International Journal of Applied Management and Technology
2011, Volume 10, Issue 1, Pages 16–33
©Walden University, Minneapolis, MN
DOI: 10.5590/IJAMT.2011.10.1.16

WALDEN UNIVERSITY
A higher degree. A higher purpose.

Identification of Patterns in Genetic-Algorithm-Based Solutions for Optimization of Process-Planning Problems Using a Data Mining Tool

Dowluru Sreeramulu
Gandhi Institute of Engineering and Technology

Chilakalapalli Surya Prakasa Rao
National Institute of Technology

This paper presents a novel use of data mining algorithms for extraction of knowledge from a set of process plans. The purpose of this paper is to apply data mining methodologies to explore the patterns in data generated by genetic-algorithm-generating process plans and to develop a rule set planner, which helps to make decisions in odd circumstances. Genetic algorithms are random-search algorithms based on the mechanics of genetics and natural selection. Because of genetic inheritance, the characteristics of the survivors after several generations should be similar. The solutions of a genetic algorithms for process planning consists of the operation sequence of a job, the machine on which each operation is performed, the tool used for performing each operation, and the tool approach direction. Among the optimal or near-optimal solutions, similar relationships may exist between the characteristics of the operation and sequential order. Data mining software known as See5 has been used to explore the relationship between the operation's sequence and its attributes, and a set of rules has been developed. These rules can predict the positions of operations in the sequence of process planning.

Keywords: *data mining, genetic algorithms, process planning*

Introduction

Process planning is an engineering task that determines the detailed manufacturing requirements for transforming a raw material into a completed part, within the available machining resources. The output of process planning generally includes operations, machine tools, cutting tools, fixtures, and machining parameters, among others. This paper presents a process-planning problem for a part is modeled in a network by simultaneously considering the selection of operations, machines, cutting tools, and operations sequence, as well as the constraints imposed by the precedence relationships between operations and available machining resources.

In recent years, information growth has proceeded at an explosive rate. While database management systems provide us with basic tools for the effective, efficient storage and lookup of large data sets, the capabilities for collecting and storing data have outpaced our abilities to analyze, summarize, and extract knowledge from this data. Traditional methods of data analysis were based mainly on direct human dealings with data. Large volumes of data overwhelm the traditional manual methods of data analysis, such as spreadsheets and ad-hoc queries, and while informative reports can be produced through these methods, they cannot further analyze the content of those

Please address queries to: Dowluru Sreeramulu, Assistant Professor, Department of Mechanical Engineering, Gandhi Institute of Engineering and Technology, Gunpur, India. Email: sreeram_india9@yahoo.com

reports. These methods help only in data collecting and computing; they do not assist in improving the analysis task. Moreover, they overemphasize the statistical aspects of data while ignoring the domain knowledge of data. As a result, traditional analysis can fail to reveal the physical natures that the data implies. The modern answer is data mining, which is being used both for analysis and to predict the future pattern. Data mining is a step in a long process chain of data analyzing that involves the evaluation and interpretation of patterns to determine what constitutes knowledge. In this paper, a See5 data mining tool (RuleQuest software, Australia) is used to induct a rule set from a classified data.

Literature Review

Literature on computer-aided process planning (CAPP) is vast, and a considerable amount of work on CAPP has been carried out over the last few decades. In the past, Alting and Zhang (1989); Steudel (1984); Weill, Spur, and Eversheim (1982); and Cay and Chassapis (1997) performed extensive studies on CAPP.

Zhang, Zhang, & Nee (1997) presented an approach that deals with process-planning problems using genetic algorithms (GAs, which are random-search algorithms based on the mechanics of genetics and natural selection) in a concurrent manner in generating the entire solution space by considering the multiple decision-making activities—i.e., operation selection, machine selection, setup selection, cutting tool selection, and operations sequencing—simultaneously. GAs are selected due to their flexible representation scheme. Specially designed crossover and mutation operators are used to get a near-optimal process plan. A space search method is used for comparison.

In a distributed manufacturing environment, factories possessing various machines and tools at different geographical locations are often combined to achieve the highest production efficiency. When jobs requiring several operations are received, feasible process plans are produced by those factories available. Li, Fuh, Zhang, and Nee (2005) used GA to get optimal or near-optimal process plans for a single manufacturing system, as well as distributed manufacturing systems. It is shown from the case study that the approach is comparative with or better than the conventional single-factory CAPP. The applications of GA for different types of process-planning problems were addressed by Goldberg & Lingle (1985), Goldberg (1989), Bruns and Forrest (1993), Falkenauer and Delchambre (1992), Dagli and Sittisathanchai (1993), Zhang (1997), and Dereli and Filiz (1999).

A new generation of techniques and tools are required to assist humans in intelligently analyzing voluminous data for pieces of useful knowledge. Knowledge discovery in databases and data mining integrate database management systems and artificial intelligence technologies to assist humans in analyzing large quantities of data. Knowledge discovery in databases is defined as the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad, 1997).

Koonce and Tsai (2000) presented a novel use of data mining algorithms for the extraction of knowledge from a large set of job shop schedules. They applied data mining methodologies to explore the patterns in data generated by a GA performing a scheduling operation and to develop a rule set scheduler that approximates the GA's scheduler. In using a GA for job shop scheduling, the solution is an operational sequence for resource allocation. Among the optimal or near-optimal solutions, similar relationships may exist between the characteristics of operations and sequential order. An attribute-oriented induction methodology was used to explore the relationship between an operation's sequence and its attributes, and a set of rules has been developed. These rules can duplicate the GA's performance on an identical problem.

