A Search Algorithm for Generating Alternative Process Plans in Flexible Manufacturing System

Hossein TEHRANI**, Nobuhiro SUGIMURA**, Yoshitaka TANIMIZU ** and Koji IWAMURA**

**Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Nakaku, Sakai, Osaka, 599-8531, Japan
m08thrni@me.osakafu-u.ac.jp

Abstract
Capabilities and complexity of manufacturing systems are increasing and striving for an integrated manufacturing environment. Availability of alternative process plans is a key factor for integration of design, process planning and scheduling. This paper describes an algorithm for generation of alternative process plans by extending the existing framework of the process plan networks. A class diagram is introduced for generating process plans and process plan networks from the viewpoint of the integrated process planning and scheduling systems. An incomplete search algorithm is developed for generating and searching the process plan networks. The benefit of this algorithm is that the whole process plan network does not have to be generated before the search algorithm starts. This algorithm is applicable to large and enormous process plan networks and also to search wide areas of the network based on the user requirement. The algorithm can generate alternative process plans and to select a suitable one based on the objective functions.

Key words: Production System, Process Planning, Flexible Manufacturing Systems (FMS), Artificial Intelligence, Engineering Optimization

1. Introduction

Manufacturing machineries and equipment have made much progress during the last decade, aiming at realizing more flexible structures and control systems which can cope with the dynamic changes in the variety of products and producing the more complicated parts. Current developments in such kind of the integrated manufacturing systems focus on linking different activities (often called islands of automation) within the enterprise in order to use the resources effectively. One of the most critical links for integration is the link between the process planning and the scheduling. Availability of alternative process plans is a key factor for integration of the design, the process planning, and the scheduling functions. The availability of the alternative process plans provides flexibility for the scheduling functions under a more relaxed set of constraints to reach better solutions in terms of the resource utilization or the production cycle minimization (1).

The issue of the process sequencing and the generation of the alternative process plans have received early attention in the process planning research. Prabhu et al. (2) proposed a method for generating the operation networks for the rotational parts based on the feature precedence constraints. They proposed a network representing the feasible feature sequences for the turning operations. The integration of the process plan networks into the

*Received 18 Jan., 2007 (No. 07-0037)
[DOI: 10.1299/jamdsm.1.706]
process planning and the shop floor control is reported in Cho et al.\(^{(3)}\). They describe the two step process planning (off-line and on-line) and the hierarchy of the manufacturing tasks. An algorithm is proposed to convert to the feature graphs into the task graphs. Sormaz and Khoshnevis\(^{(4)}\) proposed a methodology for generating the process plan networks and a method for optimal process plan selection. The proposed process plan hierarchy consists of four interconnected layers: feature layer, process layer, cutting tool orientation layer and machine layer.

In this paper, we extend the process plan domain for the advanced manufacturing systems by adding the fixture layer to the process plan hierarchy. In the manufacturing systems, we need to change the part’s fixtures based on the machining operations and the machine tool types. For the case of the prismatic parts, we have to change the part’s fixtures to change the positions and orientations of the parts against the tools. Thus, we have to consider the fixtures in the process planning problems. The generated process plans become more realistic by adding the fixture layer to the process plan domain. This important factor introduces another complexity to the process planning that has not yet been discussed in the previous papers. A class diagram is introduced for generating the process plans and the process plan networks from the viewpoint of the integrated process planning and scheduling systems. We also propose a search algorithm to generate a set of alternative process plans and to select the suitable process plans. The proposed search algorithm provides the users with an effective algorithm for generating the alternative process plans from the wide areas of the process plan networks by setting the search parameters.

2. Process Planning algorithm

2.1 Machining Features Generation

The process planning system generates the manufacturing process sequences, by which the parts are transformed from the raw materials into the finished products. The process planning task starts with extraction of machining features defined as a group of the faces, which are machined concurrently by the same combination of the machine tools, the fixtures and the cutting tools.

