	15, 15, 11.25
F_8	Milling ($Oper_8$)	+x, -x, -z	M_2, M_3, M_4	C_6, C_7, C_8, C_{11}	25, 25, 18.75
F_9	Drilling ($Oper_9$)	+z, -z	M_1, M_2, M_3, M_4	C_2, C_3, C_4	30, 25, 25, 18.75
	Reaming ($Oper_{10}$)		M_2, M_3, M_4	C_9	20, 20, 15
	Boring ($Oper_{11}$)		M_2, M_3, M_4, M_5	C_{10}	20, 20, 15, 24
F_{10}	Drilling ($Oper_{12}$)	+y, -y	M_1, M_2, M_3, M_4	C_1	9.6, 8, 8, 6
	Tapping ($Oper_{13}$)		M_2, M_3, M_4	C_5	8, 8, 6
F_{11}	Drilling ($Oper_{14}$)	+z, -z	M_1, M_2, M_3, M_4	C_9	6, 5, 5, 3.75

Two objectives described in chapter 7 are used here as the optimising direction, i.e., the makespan and the balanced machine utilisation (Eq 7.1 - Eq 7.5).

The optimisation results of the PSO algorithm are shown in figure 8.8 and figure 8.9 respectively. From these two figures, it can be seen that the PSO can optimise the Makespan and balance the machine utilisation for group 1 successfully. The optimised schedule for minimised Makespan can be achieved after nearly 3000 iterations and the optimised schedule for balanced machine utilisation can be achieved more quickly, after 200 iterations.

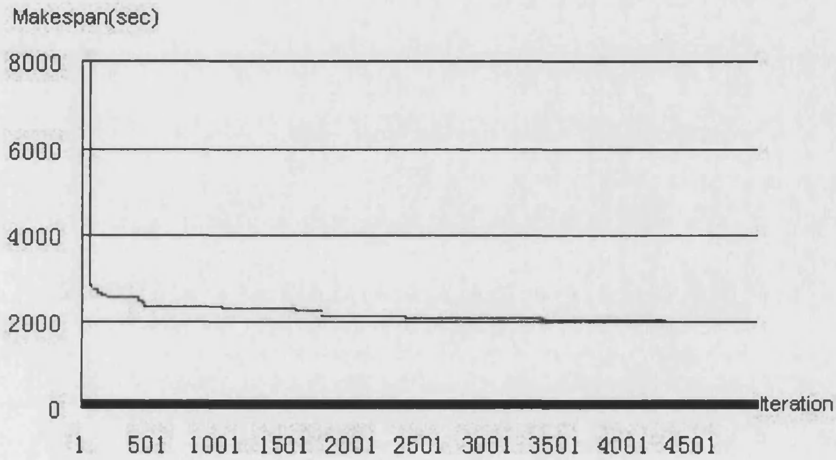


Figure 8.8 The optimisation result of Makespan for group1

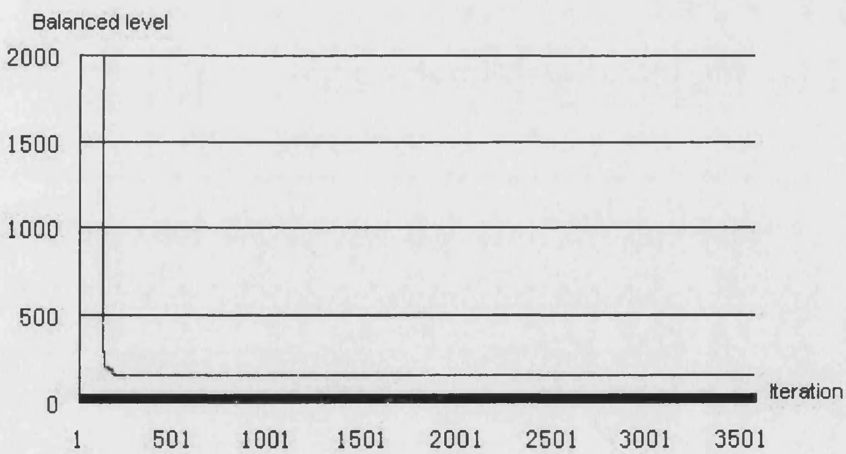


Figure 8.9 The optimisation result of balanced machine utilisation for group1

Two other evolutionary algorithms, GA and SA developed by Li (Li and McMahon 2006), are used to compare the optimised results, computation efficiency and robustness. Figure 8.10 and figure 8.11 show the optimisation results of GA, SA and PSO for two objectives respectively. The optimisation results are based on 5000 iterations for each algorithm (Here for SA, one iteration refers to an occurrence of current-plan replaced by a temp-plan; the current-plan and temp-plan are described in section 2.4.2). The population of the GA and the PSO are both set as 200.

Makespan:

Table 8.16 The comparisons of GA, SA and PSO of Makespan for group1.

Algorithm	Time for 5000 iterations	Robustness (successful optimisation trials out of 20 trials)
GA	19 min 40 sec	20
SA	59 min	14
PSO	7 min 40 sec	20

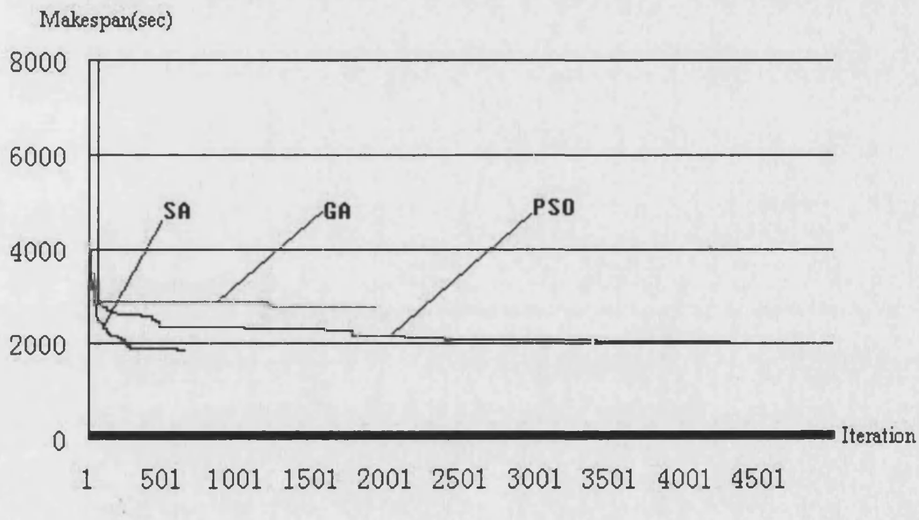


Figure 8.10 Comparison of GA, SA and PSO in Makespan for group1 (8 min's run)

As table 8.16 shows, the SA takes 59 min to finish 5000 iterations, so figure 8.10 shows the results after an 8 minute run. As shown in table 8.16 and figure

8.10, with the same time period, the SA and the PSO can achieve better results than GA, but the SA is not as robust as the GA and PSO. For 20 random consecutive trials, the SA proceeds with optimisation successfully in 14 trials, the PSO and the GA can proceed with optimisation successfully in all 20 trials.

Balanced machine utilisation:

Table 8.17 The comparisons of GA, SA and PSO of Balance machine utilisation for group1.

Algorithm	Time for 5000 iterations	Robustness (successful optimisation trials out of 20 trials)
GA	16 min 15 sec	20
SA	45 sec	20
PSO	3 min	20

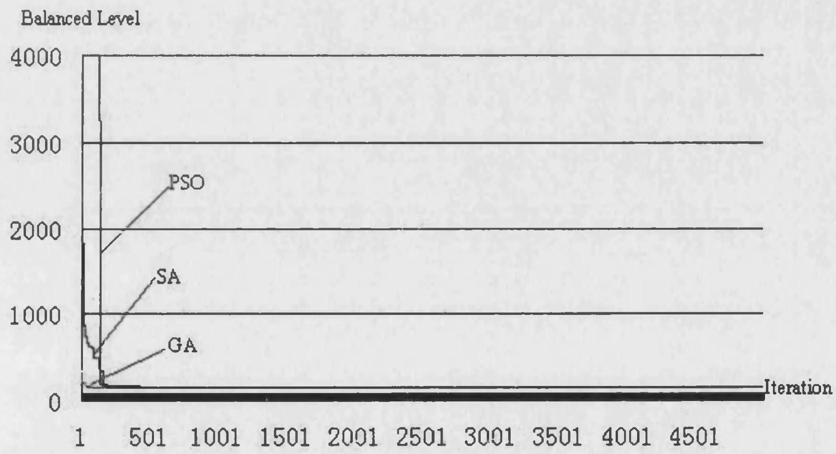


Figure 8.11 Comparison of GA, SA and PSO for group1

From table 8.17 and figure 8.11, it can be seen that all three algorithms can achieve the optimized results in all 20 trials, but the GA and the SA algorithms approach the optimised result more quickly.

Group 2:

Eight parts taken from (Li and McMahon 2006) have been used to test the algorithm under more complex conditions. The relevant specifications of the parts are given in

Table 8.18. The above two objectives have been used again, and the optimisation results are shown in figure 8.12 and figure 8.13. It can be seen that the PSO can optimise the Makespan after nearly 4000 iterations and the balanced machine utilisation after 3000 iterations.

Table 8.18 The technical specifications for the part in Group 2 (Li and McMahon 2006).

Parts	*Numbers of Operations (with Numbers of Alternative Machining Plans for Each Operation)*	*Numbers of Constraints*
1	7 (9, 9, 27, 8, 8, 9, 36)	11
2	8 (9, 9, 36, 18, 27, 8, 27, 18)	11
3	7 (9, 9, 36, 36, 18, 6, 6)	10
4	9 (9, 9, 27, 6, 36, 36, 6, 18, 18)	18
5	7 (9, 9, 36, 36, 36, 18, 6)	13
6	9 (9, 9, 36, 27, 18, 6, 27, 6, 18)	20
7	5 (9, 27, 27, 18, 9)	5
8	7 (9, 9, 27, 36, 36, 6, 6)	13

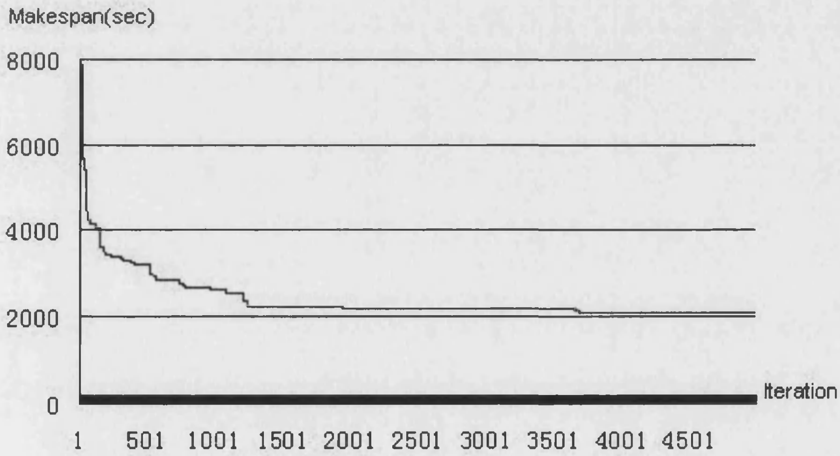


Figure 8.12 The PSO optimisation result of Makespan for group2

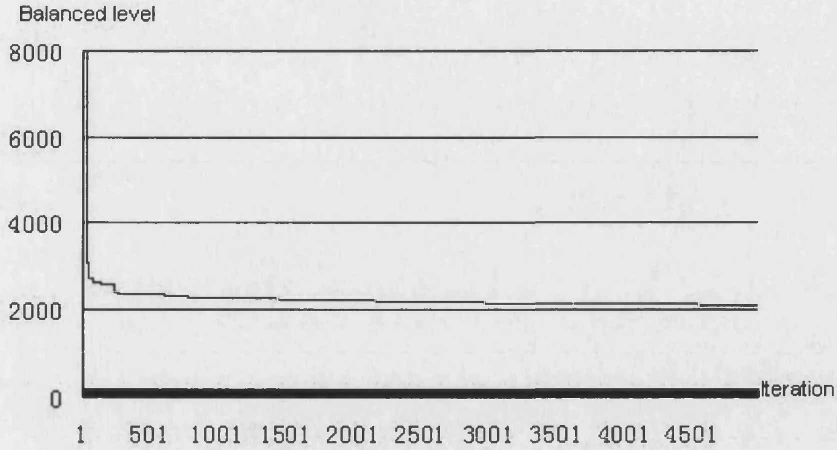


Figure 8.13 The PSO optimisation result of balanced machine utilisation for group2

The comparisons of the GA, SA and PSO designed for group 1 are used to compare the results, efficiencies and robustness for group 2 as well.

