Journal of the Korean Institute of Industrial Engineers
Vol.24, N0.1, March, 1998

Network Analysis and Neural Network Approach for the Cellular Manufacturing System Design

Network 분석과 신경망을 이용한 Cellular 생산시스템 설계

Hong-chul Lee*

이홍철*

Abstract

This article presents a network flow analysis to form flexible machine cells with minimum intercellular part moves and a neural network model to form part families. The operational sequences and production quantity of the part, and the number of cells and the cell size are taken into considerations for a 0-1 quadratic programming formulation and a network flow based solution procedure is developed. After designing the machine cells, a neural network approach for the integration of part families and the automatic assignment of new parts to the existing cells is proposed. A multi-layer backpropagation network with one hidden layer is used. Experimental results with varying number of neurons in hidden layer to evaluate the role of hidden neurons in the network learning performance are also presented. The comprehensive methodology developed in this article is appropriate for solving large-scale industrial applications without building the knowledge-based expert rule for the cellular manufacturing environment.

1. Introduction

A cellular manufacturing system which implements the Group Technology (GT) principle is the decomposition of the manufacturing system into subsystems to minimize intercellular interactions. This objective is usually achieved by grouping the machines into cells and the parts into families on the basis of similar processing requirements. In this context, a part family consists of parts requiring similar machine operations and machines within a cell work only on parts belong to the family assigned to that cell. The concept of GT provides the essential means for successful development and implementation of Computer Integrated Manufacturing (CIM) through the integration of CAD/CAM and the part family concept. Moreover, it can be successfully applied in the implementation of flexible manufacturing systems (FMS). An FMS is a totally programmable manufacturing cell dedicated to the production of a part family.

Numerous algorithms in the construction of machine

* Department of Industrial Engineering, Korea University

and part groups have been developed on the basis of GT classification and coding system [9][29], similarity-coefficient based methods [20][23][24], graph theoretic and network based method [22][28], mathematical programming formulations [4][15]and matrix based heuristic methods [6][11]. Kusiak[17] conducted a good review of each methodology developed over the last two decades. Most of all methods mentioned above are computationally inefficient and infeasible in case of large-scale industrial applications. In addition, those methods utilized the machine-part matrix indicating which machines are needed for processing each part (in this matrix an entry a_{ij} = 1 if machine j is required to process part i, a_{ij} = 0, otherwise) for the production flow analysis [5] as their input. The machine-part matrix does not consider two important factors such as the sequence of operations and production volume of the parts for the more realistic and accurate design approach.

From the current literature on the machine-part grouping problem, it can be seen that there are two methods that are generally followed : identify machine groups and part families independently and identify machine-part groups simultaneously. Once one takes into account the operation sequences of the parts, simultaneous grouping is not possible. Even when only the machine requirement (machine-part matrix) is considered, the machine-part grouping problem becomes complicated when the number of machines and the number of parts become large. The number of parts to be processed in a factory is usually more than and more likely to vary than the number of machines. Hence, it would be desirable to form part families after machine cells are identified.

Once machine cells have been formed, it is necessary to develop method to support the on going decision making concerning new parts assignment to the most appropriate existing machine cell. Automatic new part assignment requires the knowledge of similarity between processing requirements of a newly introduced part and the existing ones. To perform this task, artificial intelligence (AI) techniques such as knowledge based expert systems [8][16], the syntactic pattern recognition approach[27] have been developed. These methods used the machine-part matrix as their input for analysis. A knowledge based expert system depends on a precisely defined knowledge base with remarkable degree of accuracy. The problem of AI techniques [25] is that they are not capable of generalization if new parts do not fall into the existing domain-specific knowledge base.

A neural network [12] defined as a cognitive information processing technique consisting of neuron-like units (neurons) and the weighted connections between neurons, however, is able to generalize. Generalization [26] in neural networks is defined as the network's ability to perform the mapping of similar inputs, not contained in the training set, to an output. Some neural network models are adept at pattern classification in the presence of noise and incomplete data [13][19]. Direct application of neural networks for machine/part grouping based on machine-part matrix [3][7][18] is possible, however, under considerations of operational sequences with production volume, direct application of neural networks may not be possible. Therefore, the machine cell formation comprising all considerations is the first stage to generate a training set of neural networks for developing the part family formation and the new part assignment procedure.

This article presents a comprehensive methodology for the analysis of flexible manufacturing cell systems. A 0-1 quadratic programming is formulated to obtain machine cells with the objective of minimizing intercellular part moves. The optimal cell formation takes into account important parameters, such as the operational sequences of the parts, the production quantity of each part, the number of desired machine cells and cell size. The network flow based heuristic is developed to solve this 0-1 quadratic programming formulation. Network flow models and solution techniques can be of great value in the design, improvement, and rationalization of complex large-scale systems. The advantages of using network models include the accurate representation of real word systems and the extremely efficient solution procedures of network algorithms to some large-scale models due to the exploitation of particular structures in a model. State-of-the art optimization procedure is used for enhancing the computational efficiency of the proposed methodology. The information of cell formation is then used as a training set (input) of neural network for learning process. The output of the trained neural network is the identification of part families and the assignment of new parts to the existing machine cell.