Harrath, Chebel-Morello, and Zerhouni (2002) proposed a new method based on GA and data mining to resolve job shop scheduling problems. The developed GA generates a learning population of good solutions, which are mined by the mean of See5 classifier systems. The mining step gives decision rules that are transformed into a metaheuristic that allows the sequence of operations on the machine. Harrath, Chebel-Morello, and Zerhouni (2001) and Taaj (1997) addressed a job shop scheduling problem with the use of data mining and GAs.

Yang and Webb (2002) argue that the requirements for effective discretization differ between naïve Bayes learning and many other learning algorithms. They evaluate the effectiveness with naïve Bayes classifiers of nine discretization methods: equal width discretization, equal frequency discretization, fuzzy discretization, entropy minimization discretization, iterative discretization, proportional k-interval discretization, lazy discretization (LD), nondisjoint discretization (NDD), and weighted proportional k-interval discretization (WPKID). The authors say that, in general naïve Bayes classifiers trained on data preprocessed by LD, NDD, or WPKID achieve lower classification error than those trained on data preprocessed by the other discretization methods, but LD cannot scale to large data. This study led to a new discretization method, weighted nondisjoint discretization, which combines the advantages of WPKID and NDD.

Background

In process planning, a part is generally described by features, which are geometric forms having machinable shapes, such as holes, slots, and bosses. Given a part and a set of manufacturing resources, the process-planning problem can be defined as follows: (Zhang, Ma, & Nee, 1999).

i) **Operation selection**: For each feature, determine one or several operations required. This includes the selection of machines, cutting tools, and tool approach directions (TADs) based on the feature geometry and available machining resources.

ii) **Operation sequencing**: Determine the sequence of executing all operations required for the part so that the precedence relationships (PRs) among all the operations are maintained

The decision-making tasks in i) and ii) above must be carried out simultaneously to achieve an optimal plan against a predetermined evaluation criterion. Most of these systems, however, focus on generating the optimal plan for individual features. Recently, research has focused on process-planning optimization by considering some of the decision-making tasks concurrently. These efforts have undoubtedly achieved certain success; however, few CAPP systems have gained industry acceptance. CAPP systems based on the different criteria are as follows:

i) **Simple or limited machining environment**: Most reported CAPP systems are designed for handling planning tasks within a simple machining environment, such as a vertical milling center. To accommodate various machining environments in different companies and/or the change of machining capacity in the same company, a CAPP system must be able to handle different job shop environments.

ii) **Feature being the basic element**: Most existing CAPP systems use features as the basic elements for process planning. In practice, human planners use operations for process plans. The difference between using features and operations as the basic elements occurs when a feature needs two or more operations to be performed on different machines. In such a situation, the optimum plan in terms of minimum setups can never be reached if features are used as the basic planning elements.

iii) **Sequential decision-making**: Most existing approaches treat the various decision-making activities of process planning in a sequential manner. Although this strategy may reduce the solution space significantly, the optimum or even the feasible plans may well be lost on the way to the final solution.

When the various decision-making tasks are carried out simultaneously, process planning becomes a combinatorial problem. During the last decade, GAs have been applied to many such problems, including job shop scheduling, the travelling salesman problem (TSP is an NP-hard problem in combinatorial optimization studied in operations research and theoretical computer science), and other NP (i.e., nondeterministic polynomial time)-complete problems. A process-planning problem is similar to a traveling salesman problem in that every operation has to be traversed once and only once; although, a process-planning problem is more complicated due to precedence constraints among operations and nonfixed "distance" between operations (time required for machine, setup, and tool change). It is expected that GAs can provide a valid option for solving the process-planning problems so long as a suitable string representation and a corresponding search operator can be devised. Recently, there have been reports on applying GAs to process planning.

The Process Planning Model: A Network Representation

Given a part needing M-stages, ("machining stage" or M-stage is a general term for any OPT; a feature can then be represented by a fixed number of M-stages while each M-stage has its alternative sets of [M, T, TAD]. A part can also be represented in the similar fashion) the process-planning problem can be conveniently described by a network constrained by its PRs. The network consists of M-stages. Each M-stage consists of several combinations of machine (*M*), tool (*T*), and *TAD*.

There is a link between any two M-stages that represent the PR between them (i.e., the one that the arrow points to must be performed after the other, while a link with double arrows means that there is no PR between the two M-stages they connect). An M-stage network for a part that consists of five operations is depicted in Figure 1.

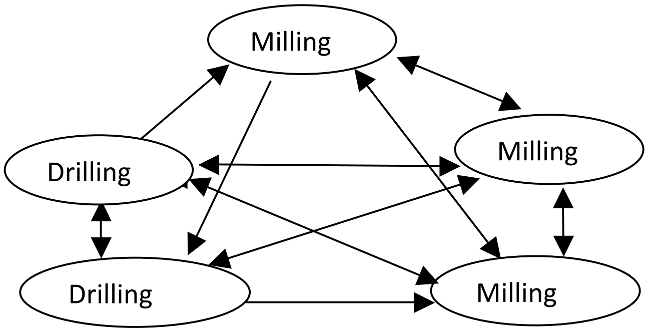


Figure 1: *Example M-Stage Network for a Part Consisting of Five Operations*

Flexible Process Plan Evaluation Criteria

The most commonly used criteria for process-plan evaluation include minimum number of setups, shortest process time, and minimum machining cost, among others. Because the detailed information on tool paths and machining time cannot be used for plan evaluation, the following five cost factors are identified as the plan evaluation criteria:

1. Machine cost (MC),

$$\mathrm{MC} = \sum_{i=1}^{n} \mathrm{MCI_i} \qquad \textit{Eq. 1}$$

where *n* is the total number of operations and $\mathrm{MCI_i}$ is the machine cost index for using machine *i,* a constant for a particular machine.