Makespan:

Table 8.19 The comparisons of GA, SA and PSO of Makespan for group2.

Algorithm	Time for 5000 iterations	Robustness (successful optimisation trials out of 20 trials)
GA	16 min 45 sec	20
SA	45 min	6
PSO	7 min	20

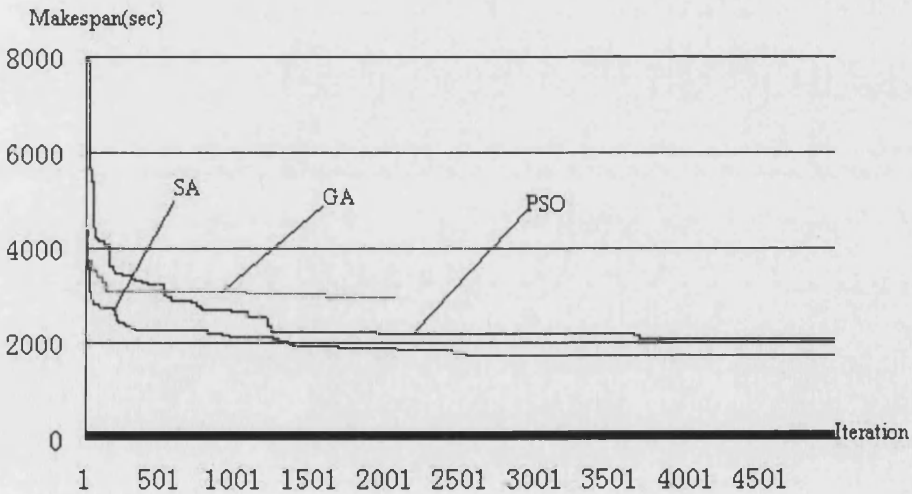


Figure 8.14 Comparison of PSO, GA and SA for group2 (in 7 min)

As shown in table 8.19 and figure 8.14, with the same time period, the PSO and the SA can achieve better results than the GA. But for 20 random consecutive trials, the SA can only proceed with optimisation successfully in 6 trials, the PSO and the GA can proceed with optimisation successfully in all 20 trials.

Balanced machine utilisation:

Table 8.20 The comparisons of GA, SA and PSO of Balance machine utilisation for group1.

Algorithm	Time for 5000 iterations	Robustness (successful optimisation trials out of 20 trials)
GA	16 min 45 sec	20
SA	22 min	6
PSO	7 min 30 sec	20

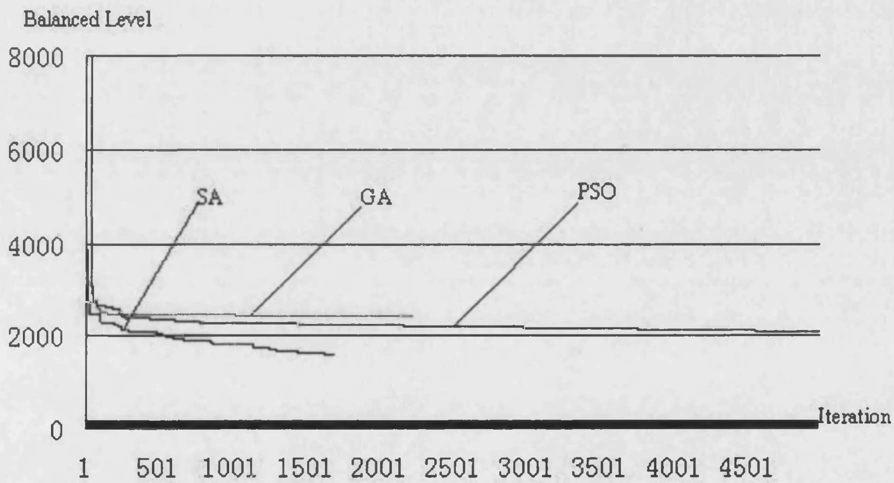


Figure 8.15 Comparison of PSO, GA and SA for group2

From table 8.20 and figure 8.15, it can be observed that all of the algorithms can reach good results, while different characteristics are shown due to the inherent mechanisms of the algorithms. The SA is much “sharper” to find optimised solutions than the GA and the PSO. The SA can achieve better results than the GA and the PSO. However, in 20 trials, the SA can only proceed with optimisation successfully in 6 trials but the GA and the PSO can proceed with optimisation successfully in all 20 trials.

From the comparisons of the GA, SA and PSO, especially from the performance of the SA, it can also be observed that the optimisation of makespan takes longer than the optimisation of balanced machine utilisation. Compare table 8.16 and table 8.17, 5000 iterations of SA for optimising makespan take 59 min but the same number of iterations for optimising balanced machine utilisation only take 45 sec. As described above, one iteration in the SA refers to an occurrence of the current-plan replaced by a temp-plan. Therefore it takes longer (more trials) to generate a temp-plan which is better than the current-plan. This means that the optimisation of Makespan is more difficult than that of balanced machine utilisation (This can be also observed from the results of the PSO for both objectives).

Summary of GA, SA and PSO algorithms

As discussed in section 8.2 and section 8.3, the GA, SA and PSO algorithms are used to optimise the operation sequencing problem and the IPPS problem. All of them can yield good results, but they have different characteristics. The GA and the PSO are both population based algorithms but the SA is not. So the optimising processes of the GA and the PSO take a longer time than that of the SA in the examples of section 8.2. It can also be observed that the PSO needs to adjust the particle dimensions' by updating the velocities and positions of them due to its intrinsic mechanism so that it needs more computation time than the GA. For the optimisation results, the SA and the PSO both outperform the GA in all the above case studies. As the complexity of the problem increases (for example when optimising IPPS problems), the SA can achieve better results than the GA and the PSO in the case studies described in section 8.3.1. But as the complexity of the problem increases, the SA is not as robust as the GA and the PSO. This is probably because the SA is not population based, so the initial plan does not have enough diversity to enable it to search the space successfully. Also as the complexity of the problem increases, it can be seen that the optimisation speed advantages of the GA and the SA over the PSO diminish. It is well known that simple mathematic operations run much faster than other position changing operations. This can probably be attributed to the fact that each iteration of the PSO algorithm uses mainly simple mathematical operators that can be finished in a shorter time than for the GA and the SA algorithms with mainly complex position changing operators. In constraints handling, the GA and the SA can use the adjust method developed by Li et al. (2002) that keep the plan feasible, but the PSO can only use the penalty method to enable the results to comply with the constraints due to its intrinsic mechanism. The above discussion is illustrated in table 8.21.

Table 8.21 The comparison of GA, SA and PSO algorithms.

Algorithm	Population based	Optimisation result (out of 10)	Optimisation speed	Constraints handling	Robustness
GA	Yes	6	Fast but get slow when complexity of problems increases	Adjust Penalty	Robust
SA	No	9	Faster but get slow when complexity of problems increases	Adjust Penalty	Not robust when complexity of problems increases
PSO	Yes	8	Fast	Penalty	Robust

8.3.2 Experiment 2

The parts in group 2 have been used to test the replanning ability of the PSO developed for IPPS under machine breaking down and new order arrival conditions. In this experiment, as new order arrivals and machine breakdowns occur, it is appropriate to set the total tardiness as the main objective, so as to make comparison under these two conditions.

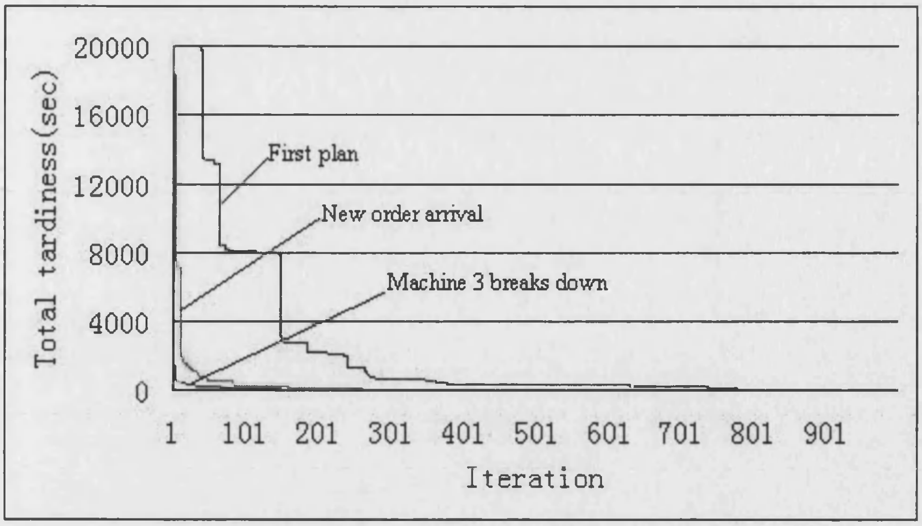


Figure 8.16 Results of optimization for first planning, replannings

after new order arrival and machine breaks down

1. First planning

The 8 parts in group 2 consist of a total of 59 operations. Here the Due Date (DD) is set as 2700.0. Table 8.22 shows the first scheduling results of the complete time for individual parts in group 2. It also can be seen from figure 8.16, the process can be optimized to achieve the DD for all the parts.

Table 8.22 Complete time for individual part after optimisation

Part	1	2	3	4	5	6	7	8
Time	2272	1958	2665	2355	2350	2505	2690	2697

2. Condition of new order arrival

At time 1000.0, a new order arrives (part 9 in this experiment which is the same as part 1) and the corresponding DD is set as 3500.0. At the time 1000.0, 18 operations have been finished and 41 operations are left. With part 9 added as a new part, 7 operations are then inserted into the total operation list which includes 48 operations.

The individual available time for all the machines is shown in table 8.23. The optimization result is shown in figure 8.16 and the individual complete time for all the 9 parts after replanning is shown in table 8.24.

