Setup Planning and Operation Sequencing Using Neural Network and Genetic Algorithm

R.S Joshi
RIMT-IET, India
ravindersinghjoshi@yahoo.com

N.kumar
TU, India
navdeepgarg_tiet@yahoo.com

Anju Sharma
TU, India
anjusharma@thapar.edu

Abstract

In process planning setup planning and operation sequencing are the major issues. In setup planning feature having same approach direction & tool commonality are grouped into setup. After making different setups operation sequencing in each setup is done for setup planning neural network is efficient. Operation sequencing problem is converted into the traveling salesman problem in which objective function is to reduce total cost. To solve these issues in an efficient manner Genetic Algorithm technique is more suitable because it is a viable means for searching the solution space of operation sequence providing a computational time on the order of a few seconds. The present work generates the results for the prismatic parts. The presented algorithm for setup planning and operation sequencing is efficient enough for its use as a module within the development of a CAPP system.

Key Words- Unsupervised learning, Genetic algorithm, CAPP, Setup planning, Operation sequencing.

1. Introduction

The process planning function of a manufacturing system is responsible for the transformation of product design specifications into process specifications. Process planning has been recognized as an interface between computer-aided design and computer-aided manufacturing. Automated process planning in a manufacturing environment is vital to achieving the ultimate goal of integrated factories in the future. Neural networks, characterized by their learning ability, provide a promising approach for automated knowledge acquisition and can be advantageously used in the building of automated process planning systems.

Operation sequencing which denotes allocation of operations in specific order is always major concern in process planning. The operation sequences generally obtained are evaluated using a number of objectives such as shortest time or minimum cost with constraints etc., and there has been a great effort to optimize the operation sequencing.

In real situations, the operations comprising the job can be performed with different tools, fixtures, setups and even other machines.

The main objective of the researchers lie in grouping for minimum number of setups and sequencing within the same setup for minimum non cutting time or cost. The grouping and sequencing are performed separately with an assumption that there is no limitation of processing time in each setup. Procedure must be such that it can perform following tasks:

Grouping of feature or setup planning
Sequence the operations in one chain
Consider the precedence relationships
Consider the tool commonality

Typical manufacturing features include pockets, slots, holes etc. while presenting part to cutting tool for machining there are more than on ways depending upon the holding devices. So this results in different setups to access all the features more than on setup may be required some feature may be accessible from more than one direction, so it is responsibility of process planner to make the group of these features according to different approach direction to reduce non–cutting time. After setup planning operation sequencing in each setup is done.

2. Literature Survey

A lot of work has been done in the field of CAPP and its activities. A number of researchers have contributed in the development of an efficient process planning technique. Various researchers have discussed regarding the modeling constraints, solving problem of operation sequencing and setup planning using different approaches. The main aim of doing this literature survey is to explore the different

978-0-7695-3099-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ITNG.2008.94

techniques used for operation sequencing and setup planning.
Laura I Burke and Poulomi Damany [8] introduced a new neural network approach for solving the traveling salesman problem (TSP). Hopfield's seminal effort to use a particular type of neural network (recurrent network) to solve a 30-city traveling salesman problem has been used lo support the case that neural networks can solve such technically difficult problems. Looking closer, it is apparent that the elegant approach lacked practical appeal (even for 30-city problems, feasibility could not be ensured).
John M.Usher and Royce O.Bowden [10] presented that Operation sequencing has long been a difficult problem in process planning. As part complexity increases, the number of potential solutions increases exponentially. An approach to operation sequence coding that permits the application of genetic algorithms for quickly determining optimal, or near-optimal, operation sequences for parts of varying complexity. This approach improves on existing techniques by utilizing common sequencing constraints to guide the coding process resulting in a further reduction in the size of the solution search space.
I.T.Kim and H.W.Suh [11] investigate the problem of determining the optimal group and sequence of operations for a multistage machining system. A method incorporating a combination of the expert system and mathematical programming is proposed to produce an optimal operation sequence minimizing the non-cutting time.
J.Chen , Y.F.Zhang and A.Y.C.Nee [13] developed an new approach to setup planning of prismatic parts using Hopfield neural net coupled with simulated annealing.
H.C.Wu and T.C.Chang [12] proposed that setup planning is a critical part of automated process planning. Setup is determined by several factors, tool approach direction and tolerance specification are two major ones.
C.L.Philp Chen and Steven .R.LeClair [9] developed a methodology of using an unsupervised learning approach for setup generation and feature sequencing. Setup generation is a pivoted step in automated process planning as it greatly influence machine/tool selection, machining sequences and fixturing configuration. The STTD (similar-tool-type-direction discovery) architecture and SADD (similar-approach direction discovery) incorporate multiple objective functions into setup generation. Intersecting and non intersecting features within a setup are identified and classified using associative memory. After the feature sequence has been determined, algorithm likes discover-and-merge and optimal tool sequence is proposed to obtain the best sequence for creating the features in setup. From the literature survey the following observations are made:-