2. *Tool cost* (TC),

$$\mathrm{TC} = \sum_{i=1}^{n} \mathrm{TCI_i} \qquad \textit{Eq. 2}$$

where $\mathrm{TCI_i}$ is the tool cost index for using tool *i*, a constant for a particular machine.

3. *Machine change cost* (MCC; a machine change is needed when two adjacent operations are performed on different machines),

$$\mathrm{MCC} = \mathrm{MCCI} \sum_{i=1}^{n-1} \Omega(\mathrm{M}_{i+1}, \mathrm{M}_i) \qquad \textit{Eq. 3}$$

where MCCI is the machine change cost index and M_i is the identification of the machine used to performed operation *i.*

$$\Omega(x,y) = \begin{array}{l} 1,\ \text{if } x \neq y \\ 0,\ \text{if } x = y \end{array} \qquad \textit{Eq. 4}$$

4. *Setup change cost* (SCC; a setup change is needed when two adjacent operations performed on the same machine have different TADs),

$$\mathrm{SCC} = \mathrm{SCCI} \sum_{i=1}^{n-1} \{[1- \Omega(\mathrm{M}_{i+1}, \mathrm{M}_i)]*\Omega(\mathrm{TAD}_{i=1},\mathrm{TAD}_i)\} \qquad \textit{Eq. 5}$$

where SCCI is the setup change cost index, a constant.

5. *Tool change cost* (TCC; a tool change is needed when two adjacent operations performed on the same machine use different tools),

$$\mathrm{TCC} = \mathrm{TCCI} * \sum_{i=1}^{n-1} \{[1- \Omega(M_{i+1}, M_i)]*\Omega(T_{i+1}, T_i)\} \qquad \textit{Eq. 6}$$

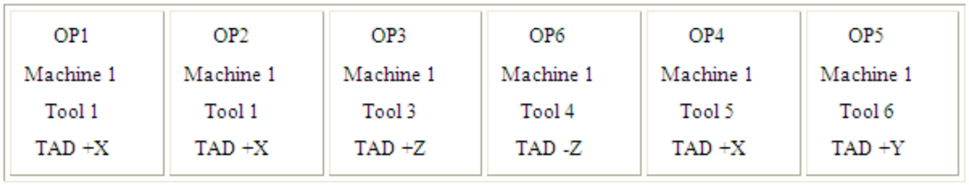
where TCCI is the tool change cost index, a constant.

These cost factors can be used either individually or collectively as a cost compound based on the actual requirement and data availability of the job shop. In summary, the process-planning problem can be rephrased as to identify a combination of M, T, and TAD from every M-stage and put them into an order that does not violate any PRs between any two M-stages, while achieving the least cost compound (cost compound is a function of the cost factors introduced earlier).

Applying GA to Process-Planning Problem

Knowledge-Based Representation of a Process Plan

The first step in formulating a GA for process planning is to map the problem solutions (process plans) to string representations. A knowledge-dependent string is used to represent a process plan. For an n-operation problem, the string is composed of n-segments. Each segment contains an M, T, and TAD from a unique M-stage and its order in the string. This representation is illustrated through an example of a six-operation problem, as shown in Figure 2. This string representation can cover all the solution space due to the selection of machine tools, cutting tools, TADs, and the sequence among operations.



OP1 Machine 1 Tool 1 TAD +X	OP2 Machine 1 Tool 1 TAD +X	OP3 Machine 1 Tool 3 TAD +Z	OP6 Machine 1 Tool 4 TAD -Z	OP4 Machine 1 Tool 5 TAD +X	OP5 Machine 1 Tool 6 TAD +Y

Figure 2: *A String Representing a Process Plan With Six Operations*

Generation of Initial Population

To generate the initial sequences of the operation that obey precedence constraints, an algorithm has been generated. The algorithm for generating a solution based on given PR is described as follows:

1. Select (at random) an operation with no predecessors.
2. From the remaining operations, select an operation at random such that it obeys the PR.
3. Repeat step 2 until all operations are completed.
4. Assign M, T, and TAD for each operation. (If any operation has more than one alternative, [e.g., OP1 can be done on machines M_1, M_2, and M_3], choose one of the machines at random.)
5. Repeat steps 1 through 4 until required initial population is achieved.

Fitness Evaluation

Once all solution strings are generated, the cost compound for the plan alternatives represented by these strings can be calculated using equations 1–6. Cost compound is used as the fitness of the solution string.

Reproduction

The present reproduction operator works in two steps. First, it applies "elitism" by copying the solution string having the lowest cost value, thus keeping the cost function nonincreasing. Second, it uses the "roulette wheel" method for the reproduction of the remaining string solutions.

Crossover

The strings obtained from reproduction are then mated at a given probability (crossover rate). To ensure that the crossover will not result in any violation of PRs and each operation in the offspring is executed only once, the cyclic crossover operator proposed by Dagli & Sittisathanchai (1993) is adopted. The algorithm for the crossover of string 1 and string 2 is described as follows:

1. Determine a cut point randomly from the all the positions of a string. Each string is then divided into two parts—the left side and the right side—according the cut point.
2. Copy the left side of string 1 to form the left side of offspring 1. The operator constructs the right side of offspring 1 according to the order of operations in string 2.
3. Copy the left side of string 2 to form the left side of offspring 2. The operator constructs the right side of offspring 2 according to the order of operations in string 1.