This article is organized as follows. The problem under consideration is formulated in Section 2 and the corresponding network flow based solution procedure is developed along with an illustrative example in Section 3. The detailed methodology of neural network approach for the part family formation is presented in Section 4. Finally, the conclusions of this article are presented in Section 5.

2. Problem Formulation

The major issue in GT problem, that of finding the part family requiring similar machine operations and forming the associated group of machines, is actually the cluster analysis. Due to high combinatorial nature of clustering problems, grouping machines is better than grouping parts. As an illustration, suppose there are 100 parts processed by some of 20 machine types, then the number of distinct partitions of 100 parts into 5 part families is approximately 10^{67} but that of 20 machines into 5 machine cells is approximately 10^{12}. Therefore, machine cells should be formed prior to the part family formation.

Direct mathematical modeling based on the operation sequence and production volume of parts may not be possible. Therefore, a 0-1 quadratic programming is formulated to minimize intercellular movements on the basis of the material flow matrix transformed from the given information, i.e., the operational sequences and production quantity of the parts. The detailed procedures of construction of the material flow matrix and the mathematical programming formulation, along with an illustrative example are described in the remaining portion of this section.

2.1 Construction of material flow matrix

The objective of this subsection is to describe a relationship for computing the material flow between machines i and j based on the operation sequences and part volume. The sequence of operations indicates the order and types of machines needed. The basic concept used in the computation of material flow follows. We consider M machines and N parts. Let η_{ij}^{k} be defined as follows:

$$\eta_{ij}^{k} = \begin{cases} 1, & \text{if } k\text{th part is processed at machine } j \text{ immediatel} \\ & \text{y after machine } i \\ 0, & \text{otherwise} \end{cases}$$

Additionally, let v_k be the production volume of the k^{th} part. The total material flow for all N parts from machine i to machine j can now be computed as indicated below:

$$f_{ij} = \sum_{k=1}^{N} v_k(\eta_{ij}^{k} + \eta_{ji}^{k}) \qquad i,j=1,2,\ldots,M \tag{1}$$

From the computation of the f_{ij} values in terms of Eq. (1), an MxM material flow matrix F can be obtained. To illustrate the procedure for calculating material flows, five parts (P1, P2, ..., P5) with the following operational sequences on five machines (1,2,...,5) and the given production volumes will be considered:

P1:	2-3-1-4-5	v1=2
P2:	2-5-1-4	v2=3
P3:	1-4-3-5-2	v3=1
P4:	5-3-4-2	v4=3
P5:	4-1-5-3	v5=1

The material flow matrix F can be computed using Eq. (1) by counting the movements of all parts between machine i and machine j. For example, the number of material moves between machine 1 and machine 4 is

$$
\begin{aligned}
f_{ij} &= \sum_{k=1}^{s} v_k(\eta_{14}^k + \eta_{41}^k) \\
&= 2(1 + 0) + 3(1 + 0) + 1(1 + 0) + 3(0 + 0) + 1(0 + 1) \\
&= 7
\end{aligned}
$$

All other f_{ij}'s are computed to determine the following material flow matrix F:

$$
F = \begin{bmatrix} 0 & 0 & 2 & 7 & 4 \\ 0 & 0 & 2 & 3 & 4 \\ 2 & 2 & 0 & 4 & 5 \\ 7 & 3 & 4 & 0 & 2 \\ 4 & 4 & 5 & 2 & 0 \end{bmatrix}
$$

As indicated earlier, this matrix represents the number of part moves from one machine to another. The objective of cell formation is to minimize the intercellular movement of material so as to reduce the set up times and material handling costs. In order to meet this objective, machines having high number of part moves should be grouped together and machines having low number of part moves should be separated. Based on matrix **F**, the Intermachine Material Flow (IMF) network with nodes and arc weights representing, respectively, the machines and the number of part moves between machines can be obtained.