Table 8.23 Machines available time when new order arrives

Machine	1	2	3	4	5
Available time	1000	1120	1078	1116	1000

Table 8.24 Complete time for individual part after replanning when new order arrives

Part	1	2	3	4	5	6	7	8	9
Time	2221	2513	2051	2659	2660	2689	2344	2654	3458

3. Condition of machine breaks down

At time 1500.0, machine 3 breaks down (repair time 300.0). Table 8.25 shows the available times for different machines. At that time, 16 operations have been finished and only 32 operations are left. The optimisation result is shown in figure 8.16 and the individual complete time for all the 9 parts after replanning is shown in table 8.26.

Table 8.25 Machines available time when machine 3 breaks down

Machine	1	2	**3**	4	5
Available time	1500	1560	**1800**	1574	1500

Table 8.26 Complete time for individual part after replanning when machine 3 breaks down

Part	1	2	3	4	5	6	7	8	9
Time	1845	2651	2497	2484	2613	2659	2334	2634	3357

Because the algorithm will not continue optimizing when it achieves the lowest objective (here total tardiness=0), it can find the earliest complete date for parts by reducing the DD. For example if DD(part 1-8) = 1500.0, DD(9) = 2500, the planning results are shown in table 8.27, 8.28 and 8.29:

Table 8.27 Complete time for individual part after first planning

Part	1	2	3	4	5	6	7	8
Time	1313	1914	2399	2242	1775	1488	1123	2095

Table 8.28 Complete time for individual part after replanning after new order arrives

Part	1	2	3	4	5	6	7	8	9
Time	1415	1738	2038	2226	1710	1572	1123	2539	2546

Table 8.29 Complete time for individual part after replanning when machine 3 breaks down

Part	1	2	3	4	5	6	7	8	9
Time	1415	1918	2063	2310	1844	1692	1123	2295	2502

From this case study, it can be seen that the modified PSO algorithm has the ability to replan when new order arrival and machine breakdowns occur. Figure 8.16 shows that with the method discussed in chapter 7, the replanning time can be reduced and the computation efficiency can be improved significantly.

8.4 System implementation and case study for 5-axis process planning

8.4.1 System implementation

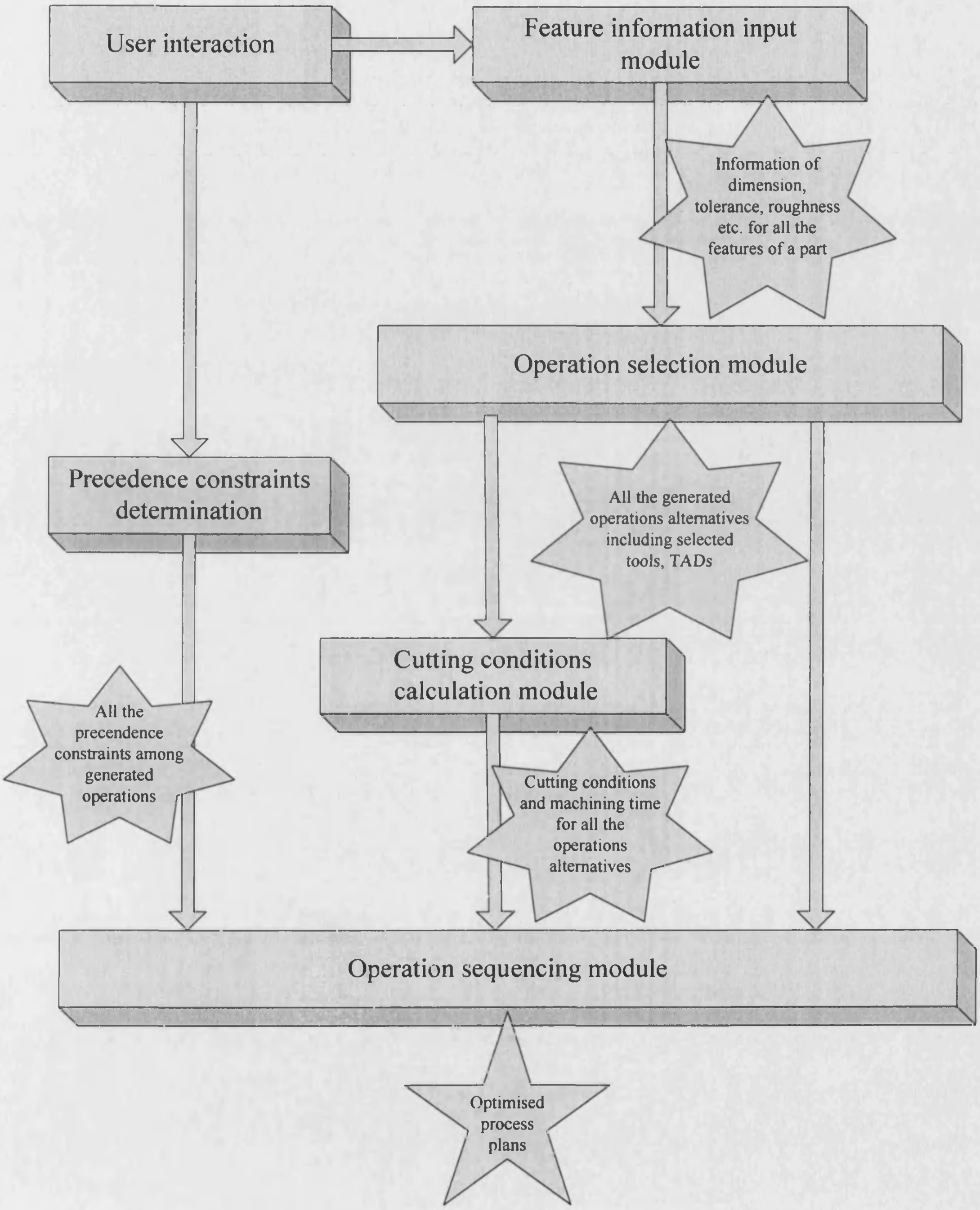


Figure 8.17 Information flow between main modules of CAPP

As discussed in section 4.2 and illustrated in figure 4.2, the CAPP system for the 5-axis CNC environment is comprised of four modules: feature information input module, operation selection module, cutting conditions calculation module and operation sequencing module. The first two modules are implemented with Visual C++ 6.0 and the last two modules are implemented with Java SDK 1.5.0 (this is for comparison with the GA and SA approaches developed by Li with Java language). The information flow between these modules in generating a process plan for a part is illustrated in figure 8.17. To achieve this, the information shown in the stars is stored in different databases and files.

As figure 8.17 shows, the user inputs all the feature information including feature types, dimensions, tolerances and roughness into the system. This information is automatically stored into the *feature.mdb* database which is shown in table 8.30.

Table 8.30 Tables in feature.mdb

Table name	Information stored
PartManage	Recording all the general information of the components that have been input into the system including part name, part number, material, production batch and part dimensions etc.
Part	Detailed features information for a specific part including feature type, sub type, all dimensions, dimension tolerances, roughness, feature original point, dimension directions and other related information

Table 8.31 Tables in Tools.mdb

Table name	Information stored
Cutters	Recording all the general information of the cutting tools that have been input into the system including tool id, tool dimensions, material and tool holders etc.
ToolTypeRecom	Store the information as table 5.3 shows.

Table 8.32 Tables in Featureagent.mdb

Table name	Information stored
PartForOperate	Store the information for the part needs to be processed.
FeatureForOperate	Store the feature information transferred from feature.mdb.
HardnessAbility	Store the hardness value of different tool material can achieve.
MaterialHardness	Store the related hardness value for different materials.
RoughnessAbility	Store the surface roughness different processes can achieve.
ToleranceAbility	Store the dimension tolerances different processes can achieve.
CuttingSpeed	Store the cutting speed ranges for different part materials and tool materials.
MillingFeed	Store the milling feed ranges for different part materials, tool materials and tool types.
DrillingFeed	Store the drilling feed rates for different hole diameters and tool materials.
OperationList	Store all the detailed information for generated operations including operation type, sub type, operation description, selected alternative tools and TADs etc.

All this information is used by the operation selection module to generate the operations for every feature of the part including the alternative tools, TADs and detailed operation descriptions etc. To implement this process, *Machine.mdb*, *Tools.mdb* and *Featureagent.mdb* are designed to include all the information needed by the operation selection module and the generated alternative operations list. *Machine.mdb* stores the specifications of the 5-axis CNC machine tool used in this research including its maximum work spindle speed, federate, maximum travel distance in X, Y and Z, and the maximum dimensions, weight of part it can machine etc. All this information indicates the capability of the machine and can be used to check if the part is machinable and the operation is executable. The design of *Tools.mdb* and *Featureagent.mdb* are shown in table 8.31 and 8.32 respectively.

The generated operations are stored in the *OperationList* table of *Featureagent.mdb*. With this information, the cutting conditions calculation module calculates the corresponding cutting speed, feed rate and machining time for all the alternative operations. This information is stored in file *operation.dat*, and is used by the operation sequencing module to generate the optimised process plan with the precedence constraints between all the operations determined by the user. The final optimised process plan can be output to the monitor or saved to file.