In this paper, we use the process planning algorithms previously proposed \(^{(5)}\). Sugimura \(^{(5)}\) has developed an object oriented models to represent the information needed in both the process planning and the scheduling. Algorithms are developed to extract the machining features based on both the part model and the model of shape generation functions. The feasible processes, the machine tools, the fixtures and the cutting tools are selected for the individual machining features, referring to the machining functions of the machining equipment. The manufacturing time of the individual feature is estimated for the combination of the machine tools, the fixture and the cutting tools. Finally, the precedence constraints among the machining features are determined based on the face ID, shapes and surface roughness. The extraction task of the machining features is basically carried out in the following four steps;

**STEP 1: Extraction of faces to be machined**
All the faces to be machined are extracted based on the part model representing the finished parts.

**STEP 2: Selection of suitable shape generation functions of faces**
Suitable shape generation functions are selected for the individual extracted faces referring to the shape generation function model representing the machining functions of the machine tools and the cutting tools.

**STEP 3: Integration of faces and generation of machining features**
Machining features are generated by grouping the faces which can be generated by the same shape generation functions. The machining features are the group of the faces machined simultaneously by the same combinations of the machine tools, the fixtures and cutting
tools. \( MF \) is a set of machining features, where \( f_j \) is the \( j \)-th machining feature of the part;

\[
MF = \{ f_j | j = 1, 2, ..., n \}, \quad n = |MF|, \quad (|A| : \text{cardinality of set } A)
\]  

The precedence constraints among the machining features are determined based on the face ID, the shapes and the surface roughness of the machining features. For examples, if there are a rough machining feature and a finished machining feature, both of which are extracted from a same face of the part, high priority is given to the rough machining feature. If there is a hole feature on a plane feature, high priority is given to the plane feature.

**STEP 4: Selection of suitable machining equipment for individual machining features**

Feasible machining equipment for the individual machining features are selected based on the surface roughness and the machining functions of the machining equipment. For the individual machining feature \( f_j \), we have the feasible set of the machining equipments \( ME_j \), which are capable to manufacture \( f_j \), and represented by the combinations of the machining tools, the fixtures and cutting tools;

\[
ME_j = \{ me_l | l = 1, 2, ..., n_j \}, \quad n_j = |ME_j|, \quad (2)
\]

\[
MT = \{ mt_l | l = 1, 2, ..., m \}, \quad m = |MT|, \quad (3)
\]

\[
FI = \{ fi_l | l = 1, 2, ..., F \}, \quad F = |FI|, \quad (4)
\]

\[
CT = \{ ct_t | t = 1, 2, ..., T \}, \quad T = |CT|, \quad (5)
\]

where,

\( mt_l \) : ID of machine tools.

\( fi_l \) : ID of fixtures.

\( ct_t \) : ID of cutting tools.

### 2-2 Generation of Process Plan Network

The goal of process planning is to solve the problems of the machining equipment selection and the process sequencing. For this purpose, process plan network is defined as a representation scheme that captures the generated process plans in a non-linear and hierarchical way\(^4\). It includes all the alternative process routes while satisfying the precedence constraints of the machining features. In the generation of the process plan networks, we have to consider the equality of machine tools, fixtures and cutting tools. The equality of them is verified based on the equality of the IDs of the machine tools, the fixtures and the cutting tools.

The generation algorithm starts from the initial node representing the raw material and generates a set of nodes representing the candidate machining features that could be machined in the first stage. The candidate machining features are clustered based on the IDs of the machine tools, the fixtures and the cutting tools. The algorithm is repeated until all the features are included in the machining sequence which is consistent with the precedence constraints. Note that the network does not provide all the feature sequences, because in each machine tool the set of assigned features could be processed in any order. The algorithm for generation of process plan network is mentioned in the followings.