- Use of mathematical programming techniques, is not suitable from computational point of view due to use of large number of variables and equations.
- Formulation of Computer Aided Process Planning depends on part complexity. As part complexity increases it becomes critical activity which requires various optimization techniques.
- In CAPP operation sequencing is one of most critical and time consuming activity for manufacturing a part.
- On the other hand the use of Hopfield net is also limited because it adaptive only at the run time.
- Simulated annealing incorporated with Hopfield net to search towards global searching.
- Setup planning is required because there can be two same features but requires change in setup, so to avoid this grouping of feature must be done before operation sequencing.
- In CAPP system, an efficient search is required to explore the large solution space of valid operation sequences under various interacting constraints.

3. Methodology and algorithm

To satisfy the above criteria, the present work should be exported by following the steps mention below:-

- First the part drawing is analyzed to identify the form features to be machined, with details of geometric specifications and technological information.
- Then a table is made which shows tool approach directions and tools required for the features. Tool approach directions and tool patterns are the feature values of features.
- Then setup planning i.e. grouping of feature is done according to tool approach direction and tool commonality.
- After making setups operation sequencing in each setup is done.
- The precedence relationship among the machining operation is obtained. Considering various feasibility constraints.
- Determine the best sequence of operations for the prevailing production environment by utilizing the application of genetic algorithms.

3.1 Unsupervised learning approach for setup planning

It is proposed to use unsupervised approach for setup planning. The approach direction of a feature is an unobstructed path that a tool can take to access feature attachment face in the workpiece. Some features may have more than one approach direction. Because of the possibility of multiple approach directions one feature may be grouped to more than one cluster. The problem of clustering can be solved using the suggested approach of unsupervised learning. Suppose that a feature F_i needs a set of tools Ti to process, we represent a set of tools Ti by an ordered tuple < ${}^{i}t_1$, ${}^{i}t_2$, ${}^{i}t_3$….,${}^{i}t_n$ > each element in the n-tuple represents the tool needed for the processing of the feature F_i, we define binary values 0 or 1 for the value of each element. If the value of the element is 1, the corresponding tool is needed for the processing of the feature F_i, otherwise the corresponding tool bit is set to 0. An approach direction is determined similarly. If we fix the starting feature of material to be machined in a fixed orientation, we can define the approach direction of a feature by six digits of ordered binary pattern <+x, +y, +z, -x, -y, -z>. if a digit is 1, the feature can be made using the corresponding approach direction to process it. The feed direction of a feature (i.e. the direction of the tool motion necessary to cut the feature) can be determined on the basis of the geometry of the feature and the workpiece material.

Unsupervised learning type of neural network groups input on the basis of similarity of their feature values. Feature values to be considered for the setup generation are approach direction and tool pattern defined.

Weight update: AND and OR update rules [9]

To determine clusters of the features that have the same approach direction and common tools, we represent a feature by 6+ n binary ordered patterns. We divide neural network into two subnets: an approach direction subnet and tool subnet. Each subnet has its own weight updating rule. For tool approach direction the logical AND operation is used, while for the tools subnet, we use logic OR operation.

In AND update rule if we have an approach direction subpattern<1, 0, 1, 0, 0, 0>, which corresponds to an approach from +x and +z directions, and it forms a cluster. If other subpattern with an approach direction <1,1,0,0,0,0> enters the network, the updated weight is <1,0,0,0,0,0>, which means that this cluster has a common approach direction on the +x direction.

The approach direction subpattern dominates the cluster formation. Two features cannot be clustered together if two features have common tools but not common approach directions. The SADD (similar –approach-direction discovery) subnet takes the approach direction subpattern as input and temporarily assigns the feature patterns to clusters that have similar approach directions.

The STTD (similar-tool-type-direction discovery) subnet takes the tool type subpattern as input, and assigns the feature patterns to cluster that store the information of both similar approach directions and maximum tool type overlap. Similar to AND rule OR operation is also applied to bit value. For example if a tool subpattern<1, 0, 1, 0, 0, 1>, then this feature requires tools t_1, t_3 and t_6. Suppose this tool set form a cluster. If another subpattern with a similar tool subpattern <0,1,1,0,0,0>,enters the network , and is assigned to this cluster, the weight updated is <1,1,1,0,0,1>, which means that this cluster need tools t_1,t_2, t_3 and t_6. if there are other pattern whose subpatterns share this information, we can assign these pattern to the same cluster on the basis of the similarity measure and the threshold value. Figure 1 shows proposed architecture for neural network.