2.2 Problem Formulation

The problem of machine cell formation is formulated as a 0-1 quadratic programming formulation originally developed by Kumar *et al.* [14] for the network decomposition problem. Let **F** be a M x M matrix where M is the number of machines and an element of **F**, f_{ij} denotes the number of part moves between machine i and machine j. Let p be the number of desired cells and u and l be the upper bound and the lower bound on the total number of machines in each cell. With this notation, a 0-1 quadratic programming formulation for the problem under consideration is shown below:

$$
\text{Maximize} \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \sum_{l=1}^{p} f_{ij} x_{il} x_{jl} \tag{2}
$$

subject to

$$
\sum_{j=1}^{p} x_{ij} = 1, \qquad i = 1,\dots,M \tag{3}
$$

$$
l \le \sum_{i=0}^{M} x_{ij} \le u, \qquad j = 1,\dots,p \tag{4}
$$

$$
x_{ij} \in \{0, 1\}, \qquad \forall\ i, j \tag{5}
$$

where

$$
x_{ij} = \begin{cases} 1, & \text{if machine } i \text{ is in cell } j \\ 0, & \text{otherwise} \end{cases}
$$

This formulation maximizes the sum of all number of part moves that belong to the p cells. Since machine i and machine j in a cell are represented by the node product $x_{il}x_{jl}$, machines i and j are included in a cell if and only if $x_{il} = x_{jl} = 1$. Constraint set (3) ensures that each machine i is in exactly one cell and constraint set (4) restricts the number of machines in each cell. Constraint (5) imposes integrality.

It is unlikely that an efficient exact algorithm will ever be found because this model is *NP-Complete*. A practical solution to the problems can be obtained by the network flow based heuristic. In the next section an efficient solution procedure is presented.

3. Network Flow Based Solution Procedure

The basic network flow concept to solve the mathematical formulations is as follows. A network $G = (N, A)$ is said to be *bipartite* if node set N can be partitioned into disjoint subsets N_a and N_b such that each of its arcs has one endpoint in N_a and the other in N_b. Node i in the original network is split into two nodes i_a and j_b to transform the original network into the *bipartite* network having $2M$ nodes ($N_1=\{i_a\}$ and $N_2=\{i_b\}$) and directed branches (i_a, j_b), $i, j = 1,2,\cdots,M$. The solution procedure will be developed after representation of the *bipartite* network as a capacitated circulation network. In order to solve the cell formation problems using a network flow approach, the following a capacitated circulation network construction procedure can be conducted:

3.1 Network Transformation Procedure

1) Create $2M$ nodes represented by $i_a, j_b, j = 1,2,\cdots,M$, and a super source node S and a super sink node T.
2) Structure the rest of the network according to the following rules. Each arc is assigned three values U, L, C to indicate an upper bound on its flow, a lower bound on its flow and the per-unit cost of flow, respectively.
   ① For each node i_a, create an arc directed from i_a to j_b with capacity-cost triplet defined by $[U, L, C]$ = $[1, 0, -c_{ij}]$, $i \neq j$ and $[U, L, C]=[1, 0, 0]$ for $i=j$.
   ② For each node i_a, create an arc directed from S to i_a with capacity-cost triplet defined by $[U, L, C]$= $[1, 1, 0]$.
   ③ For each node j_b, create an arc directed from j_b to T with capacity-cost triplet defined by $[U, L, C]$= $[\infty, 0, 0]$.
3) Create a return arc (S, T) with capacity-cost triplet $[U, L, C]=[M, M, 0]$ to transform the network $G=(N, A)$ into a circulation network.

A circulation network representation for the cell formation problem is shown in Figure 1. As can be seen in the figure, the network configuration is the capacitated circulation network which can be solved by the minimum cost flow algorithm. In this network model, the flow along arc (S, i_a) is always one such that each source node i_a can supply one unit of flow to one destination node j_b by the flow conservation condition. The cost of shipping

