# Intelligent process planning system for optimal CNC programming - a step towards complete automation of CNC programming

Yeung, Millan K.



**Publisher's version / Version de l'éditeur:**
https://doi.org/10.1108/17410380310698478
*Integrated Manufacturing Systems, 14, 7, pp. 593-598, 2003*

# Intelligent Process Planning System for Optimal CNC Programming – A Step Towards Complete Automation of CNC Programming

Millan K. Yeung
Group Leader, Shape Transfer Processes
Integrated Manufacturing Technologies Institute
National Research Council Canada
800 Collip Circle, London, Ontario, N6G 4X8.
Email: Millan.Yeung@NRC.CA

## 1. Abstract

One of the bottlenecks of CNC machining is the CNC programming. It relies on the experience and skills of the CNC programmer for the generation of the CNC program. The intelligent process planning system described in this paper generates a process plan automatically for CNC programming. It utilizes artificial intelligent technologies such as knowledge base, blackboard system and machine learning to extract machine-able features, proposes and selects optimal tools for the machining of the given part. Its flexibility and simplicity provides a convenient way to include new techniques and knowledge. The incorporation of this system with other CAD/CAM tools could effectively automate the CNC programming process.

Keywords: CNC programming, process planning, artificial intelligence, knowledge base, blackboard system, machine learning.

## 2. Introduction

CNC machining is one of the most important developments for manufacturing technologies in the 20$^{th}$ Century. It provides high efficiency for mass production of consumer products and flexibility for low quantity production of specialized parts and components. However, one of the bottlenecks of CNC machining is the CNC programming. It requires a skillful programmer who should not only be a CAD/CAM and computer literate but also a machining expert. While CNC tool paths can be generated by most CAD/CAM systems efficiently, planning for the machining process is tedious and almost completely relies on the expertise of the programmer. Decisions such as tools used for roughing and for finishing, number of passes and sequence of cuts are completely dependent on the knowledge of the programmer. Recent publications have started to address this shortcoming of CNC programming and many research works have been conducted to find ways to reduce this deficiency but they mostly dealt with specific environments and conditions [1-9]. This paper proposes an intelligent process planning system based on a generic methodology and algorithm to automate and optimize the planning process for CNC machining.

The system utilizes the knowledge base, blackboard system and generic learning technologies. The knowledge base captures the experience of the programmer and facts relevant to the machine tools. The combination of the blackboard system, learning structure and algorithm defines the process plan and optimizes it for the operation. The implemented system would be very flexible. It has the capability to store and utilize new and specialized knowledge and facts or rules. Optimization is achieved within the defined knowledge set. With the capability and flexibility to expand and learn, the system could fully automate and optimize the CNC planning process within its given machining domain.

## 3. <u>The system</u>

The system divides the CNC planning process into three modules. The first module "Feature Extraction" is a knowledge base [10] that captures and stores part features, common machining techniques and logics for the given CNC machine. The second module "Tool Competition" is a blackboard system [11] that considers each cutting tool preset on the CNC machine as a 'competitor'. It competes the 'job' to machine the part that is posted onto the blackboard. The third module "Tool Optimization" is a learning system [12] that evaluates and selects the optimal cutting tools from the successful competitors to machine the part. An optimal cutting tool is selected for each pass of the tool path and therefore the tool path is optimized.

### 3.1 The Feature Extraction module

The core of this module is a knowledge base that consisted of a series of parallel nodes. Each of these nodes represents a family or class of machine-able parts (or sections of a part). Rules, facts, topological machining logics and constraints for the machining of the part are stored in a tree under the node. The leaf of the tree is the feature to be machined. To plan a machining job, the part to be machined is divided into features according to their machining criteria and procedures. Subsequently, the features are extracted in a topological order for the actual machining. The granularity of feature or part division can be adjusted for specific preference and application with appropriate design and implementation. The knowledge base $S$ is a set of tree structures that represents different classes of parts.

$$S = \{ x_1, x_2, x_3, \ldots, x_n \};$$

Where $x_i \cap x_j = \phi; i, j = 0, 1, 2, \ldots, n; i \neq j$ and $x_i$ is the root of a tree such that

$$x_i = ( y_1, y_2, \ldots, y_k );$$

Where $y_i$ is a sub-tree that represents a sub-assembly of features or a leaf that represents a feature to be machined, $k$ is the number of sub-assemblies or features.