8.4.2 Case study

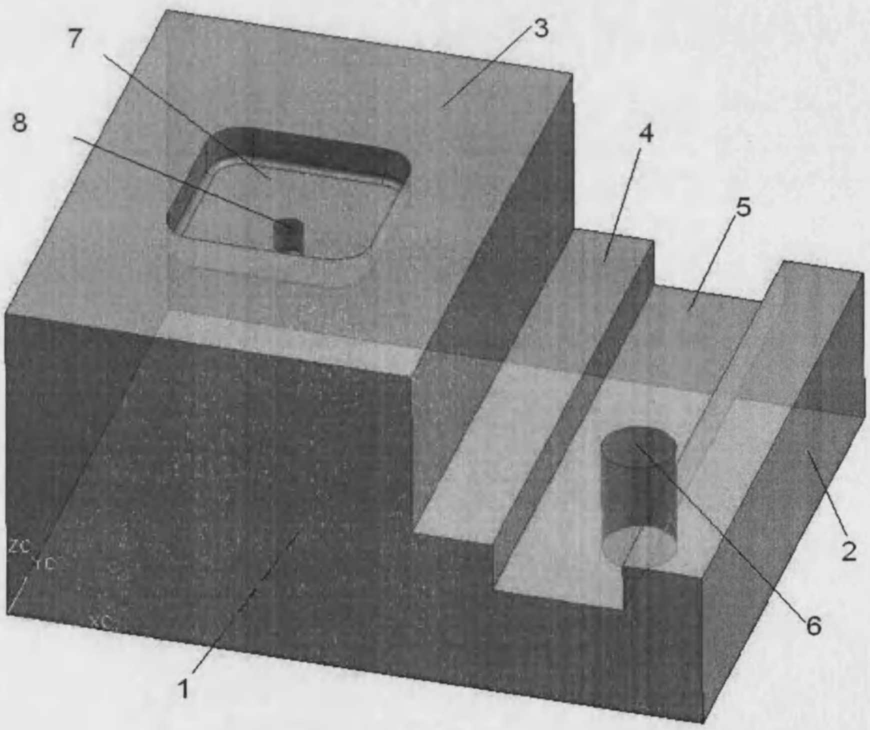


Figure 8.18 Example part with 8 features

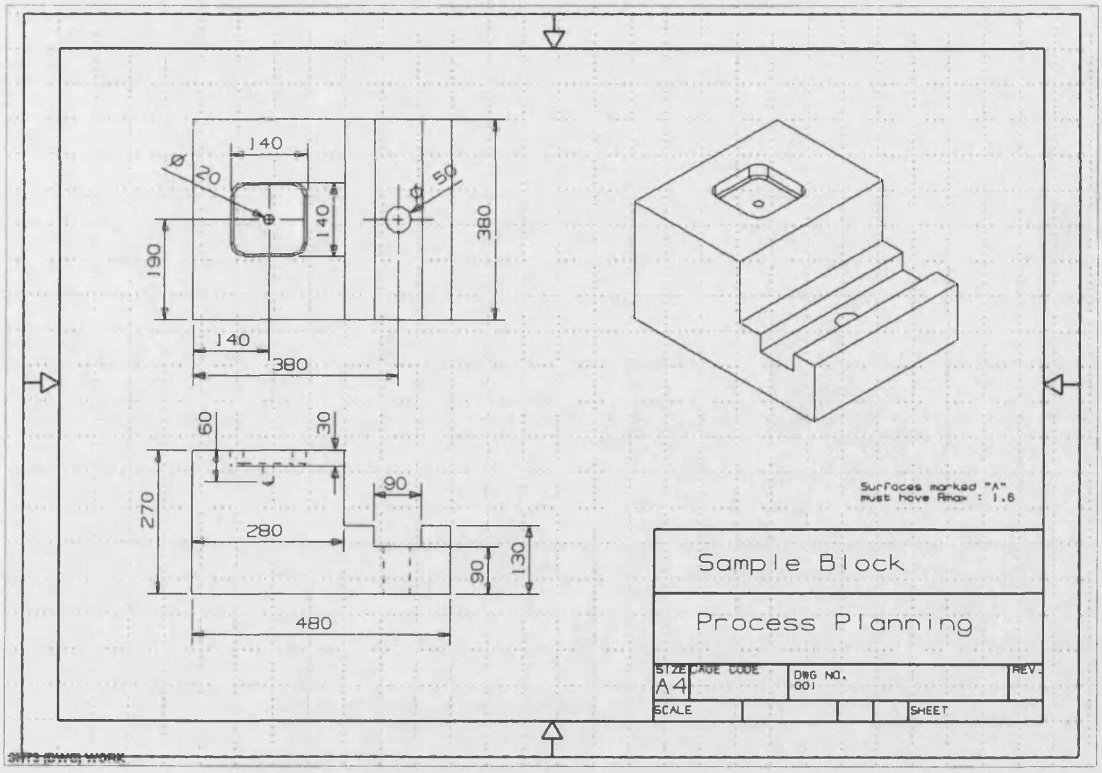


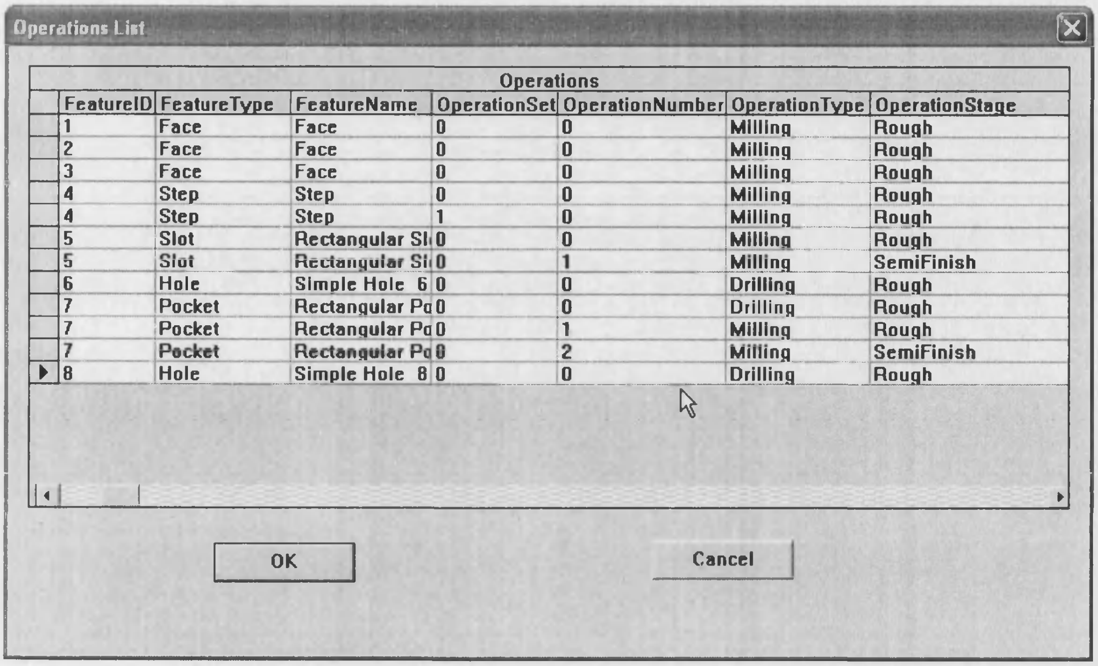
Figure 8.19 Specifications of example part

A part with 8 features is used as an example in this case study as figure 8.18 shows. Table 8.33 shows the description of features and corresponding roughness and depth. Other dimensions can be seen in figure 8.19.

Table 8.33 Descriptions of features in example part

Features	Feature Descriptions	Depth(mm)	Roughness (μm)
F_1	A face at the bottom	5	6.3
F_2	A face at the right side	5	12.5
F_3	A face on the top	6	6.3
F_4	A step	140	6.3
F_5	A slot	40	1.6
F_6	A through hole	90	12.5
F_7	A pocket on the top	30	3.2
F_8	A blind hole	30	12.5

The operation selection module generates all the alternative operation with all applicable tools and TADs as figure 8.20 shows.



FeatureID	FeatureType	FeatureName	OperationSet	OperationNumber	OperationType	OperationStage
1	Face	Face	0	0	Milling	Rough
2	Face	Face	0	0	Milling	Rough
3	Face	Face	0	0	Milling	Rough
4	Step	Step	0	0	Milling	Rough
4	Step	Step	1	0	Milling	Rough
5	Slot	Rectangular Sl	0	0	Milling	Rough
5	Slot	Rectangular Sl	0	1	Milling	SemiFinish
6	Hole	Simple Hole 6	0	0	Drilling	Rough
7	Pocket	Rectangular Pc	0	0	Drilling	Rough
7	Pocket	Rectangular Pc	0	1	Milling	Rough
7	Pocket	Rectangular Pc	0	2	Milling	SemiFinish
8	Hole	Simple Hole 8	0	0	Drilling	Rough

Figure 8.20 Generated operations list from operation selection module

Table 8.34 The generated operations for example part.

Features	*Operation Set*	*Operations*	*Depth To Cut/Diameter To Cut*	*Tool Candidate IDs*	*TAD Candidates*
F_1	0	Rough Milling ($Oper_1$)	5	5	+z
F_2	0	Rough Milling ($Oper_2$)	5	4	+x
F_3	0	Rough Milling ($Oper_3$)	6	5	-z
F_4	0	Rough Milling ($Oper_4$)	140	4	-z
	1	Rough Milling ($Oper_4$)	200	3	-x
F_5	0	Milling ($Oper_5$)	39	12, 17	-z
		SemiFinish Milling ($Oper_6$)	1	12, 17	-z
F_6	0	Rough Drilling ($Oper_7$)	90	20	-z, +z
F_7	0	Rough Drilling ($Oper_8$)	30	23	-z
		Rough Milling ($Oper_9$)	29.6	11, 14	-z
		SemiFinish Milling ($Oper_{10}$)	0.4	11, 14	-z
F_8	0	Rough Drilling ($Oper_{11}$)	30	19	-z, +z

Table 8.35 All the operations alternatives with machining time.

Feature ID	*Operation Set*	*Operation No.*	*Tool ID*	*TAD*	*Machining time(min)*
F_1	0	1	5	+z	1.60
F_2	0	2	4	+x	1.49
F_3	0	3	5	-z	1.60
F_4	0	4	4	-z	12.14
	1	4	3	-x	9.6
F_5	0	5	12	-z	9.2
			17	-z	4.6
		6	12	-z	1.8
			17	-z	0.9
F_6	0	7	20	-z	0.42
				+z	0.42
F_7	0	8	23	-z	0.15
		9	11	-z	4.18
			14	-z	1.98
		10	11	-z	1.04
			14	-z	0.49
F_8	0	11	19	-z	0.09
				+z	0.09

Table 8.34 shows the detailed generated operations information that is the input of the cutting conditions calculation module. Then the cutting conditions are selected and machining times for all the operations alternatives are calculated as table 8.35 shows.

Table 8.36 Precedence constraints between machining operations.

Constraints	**Descriptions**
Datum interactions	$Oper_1$ is the first operation. $Oper_2$ is prior to $Oper_4$, $Oper_5$, $Oper_6$ and $Oper_7$. $Oper_3$ is prior to $Oper_4$, $Oper_5$, $Oper_6$, $Oper_7$, $Oper_8$, $Oper_9$, $Oper_{10}$ and $Oper_{11}$.
Material removal interactions	$Oper_5$ and $Oper_6$ are prior to $Oper_{11}$. $Oper_8$, $Oper_9$ and $Oper_{10}$ are prior to $Oper_{11}$.
Feature priority	$Oper_4$ is prior to $Oper_5$, $Oper_6$ and $Oper_7$.
Fixed order	$Oper_5$ is prior to $Oper_6$ $Oper_8$ is prior to $Oper_9$ and $Oper_{10}$, and $Oper_9$ is prior to $Oper_{10}$.

The precedence constraints between these operations are stated in table 8.36. When considering these constraints, the task of the operation sequencing module is to generate the optimised process plan so as to achieve the least total machining time. In this process, four methods of set-up determination discussed in chapter 6 are tested. It is assumed that all 6 directions can be used as the set-up direction. The set-up change time, tool change time and table turn time are set as 1200 sec, 12 sec and 3 sec respectively.

Figure 8.21 shows the optimisation process of total machining time in which the set-up is determined by the TR method (Total Random method presented in chapter 7). It can be seen that within 100 iterations the optimised result can be achieved. Table 8.37 and table 8.38 show two optimised process plans. Neither process plan needs a set-up change and both require 7 tool changes.