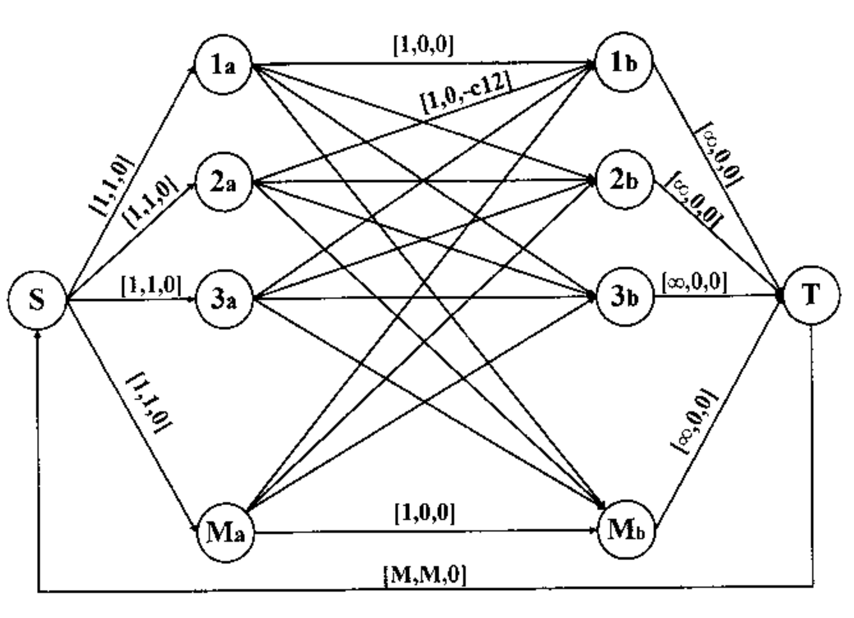

Figure 1. Initial Capacitated Circulation Network for Machine Cell Formation

one unit of flow from each node i_a to the node j_b is known as the weight of the arcs (number of part moves) in the original IMF network. Having properly represented the problem as a capacitated circulation network, the relaxation algorithm developed by Bertsekas and Tseng [1] can be used to solve the corresponding minimum-cost problem. This relaxation method is a dual ascent procedure; it works on the dual formulation trying to improve the value of the dual solution at every iteration for the minimal cost network flow problems. The computerized version of this procedure is known as RELAXT-III [2]. After transforming the IMF network to the capacitated circulation network, the proposed network flow based heuristic consisting of the following five steps can be applied.

3.2 Algorithm Steps

Step 1: *Identification of minimal cost arc flows*: Solve the transformed network using the relaxation algorithm and identify the demand node set $\{j'_b\}=\{j_b | f\ (i_a, j_b)=1\}$.

Step 2: *Computing the total cost on each demand node*: If $f\ (i_a, j'_b)=1$, for all arcs $(i_a, j_b) \in A'$, compute the sum of costs on each demand node:

$$S(j_b) = \sum_{i_a} f(i_a, j_b)\, c\,(i_a, j_b)\, c\,(i_a, j_b)\ \forall\ j_b \in \{j'_b\}$$

Step 3: *Selection of the p demand nodes having the p smallest sums of costs*: For given p, order $S(j'_b)$'s as $S_1 \leq S_2 \leq \cdots \leq S_p$ and find demand node set

$$(j^*_b) = \{j'_b \,|\, S(j'_b) \in \{S_1, S_2, \ldots S_p\}\}.$$

Step 4: *Parameter change procedure*: Remove all arcs (j'_b, T) with $j_b' \in \{j^*_b\}$ by setting their upper bound equal to zero and reset $[\infty, 0, 0]$ to $[u, l, 0]$ for all arcs (j^*_b, T). In addition, set lower bound equal to one on arcs(i_a, j^*_b) for $i=j$.

Step 5: *Identification of cells*: Solve again minimal cost problem and identify cells by tracing positive flows from the source to the demand nodes.

The main idea of above solution procedure is to devise an assignment network to assign machines to other machines. The arcs for this network (a *bipartite* network) are determined by whether or not two given machines are connected in the IMF network. After transforming the *bipartite* network into a capacitated circulation network to choose p subsets representing p cells, the machines are allocated to cells subject to a constraint on cell size. Cells correspond to loops having positive flows.

3.3 An illustrative example

As an illustrative example, 16 parts with operational sequences on 12 machines and production volumes as given in Table 1 will be considered. The intermachine material flow (IMF) network based on Eq. (1) is shown in Figure 2. The number assigned to the arcs represents the arc weight (number of part moves) between two nodes

Table 1. The Operational Sequences of the Parts

Part i	Operational Sequence	Production Volume
1	1-4-8-9	2
2	1-2-6-4-8-7	3
3	1-2-4-7-8-9	1
4	1-4-7-9	3
5	1-6-10-7-9	2
6	6-10-7-8-9-	1
7	6-4-8-9	2
8	3-5-2-6-4-8-9	1
9	3-5-6-4-8-9	1
10	3-6-4-8	2
11	11-7-12-11	1
12	11-7-10-12	1
13	11-7-10	3
14	11-10	1
15	11-7-12	1
16	6-7-10	3

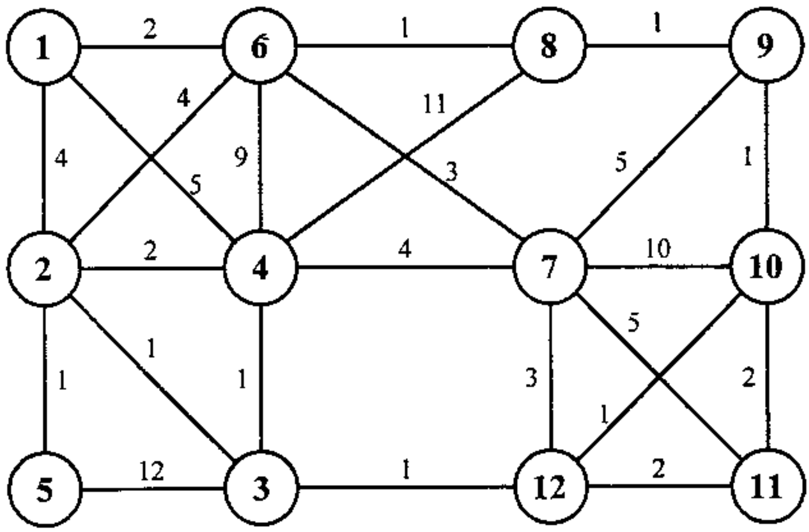

Figure 2. An Illustrative IMF Network

(machines). It is desired to find three cells with the size restriction $l=2$ and $u=4$.