**Initialization :**

- Set \( RMF \) and \( AMF \). The \( RMF \) is the set of the remaining machining features. The \( AMF \) is the set of the available machining features that do not have any preceding machining features and could be done firstly considering the precedence constraints among the nodes of the process plan networks. (\( AMF \subseteq RMF \)).
- Set both \( CNL \) and \( ONL \) as empty. The \( CNL \) is the set of the closed nodes and the \( ONL \) is the set of the open nodes.
- Set a initial node \( N_0 \). In the \( N_0 \), \( RMF = \{ \text{set of all machining features} \}, \ AMF = \{ \text{set of all machining features without any predecessors} \}. \)
At the initial node $N_0$, we have $RMF = \{MF1,MF2,MF3\}$, $AMF = \{MF1,MF3\}$. There are feasible machining equipment sets $(mt_1,fi_1,ct_1)$, $(mt_1,fi_1,ct_2)$, $(mt_1,fi_1,ct_1)$, $(mt_2,fi_2,ct_2)$, $(mt_2,fi_2,ct_2)$ for the machining features.
A portion of process plan network for the example is shown in Fig. 2. We start with initial node $N_0$, and four successor nodes $N_i (i=1,2,3,4)$ from the initial node $N_0$. We put these four nodes in the $ONL$, and update the $RMF$ and $AMF$ for these successor nodes. At each step of the algorithm, we select a node from $ONL$ and generate its successor nodes. If the node $N_1$ is selected from the $ONL$, there are four successor nodes which we put them in the $ONL$, and put the node $N_1$ in $CNL$. The same processes are repeated until the $ONL$ becomes empty. When we reach to the last level of the process plan network, as you can see in Fig.2, all the nodes move to $CNL$, and the search algorithm stops. It is because that there are no remaining machining features in the nodes of the $ONL$.

### 2-3 Data Model for Process Planning

The UML class diagram for generating process plans and process plan networks is shown in Fig. 3. The UML is a standard modeling language for visualizing, and specifying the data structures, the dynamic behaviors and the model organizations (6).

This class diagram is an extension of object model which is proposed by Sugimura (5). The boxes and the arcs, in the Fig. 3, represent the class, the class attributes and their relationships, respectively. The job class represents the information about the parts. The machining equipment class gives the information about the shape generation functions of the machining equipment and the component of the machining equipment, such as the machine tools, the cutting tools and the fixtures.

The class of the shape generation functions gives the information about the relationships between shape generation functions of the machining equipment and the faces of the jobs to be machined. This class is referred to in selection of suitable shape generation functions and machining equipment for the individual faces. The machining feature class
integrates the faces, which could be generated simultaneously by the same combinations of the machine tools, the cutting tools and the fixtures.

For generating the process plan networks under the precedence constraints, the machining features are clustered based on the equality of IDs of the machine tools, the fixtures and the cutting tools. The process plan nodes also contain the addresses of the followed process plan nodes (successor nodes) and the preceding process plan nodes (predecessor nodes) in the process plan networks. The process plan networks contain the addresses of the process plan nodes in the first level of the process plan networks. We can search all the information of the process plan networks from the process plan network class and the process plan node class, because the addresses of the successor nodes and the predecessor nodes for each node are stored in the process plan nodes, and we can access to the first level nodes of the process plan networks through the process plan network class. The process plan network class also contains a search function for generating the alternative process plans and for selecting the suitable process plans.
3. Process Plan Selection

Once a process plan network is generated, the next step is to select suitable process plans to be used for the manufacturing. The goal of this task is to find suitable process plans, based on the objective function. At present, we consider only the manufacturing time, and our goal is to find the process plan with the minimum manufacturing time.

It was remarked (2) that the Hamiltonian method enumerates all the possible paths in a process plan network. Hamiltonian path in the graph theory is a path that passes through every vertex only once. Hamiltonian path method was adopted to find the number of paths (7). This method needs long computation time for building whole the process plan network, and is not applicable to large problems. A* algorithm (8) is a general search algorithm that explores the search space in a best-first manner and always guarantee to find the optimum solutions (9,10). The benefit of this algorithm is that the whole process plan network does not have to be generated before the search algorithm starts. The network is generated and expanded along the search path and only the portions of the network are searched.