The output $F$ of this module is a set of $m$ ordered pairs of machine-able feature $z_i$ and updated blank configuration $b_i$ for the part to be machined.

$$F = \{ (z_1, b_1), (z_2, b_2), (z_3, b_3), \ldots, (z_m, b_m) \};$$
$$and\ z_i \cap z_j = \phi;\ i \neq j;\ b_p \in b_q;\ b_q \notin b_p;\ p < q;\ i, j, p, q = 0, 1, 2, \ldots, m;$$

The operation starts by setting $z_0$ to null, $b_0$ to the blank and $p_0$ to the part. Then

$$z_0 = \phi;$$
$$b_0 = blank\_configuration; \quad size, material, shape, etc.$$
$$p_0 = \{ f_1, f_2, \ldots, f_m \}; \quad the\ part,\ represented\ by\ features$$
$$z_{i+1} = g( S, p_i );$$
$$b_{i+1} = b_i + z_i;$$
$$p_{i+1} = p_i - z_i;$$
$$and\ f_i \subseteq z_j;\ i, j = 0, 1, 2, \ldots, m;$$

Where $g$ is a function (the inference engine) that extracts the machine-able feature $f$ from the part $P$ according to the machining topology and knowledge stored in $S$ and outputs it along with the specific rules (if there are any) to $z$. It can be as simple as

$$g( S, p_i ) = extract\_leaf( x_i \leftarrow p_i\ or\ f_j \in S );$$

The procedure re-iterates until all the features are extracted.

### 3.2 The Tool Competition module

The main element of this module is a blackboard or bulletin board structure where the feature and requirements extracted from the Feature Extraction module are posted. Each of the cutting tools preset on the machine is represented by a 'vendor' (a bidder to the job listed on the bulletin-board) or tool-node that will 'compete' to machine the feature based on its capabilities and constraints. Each tool-node is equipped with a knowledge base that stores these capabilities and constraints. The structure of the knowledge base is similar to the one in the Feature Extraction module. The capabilities are represented by a set of parallel nodes and their constraints are stored in tree structures under the nodes. A tool $T$ is represented by a set of $k$ capabilities $C$.

$$T \supset C;$$
$$C = \{ c_1, c_2, c_3, \ldots, c_k \};$$

Where $c_i \cap c_j = \phi;\ i, j = 0, 1, 2, \ldots, k;\ i \neq j$ and $c_i$ is the root of a tree such that

$$c_i = ( r_1, r_2, \ldots, r_n );$$

Where $r_i$ is a sub-tree that represents an intermediate constraint or a leaf. The leaf could be restrictions, limits of the functional/operational range, quantified finishing qualities, etc.

To machine a feature $F$ being posted in the bulletin board, tool $T$ examines its capabilities and decides whether to compete the machining work or not. The output $Q$ of this module would be a set of tools that can machine $F$.

The algorithm is relatively simple for this process.

$$F = ( f, b );$$
$$Q = \phi;$$
$$t_i( f, b ) = \begin{cases} T_i; & if\ c_i \rightarrow f;\ \text{i.e.}\ T_i\ is\ to\ compete \\ \phi; & Otherwise;\ \text{i.e.}\ T_i\ is\ not\ to\ compete \end{cases}$$
$$Q = Q + t_i( f, b );$$
$$for\ i = 1, 2, \ldots, m;$$

Where $f$ is the feature to be machined, $b$ is the current configuration of the blank, $t_i$ is the inference engine of the $i^{th}$ tool that decides if $T_i$ is capable to machine $F$ and $m$ is the number of tools preset for the machine.