The original network is transformed into the capacitated circulation network. After applying the network algorithm, seven demand nodes have positive flows. Three demand nodes (4,7, and 8) are selected after computing the associated cost. Four demand nodes are eliminated by setting the upper bound ∞ equal to zero on arcs (j'_h, T). Also, the lower bound on arcs (i_a, j^*_h) is set to one. Then, the flows are rerouted to obtain the required solution by applying the network flow algorithm. Three machine cells (MCs) can be obtained:

(1) MC 1 consists of machine 1,4,6,8
(2) MC 2 consists of machine 7,9,10,11
(3) MC 3 consists of machine 2,3,5,12

This cell information can be used as the training set of neural network's learning process for the identification of part families and the automatic assignment of new incoming parts. Detailed description of neural networks and its theoretical foundations are described in the following section, along with the example considered here.

4. Neural Networks for part family formation and new part assignment

The role of the neural networks is the assignment of parts to machine cells based on the learning process. It takes an input numeric pattern (input vector) and outputs an numeric pattern (output vector). In the context of GT problem, machine requirement of each part is the input and assignment of a suitable machine cell is the output of neural networks. The properties of neural networks and methodology are given following two subsections.

4.1 Property of neural networks

A neural network can be thought of as a mapping procedure, an M-dimensional input space are transformed into corresponding an p-dimensional output space. It requires a training set $\Omega : \Omega = \{(x_1, o_1), \cdots, (x_n, o_n)\}$, where $o_i = \phi(x_i)$ and $\phi(\cdot)$ is the mapping function $\phi : \Re^m \rightarrow \Re^p$. The x_i are called input vectors and the o_i are called target vectors or desired outputs.

A neural network implements mapping by many processing elements (neurons) organized into layers and weighted connections (weights) between layers. A typical neural network consists of a sequence of layers with full or partial connections between consecutive layers. The input layer receives the input vector and the output layer forms the numeric output of the network to a given input. Layers between input layer and output layer are called hidden layers. Inputs (x_i) are supplied to the input neurons, then each input is multiplied by a corresponding weight (w_{ij}), and all of the weighted inputs are summed to process the activation function $f_o(\cdot)$.The output value of the activation function (y_j) is the input of consecutive layer and proceeds as before to the output layer.

Learning is the most important property of neural networks. Learning is the process of adapting or modifying the connection weights to map a set of input vectors onto a corresponding set of output vector by means of the gradient descent algorithm [21]. This algorithm gradually adapts the connection weights in order to reduce difference between the actual network output and the desired output. This kind of learning based on input and desired output is referred to as the supervised learning. If no desired output is shown in the learning process, then it is called the unsupervised learning. As far as it concerns the GT problem, supervised learning can be considered since information concerning correct machine cell by means of the network analysis are available. Based on this requirement, the multi-layer neural network with one hidden layer along with the backpropagation learning algorithm was chosen.

4.2 Methodology of backpropagation neural network

The typical backpropagation (BP) network always has an input layer, an output layer and at least one hidden layer. The objective of the learning process is to minimize the squared error E given in Eq. (6) by making connection weight adjustment according to the error between the actual output value and the desired output value.

$$E = \frac{1}{2}\sum_{k=1}^{p}(o_k - \hat{o}_k)^2 \qquad (6)$$

Where o_k is the desired output value, $\hat{o}_k$ is the actual output value of network and p is the number of output neurons. The error is determined by performing the forward computations and this error is propagated backward through the network by means of the gradient descent algorithm called the generalized delta rule. The detailed algorithm of BP network for implementation can be founded in Haykin [10].

In the learning process the weights starting off with random values are changed using the generalized delta rule which accounts for error between the actual and desired output value. This error is decreased gradually as the number of iterations is increased. After several complete presentation of all vectors in the training set, the network converges to a steady set of weights, which have only small system error in value. The system error can be defined as mean squared error (MSE):

$$MSE = \frac{1}{p}\sum_{i=1}^{p}(e_i - \bar{e})^2 \qquad (7)$$

where

$$\bar{e} = \frac{1}{p}\sum_{i=1}^{p} e_i \ and\ e_i = o_i - \bar{o}_i$$

The track of system error (MSE) versus the number of iterations is defined as a learning curve which represents the learning performance of network. The network convergence is determined by a learning tolerance of 0.01. i.e., if MSE is less than 0.01, then network was trained.