3.2 Genetic Algorithm for operation sequencing [14]

Procedure for genetic algorithm:-
Initialization: Randomly generate a population.
Evaluation: The each individual by using the objective function; compute a fitness value that measures how well the individual optimizes the function.
Parent selection: Choose pairs of individuals from the population in such a way that those with higher fitness values will be chosen more frequently.
Reproduction: Reproduce children from each pair of parents. Each parent contributes halls of its genetic makeup to each child.
Mutation: Randomly change a tiny amount of the genetic information n each child.

3.3 Assumption

Operations used are only prismatic components but the first module of i.e of setup planning can be used for both rotational as well as prismatic parts.
Maximum no. of groups will not be more than 6.
For grouping tool approach direction is preferred as compared to tool patterns.

The no. of operations in a group should not be more than 20.
If any operation having same specification, than these operations are considered as one operation. For example:- two hole having same size, requiring the same tool or if there are two steps than these two are considered to be one.
We take probability of both crossover and mutation as 1 so that the numbers of generation of feasible string are less and thus it will take less computer time.

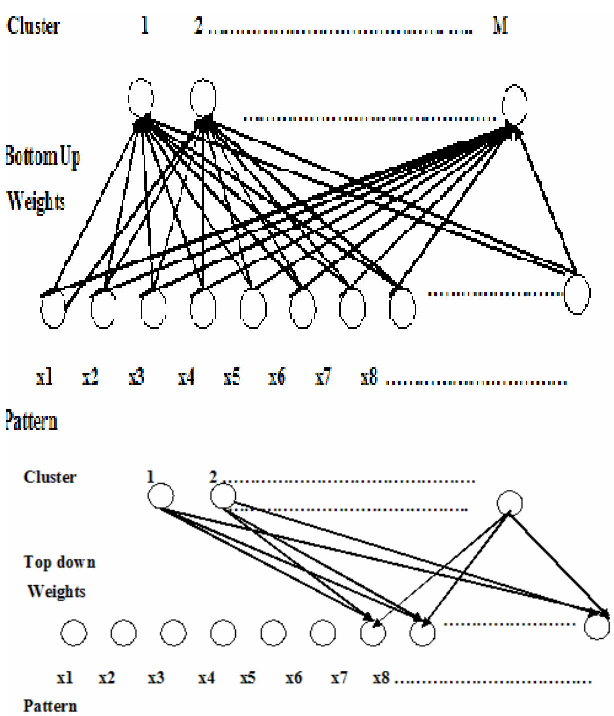

Figure 1: Proposed architecture of neural network

3.4 Algorithm

1. Let ${}^a b_1 = {}^a b_{i1}(0) = {}^a x_i^{(1)}$, ${}^t b_1 = {}^t b_{i1}(0) = {}^t x_i^{(1)}$, $p \leftarrow 2$, and set the similarity measures =1, where ${}^a b_{ij}$ is the connective weight between the input ${}^a x_i$(i.e. the approach direction) and the cluster j, and ${}^t b_{ij}$ is the connective weight between the input ${}^t x_i$(i.e. the tool type) and the cluster j.
2. Apply a new input $x^{(p)} = ({}^a x^{(p)}, {}^t x^{(p)})$, and set the top down weight $v_m = v_{im} = 0$, m=1,2,….M.
[Apply a new input $x^{(p)}$]
3. Select all the possible clusters for which

$$\sum_{i=1}^{Na} {}^a x_i(p) *^a b_{im} >= \Gamma$$

For m=1,2,….M, where m is the number of clusters that have been discovered. Let j be the number of selected clusters, and set $v_{im}(t) = {}^t b_{im}$, m=1,2,….J. store this patter to these clusters temporarily, and continue to step 4; otherwise, no such clusters can be found; go to step 6. in this step, the approach direction of all the clusters that match with the given input, even with only one digit are selected. The top down weights of the selected clusters v_{im}, m=1, 2, …J. and set to ${}^t b_{im}$.
[Compute the similarity and threshold test.]
4. The decision for assigning the input to a cluster is performed as follows.

$$y_m = \sum_{i=1}^{Ni} v_{im} *^t x_{i(p)} \ldots\ldots\ldots\ldots m = 1,2\ldots\ldots.M$$