### 3.3 The Tool Optimization module

The Tool Optimization module works in conjunction with the Tool Competition module to select optimal tools for the machining of the part. At this early design of the system, the environment for the evaluation is the simulation of machining path/pass of the selected tool and the goal is to achieve the fastest cycle time. A modified Stochastic Learning Automata [13] structure with generic learning algorithm is chosen to be the foundation of the implementation because of its flexibility and simplicity. Future design of the learning module will include multiple goals in various properties such as surface finishing, machining dynamics, tool characteristics, tool longevity, etc. The standard Stochastic Learning Automata structure is defined as the following quintuple.

$$SLA = \{ \alpha, \beta, p, T, c \};$$

Where $a$ and $\beta$ represent the sets of input and output states, $p$ is the corresponding probability vectors, $T$ is the learning algorithm and $c$ is the corresponding penalty probability defining the environment. The modified SLA for this system is much simpler with only three elements.

$SLA = \{ \beta, P, T \};$

Where $\beta$ is the output state, $P$ is the probability vector for the cutting tools of the machine and $T$ is the learning algorithm. A tool is represented by an element $p_i$ of the probability vector $P$.

$P = \{ p_1, p_2, \square\ , p_n \};$

Where $n$ = number of tools preset on the machine that is qualified for the job.

$T$ evaluates the return from the simulation and updates $P$ accordingly. After the $i^{th}$ iteration, $T$ updates $P^i$ to $P^{i+1}$ for the next simulation based on the $\beta^i$.

$P^{i+1} = T\{ P^i, \beta^i \};$

The expansion of $T$ can be described in two reactions to counter $\beta^i$.

Favorable reaction is when the return from the $i^{th}$ simulation is positive and $k^{th}$ tool was chosen:

$$p_j^{i+1} = p_j^i - ap_j^i; \quad \forall j, j \neq k, 0 < a < 1$$

$$p_k^{i+1} = p_k^i + \sum_{j=1, j \neq k}^{n} ap_j^i;$$

Unfavorable reaction is when the return from the $i^{th}$ simulation is negative and $k^{th}$ tool was chosen:

$$p_j^{i+1} = p_j^i + \left\{ \frac{b}{(n-1)} - bp_j^i \right\}; \quad \forall j, j \neq k, 0 \leq b < 1$$

$$p_k^{i+1} = p_k^i - \sum_{j=1, j \neq k}^{n} \left\{ \frac{b}{(n-1)} - bp_j^i \right\};$$

Where $a$ is the reward parameter and $b$ is the penalty parameter. They can be strategically set to offset the weights of the favorable or unfavorable return according to the desired scenario.

Initially, all qualified tools are set to have equal probabilities so that they would have an equal chance to be chosen to perform the next pass of the cutting path. Based on the result from the simulation, the tool returns the best performance (e.g. shortest cycle time) is chosen for the pass and its probability is asymptotically increased while probabilities for other tools are decreased. On the other hand, if the performance of the chosen tool is degraded from its previous pass, the opposite actions are applied. Note that the reward would be greater than the penalty if the penalty parameter were set to equal to the reward parameter. This gives the degraded tool other chances to prove itself until its probability is penalized to below the level of the others. This tool selection iterates until the path is completed.

## 3.4 Integration

The three modules described work cooperatively from extracting the machine-able features to generating the optimal CNC programming plan. While the modular nature of the system permitted the implementation of these modules individually, an effective integration of these modules to achieve the planning goal is needed. The classical goal of integration of modular systems is high cohesion but low coupling. This allows the autonomous operation of each module and lessens the interferences between modules. The design of these modules follows this principle and the goal of integration can be achieved.

The operation and integration of the Feature Extraction module is straightforward. It represents and stores the machine-able features and the updated blank configurations independently from other operations. The input could be a CAD model of the part to be machined and/or specific preferences from the operator or programmer. The output is a simple set of feature and updated-blank pairs. Hence the integration of this module into the system is simple. The capturing techniques and data representation of the input is outside the scope of this paper. Relevant information will be described in future publications.

The integration between the Tool Competition module and the Tool Optimization module is also simple but with an added element. Their operations must be synchronized in order to generate an optimal path with multiple passes. The input of the Tool Competition module is the feature and updated blank pair from the Feature Extraction module. It outputs a set of capable tools that can machine the given feature that in turn becomes the input for the Tool Optimization module. The Tool Optimization module examines each tool in the set for each pass in the cutting path of the feature and outputs the best tool for that pass. These cooperative operations iterate until the feature is completely machined. While the generated tool set being examined the Tool Competition module can process the next feature from its input. However, it must output the next capable tool set in synchronization with the input of the Tool Optimization module to reduce storage and processing overheads. This tool competition and optimization processes propagated to all the features of the part to be machined. Integration of these two modules into the system can then be carried out as a unified module. Figure 1.0 shows the schematics of the integration.