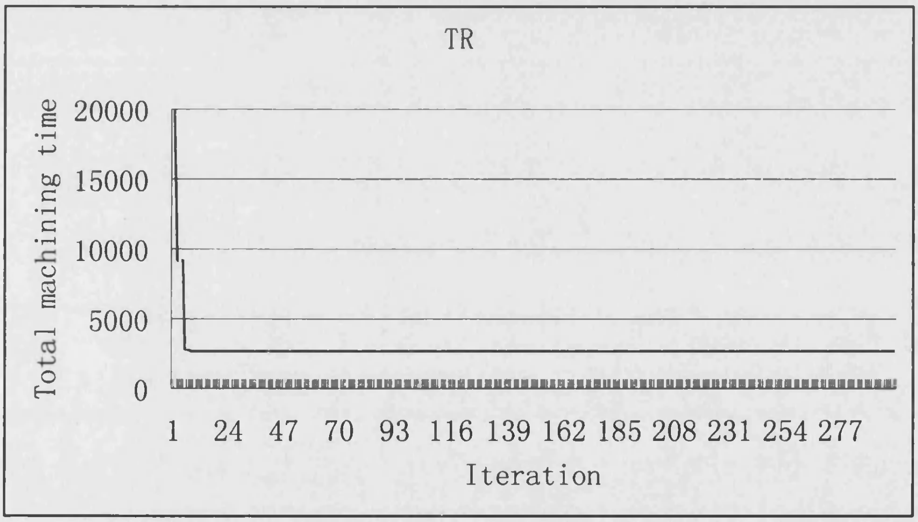


Figure 8.21 Optimisation result of total machining time (set-up determination by TR)

Table 8.37 An optimised plan for the part (total machining time = 2673.0 sec).

Sequenced operation No.	1	3	8	2	4	9	10	5	6	11	7
Tool	5	5	23	4	3	14	14	17	17	19	20
TAD	+z	-z	-z	+x	-x	-z	-z	-z	-z	-z	-z
Set-up	-y	-y	-y	-y	-y	-y	-y	-y	-y	-y	-y

Table 8.38 An optimised plan for the part (total machining time = 2673.0 sec).

Sequenced operation No.	1	3	8	2	4	9	10	5	6	7	10
Tool	5	5	23	4	3	14	14	17	17	20	19
TAD	+z	-z	-z	+x	-x	-z	-z	-z	-z	-z	-z
Set-up	+y	+y	+y	+y	+y	+y	+y	+y	+y	+y	+y

Similar optimisation result can be achieved when determining the set-up with FR method (First Random method presented in chapter 7) as figure 8.22 shows (the same optimised result 2673.0 can be achieved).

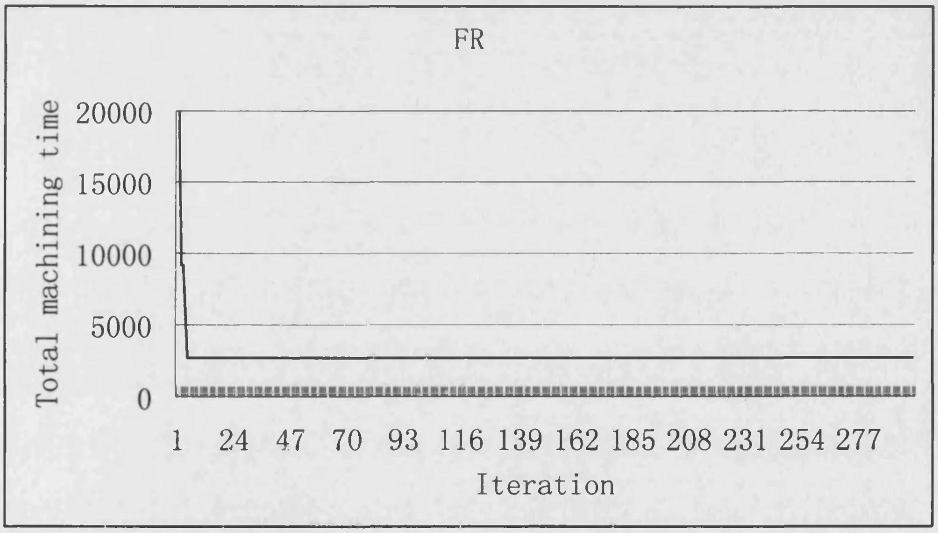


Figure 8.22 Optimisation result of total machining time (set-up determination by FR)

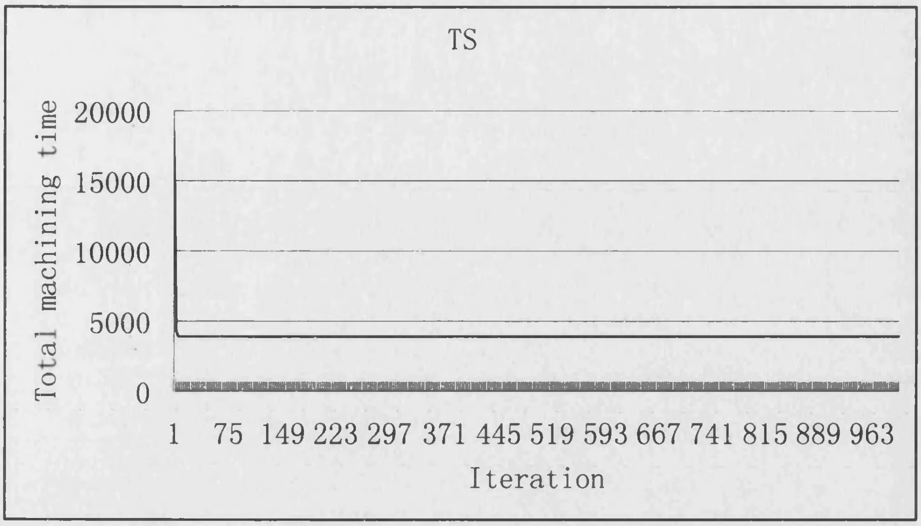


Figure 8.23 Optimisation result of total machining time (set-up determination by TS)

Figure 8.23 shows the optimisation results using the TS method (Total Set method presented in chapter 7) to determine the set-up. It can be observed that even in 1000 iterations, the algorithm can only achieve an optimised result 3870.0 which means it needs one set-up change in the process. Table 8.39 shows a generated process plan with the TS method.

Table 8.39 An optimised plan for the part with TS method (total machining time = 3870.0 sec).

Sequenced operation No.	1	3	8	2	4	5	6	7	9	10	11
Tool	5	5	23	4	3	12	12	20	14	14	19
TAD	+z	-z	-z	+x	-x	-z	-z	-z	-z	-z	-z
Set-up	+z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z

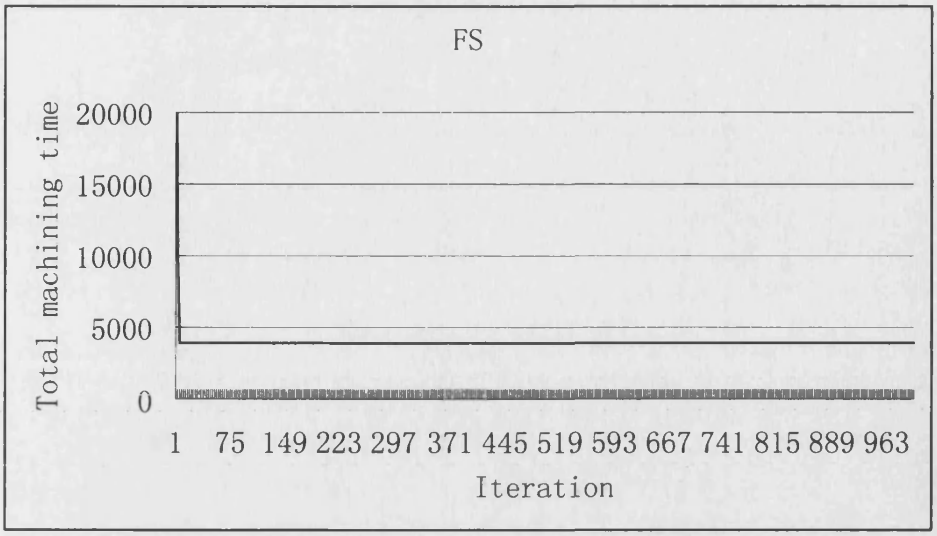


Figure 8.24 Optimisation result of total machining time (set-up determination by FS)

Table 8.40 An optimised plan for the part with FS method (total machining time = 3870.0 sec).

Sequenced operation No.	1	3	2	4	8	5	6	9	10	11	7
Tool	5	5	4	3	23	17	17	14	14	19	20
TAD	+z	-z	+x	-x	-z	-z	-z	-z	-z	-z	-z
Set-up	+z	+y	+y	+y	+y	+y	+y	+y	+y	+y	+y

Similar results can also be achieved with the FS method (First Set method presented in chapter 7) to determine the set-up as shown in figure 8.24 and table 8.40. It can be observed that by constraining the first set-up direction, the flexibility of selecting the set-up directions is reduced so that the optimised result can not be achieved (here, with set-up as +2 or -2, the part can be machined within one single set-up). Therefore, when all the 6 directions can be used as set-up directions, it is better to determine the set-up with the TR or FR methods. However, there may be some directions that can not be used as set-up directions. For example, if it is assumed that only certain set-up directions can be selected in (-3, 3, -1), TR, FR, TS and FS can all achieve the optimised result of 3870.0 sec which means there is at least one set-up change. Table

8.41 shows an optimised plan using the TS method. After operation 1 is executed, a set-up change is required.

Table 8.41 An optimised plan for the part with FS method (total machining time = 3870.0 sec).

Sequenced operation No.	1	3	8	2	4	5	6	7	9	10	11
Tool	5	5	23	4	3	17	17	20	14	14	19
TAD	+z	-z	-z	+x	-x	-z	-z	-z	-z	-z	-z
Set-up	+z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z

8.5 Summary

In this chapter, the implementations of the operation sequencing module for 3-axis machining, the IPPS optimisation module and the CAPP system for 5-axis machining are illustrated. For the operation sequencing module for 3-axis machining, two experiments using three parts are designed to determine the PSO algorithm parameters and verify the efficiency of the PSO algorithm. Two groups of different parts are used to verify the efficiency of the PSO algorithm for optimising the IPPS problem. By comparing the GA, SA and the PSO algorithms for these two modules, the benefits and drawbacks of these algorithms are discussed. It can be seen that the PSO algorithm can obtain better computation results than GA in the operation sequencing problem and the IPPS optimisation problem and is more robust than the SA. At this point in time the conclusions are limited by this computational experience, and more theoretical analysis needs to be made in the future.

Finally the process of generating an optimised process plan for an example part by the developed 5-axis CAPP system is illustrated by a case study. From the case study, it can be seen that the system can generate the optimised process plan to achieve the minimised total machining time.

Chapter 9 Conclusions and Future Work

9.1 Conclusion and Contributions

The developed CAPP system and the stand alone IPPS module are considered to meet the research objectives given in section 3.2. The primary aim of developing a prototype CAPP system for common prismatic components in a 5-axis CNC machining environment has been achieved. The workflow and general structure of the system has been given in chapter 4. The four major modules that form the prototype CAPP system, namely *Feature information input module*, *Operation selection module*, *Cutting conditions calculation module* and *Operation sequencing module*, have been designed and implemented. A case study shown in chapter 8 has proved that the representation model of process planning for common prismatic component in 5-axis machining environment works correctly and the optimised process plans can be achieved by using a Particle Swarm Optimisation (PSO) algorithm. Furthermore, a stand alone operation sequencing module for 3-axis machining and a stand alone adaptive IPPS module have been designed and implemented. Case studies have been used to verify and test these two modules and show that these two problems have been modelled properly and the sequence and schedules can be optimised with the PSO algorithm. Through case studies, a comparison has been made between the result of the modified PSO algorithm and previous published results using the Genetic Algorithm (GA) and the Simulated Annealing (SA) algorithm respectively, and for these cases the PSO algorithm has been shown to outperform both the GA and SA in the majority of applications by consideration of computation efficiency, optimisationability and robustness.