A BP network architecture for the part family formation is shown Figure 3. The number of neurons in the input layer is the total number of machine types (M). The input vectors to this network are M-dimensional binary vectors where an element of m_{ij}=1 indicates machine cell i has machine j and m_{ij}=0 indicates machine cell i does not, $j=1,\cdots,M$. The number of neurons in the output layer is the number of machine cells (p). The target vectors (desired output) correspond to input vectors are the p-

dimensional binary vectors where an element of $o_{ij}=1$ indicates the machine cell i and $o_{ij}=0$ does not, $j=1,\cdots,p$. As an illustration, consider 3 machine cells from the previous example. Machine cell 1 consists of machine (1,4,6,8). The input vector for this cell is (1 0 0 1 0 1 0 1 0 0 0 0) and the desired output vector is (1 0 0). Similarly, the remaining machine cells are transformed into the following binary vectors (training set) presented in Table 2.

The number of neurons in hidden layer is the important problem in the design of BP network[21] . Smaller number of neurons in hidden layer does not make network converge and larger number of neurons is useful for enhancing mapping accuracy, but increases the learning iterations impractically, which means it takes long time to converge, or the network cannot be trained at all. Based on these guideline, simulation study was conducted to train networks with varying number of hidden layer to

Table 2. The List of Input and Target Vectors (Training Set)

Cell	Machines in Cell	Input Vector	Target Vector
1	(1,4,6,8)	(1 0 0 1 0 1 0 1 0 0 0 0)	(1 0 0)
2	(7,9,10,11)	(0 0 0 0 0 0 1 0 1 1 1 0)	(0 1 0)
3	(2,3,5,12)	(0 1 1 0 1 0 0 0 0 0 0 1)	(0 0 1)

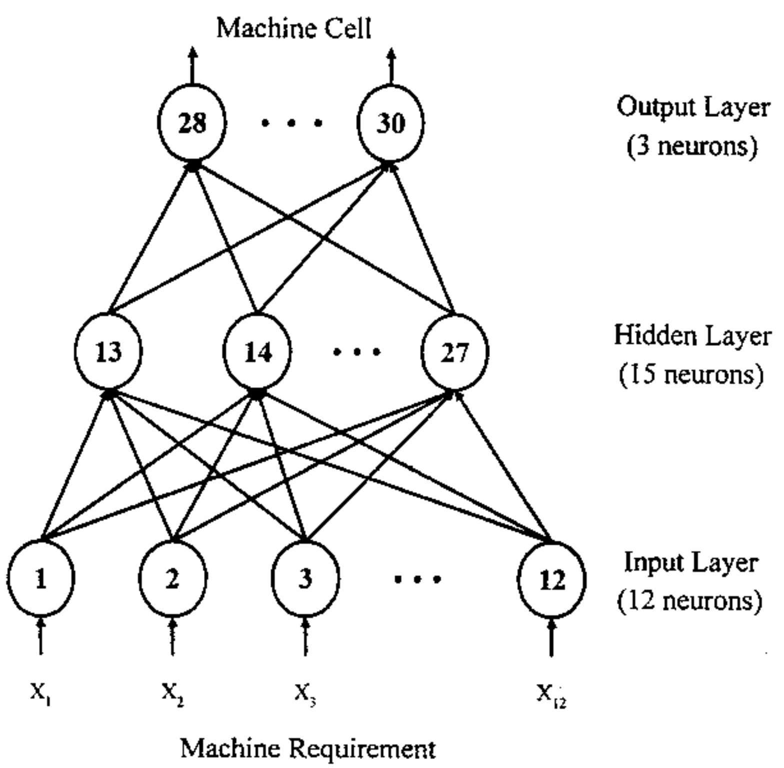

Figure 3. A Three-Layer Backpropagation Network for Part Family Formation

evaluate learning performance of each network. Networks with 3,6,10,15 and 20 hidden neurons were trained using the same training set and their learning performances are presented in Figure 4.

the set of parts assigned to the same cell. The testing set of 16 binary vectors were submitted to the trained network. The output of network, the assignment of machine cell for each part, can be produced by the