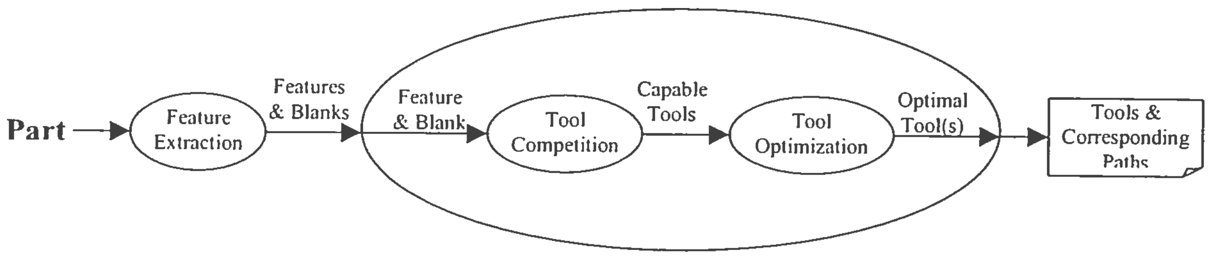


**Figure 1.0 Integration of the Feature Extraction, Tool Competition and Tool Optimization modules.**

## 4. <u>Machining Parts</u>

To machine a part, the Feature Extraction module divides the part into a number of machine-able features and arranges them into an ordered set along with the corresponding blank configurations according to the machining topology or operational preferences. For each of this ordered pair, the Tool Competition module recommends a set of capable tools to machine the feature and the Tool Optimization module, through simulation, evaluates and selects the optimal tools among the recommended tools. The process of tool recommendation and selection iterates until tools for all the features are selected. The outcome is an optimal plan for the CNC programming. Figure 2.0 shows the schematic of the operation of the system.

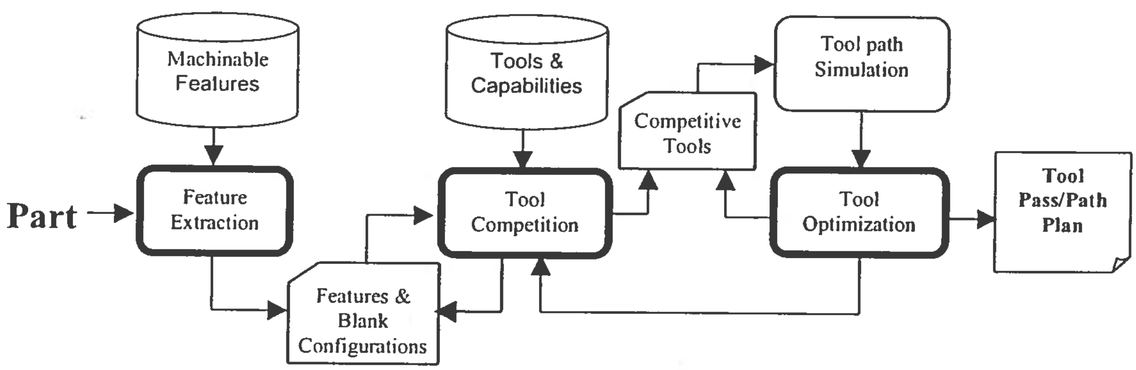


**Figure 2.0 Operation of the Intelligent Process Planning System.**

Following is an example to illustrate the operation of this intelligent planning process for CNC programming and to verify its feasibility.

The part: 2” ∅ x 2” long cylindrical sleeve with a 1½” ∅ through hole.
$p_0 = \{ f_1 = bore,\ f_2 = o.d.,\ f_3 = front\_face,\ f_4 = back\_face \};$

The blank: 2¼” ∅ x 4” cold rolled steel bar with 1½” chucking length.
$B = as\_described;$

The machine: 2-axis CNC lathe with 8-tool turret ATC; spindle speed and feed rate are negligent for this example.