In this step, the selection of the best matching existing cluster j is performed according to the maximum criteria as follow.

$$y_j = \max y_m \ldots\ldots\ldots m = 1,2\ldots\ldots\ldots\ldots..j$$

[Discover a cluster on the basis of tool commonality.
5. Record this pattern to the cluster (i.e. j) and update

$${}^a b_{ij}(s+1) = {}^a x_i(p) *^a b_{ij}(s)$$

$${}^t b_{ij}(s+1) = f({}^t x_i(p) + {}^t b_{ij}(s))$$

Where f (n) =1 if n>=1; otherwise, f (n) =0. In this step, only the weight of the cluster j with the maximum matching tools is updated. If all the input patterns have been applied, stop; otherwise set $p \leftarrow p+1$ and go to step 2.
[Update the connective weight.]
6. Let ${}^a b_{ik}(0) = {}^a x_i^{(p)}$, ${}^t b_{ik} = {}^t x_i^{(p)}$. If all the input patterns have been applied stop; otherwise set $p \leftarrow p+1$, $k \leftarrow k+1$ and go to step 2.
[Create a new cluster k.]
7. Now operation sequencing of each group is done.
8. Making the standard operation library.
9. Entering the number off operations.
10. If the numbers of features are more than 20, go to step 9.
11. Entering the operation code from the standard operation library.
12. If operation code entered is out of operation library, go to step 11.
13 Generation of penalty cost matrix for all the operation entered by the user.
14. Sorting all the operations in increasing order.

15. Grouping of operations based on their relationship(for example hole and boring comes under one group).
16. Identification of form operations from the step 15 based on their precedence relationship (for example boring can be done after making of hole).
17. Randomly Generation of all feasible string based on there precedence relationship (initialization).
18. Calculation of cost of each feasible string.
19.Finding out the string which has maximum cost [M].
20. Calculating the fitness value for each string. $(u_{max})_i=M-(u_{min})_i$
21. Calculating the actual count for each feasible string. $E_i=(u_{max})_i/u_{avg}$ (if Ei is in decimal than round it off)
22. Selecting two strings (parents) randomly from the feasible string. 16. Check if two parents have same cost. If 'yes' go to step 18 if 'no' go to step 19.
23. Mutate.
24. Crossover.
25. If after crossover the cost any of parent 1, parent 2, offspring 1, and offspring 2 is same then go to step 17 else continue.
26. Repeat the step 12 to step 20 for a specified no. of generations. At the end of the last generation, the string(s) to the min value is taken as the optimal sequence.
27. If group is left than go to step 8 otherwise stop.

4. Results and discussion

Now let us take a case study by C. L. Philip Chen et. al. [9] shown in figure 2. Feature of the part is labeled as f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11 and f12.

Tool approach direction and tools required for each Feature are shown
in Table 1. Feature 1-4 are open pockets, feature 5 is shoulder, features 6 and 7 are blind holes, features 8 to 11 are through hole, and feature 12 is a shoulder. Total no: of feature are 12.

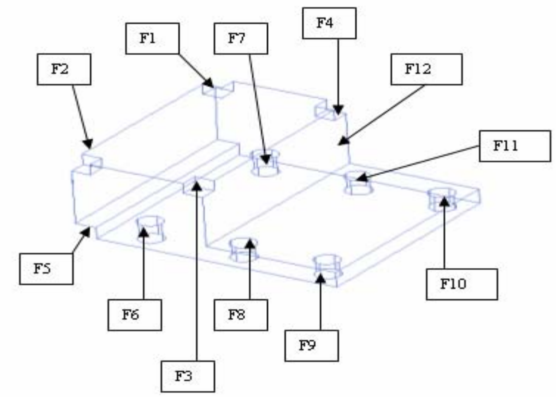

Figure 2: Example part

1	F1,F2,F3,F4,F12	T1,T2,T4,T5 T7,T8
2	F5,F6,F7,F8,F9 F10,F11	T1,T2,T3,T4 T5,T6

Table 2: Final outputs after grouping

Feature	+X	-X	+Y	-Y	+Z	-Z	T1	T2	T3	T4	T5	T6	T7	T 8
f1	1	0	1	0	1	0	1	1	0	1	0	0	0	1
f2	0	0	1	1	1	0	1	1	0	1	0	0	0	1
f3	1	0	0	0	1	0	1	1	0	1	0	0	0	1
f4	0	0	0	1	1	0	1	1	0	1	0	0	0	1
f5	1	1	1	1	0	0	1	1	0	1	0	1	0	0
f6	0	1	0	0	0	0	1	0	1	1	0	0	0	0
f7	0	1	0	0	0	0	1	0	1	1	0	0	0	0
f8	0	1	0	0	1	0	1	0	1	1	0	1	0	0
f9	0	1	0	0	1	0	1	0	1	1	0	1	0	0
f10	0	1	0	0	1	0	1	0	1	1	0	1	0	0
f11	0	1	0	0	1	0	1	0	1	1	0	1	0	0
f12	1	0	0	1	1	1	0	1	0	0	1	1	1	1