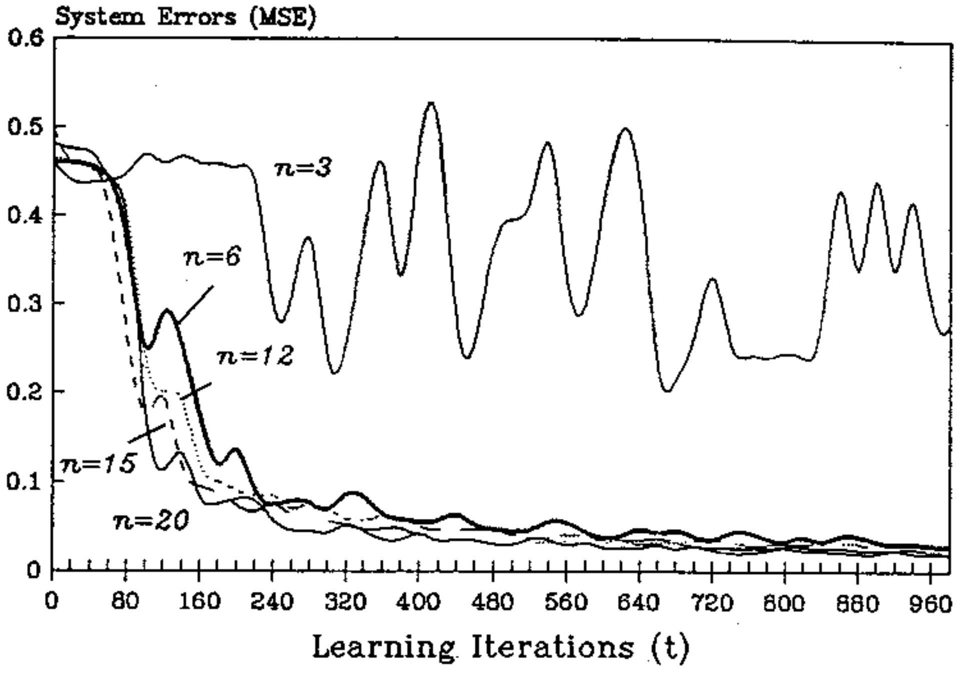

Figure 4. Learning Performance as a Function of Number of Neurons in Hidden Layer

The learning performance improved with the number of neurons in hidden layer up to 15 hidden neurons. No further improvement was achieved more than 15 hidden neurons. After training the network with 15 hidden neurons, the part family formation and the new part assignment can be done on the basis of the generalization ability of neural networks. As previously mentioned, mapping networks can produce reasonable output for input vectors, not contained in the training set.

Once machine cells are formed, a part should be assigned to the machine cell by the machine requirement of part so that this part can be fully processed in the assigned machine cell. In this context, a part family is

forward computation. If new parts are introduced, the same procedure can be applied to assign these parts to the most appropriate machine cell without repeating the entire computational process.

This methodology was written and complied the C-language using an IBM/PC computer. Table 3 shows the whole testing set, new parts and output vectors of the trained network. As can be seen, three part families can be identified and new parts are assigned automatically to the one of the machine cells, to which these parts should belong.

Table 3. The List of Test Vector and Output of Neural Network

Part	Test Vector	Output	Cell Assignment
1	(1 0 0 1 0 0 0 1 1 0 0 0)	(0 1 0)	2
2	(1 1 0 1 0 1 1 1 0 0 0 0)	(1 0 0)	1
3	(1 1 0 1 0 0 1 1 1 0 0 0)	(1 0 0)	1
4	(1 0 0 1 0 0 1 0 1 0 0 0)	(1 0 0)	1
5	(1 0 0 0 0 1 1 0 1 1 0 0)	(1 0 0)	1
6	(0 0 0 0 0 1 1 1 1 1 0 0)	(0 1 0)	2
7	(0 0 0 1 0 1 0 1 1 0 0 0)	(0 1 0)	2
8	(0 1 1 1 1 1 0 1 1 0 0 0)	(1 0 0)	1
9	(0 0 1 1 1 1 0 1 1 0 0 0)	(1 0 0)	1
10	(0 0 1 1 0 1 0 1 0 0 0 0)	(1 0 0)	1
11	(0 0 0 0 0 0 1 0 0 0 1 1)	(0 0 1)	3
12	(0 0 0 0 0 0 1 0 0 1 1 1)	(0 0 1)	3
13	(1 0 0 0 0 0 1 0 0 1 1 0)	(0 0 1)	3
14	(1 0 0 0 0 0 0 0 0 1 1 0)	(0 0 1)	3
15	(0 0 0 0 0 0 1 0 0 0 1 1)	(0 0 1)	3
16	(0 0 0 0 0 1 1 0 0 1 0 0)	(1 0 0)	1
new1	(1 0 1 1 0 0 0 1 1 1 0 0)	(1 0 0)	1
new2	(0 0 0 1 0 0 1 1 0 0 1 1)	(0 0 1)	3

5. Conclusions

A new approach for the design of manufacturing cells employing a neural network combined with the network analysis has been presented. A network flow based solution procedure was developed to form machine cells on the basis of the material flow matrix. This matrix is obtained from given set of parts having the generic operational sequences and production volume. The objective of machine cell formation is to minimize intercellular movements of parts subject to the restrictions on the limited number of machine cells and cell sizes. The cell information was then used as a training set of neural networks for the part family formation and new part assignment. Simulation study for selecting the best internal representation (hidden neurons) was also conducted. The neural networks successively implemented in this article can be generally applied to the cell assignment of newly introduced part without building the previous coded knowledge or expert rule and repeating the entire computational process.