The K-base:
$$S = \left\{ x_1 \left( \begin{array}{l} tubular\_part; \\ y_1 \begin{pmatrix} rough\_cut; \\ z_1 = bore, \\ z_2 = o.d., \\ z_3 = front\_face \end{pmatrix}, y_2 \begin{pmatrix} finish\_cut; \\ z_4 = front\_face, \\ z_5 = bore, \\ z_6 = o.d., \\ z_7 = back\_face, \\ z_8 = parting \end{pmatrix} \end{array} \right), \square\ , x_n(\square\ ) \right\};$$

Preset tools: 1" ∅ drill with 4" drilling depth

$$T_1 = \left\{ c_1 \begin{pmatrix} drill\_hole; \\ r_1 = 1"\phi\ \ hole, \\ r_2 = \leq 4"\ deep \end{pmatrix} \right\};$$

½" ∅ x 4" finish-cut boring bar; optimal cutting depth 0.005" – 0.01".

$$T_2 = \left\{ c_1 \begin{pmatrix} machine\_through\_bore; \\ r_1 = requires \geq 0.625"\varnothing\ existing\ hole, \\ r_2 = 0.005"\text{–}0.01"\ optimal\ cutting\ depth, \\ r_3 = bore \leq 4"\ long, \\ r_4 = finish\ cut\ quality;\ 0"\ under\_\phi\_size \end{pmatrix} \right\};$$

¾" ∅ x 4" rough-cut boring bar; optimal cutting depth 0.02" – 0.05".

$$T_3 = \left\{ c_1 \begin{pmatrix} machine\_through\_bore; \\ r_1 = requires \geq 0.875"\varnothing\ existing\ hole, \\ r_2 = 0.02"\text{–}0.05"\ optimal\ cutting\ depth, \\ r_3 = bore \leq 4"\ long, \\ r_4 = rough\ cut\ quality;\ \cong 0.01"\ under\_\phi\_size \end{pmatrix} \right\};$$

Left side rough-cut tool; optimal cutting depth 0.02" – 0.1".

$$T_4 = \left\{ c_1 \begin{pmatrix} machine\_O.D.; \\ r_1 = 0.02"\text{–}0.1"\ optimal\ cutting\ depth, \\ r_2 = rough\ cut\ qaulity;\ \cong 0.01"\ over\_\phi\_size \end{pmatrix} \right\};$$

Left side finish-cut tool; optimal cutting depth 0.005" – 0.02".

$$T_5 = \left\{ c_1 \begin{pmatrix} machine\_O.D.; \\ r_1 = 0.005"-0.02" \text{ optimal cutting depth}, \\ r_2 = \text{finish cut qaulity}\ ; 0" \ over\_\phi\_size \end{pmatrix} \right\};$$

¼" x ¾" parting tool.

$$T_6 = \left\{ c_1 \begin{pmatrix} parting\_or\_grooving\_and\_facing\_both\_sides; \\ r_1 = ¼\text{" cutting width}, \\ r_2 = \leq ¾\text{" cutting deep} \end{pmatrix} \right\};$$

With this setup, the Feature Extraction module returns the set

$$F = \{ (z_1, b_1), (z_2, b_2), (z_3, b_3), (z_4, b_4), (z_5, b_5), (z_6, b_6), (z_{7,8}, b_7) \};$$
$$with\ b_{i+1} = b_i + z_{i-1}\ ; i = 1,2,\ldots,8\ ; z_0 = \phi\ ; b_1 = b_0;$$

For each of the $(z_i, b_i)$, the Tool Competition and Tool Optimization modules work together to define the optimal set of tool-passes. Each tool-pass is represented by the ordered pair *(Tool$_i$, Pass$_i$) or $(T_i, P_i)$*. The complete set of tool-passes generated by the system for the machining of the cylindrical sleeve is

$$CNC\_Plan = \left\{ \begin{matrix} (T_1, P_1), (T_6, P_2), (T_3, P_{3,4,5,6}), (T_3, P_7), (T_4, P_8), (T_4, P_9), (T_6, P_{10}), (T_2, P_{11}), \\ (T_5, P_{12}), (T_6, P_{13}) \end{matrix} \right\};$$