This process—cyclic crossover selecting a string pair with probability P_c randomly selecting a cut point—is illustrated in Figure 3. A pair of strings (parent 1 and parent 2) is under the crossover operation in which the cut point is chosen between positions 3 and 4. The left side of parent 1, *op4-op1-op2*, is used to form the left side of offspring 1. The order of the right side of parent 1, *op5-op7-op8-op3-op6-op9*, is adjusted according to the order of parent 2 to form the right side of offspring 1. By doing so, the sequences among the operations in both parent 1 and parent 2 are maintained in offspring 1. A similar operation is applied to parent 2 and parent 1 to form offspring 2.

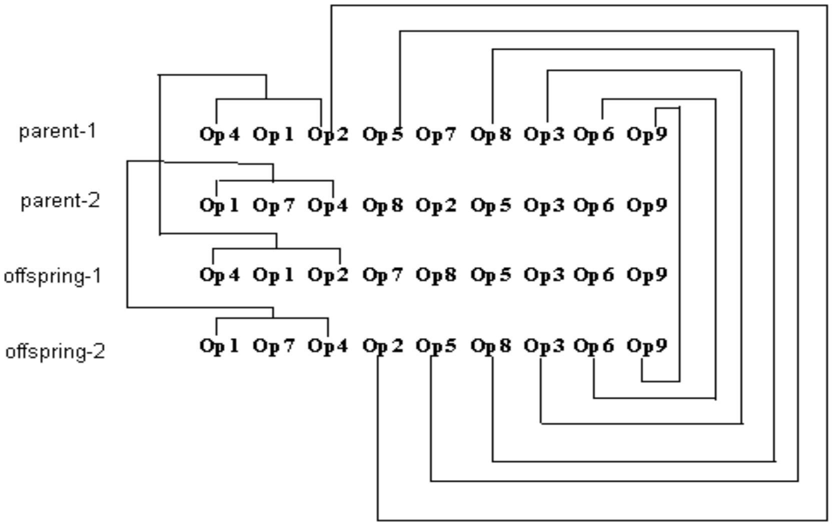

parent-1
Op4 Op1 Op2 Op5 Op7 Op8 Op3 Op6 Op9
parent-2
Op1 Op7 Op4 Op8 Op2 Op5 Op3 Op6 Op9
offspring-1
Op4 Op1 Op2 Op7 Op8 Op5 Op3 Op6 Op9
offspring-2
Op1 Op7 Op4 Op2 Op5 Op8 Op3 Op6 Op9

Figure 3: *An Example of Applying the Cyclic Crossover for Changing Operations Sequence*

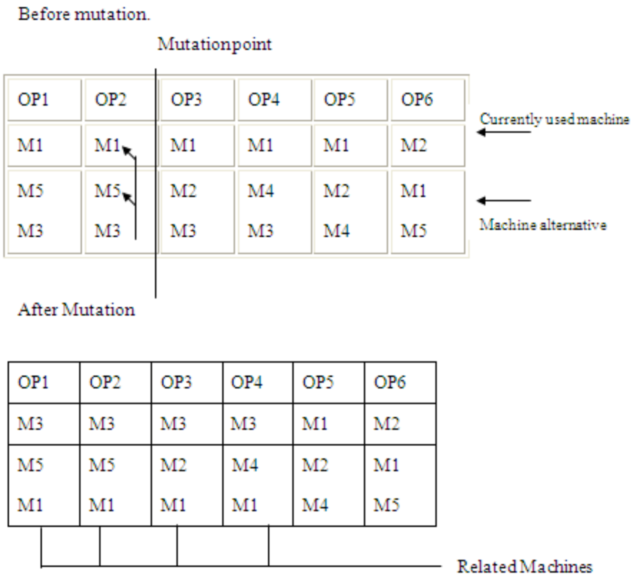
Mutation

Three mutation operations were developed in which the process-planning heuristics are employed: machine mutation, tool mutation, and TAD mutation.

Machine mutation is used to change the machine to perform an operation if more than one machine can be applied. To reduce the total number of machine changes, machine mutation does not stop at the selected position. Instead, the machine alternatives for every other operation are also checked to determine if a mutation can heuristically reduce machine changes. The algorithm for machine mutation is described as follows:

1. For every solution string, select an operation (a position in the string) randomly and use a predetermined probability (mutation rate) to determine whether or not the machine needs to be changed.
2. Randomly choose a machine (M_b) from all the alternatives to replace the current machine (M_a).
3. Identify all the other operations in the same string whose current machine is M_a. If any one of these operations has M_b as an alternative, assign M_b to replace M_a.

An example of this machine mutation is illustrated in Figure. 4. It can be seen that op3 (M_1) is selected for mutation where M_1 is the current machine. M_3 is then assigned to op3 to replace M_1. It is also found that M_1 is currently used by op1, op4, op5, and op2. Among them, op1, op4, and op5 have M_3 as one of their alternative machines; therefore, M_3 is assigned to op1, op4, and op5 to replace M_1.



OP1	OP2	OP3	OP4	OP5	OP6
M1	M1	M1	M1	M1	M2
M5	M5	M2	M4	M2	M1
M3	M3	M3	M3	M4	M5

OP1	OP2	OP3	OP4	OP5	OP6
M3	M3	M3	M3	M1	M2
M5	M5	M2	M4	M2	M1
M1	M1	M1	M1	M4	M5

Figure 4: *An Example of Machine Mutation With Size Operations*

Tool mutation operates on the solution strings after machine mutation. It has a similar mechanism. *TAD mutation* operates on the solution strings after both machine mutation and tool mutation. It also has a similar mechanism.