The following contributions are considered to have been made.

(1) A generative CAPP prototype system for common prismatic parts in 5-axis machining environment has been developed.

- A feature-based model has been developed to assist the user to input the features information from a CAD model of a part. This model can also help deal with adding, editing and deleting features to modify the information of a part.
- In order to simplify the algorithm generation and development time, the research work presented in this thesis has been carried out using a restricted set of only 5 features. These features are namely: planar face, pocket, slot, hole and step. By representing these features with an Object Oriented Programming Strategy (OOPS), it is possible to organise and represent the feature information for easy message processing which offers the flexibility to modify the definition of a feature, its structure, variables and functions without affecting the rest of the system configuration. Also it is easier to expand the system to include other feature types without affecting the existed feature types.
- The feature information required for downstream process planning activities has been summarised including Identifier, Dimensions, Location and Technical specifications that have been set as the variables for different feature classes.
- The logic and algorithm to select the machining operations for each feature have been discussed and organised in section 5.2. By executing the following steps: a) Feature information extraction, b) Operation Type (OPT)

Selection, c) Tool selection and d) TAD determination, the operations including the selected tools and TADs can be determined for each feature.

- To support global optimisation of the process plans, the operation selection module can generate all the applicable operations including all the available alternative machining resources for each type of feature. For simplification and efficiency, only 2 tools will be selected in the procedure according to tools' materials and sizes. The hierarchy for generating alternative operations is shown in figure 5.15.
- The cutting conditions including cutting speed, feed rate and machining time can be selected and calculated based on the knowledge generated from different literature (handbooks, catalogues).
- To support the operation selection module and cutting conditions calculation module, all the knowledge needed for decision making has been stored in the database which it is easy to modify according to the changes of the manufacturing environment. The rules for realising the logic have been coded in the program to generate the operations.
- Operation sequencing is one of the crucial tasks in process planning. However, it is an intractable process to identify an optimised operation sequence with minimal machining cost in a vast search space constrained by manufacturing conditions. The complicated operation sequencing process has been modelled as a combinatorial optimisation problem, and an expanded model to represent the process plans for 5-axis machining has been proposed.
- A modern evolutionary algorithm, the Particle Swarm Optimisation (PSO) algorithm, has been employed and modified to solve it effectively. Initial

process plan solutions are formed and encoded into particles of the PSO algorithm. The particles "fly" intelligently in the search space to achieve the best sequence according to the optimisation strategies of the PSO algorithm. Meanwhile, to explore the search space comprehensively and to avoid being trapped into local optima, several new operators have been developed to improve the particles' movements to form a modified PSO algorithm.

- The set-up determination for 5-axis machining is considered to be the most difficult problem in the move from 3-axis process planning. Four methods to determine the set-up directions have been proposed and discussed.
- A case study shown in chapter 8 is used to test and verify the CAPP system. Through the case study, it can be seen that the different modules proposed can achieve the expected results. The optimised process plan with suitable operations and machining resources can be achieved. A comparison of the methods to determine the set-up has been conducted, it can be observed that without set-up constraints, the TR and FR methods can both achieve the optimised result whereas the FS and TS can not.

(2) An independent operation sequencing module has been developed and the PSO algorithm is used to optimise the combinational operation sequencing problem. Three example parts have been used to verify the efficiency of the PSO and compare it with two other popular algorithms, the Genetic Algorithm (GA) and Simulated Annealing algorithm (SA). Through the case study, the parameters for the PSO have been determined and the benefits and drawbacks of different algorithms have been determined.

(3) The problem of Integrated Process Planning and Scheduling (IPPS) optimisation has been defined as the optimisation of an extension to the operation sequencing problem with scheduling objectives. The procedure of evaluating the performance of the schedule from sequenced operations has been discussed. Through 2 case studies, the computation results and efficiencies of the PSO algorithm and the comparisons with the GA and the SA have shown the benefits and drawbacks of the different algorithms to be as follows:

- Compare the computation efficiency of the PSO, GA and SA, in the case of operation sequencing optimisation problem, the optimising processes of the PSO take a longer time than those of the GA and SA in the examples of section 8.2.
- As the complexity of the problem increases (for example when optimising IPPS problems), the optimisation speed advantages of the GA and the SA over the PSO diminish. The above two points can probably be attributed to the facts: a) The GA and the PSO are both population based algorithms but the SA is not; b) each iteration of the PSO algorithm uses mainly simple mathematical operators that can be finished in a shorter time than for the GA and the SA algorithms with mainly complex position changing operators.
- The SA and the PSO both outperform the GA in all the case studies experimented in chapter 8.
- As the complexity of the problem increases, the SA can achieve better results than the GA and the PSO in the case studies described in section 8.3.1. But as the complexity of the problem increases, the SA is not as robust as the PSO and the GA.

9.2 Limitations

The methodology presented and the system developed in the thesis has, however, got certain limitations, which are described below.

1) Restricted component geometry, i.e. only 3D prismatic components are considered.
2) Limited standard feature classes. Only five basic features, including planar faces, holes, slots, pockets and steps are included in the research currently. It has not yet been developed to plan interacting features and contoured 3-D surfaces which will be subject to future research.
3) Lack of an automatic validity check for features information input and precedence constraints determination.
4) Lack of consideration of fixturing.

9.3 Recommendation for future work

The proposed system works, but future work is still required to increase its capabilities:

(1) Automatic feature extraction from a CAD model

A feature modeller has been developed to input the feature information of the part for downstream process planning but it requires the user to input it manually. It is error-prone compared to automatic feature recognition or extraction from feature-based design tools.

(2) Automatic precedence constraints generation

When generating the precedence constraints manually, it is possible to miss some constraints which will result in the optimised process plans being unfeasible. So it is better to be able to generate the precedence constraints between operations automatically according to the characteristics of features and operations. To make this happen, more information is required when inputting the feature information.

(3) Extension of component geometry and feature classes

It is required to find a method to represent contoured 3-D surface features and apply corresponding operation selection methods for it. The current feature classes are also needed to be further extended and the interacting features need to be considered as well.

(4) Consideration of fixturing

Fixturing may affect process planning dramatically. Future effort should be paid to fixturing constraints and corresponding clamping strategy. With automatic fixture selection, the set-up time can be determined more accurately so as to increase the accuracy of total machining time calculation and schedule determination.

(5) Improvement of PSO algorithm

The PSO algorithm can be further improved by employing a new sequence adjustment method to reduce the computation time in more complex operation sequencing and IPPS problems. It is possible to improve the algorithm efficiency and computation results by using hybrids of the algorithms introduced in this thesis.

(6) Interfacing with NC program

The research can be extended to interface with NC systems to cover toolpath generation and generate the NC code which can be directly used on machines.

(7) Expand the IPPS module into the 5-axis environment

With the current methodology introduced in the 3-axis CAPP system and IPPS module, it can be easy to expand the IPPS module into the 5-axis environment with expanded representations for process plans and schedules and set-up determination methods.

(8) More flexible CAPP and IPPS systems.

The current system can be made more flexible with a distributed organisation. For a very complex component, it is possible to input the feature information collaboratively by different users on different computer at the same time (it is similar to collaborative design). The IPPS system should respond to the changed situations in real time so a distributed real-time structure should be applied.

(9) Application of the PSO to other Manufacturing problems

The results presented in this thesis have shown the merits of optimisation using the PSO algorithm compared with GA and SA. The application within this thesis has concentrated on one small area of the manufacturing paradigm. Other opportunities exist for the application of the PSO to increase the efficiency of search optimisation, such as supply chain optimisation and buffer size optimisation etc.

References:

1. **Aldakhilallah K.A.** and **Ramesh R.**, Computer-Integrated Process Planning and Scheduling (CIPPS): intelligent support for product design, process planning and control, International Journal of Production Research, 1999, vol. 37, No. 3, 481-500.
2. **Baptista R.** and **Antune Simoes J.F.**, Three and five axes milling of sculptured surfaces, Journal of Materials Processing Technology, 2000, 103, 398-403.
3. **Bhandarker P.M** and **Nagi R.,** "STEP-based feature extraction from STEP geometry for Agile Manufacturing", Computers in Industry, Vol. 41, pp. 3-24, 2000.
4. **Bhaskara Reddy S.V., Shunmugam M.S.** and **Narendran T.T.**, Operation sequencing in CAPP using genetic algorithms, 1999, International Journal of Production Research, Vol.37, No.5, 1063-1074.
5. **Bohez E.L.J.**, Five-axis milling machine tool kinematic chain design and analysis, International Journal of Machine Tools and Manufacture, 2002, 42, 505-520.
6. **Brandimarte, P.** and **Calderini, M.**, A hierarchical bicriterion approach to integrated process plan selection and job shop scheduling. 1995, International Journal of Production Research, 33(1), pp. 161-181.
7. **Bronsvoort W.F. and Jansen F.W.**, "Feature modelling and conversion - key concepts to concurrent engineering", 1993, Computers in Industry, Vol. 21, pp. 61-86.

8. **Brown, K.** and **Cagan, J.**, Optimised process planning by generative simulated annealing, 1997, Artificial Intelligent for Engineering Design, Analysis and Manufacturing, 11, 219-235.

9. **Bryman A**, Research Methods and Organisational Studies, 1989, Unwin Hyman: London pp 230-233.

10. **Burrell G** and **Morgan G**, Sociological Paradigms and Organisational Analysis, 1979, Heineman: London, pp 2-9.

11. **Cagnina L.**, **Esquivel S.** and **Gallard R.**, Particle Swarm Optimization for Sequencing Problems: A Case Study, 2004, CEC2004, 1, 536-541.

12. **Carpenter, I.D.** and **Maropoulos, P.G.**, A flexible tool selection decision support system for milling operations, 2000, Journal of Materials Processing Technology, 107, pp 143-152.

13. **Case. K.** and **Acar B.S.**, The manufacturing features approach to design and its impact on the learning and use of Computer Aided Design systems, Contemporary Ergonomics, Taylor and Francis, London, 1989.

14. **Case, K.**, **Gao, J.X.** and **Gindy, N.N.Z.**, Implementation of a feature-based component representation for CAD/CAM integration. 1994, Proc. Instn Mech. Engrs, Part B, 208(B1), 71-80.

15. **Case K.** and **Harun W.A. Wan**, Feature-based representation for manufacturing planning, 2000, International Journal of Production Research, Vol. 38, No.17, 4285-4300.

16. **Chang P.T.** and **Chang C.H.**, An integrated artificial intelligent computer aided process planning system, 2000 International Journal of Computer Integrated Manufacturing, Vol. 13, No. 6, 483–497

17. **Chang T.C.** and **Wysk R.A.**, An introduction to Automated Process Planning

Systems, 1985, Prentice-Hall Inc., Englewood, Cliffs, New Jersey, USA.