Where
- $P_1$ = drilling;
- $P_2$ = roughing front face;
- $P_{3\text{-}6}$ = rough-boring at 0.05" cutting depth each;
- $P_7$ = rough-boring at 0.045" cutting depth;
- $P_8$ = roughing O.D. at 0.1" cutting depth;
- $P_9$ = roughing O.D. at 0.02" cutting depth;
- $P_{10}$ = finishing front face;
- $P_{11}$ = finishing bore;
- $P_{12}$ = finishing O.D.;
- $P_{13}$ = parting and finishing back face;

The plan can be incorporated into CAD/CAM systems for the CNC programming of the actual machining.

## 5. Conclusion

An automated process planning system can reduce the bottleneck burden of CNC machining. The system presented utilizes knowledge base, blackboard system and machine learning technologies to automatically plan and optimize the CNC programming process. Its flexibility and simplicity allow the inclusion of meta-knowledge and operation specific preferences, as well as dynamic updating. Users can adjust the granularity of the knowledge and information input into the system to satisfy their needs and preferences. It can also be incorporated or integrated into CAD/CAM systems for direct CNC programming. Future expansion of this system will include the development of new machining and automation technologies and relevant knowledge such as high-speed-machining, motion dynamics, geometric error compensation, characteristics of machines, tools and materials, etc. The intelligent process planning system will accommodate and facilitate the implementation and operation of these technologies and processes. It is a true fully automatic planning system for CNC programming and automation-controls within its knowledge domain. It can evolve and grow dynamically along with the acquisition of new knowledge and techniques.

## 6. References


[1] Chin Sheng Chen, "Developing a feature based knowledge system for CAD/CAM integration", Computers & Industrial Engineering. vol.15, pp.34-40, 1988.

[2] Mantyla M., "Feature-based product modeling for process planning", Organization of Engineering Knowledge for Product Modeling in Computer Integrated Manufacturing, 2nd Toyota Conference. Elsevier, Amsterdam, Netherlands, xii+461, pp.303-324. 1989.

[3] Martin J.M., "A strategy for NC programming", Manufacturing-Engineering. vol.102, no.2, pp.82-84. Feb. 1989.

[4] Pande S.S. and Prabhu B.S., "An expert system for automatic extraction of machining features and tooling selection for Automats", Computer Aided Engineering Journal, vol.7, no.4, pp.99-103, Aug. 1990.

[5] Budde W. and Imbusch K., "EXAPT process planning and NC planning with database-supported management of production data", IFIP Transactions B (Applications in Technology), vol.B-3, pp.119-30, 1992.

[6] Norton N., "Controlling the Shop Floor", Manufacturing-Engineer, vol. 72, no. 6, pp. 272-275. December 1993.

[7] Yeo S.H., "A multipass optimization strategy for CNC lathe operations", International Journal of Production Economics, vol.40, no.2-3, pp.209-218, Aug. 1995.

[8] Fuh J.Y.H., Ji P. and Zhang Y.F., "Future development trends in CAM/CAPP-NC systems", International Journal of Computer Applications in Technology, vol.8, no.3-4, pp.203-210, 1995.

[9] Christensen G.K. and Mogensen O.B., "Backward form feature recognition and removal for an automatic CNC-programming system-BCAM", Advanced CAD/CAM Systems – State-of-the-art and Future Trends in Feature Technology. Chapman & Hall, London, U.K., pp. 205-216, 1995.

[10] Waterman Donald A., "A Guide to Expert Systems", Addision-Wesley Publishing Company, U.S.A., pp.392, 1986.

[11] Waterman Donald A., "A Guide to Expert Systems", Addision-Wesley Publishing Company, U.S.A., pp.388-389, 1986.

[12] Tanimoto Steven L. "The Elements of Artificial Intelligence", Computer Science Press, W.H. Freeman and Company, New York, U.S.A., pp.311-313, 1990.
[13] Narendra K.S. and Thathatchar M.A.L., "Learning Automata – An Introduction", Prentice-Hall International Inc., 1989.