The settings of GA parameters for the example part (Figure 5) were as follows:

- *Population size:* 100
- *Crossover probability:* $P_c = 0.7$. In the present GA formulation, crossover is equivalent to a change of operation sequence that should therefore be vigorously performed to traverse more points in the solution space.
- *Mutation probability*: The three mutations play a similar role as the crossover since the extended solution space due to the availability of alternative machines, tools, and TADs must be adequately traversed in the optimizing process; therefore, the three mutation probabilities should be similar (e.g., $P_m = 0.35$).
- *Stopping criterion:* According to the observation, all the cases tested achieved very good results after 6,000 generations. The stopping criterion of 6,000 generations was therefore selected.

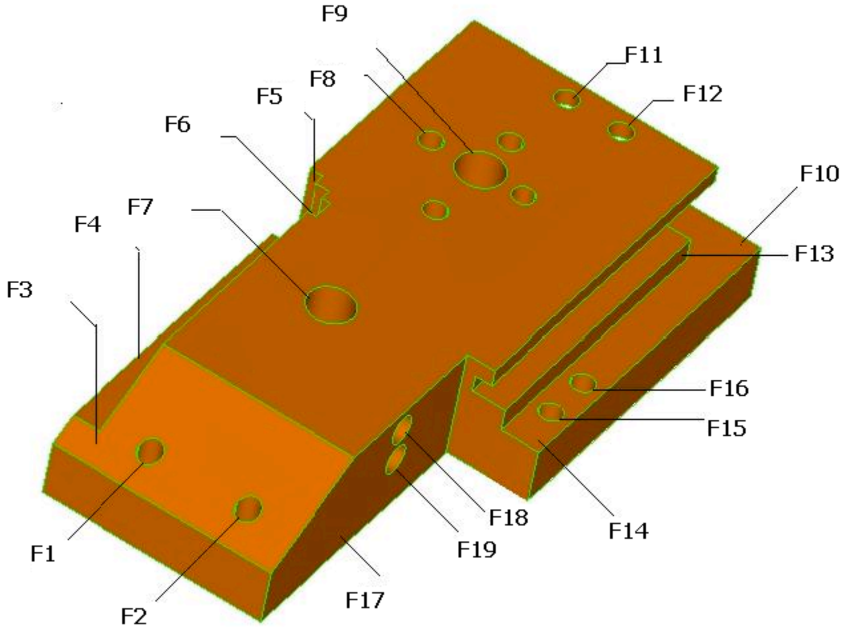


Figure 5: *A Prismatic Part and Its 19 Features*

Process Planning for a Sample Part

The authors have considered the same part, resources, and other constraints as those taken by Zhang, Zhang, and Nee (1997). Figure 5 shows a prismatic part and descriptions of its 19 features. Based off of Figure 5, the machining resources and the results of operations selection are given in Table 1, where columns 1 and 2 show the features and the operations to which they are mapped, column 3 shows the possible TADs for each operation, the machine alternatives for each operation are shown in column 4, and the tool alternatives for each operation are shown in column 5. Table 2 depicts machine, tool, and other cost indices for the example prismatic part. The PRs between the operations are then obtained as shown in Table 3.

Table 1: *The Operation Selection Results for the Example Prismatic Part*

Features	Operation	TAD	Machines	Tool
	Candidates	Candidates	Candidates	Candidates
F1	Drilling (op1)	-z, +z	M1, M2, M3	T1
F2	Drilling (op2)	-z, +z	M1, M2, M3	T1
F3	Milling (op3)	-z, +z	M2, M3	T7
F4	Milling (op4)	-z, +y	M2, M3	T5, T6
F5	Milling (op5)	+y	M2, M3	T5, T6
F6	Milling (op6)	+y	M2, M3	T5, T6
F7	Drilling (op7)	-z, +z	M1, M2, M3	T2
	Reaming (op8)	-z, +z	M1, M2, M3	T3
	Boring (op9)	-z, +z	M3, M4	T4
F8	Drilling (op10)	-z	M1, M2, M3	T1
F9	Drilling (op11)	-z, +z	M1, M2, M3	T2
	Reaming (op12)	-z, +z	M1, M2, M3	T3
	Boring (op13)	-z, +z	M3, M4	T4
F10	Milling (op14)	+x	M2, M3	T5, T6
F11	Drilling (op15)	-z	M1, M2, M3	T1
F12	Drilling (op16)	-z	M1, M2, M3	T1
F13	Milling (op17)	-y, -z	M2, M3	T5, T8
F14	Milling (op18)	-y, -z	M2, M3	T5, T6
F15	Drilling (op19)	-z, +z	M1, M2, M3	T1
F16	Drilling (op20)	-z, +z	M1, M2, M3	T1
F17	Milling (op21)	-y	M2, M3	T5, T6
F18	Drilling (op22)	-y	M1, M2, M3	T1
F19	Drilling (op23)	-y	M1, M2, M3	T1

Note: TAD = tool approach direction (x = horizontal; y = vertical; z = upward); F = feature; op = operation; M = machine; T = tool.