We consider the fixtures and the fixture changing processes during the process plan generation that increases the complexity and the volume of the process plan networks. The sizes of process plan networks are enormous even for simple example. Therefore, the limited-expansion A algorithm (11,12) is adopted, in this paper, to solve the problem. An algorithm, called limited-expansion A algorithm, is a modified algorithm based on the A* algorithm to limit the search areas, aimed at reducing the computation time. The relation between the limited-expansion A algorithm and the A* algorithm is analogous to that between the beam search algorithm and the breadth-first search which are applied to the routing problems in the graph theory.

The idea of limited-expansion A algorithm is similar to the stage search (13) which assumes that the ONL in the process plan networks given in section 2.2 has a maximum capacity $m$. If there are more than $m$ nodes on ONL, a pruning procedure takes place and only a specified number $b$ of the “best” nodes are kept for further processing, based on the objective function. Obviously, it is not guaranteed that the algorithm finds the optimal process plan, by using the algorithm. If $m$ and $b$ becomes unlimited then the limited-expansion A equals to A*, and it is possible to find the optimal process plan. The following search algorithm is proposed based on the limited-expansion A algorithm, in order to search suitable process plans.

STEP 1: Set search parameters $m$ and $b$. $m$ is the maximum number of the nodes in the ONL. $b$ is the number of the nodes which are kept for searching the process plans

STEP 2: Generate a first level of process plan network and put all the generated nodes in the open nodes list (ONL).

STEP 3: If ONL is empty, terminate.

STEP 4: Choose a node $N_i$ from the ONL, $i \in \{0,1,2,\ldots,|ONL|\}$ with minimal estimation of the manufacturing time and move it from the ONL to the close nodes list (CNL).

- If there are no remaining features in node $N_i$, it is an end node, and constructs a process plan by the path from the start node to the end node.
- Else generate the next level of the process plan network and store the links of the successor nodes in $N_i$, and move all generated nodes to ONL.

STEP 5: If there are more than $m$ nodes in the ONL, only $b$ numbers of the “best” nodes are kept for further processing, based on the estimated manufacturing time.

STEP 6: Go to the STEP 3.

In the above algorithm, the manufacturing time is estimated based on the information of the node $N_i$ by applying the following function.

$$f(N_i) = g(N_i) + h(N_i)$$  \hspace{1cm} (6)
where,

\( g(N_i) \) : Manufacturing time from the initial node to the node \( N_i \).

\( h(N_i) \) : Estimation of the manufacturing time from the node \( N_i \) to the end node, which are estimated based on the shortest manufacturing time of the remaining machining features.

4. Experimental Results

Experimental results are presented here to verify the effectiveness of the proposed algorithm. Consider the flexible manufacturing system shown in Fig. 4, for producing several parts including four cases shown in Fig. 5. The detail information of machining features and the machining equipment of all four case studies are brought in the previous paper (14).

A computer program is prepared by C++ for generating the process plan networks and
the alternative process plans. The objective function is set to be the manufacturing time. We apply A* algorithm for finding the optimal manufacturing time of the case studies. When the size of process plan networks are too large, we apply the limited-expansion A algorithm to find the suitable process plans.

4.1 Results from A* algorithm

The results of the A* algorithm are compared with the previously proposed algorithm that applies Genetic Algorithm (GA) and Dynamic Programming (DP) for process planning\(^{14}\). Table 2 summarizes the minimum manufacturing time and the computation time for each case of Fig. 5. The proposed algorithm generates optimum solution for all the three cases of Fig. 5, and dominates the previous algorithm. The proposed algorithm also generates all alternative process plans and sorts them based on the manufacturing time.

4.2 Results from Limited-expansion A algorithm

When the sizes of process plan networks are too large to find the optimum solutions, it is required that the search algorithm expands different portions of the process plan networks and finds suitable process plans. The sizes of the process plan networks are mainly related to the following parameters.

(1) Number of machining features.
(2) Number of available machining equipment sets for each machining feature.