18. **Chang T.C.**, "Expert Process Planning for Manufacturing", 1990, Addison-Wesley Publishing Company.
19. **Chang, T.C.**, **Wysk, R.A.**, and **Wang, H.P.**, 1998, Computer-Aided Manufacturing (Second Edition), Prentice-Hall.
20. **Chatting Ian**, An attribute management process to enhance concurrent engineering in the aerospace industry, 2001, PhD thesis.
21. **Chen, J.**, **Zhang, Y.F.** and **Nee, A.Y.C.**, Set-up planning using Hopfield Net and Simulated Annealing, 1998, International Journal of Production Research, 36, 981-1000.
22. **Chep A** and **Tricarico L.**, Object-oriented analysis and design of a manufacturing feature representation, 1999, International Journal of Production Research, Vol.37, No.10, 2349-2376.
23. **Chu, C.C.P.**, and **Gadh, R.**, Feature-based approach for set-up minimization of process design from product design. 1996, Computer-Aided Design, 28(5), 321-332.
24. **Clarke P**, Action Research and Organisational change, 1972, Harper & Row: London, pp6-25.
25. **Clerc M.**, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. Proceedings of the Congress on Evolutionary Computation, Washington, DC, 1999, 1951-1957.
26. **Deneux, D.**, **Maranzana, R.** and **Soenen, R.**, The material angle: a part-level criterion for tool-feature extraction, 1994, Proceedings of the IFIP International Conference on Feature Modeling and Recognition in Advanced CAD/CAMSystems, Valenciennes, France.

27. **Ding L.**, Feature Technology and its Applications in Computer Integrated Manufacturing, PhD thesis, 2003.

28. **Ding L.**, **Yue Y.**, **Ahmet K.**, **Jackson M.** and **Parkin R.**, Global optimization of a feature-based process sequence using GA and ANN techniques, 2005, International Journal of Production Research, 43(15), 3247-3272.

29. **Du, P.**, and **Huang, N.-K.**, 1990, Principles of Computer-Aided Process Planning (Beijing: Beijing University of Aeronautics & Astronauts Press).

30. **Faheem, W. Hayes, C.C. Castano, J.F.** and **Gaines, D.M.**, What is a manufacturing interaction? 1998, Proceedings of 1998 ASME Design Engineering Technical Conferences, Atlanta, pp. 1-6.

31. **Fu M.W.**, **Ong S.K.**, **Lu W.F.**, **Lee I.B.H.** and **Nee A.Y.C.**, An approach to identify design and manufacturing features from a data exchanged part model, 2003, Computer-Aided Design, 35, 979-993.

32. **Gao, J. X.** and **Case, K.**, Information mapping between a feature-based design system and an integrated process planning system. 1993, Proceedings of the 30th International Matador Conference, Manchester, UK, pp. 551-558.

33. **Gindy, N.N.Z.**, A hierarchical structure for form features. 1989, International Journal of Production Research, 27(12), 2089-2103.

34. **Gonzalez F.** and **Rosado P.**, General and flexible methodology and architecture for CAPP: GF-CAPP system, 2003, International Journal of Production Research, vol.41, no.12, 2643-2662.

35. **Groover M.P.**, Fundamentals of Modern Manufacturing, 2002, 2nd Edition, John Wiley and Sons, INC.

36. **Gu P.**, **Balasubramanian S.** and **Norrie D. H.**, Bidding-Based Process Planning and Scheduling in A Multi-Agent system, 1997, Computers and Industry

Engineering, Vol.32, No.2, pp. 477-496.

37. **Han J.H., Kang M. and Choi H.,** "STEP-based feature recognition for manufacturing cost optimization", Computer-Aided Design, Vol. 33, pp. 671-686, 2001.

38. **Ho M.C.**, **Hwang Y.R.**, Machine codes modification algorithm for five-axis machining, Journal of Materials Processing Technology, 2003, 142, 452–460.

39. **Howard K**, and **Peters J.** Managing Management Research, Management Decision, 1990, MCB University Press, 28(5).

40. **Huang, S.H.**, **Zhang, H.C.** and **Smith, M.L.**, A progressive approach for the integration of process planning and scheduling. IIE Trans., 1995, 1, 456-464.

41. **Irani, S.A.**, **Koo, H.Y.** and **Raman, S.**, Feature-based operation sequence generation in CAPP. 1995, International Journal of Production Research, 33, 17-39.

42. **Jerald J.**, **Asokan P.**, **Prabaharan G.** and **Saravanan R.**, Scheduling optimization of flexible manufacturing systems using particle swarm optimization algorithm, 2005, International Journal of Advanced Manufacturing Technology, 964-971.

43. **Johnson G** and **Johnson P.**, Research Methods for Managers, 1991, Sage.

44. **Jong De**, **F.**, **Reitsema, T.** and **Hoogeboom, R.**, Feature-based modelling: A way to improve the CAD/CAM-process. 1992, Revue Internationale de CFAO et d'Infographie, **7**, 209- 234.

45. **Kalpakjian, S.**, Manufacturing Engineering and Technology, 3rd edn, 1995, Addison-Wesley.

46. **Kang M**, **Han J**, and **Moon J.G.**, An approach for interlinking design and process planning, Journal of Materials Processing Technology, 2003, 139, 589-595.

47. **Karinthi, R.**, **Nau, D.**, and **Yang, Q.**, Handling feature interactions in process-planning. 1992, Applied Artificial Intelligence, 6, 389-415.

48. **Kempenaers J.** and **Pinte J.**, A collaborative process planning and scheduling system, Advances in Engineering Software, 1996, 25, 3-8.

49. **Kennedy J.** and **Eberhart R.** "Particle Swarm Optimization", 1995, Proceedings of the IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Centre, Piscataway, NJ, IV: 1942-1948.

50. **Kim, Y.K.**, **Park, K.** and **Ko, J.**, A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. 2003, Computers & Operations Research, 30, pp. 1151-1171.

51. **Kim Y.S.**, **Wang E.** and **Rho H.M.**, Geometry-based machining precedence reasoning for feature-based process planning, 2001, International Journal of Production Research, Vol. 39, No.10, 2077-2103.

52. **Kruth, J.P.**, and **Detand, J.**, A CAPP system for nonlinear process plans. 1992, Annals of the CIRP, 41(1), pp. 489-492.

53. **Kumar, M.** and **Rajotia, S.**, Integration of scheduling with computer aided process planning. 2003, Journal of Materials Processing Technology, 138, pp. 297-300.

54. **Kuhn T S**, The structure of Scientific Revolutions, 1962A, University of Chicago Press: Chicago, pp 10.

55. **Kuhn T S**, The structure of Scientific Revolutions, 1962B, University of Chicago Press: Chicago, pp 103.

56. **Lee D.H.**, **Kiritsis D.** and **Xirouchakis P.**, Branch and fathoming algorithms for operation sequencing in process planning, 2001, International Journal of Production Research, Vol.39, No.8, 1649-1669.

57. **Lee D.H.**, **Kiritsis D.** and **Xirouchakis P.**, Search heuristics for operation sequencing in process planning, 2001, International Journal of Production Research, Vol.39, No.16, 3771-3788.

58. **Lee J.Y. and Kim K.**, "Generating alternative interpretations of machining features", International Journal of Advanced Manufacturing Technology, Vol. 15, pp. 38-48, 1999

59. **Li, W.D.** and **McMahon, C.A.**, A Simulated Annealing-based Optimization Approach for Integrated Process Planning and Scheduling, 2006, International Journal of Computer Integrated Manufacturing.

60. **Li W.D**, **Ong S.K** and **Nee A.Y.C**, Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts, 2002, International Journal of Production Research, Vol.40, No.8, 1899-1922.

61. **Li W.D.**, **Ong S.K.** and **Nee A.Y.C.**, optimization of process plans using a constraint-based tabu search approach, 2004, International Journal of Production research, vol. 42, No. 10, 1955-1985.

62. **Lin C.-J.** and **Wang, H.-P.**, Optimal operation planning and sequencing: minimization of tool changeovers. 1993, International Journal of Production Research, 31, 311-324.

63. **Link C.H.**, CAPP CAM-I automated process planning system, Proc. Of the 1976 NC Conference, CAM-I Inc, Arlington, Texas, USA, 1976.

64. **Lo C.C.**, Real-time generation and control of cutter path for 5-axis CNC machining, International Journal of Machine Tools & Manufacture, 1999, 39, 471-488.

65. **Ludema, K.C.**, **Caddell, R.M.** and **Atkins, A.G.**, Manufacturing Engineering – Economics & Processes, 1987, Prentice-Hall International.

66. **McMahon, C.A.** and **Browne, J.**, CAD/CAM – Principles, Practice and Manufacturing Management (2rd Edition), 1998, Addison-Wesley.

67. **Ma G. H.**, **Zhang Y. F.** and **Nee A. Y. C.**, A simulated annealing-based optimization algorithm for process planning, 2000, International Journal of Production Research, Vol.38, No.12, 2671-2687.

68. **Mahbubur R.MD.**, **Heikkala J.**, **Lappalainen K.**, and **Karjalainen J.A.**, Positioning accuracy improvement in five-axis milling by post processing, International Journal of Machine Tools & Manufacture, 1999, 37(2), 223-226.

69. **Tolouei-Rad M.**, An efficient algorithm for automatic machining sequence planning in milling operations, 2003, International Journal of Production Research, Vol. 41, No.17, 4115-4131.

70. **Mantyla M.**, **Nau D.S.** and **Shah, J.J.**, Challenges in feature-based manufacturing research. CACM, 1996, 39(2), 85-103.

71. **Maropoulos, P.G.** and **Baker, R.P.**, Integration of tool selection with design (Part 1: Feature creation and selection of operations and tools, 2000, Journal of Material Processing Technology, 107, pp 127-134.

72. **Maropoulos P.G.**, **Baker R.P.** and **Paramor K.Y.G.**, Integration of tool selection with design (Part 2. Aggregate machining time estimation), 2000, Journal of materials processing technology, 107, 135-142.

73. **Maturana F., Shen W.** and **Norrie D.H**, MetaMorph: an adaptive agent-based architecture for intelligent manufacturing, 1999, International Journal of Production Research, Vol.37, No.10, 2159-2173.

74. **Mebrahtu H., Walker R.** and **Mileham T.**, Manufacturing simulation optimization: an expert mechanism approach, 2005, PhD thesis.

75. **Moon C.** and **Seo Y.**, Evolutionary algorithm for advanced process planning and scheduling in a multi-plant, 2005, Computers and Industrial Engineering, 48, 311-325.

76. **Morad, N.** and **Zalzala, A.**, Genetic algorithms in integrated process planning and scheduling. 1999, Journal of Intelligent Manufacturing, 10, pp. 169-179.

77. **Niebel, B.W.**, "Mechanized process selection for planning new design", 1965, ASME paper, No 737.

78. **Oberg E., Jones, Franklin D., Horton, Holbrook L.,** and **Ryffel H. H.**, Edition Machinery's Handbook, 27^{th}, 2004, Industrial Press Inc., New York.

79. **Ong, S.K., Li, W.D.**, and **Nee, A.Y.C.**, STEP-based integration of feature recognition and design-by-feature for manufacturing applications in a concurrent engineering environment, 2001, International Journal of Computer Applications in Technology – Special Issue on Applications in Industry of Product and Process Modelling Using Standards.

80. **Onwubolu G.C.** and **Clerc M.**, Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization, International Journal of Production Research, 2004, 42(3), 473-491.

81. **Owen J**, STEP: An Introduction, Information Geometers, UK, 1993.