Table 2: *Cost Indices for the Example Prismatic Part*

Machine Cost Indices	Tool Cost Indices	Other Indices
M1 (10): Drill press	T1 (3): Drill 1	MCCI = 300
M2 (35): Vertical milling	T2 (3): Drill 2	SCCI = 90
M3 (60): Vertical CNC milling	T3 (8): Reamer	TCCI = 10
M4 (50): Boring machine	T4 (15): Boring tool	
	T5 (10): Milling cutter 1	
	T6 (15): Milling cutter 2	
	T7 (10): Chamfer tool	
	T8 (10): Slot cutter	

Note: M = machine; T = tool; CNC = computer numerical control; MCCI = machine change cost index; SCCI = setup change cost index; TCCI = tool change cost index.

Table 3: *The Precedence Matrix Between Operations for the Example Prismatic Part*

	Op01	Op02	Op03	Op04	Op05	Op06	Op07	Op08	Op09	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23
Op01		1	1																				
Op02			1																				
Op03																							
Op04			1																				
Op05				1		1									1	1	1	1	1	1	1	1	1
Op06				1											1	1	1	1	1	1	1	1	1
Op07								1	1	1	1	1	1										
Op08									1	1	1	1	1										
Op09										1	1	1	1										
Op10																							
Op11										1		1	1										
Op12										1			1										
Op13			1	1						1							1	1	1	1	1	1	1
Op14																1							
Op15																							
Op16																							
Op17																							
Op18																	1		1				
Op19																							
Op20																							
Op21	1	1	1				1	1	1	1	1	1	1				1	1	1			1	1
Op22																							1
Op23																							

Note: Op = operation.

Results From GA Against Evaluation Criteria

Two evaluating criteria were used to test the GA's capability and flexibility of handling process-planning problems under different requirements. The results follow.

Criterion 1: Minimizing Total Machining Cost (e.g., CC = MC + TC + MCC + SCC + TCC)
The average machining cost over 60 trials is $1,751, with the minimum being $1,739 and the maximum being $1,789. One of the process plans with a machining cost of $1,739 is shown in Table 4, together with its number of machine changes, setup changes, and tool changes.

Table 4: *Process Plan Against Criterion 1*

Op	14	5	6	4	21	18	17	22	23	15	1	16
M	2	2	2	2	2	2	2	2	2	2	2	2
T	5	5	5	5	5	5	5	1	1	1	1	1
TAD	+x	+y	+y	+y	-y	-y	-y	-y	-y	-z	-z	-z
Op	2	19	20	7	8	3	9	11	12	13	10	
M	2	2	2	2	2	2	3	3	3	3	3	
T	1	1	1	2	3	7	4	2	3	4	1	
TAD	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	

Note: Op = operation; M = machine; T = tool; TAD = tool approach direction (x = horizontal; y = vertical; z = upward); number of machine changes = 1; number of setup changes = 3; number of tool changes = 8; total machining cost = $1,739.

Criterion 2: Minimizing Number of Machine Changes only (e.g., CC = MCC)

Each of the 60 trials finds a process plan with zero machine changes. One of the process plans is shown in Table 5. It can be seen that only one machine is selected, for example, the computer numerical control milling machine.

Table 5: *The Process Plan Against Criterion 2*

Op	14	5	6	4	21	22	23	20	1	2	15	18
M	3	3	3	3	3	3	3	3	3	3	3	3
T	5	5	5	5	5	1	1	1	1	1	1	5
TAD	+x	+y	+y	+y	-y	-y	-y	-y	-y	-z	-z	-z
Op	17	16	19	7	8	9	11	12	13	10	3	
M	3	3	3	3	3	3	3	3	3	3	3	
T	8	1	1	2	3	4	2	3	4	1	7	
TAD	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	

Note: Op = operation; M = machine; T = tool; TAD = tool approach direction (x = horizontal; y = vertical; z = upward); number of machine changes = 0; number of setup changes = 3; number of tool changes = 12; total machining cost = $1,929. This total machining cost is lower than the $2,664 cost discovered by Zhang et al. (1997).

Data Mining

Data mining is an application, under human control, of low-level induction algorithms that are used to extract patterns from data in specific categories. Data mining is step of a long process chain of data analyzing called knowledge discovery in databases, which involves the evaluation and interpretation of patterns to determine what constitutes knowledge.

Most data mining algorithms are derived from machine learning, pattern recognition, and statistics. These algorithms include classification, clustering, and graphical models. The primary goals of knowledge discovery are prediction and description. Prediction involves using variables to forecast unknown future values of other variables or attributes. For example, some of its characteristics, such as size, style, location, and number of rooms, can predict the monetary value of a house. Description focuses on finding human-interpretable patterns describing the data, such as finding patterns for "good planning." The goal of applying data mining in this work is to predict patterns in a set of process plans and to develop a rule set, which will help in making decisions in odd circumstances (Koonce & Tsai, 2000).

Data Preparation for Data Mining

The 17 unique process plans generated by GA, which satisfy the optimal criteria, are used for data preparation. These process plans are generalized to higher-level concepts. This generalized data is fed through the See5 data mining software to get a set of rules.

The following six parameters are chosen as attributes for data mining (Hand, Mannila, & Smyth, 2001):

1. Operation
2. Precedence
3. Machine index
4. Tool index
5. TAD
6. Priority.

Operation

The operation attribute represents the number of the operation. As the part consists of 23 operations, this attribute value varies from 1 to 23; therefore, it is generalized as a "continuous" attribute.

Precedence

The precedence attribute represents the PR among the operations. The precedence attribute is generalized as "Yes" or "No." If any operation has precedence, it is denoted by "Yes," otherwise "No."

Machine Index

The machine index represents the cost of unit volume material removal by a particular machine. In our problem, the machine indices of different machines are shown in the Table 6.