Consider the shaft shown in Fig. 5(d), it has 23 machining features, and 3 machining equipment sets in average for individual machining features. Therefore, the process plan network becomes very huge, and it takes long time to find optimum solution. The search parameters of \(m\) and \(b\) are set to reduce the computation time. Five cases are considered by

<table>
<thead>
<tr>
<th>Case</th>
<th>Proposed algorithm</th>
<th>Shortest Manufacturing Time by (GA + DP) algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimum Manufacturing Time by A* Algorithm</td>
<td>Computation Time of A* Algorithm (s)</td>
</tr>
<tr>
<td>a</td>
<td>8211</td>
<td>1.5</td>
</tr>
<tr>
<td>b</td>
<td>20057</td>
<td>1.2</td>
</tr>
<tr>
<td>c</td>
<td>7662</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2 Generated process plans for different cases of Fig. 5.

<table>
<thead>
<tr>
<th>Trail number</th>
<th>Size of (m) set</th>
<th>Size of (b) set</th>
<th>Minimum manufacturing time</th>
<th>Number of generated process plans</th>
<th>Number of generated nodes of process plan network</th>
<th>Computation time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
<td>500</td>
<td>8155.8</td>
<td>2177</td>
<td>88855</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>2000</td>
<td>1000</td>
<td>7526</td>
<td>2782</td>
<td>214403</td>
<td>66</td>
</tr>
<tr>
<td>3</td>
<td>3000</td>
<td>1000</td>
<td>7522.3</td>
<td>4436</td>
<td>216961</td>
<td>86</td>
</tr>
<tr>
<td>4</td>
<td>7000</td>
<td>2500</td>
<td>7838.3</td>
<td>6352</td>
<td>517971</td>
<td>572</td>
</tr>
<tr>
<td>5</td>
<td>10000</td>
<td>3000</td>
<td>7202.6</td>
<td>14551</td>
<td>608705</td>
<td>883</td>
</tr>
</tbody>
</table>

Table 3 Different trial of generating process plan for part (d) of Fig. 5.
changing the \( m \) and \( b \) sizes. Table 3 summarizes the minimum manufacturing time and the number of the generated process plans, at each trial. The proposed algorithm provides the users with better solution by increasing \( m \) and \( b \) sizes, although the computational time is increased.

In the trial number four, the best solution is not improved in comparison with the trial number three, although the sizes of \( m \) and \( b \) are increased. The sizes of \( m \) and \( b \) are limited and the algorithm contains a limited area of the process plan networks for finding local optimum in that area. In the trial number four, the search algorithm enters to a new portion of the process plan networks that is different from the previous trial. The best solution in new portion is worse in comparison with the best solution of the previous trial. By increasing \( m \) and \( b \) in trial number five, the algorithm enters to another portion of the process plan networks that contains better solutions. The following guidelines are obtained through the experiments.

1- Start with small values for \( m \) and \( b \) (\( bm \geq \))
2- Increase the values of \( m \) and \( b \) until the following conditions are satisfied.
   - The solutions are not significantly improved by increasing the values \( m \) and \( b \).
   - The calculation time becomes too much.

The main goal of the developed algorithm is to expand the different portions of the process planning networks and to find the local optimum in that area. This is an important issue in the point of the practical use and the user can generate variety of alternative process plans based on the manufacturing system status and the due dates.

5. Conclusion

This paper deals with an extension in the process plan networks and the domain of the process planning is expanded by adding the fixture layer to the process planning. A class diagram is proposed for process planning and generating the process planning networks. A search algorithm is developed for generating and searching the process plan networks. The benefit of this algorithm is that the whole process plan networks do not have to be generated, and the networks are generated and expanded only along the search paths. Some case studies have been done to evaluate and to compare the algorithm with the previous one. In most cases, the proposed algorithm generates the optimum solutions. The algorithm is efficient for the large and vast process plan networks that the previous search algorithms fail to handle it. The user can expand the different portions of process plan networks by adjusting the search parameters.

This paper deals with only the first step for developing a dynamic and real time process planning and scheduling algorithm for the flexible manufacturing systems. The autonomous agents are able to initiate actions, negotiate and make decisions for the process planning and control tasks.

References