82. **Pang W., Wang K.P, Zhou C.G.** and **Dong L.J.**, Fuzzy Discrete Particle Swarm Optimization for Solving Travelling Salesman Problem, Proceedings of

the Fourth International Conference on Computer and Information Technology (CIT'04), 2004.

83. **Patil L.** and **Pande S. S.**, An intelligent feature-based process planning system for prismatic parts, 2002, International Journal of Production Research, Vol. 40, No.17, 4431-4447.

84. **Philips E. M.** and **Pugh D. S.**, How to get a Ph.D., 1987, Milton Keynes, Open University, pp 45.

85. **Qiao L.**, **Wang X.-Y.** and **Wang S.-C.**, A GA-based approach to machining operation sequencing for prismatic parts, 2000, International Journal of Production Research, Vol.38, No.14, 3283-3303.

86. R**embold U.**, **Nnaji B.O.** and **Storr A.**, Computer Integrated Manufacturing and Engineering, 1993, Addison-Wesley Publishing Company.

87. **Requicha A.A.G.** and **Vandenbrande J.**, Automated Systems for Process Planning and Part Programming, IFS Publication Ltd., Springer-Verlag, 1988.

88. **Rose G.**, Deciphering Sociological Research, 1982, Macmillan: London, pp 14-18.

89. **Rustom E.A.**, BEPPS-GSCAPP Generative System of Computer Aided Process Planning for Prismatic Components, 1992, PhD Thesis, Bath University.

90. **Rustom E.A.** and **Mileham A.R.**, The Development of a Generative System of Computer Aided Process Planning for Prismatic Components, 1989, Proceedings of the 5th National Conference on Production Research, pp259-263, Huddersfield, UK.

91. **Rustom E.A.** and **Mileham A.R.**, Feature Ordering and Operation Sequencing for Automated Process Planning, 1990, Proceedings of the 6th National Conference on Production Research, pp259-263, Glasgow, UK.

92. **Rustom E.A.** and **Mileham A.R.**, Automated Decision Making for Process Planning Systems, 1994, Proceedings of the Joint International Conference FAIM 92, Flexible Automation and Information Management, Virginia, USA.

93. **Sadaiah, M.**, **Yadav, D.R.**, **Mohanram, P.V.**, **Radhakrishnan, P.**, A generative computer-aided process planning system for prismatic components, International Journal of Advanced Manufacturing Technology, 2002, v20, n10, p709-719.

94. **Sandvik**, Rotating Tools Catalogue, Sandvik Cormant UK, Manor West, Halesowen, West Midlands B62 8QZ, 1998.

95. **Scallan P.**, Process planning: the design/manufacture interface, 2003, Butterworth Heinemann.

96. **Schaffer G.**, GT via Automated Process Planning, American Machinist, pp119-122, 5/1980.

97. **Schenk, D.E.**, 1966, "Feasibility of automated process planning", American Machinist, May, Vol.124, pp119-122.

98. **Schey, J.A.**, Introduction to Manufacturing Processes, 2nd edn, 1987, McGraw-Hill.

99. **Seco**, Milling Catalogue and Technical Guide, Seco Tools UK, Alcester, Warwickshire B49 6EL, 1998.

100. **Shah, J.J.** and **Mantyla M.,** Parametric and Feature-based CAD/CAM, 1995 (Wiley Interscience, New York)

101. **Shah J.J.** and **Rogers M.T.**, Expert form feature modelling shell. 1988, Computer Aided Design, 20, 515-524.

102. **Sharma R.** and **Gao J. X.**, Implementation of STEP Application Protocol 224 in an automated manufacturing planning system, 2002, Proc Instn Mech

Engrs, Vol 216 Part B: J Engineering Manufacture.

103. **Shpitalni M.** and **Fisher A.**, CSG representation as a basis for extraction of machining features, 1991, Annals of the CIRP, 40, 157-160.

104. **Sormaz D.** and **Khoshnevis B.**, Generation of alternative process plans in integrated manufacturing systems. 2003, Journal of Intelligent Manufacturing, 14, pp. 509-526.

105. **Stacey A.**, **Jancic M.** and **Grundy I.**, Particle Swarm Optimization with Mutation, 2003, The 2003 Congress on Evolution Computation, 2, 1425-1430.

106. **Stellram**, Stellram Milling Program, Stellram, Hercules Way, Bowerhill Industrial Estate, Wiltshire SN12 6TS, 1998.

107. **STEP Tools Inc.**, http://www.steptools.com.

108. **STEP**, STandard for External representation of Product data, ISO 10303-224 (Industrial automation systems and integration. Product data representation and exchange: mechanical product definition for process plans using machining features), 1999.

109. **Sun J.**, **Zhang Y.F.** and **Nee A.Y.C.**, A distributed multi-agent environment for product design and manufacturing planning, 2001, International Journal of Production Research, Vol.39, No.4, 625-645.

110. **Tan, W.** and **Khoshnevis, B.**, Integration of process planning and scheduling – a review. 2000, Journal of Intelligent Manufacturing, 11, pp. 51-63.

111. **Tonshoff, H.K.**, **Beckendorff, U.** and **Andres N.**, FLEXPLAN: A concept for intelligent process planning and scheduling. 1989, Proceedings of the CIRP International Workshop, Hannover, Germany, pp. 319-322.

112. **Trafford**, An Unravelling of Research Methodology, 2001, Earlybrave Publications Ltd., pp 7-12.

113. **Tseng Y.J. and Joshi S.B.**, "Recognition of interacting rotational and prismatic machining features from 3-D mill-turn parts", International Journal of Production Research, Vol. 36, No. 11, pp. 3147-3165, 1998.

114. **Usher, J.M.** and **Bowden, R.O.**, The application of genetic algorithms to operation sequencing for use in computer-aided process planning, 1996, Computers and Industrial Engineering, 30(4), 999-1013.

115. **Usher, J.M.**, Evaluating the impact of alternative plans on manufacturing performance. 2003, Computers & Industrial Engineering, 45, pp. 585-596.

116. **Walton R.** and **Gaffney M.**, Innovating to Complete: Lessons for Diffusing and Managing Change in the Work Place, 1991, Jossey-Bass.

117. **Wang, H-.P.** and **Chang, H.**, "Automated classification and coding based on extracted surface features in a CAD data", 1987, International Journal of Advanced Manufacturing technology, Vol.2, No.1, pp25-38.

118. **Wang, H-.P.**, and **Wysk, R.A.**, "AIMSI: a prelude to a new generation of integrated CAD/CAM systems", 1988, International Journal of Production Research, Vol.26, No.1, pp119-131.

119. **Wang K.P.**, **Huang L.**, **Zhou C.G.** and **Pang W.**, Particle Swarm Optimization for Travelling salesman problem, 2003, Proceedings of the Second International Conference on Machine Learning and Cybernetics.

120. **Wang L.H.** and **Shen W.M.**, DPP: An agent-based approach for distributed process planning, 2003, Journal of Intelligent Manufacturing, 14, 429-439.

121. **Wong T.N.**, **Leung C.W.**, **Mak K.L.** and **Fung R.Y.K.**, An agent-based negotiation approach to integrate process planning and scheduling, International Journal of Production Research, 2006, vol. 44, No. 7, 1331-1351.

122. **Wong, T.N.**, and **Siu, S.L.**, A knowledge-based approach to automated machining process selection and sequencing. 1995, International Journal of Production Research, 33(12), 3465-3484.

123. **Wong T.N.** and **Wong K.W.**, A feature-based design system for computer-aided process planning, Journal of Materials Processing Technology, 1995, 52, 122-132.

124. **Wu R.R.**, **Ma L.**, **Mathew J.** and **Duan G.H.**, Optimal operation planning using fuzzy Petri nets with resource constraints, 2002, International Journal of Computer Integrated Manufacturing, vol.15, no.1, 28-36.

125. **Yan, H.S.**, **Xia, Q.F.**, **Zhu, M.R. Liu, X.L.** and **Guo, Z.M.**, Integrated production planning and scheduling on automobile assembly lines. 2003, IIE Transactions, 35, pp. 711-725.

126. **Yang Y.-N**, **Parsaei H.R.** and **Leep H.R.**, A prototype of a feature-based multiple-alternative process planning system with scheduling verification, computers & industrial engineering, 2001, 39, 109-124.

127. **Yin R.**, Applications of Case Study Research, 1993, Sage.

128. **Yin R.**, Case Study Research, Design and Methods, 1994, Sage.

129. **Yip-Hoi D. and Dutta D.**, "A genetic algorithm application for sequencing operations in process planning for parallel machining", IIE Transactions, 1996, Vol. 28, No. 1, pp. 55-68.

130. **Zhang F**, **Zhang Y.F** and **Nee A.Y.C.**, Using genetic algorithms in process planning for job shop machining, 1997, IEEE Transactions on evolutionary computation, Vol.1, No.4.

131. **Zhang H.** and **Mallur S.**, An integrated model of process planning and production scheduling. 1994, International Journal of Computer Integrated Manufacturing, 7(6), pp. 356-364.

132. **Zhang H.** and **Mallur S.**, An integrated model of process planning and production scheduling. 1994, International Journal of Computer Integrated Manufacturing, **7**(6), pp. 356-364.

133. **Zhang X.D.** and **Yan H.S.**, Integrated optimization of production planning and scheduling for a kind of job-shop. 2005, International Journal of Advanced Manufacturing Technology, in press.

134. **Zhang Y.F.**, **Saravanan A.N.** and **Fuh J.Y.H.**, Integration of process planning and scheduling by exploring the flexibility of process planning. 2003, International Journal of Production Research, 41(3), pp. 611-628.

135. **Zhang Y.** and **Mileham A.R.**, An Automated Generator of NC Part Programs for Rotational Parts from CAD Product Model, 1991, Proceedings of the 7th National Conference on Production Research, pp11-15.

136. **Zhang Y.Y.**, **Feng S.C.**, **Wang X.K.**, **Tian W.S.** and **Wu R.R.**, Object oriented manufacturing resource modeling for adaptive process planning, 1999, International Journal of Production Research, Vol. 37, No.18, 4179-4195.

137. **Zhang Y.F.**, **Nee A.Y.C.**, and **Fuh J.Y.H.**, Hybrid approach to computer-aided process planning for prismatic parts. 1994, Computers in Engineering, Proceedings of the International Conference v/l, ASME, New York, USA, pp. 437-442.

138. **Zhang Y.** and **Mileham A.R.**, BEPPS-NC: A rule based expert system for process planning, 1989, Proceedings of the 5th National Conference on Production Research, pp333-357, UK.

139. **Zhang Y.** and **Mileham A.R.**, A CAD Interpreter for interfacing CAD and CAPP of 2-D Rotational Parts, 1991, Proceedings of the 6th International Conference on CAD/CAM, Robotics and Factories of the Future, UK.

140. **Zijm W.H.M.**, The integration of process planning and shop floor scheduling in small batch part manufacturing. 1995, Annals of CIRP, 44, pp. 429-432.

Related Publication

Yanwu Guo, **A R Mileham, G W Owen** and **W D Li**, Operation Sequencing Optimisation using a Particle Swarm Optimisation Approach, 2006, Proceedings of the Institution of Mechanical Engineers, Vol. 220, Part B, Journal of Engineering Manufacture, 220(B12), 1945-1958.