Table 6: *Machine Cost Index Table*

Machine	Index	Classes
M1: Drill press	10	Low
M2: Vertical milling	35	Medium
M3: Vertical CNC milling	60	High
M4: Boring machine	50	High

Note: M = machine.

The values of machine indices are varying from 10 to 60. The range of these values is divided into three equal classes. Machine index values varying from 10 to 27 are classified as "Low," those from 28 to 44 are classified as "medium," and those from 45 to 60 classified are as "high."

Tool Index

The tool index represents the cost of unit volume material removal by a particular tool. In our problem, the tool indices of different tools are shown in the Table 7. The values of tool indices vary from 3 to 15. The range of these values is divided into three equal classes, such that tool index values varying from 3 to7 are classified as "Low," those from 8 to 11 are classified as "medium," and those from 12 to 15 are classified as "high."

Table 7: *Tool Cost index Table*

Tool	Index	Classes
T1: Drill 1	3	Low
T2: Drill 2	3	Low
T3: Reamer	8	Medium
T4: Boring tool	15	High
T5: Milling cutter 1	10	Medium
T6: Milling cutter 2	15	High
T7: Chamfer tool	10	Medium
T8: slot cutter	10	Medium

Note: T = tool.

TAD

The TAD attribute represents the direction of the tool in which it is performing the operation. This attribute has +x and -x (horizontal), +y and -y (vertical), and +z and -z (upward/downward) as tool approach directions.

Priority

The mining task is to find the relationship between an operation's characteristics and its order in the GA solution sequence. That is, we seek to predict the sequence position of an operation given its characteristics. As each operation has different possible sequence positions, it was decided that five abstract concepts would be substituted during the generalization operation sequence. The attribute priority is defined as a range of sequence positions in the GA solution. As the part has 23 operations, there will be a maximum of 23 possible sequence positions for an operation; thus, the value of position is classified, according to the equal-width concept, into four classes: {1,2,3,4,5,6} as First Class, {7,8,9,10,11,12} as Second Class, {13,14,15,16,17,18} as Middle class, and {19,20,21,22,23} as Last Class (Table 8).

Table 8: *Priority Table for a Process Plan Given by Genetic Algorithm*

	First Class	Second Class	Middle Class	Last Class
Operation Position	1, 2, 3, 4, 5, 6	7, 8, 9, 10, 11, 12	13, 14, 15, 16, 17, 18	19, 20, 21, 22, 23
Sequence of Operations	14, 5, 6, 4, 21, 18	17, 22, 23, 15, 1, 16	2, 19, 20, 7, 8, 3	9, 11, 12, 13, 10

Results

The structured data is manipulated by the classifier system See5, which produced a decision tree, or a decision rules. The induced decision rules are all of the same IF – THEN structure. Shown here are the obtained rules from the See5 software, when the rate given in the end of each rule is rate of good classification.

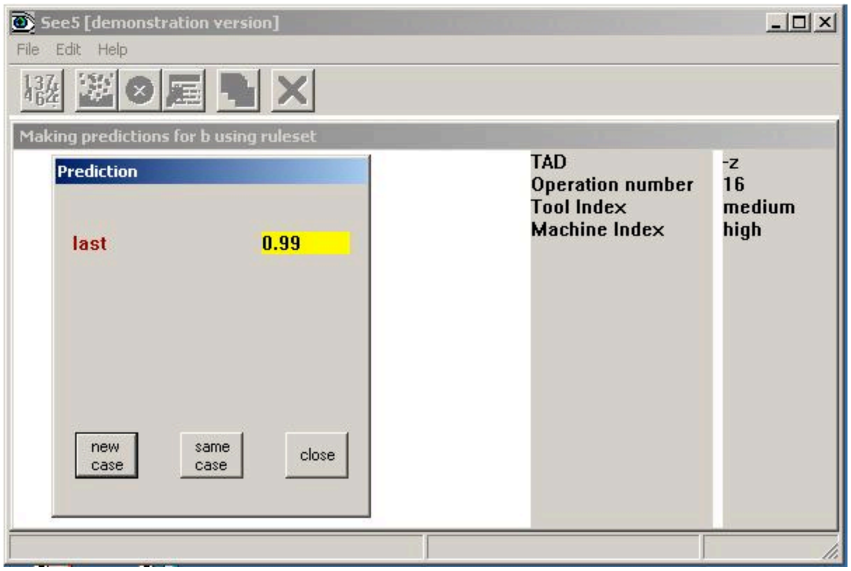
Rule 1: Machine index = medium
-> Class = first [0.334]
Rule 2: Tool index = low
TAD = -y
-> Class = second [0.972]
Rule 3: Operation number <= 17
TAD = -y

-> Class = second [0.947]
Rule 4: Operation number > 11
Operation number <= 15
Machine index = medium
TAD = -z
-> Class = second [0.895]
Rule 5: Operation number > 19
TAD = -z
-> Class = second [0.789]
Rule 6: Machine index = medium
TAD = -z
-> Class = middle [0.665]
Rule 7: Machine index = high
-> Class = last [0.989]
Default class: first

Application of Rule Set

The data generated from the process plan can also be used for predicting the sequence of the operation when there is a machine breakdown. Think that the tool or machine, which is performing the operation, is broken-down; then there are two options to adopt—one is to wait until the machine is repaired, and other is to perform same operation using another tool or machine. In some situations, the first option (i.e., to repair a machine) may take much time; therefore, the better choice is to follow the second option of performing the operation using alternative tools or machines.