Table 1: Tool approach directions and type of tools for feature

Table 2 shows final output after grouping.
After making groups now the operation sequencing is done for each group. Tool required for each group is converted into operations because each tool shows one operation. A code no. is assigned to operation based on there precedence relationship. The iterative process is going on till the optimal sequence is reached. Table 3 shows operation code for group 1. Optimal operation sequence after iterations is 0-1-4-14-2-10 with a total relative cost of 5 units.

Operation	Code
Plane milling (A)	0
Rough milling (B)	1
Plane milling (C)	10
Slot cutting (D)	4
Slot cutting (E)	14
Finish milling (F)	2

Table 3: Operation codes for group 1 is

Operation	Code
Plane milling (A)	0
Rough milling (B)	1
Slot cutting (C)	4
Hole (D)	5
Rough boring (E)	6
Plane milling (F)	10

Table 4: Operation codes for group 2

Table 4 shows operation codes for group 2. Optimal operation sequence after iterations for group 2 is 0-1-4-10-5-6 with a total relative cost of 5 units.

5. Scope of future work

The present work provides efficient results for setup planning for prismatic components only and can be further enhanced by the following ways:-

- The algorithm can be extended for rotational components also the cutting speed and feed can also been taken into account.
- The generation of setups not only on the basis of tool type and approach direction, but also on tolerance information, materials, machining operations.
- The present work can also be interfaced with clamping sequence problem of prismatic components.
- This algorithm can be integrated with an efficient CAD oriented model. This algorithm directly picked the features described by the feature based CAD model.
- It can also be interfaced with fuzzy logic. With the implementation of fuzzy approach, the precedence graph analysis can be easily interpreted and this will further enhance the performance of Genetic Algorithm to find out the optimal sequence in very short of time.

References:

[1] Simon Haykin, “Neural Networks” Sai Printopack Pvt. Ltd. New Delhi, 2004.
[2] P.N. Rao, N.K. Tewari and T.K. Kundra, 'Computer Aided manufacturing', Tata McGraw Hill, New Delhi 2003.
[3] Mikell P. Groover and Emory W. Zimmers, "CAD/CAM : Computer -Aided Design and manufacturing" PHI, New Delhi 2000.
[4] M.J. Zurada, "Introduction to Artificial Neural System “ Jacbio Publishing Mouse, Delhi 1999.
[5] Tien-Chein Chang, Richard A.Wysk, Hsu-Pin Wang “Computer Aided Manufacturing “ Prentice Hall of India, 1998.
[6] Kunwoo LEE “Principles of CAD/CAM/CAE Systems” Addition-Wesley 1993.
[7] Kevin Knight, “Artificial Intelligence” Tata McGraw-Hill Publishing New Delhi, 1991.
[8] Laura I.Burke, Poulomi Damany “The Guilty Net for the Traveling Salesman Problem” *Computer Operation Research*, Vol.19, No. 3, pp 255-265, 1992.
[9] C.L.Philip Chen, Steven R LeClair “Solving Setup Generation and Feature Sequencing Using An Unsupervised-learning approach” *Computer Aided Design*, Vol.26, No.1, pp 59-75, 1994.
[10] John M.Usher, Royce O.Bowden “The Application of Genetic Algorithms to Operation Sequencing for Use in CAPP” *Computer Industrial Engineering*, Vol.30, No.4, pp 999-1013, 1996.
[11] I.T.Kim, H.W.Suh “Optimal Operation Grouping and Sequencing Technique for Multistage Machinig Systems” *International Journal of Production Research*, Vol.36, No.8, pp 2061-2081, 1998.
[12] H.C.Wu, T.C.Chang “Automated Setup Selection in Feature-Based Process Planning” *International Journal of Production Research*, Vol.36, No.9, pp 2325-2341, 1998.
[13] J.Chen, Y.F.Zhang, A.Y.C.Nee “Setup Planning Using Hopfield Net and Simulated Anealing” *International Journal of Production Research*, Vol.36, No. 4, pp 981-1000, 1998.
[14] Whitely, D., "A Genetic Algorithm Tutorial", Technical Report CS-93-103, Colorado State University, 1993.