After selecting the alternative tool or machine, the See5 classifier gives a worthy suggestion as to when this operation could be completed—that is, the appropriate class for the selected alternatives. For example, operation 16 (see Table 4, as this operation falls in second class during the classification) is performed on machine number 2 (vertical milling) using a drilling tool (T_1) in the '-z' direction. If machine 2 (classified as medium during the classification of data) is broken down, then it can perform the same operation using milling cutter 1 (classified as medium) on machine number 3 (classified as high). So, for these alternatives, See5 gives *last* as the predictive class. We can perform this operation along with the earlier-planned operations on machine number 3. Figure 6 shows the predictive class with confidence level.



Making predictions for b using ruleset

Prediction

last 0.99

TAD	-z
Operation number	16
Tool Index	medium
Machine Index	high

Figure 6: *Predicting the Alternative Class for Operation 16.*

Conclusions

In this paper, an attempt has been made to use data mining techniques to address for the first time the process-planning problem. The work proposed in this paper indicates the initial step toward the extraction of knowledge patterns for process planning. The developed rule set and See5 software help the process planner to make instant decisions in unusual circumstances without stopping production. By analyzing the historical production data, planning knowledge can be extracted and then be expressed in IF – THEN rules. This form of knowledge representation provides clear indications as to which factors are most influential in predicting planning and how it can be affected by various levels of critical factors.

In this paper, the authors employed GA for data mining for knowledge extraction of process-planning problems with minimum possible error. The error so far reported is 6.4%. Data mining requires an understanding of the problem domain, knowledge of mining algorithms, and an insight into which attributes might be significant.

References

Alting, L., & Zhang, H. (1989). Computer-aided process planning: the state-of-the-art survey. *International Journal of Production Research*, *27*, 553–585.

Bruns, R., & Forrest, S. (1993). Direct chromosome representation and advanced genetic operators for production scheduling. *Proceedings of the 5th International Conference on Genetic Algorithms* (pp. 352–359). San Mateo, CA: Morgan Kaufmann Publishers Inc.

Cay, F., and Chassapis, C. (1997). An IT view on perspectives of computer aided process planning research. *Computers in Industry*, *34*, 307–337.

Dagli, C., & Sittisathanchai, S. (1993). Genetic neuroscheduler for job shop scheduling. *Computers and Industrial Engineering, 25*, 67–270.

Dereli, T., & Filiz, I. H. (1999), Optimisation of process planning functions by genetic algorithms. *Computers and Industrial Engineering, 36*, 281–308.

Falkenauer, E., & Delchambre, A. (1992). A genetic algorithm for bin packing and line balancing. *Proceedings of the 1992 IEEE International Conference on Robotics and Automation* (pp. 1186–1192). Los Alamitos, CA: IEEE Computer Society Press.

Fayyad, U. M. (1997). Data mining and knowledge discovery in databases: Implications for scientific databases. *Proceedings of the Ninth International Conference on Scientific and Statistical Database Management* (pp. 2–11). Washington, DC: IEEE Computer Society.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.

Goldberg, D. E., & Lingle, R. (1985). Alleles, loci and the traveling salesman problem. *Proceedings of an International Conference on Genetic Algorithms and Their Application*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining.* Prentice Hall of India.

Harrath, Y. Chebel-Morello, B. & Zerhouni, N. (2001). A genetic algorithm and data mining to resolve a job shop schedule. *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation*. Antibes-Juan les Pins, France: IEEE.

Harrath, Y. Chebel-Morello, B. & Zerhouni, N. (2002). A genetic algorithm and data mining based meta-heuristic for job shop scheduling problem. *IEEE International Conference on Systems, Man and Cybernetics*. Hammamet, Tunisia: IEEE.

Koonce, D. A., & Tsai, S. C. (2000). Using data mining to find patterns in genetic algorithm solutions to a job shop schedule. *Computers & Industrial Engineering, 38, 361–374.*

Li, L., Fuh, J. Y. H., Zhang, U. F., & Nee, A. Y. C. (2005). Application of genetic algorithm to computer-aided process planning in distributed manufacturing environments. *Robotics and Computer Integrated Manufacturing*, *21*, 568–578.

Steudel, H. J. (1984). Computer-aided process planning: past, present and future. *International Journal of Production Research*, *22*, 253–266.

Taaj, S. C. (1997). *Using data mining to explore the regularity of genetic algorithms in job shop scheduling problems* (Unpublished thesis). Ohio University, Athens, Ohio.

Weill, R., Spur, G., & Eversheim, W. (1982). Survey of computer-aided process planning systems. *Annals of the CIRP*, *31*, 539–551.

Yang, Y., & Webb, G. I. (2002). A Comparative study of discretization methods for Naive-Bayes classifiers. In T. Yamaguchi, A. Hoffmann, H. Motoda, & P. Compton (Eds.), *Proceedings of the 2002 Pacific Rim Knowledge Acquisition Workshop* (pp. 159–173). Tokyo, Japan: Japanese Society for Artificial Intelligence.

Zhang, F. (1997). *Genetic algorithm in computer-aided process planning* (Master's thesis). National University of Singapore.

Zhang, F., Zhang, Y. F., & Nee, A. Y. C. (1997). Using genetic algorithms in process planning for job shop machining. *IEEE Transactions on Evolutionary Computation,* 1, 278–289.

Zhang, Y. F., Ma, G. H., & Nee, A. Y. C. (1999). Modeling process planning problems in an optimization perspective. *Proceedings of 1999 IEEE International Conference on Robotics and Automation* (pp.1764–1769). Detroit, MI: IEEE.