The methodology developed in this article was applied to an industrial company. The production facilities included approximately 150 machines to manufacture hydraulic cylinders and pistons, power transmission gears, links, and other mechanical parts. About 40,000 parts are in process and 5,500 unique operation routings of parts are used for the production of a lift and hoist. The computer running time of about 30 seconds for the cell formation in the industrial problem considering 150 machines and 5,500 operational sequences seems to be attractive. Computational results indicate that the proposed approach is appropriate for solving large-scale industrial problems including up to several hundreds machines and several thousands of parts in a microcomputer environment. Finally, the major contributions of proposed method are efficient computational performance by employing the network flow algorithm for machine cell formation and effective in solving large-scale part family formation by exploiting the parallel architecture of neural networks. Therefore, this methodology is not significantly influenced by the size of problems. In addition, this methodology has potential applications in other areas such as the Computer Aided Process Planning System to retrieve the process plan of new part, scheduling of job dispatching rules to minimize the number of tool switchings in the FMS environment, and planning the PCB (Printed Circuit Board) assembly operations.

References

[1] Bertsekas, D., and Tseng, P., "Relaxation methods for minimum cost ordinary and generalized network flow problems", *Operations Research*, 36, 1988, pp.93-114.

[2] Bertsekas, D., and Tseng, P., "RELAXT-III: A new and improved version of the RELAXT code", *Laboratory for Information and Decision System Report LIDS-P-1990*, MIT, Cambridge, MA. 1990.

[3] Burke, L, and Kamal, S., "Neural networks and Part Family/Machine Group Formation Problem in Cellular Manufacturing: A Framework using Fussy ART", *Journal of Manufacturing Systems*, 14, 1995, pp. 148-159.

[4] Boctor, F. F., "A linear formulation of the machine-part cell formation problem, *International Journal of Production Research*", 29, 1991, pp. 343-356.

[5] Burbidge, J. L., *The Introduction of Group Technology*. New York: John Wiley & Sons. 1975.

[6] Chan, H. M., and Milner, D. A., "Direct clustering algorithm for group formation in cellular manufacturing", *Journal of Manufacturing Systems*, 1, 1982, pp. 65-75.

[7] Chu, C., "Manufacturing Cell Formation by Competitive Learning", *International Journal of Production Research*, 31, 1993, pp. 829-843.

[8] ElMaraghy, H. A., and Gu, P., "Feature based expert parts assignment in cellular manufacturing", *Journal of Manufacturing Systems*, 8, 1989, pp.139-152.

[9] Han, C., and Ham, I., "Multiobjective cluster analysis for part family formation", *Journal of Manufacturing Systems*, 5, 1986, pp. 223-230.

[10] Haykin, S., *Neural Networks: A Comprehensive Foundation*. McMillan, 1994.

[11] King, J.R., and Nakornchai, V., "Machine-component group formation in group technology: Review and extension", *International Journal of Production Research*, 20, 1982, pp. 117-133.

[12] Kohonen, T., "An introduction to neural computing", *Neural Networks*, 1, 1988, pp. 1-16.

[13] Kosko, B., "Unsupervised learning in noise", *IEEE Transactions on Neural Networks*, 1, 1990, 44-57.

[14] Kumar, K. R., Kusiak, A., and Vannelli, A., "Grouping of parts and components in flexible manufacturing systems", *European Journal of Operational Research*, 24, 1986, pp. 387-397.

[15] Kusiak, A., "The generalized group technology concept", *International Journal of Production Research*, 25, 1987, pp. 561-569.

[16] Kusiak, A., "Artificial intelligence and operation research in flexible manufacturing system", *Information Systems and Operations Research*, 25, 1987, pp. 2-12.

[17] Kusiak, A., *Intelligent Manufacturing Systems*. New Jersey: Prentice-Hall, Inc. 1990.

[18] Liao, T., and Chen, L., "An Evaluation of ART1 neural models for GT part family and Machine Cell

Formation", *Journal of Manufacturing Systems*, 12, 1993, pp. 282-290.

[19] Lippman,R.P., "An introduction to computing with neural nets", *IEEE ASSP Magazine*, 1987, pp. 4-22.

[20] Luong, L., "A Cellular Similarity Coefficient Algorithm for the Design of Manufacturing Cells", *International Journal of Production Research*, 31, 1993, 1757-1766.

[21]Pao, Y., *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, MA. 1989.

[22] Rajagopalan, R., and Barta, J. L., "Design of Cellular Production System: A Graph Theoretic Approach", *International Journal of Production Research*, 13, 1982, pp. 567-579.

[23] Seifoddni, H.,and Wolfe, P, M., "Application of the similarity coefficient method in group technology", *IEEE transactions*, 18, 1986, pp. 271-277.

[24] Seifoddini, H., and Hsu, C., "Comparative Study of Similarity Coefficients and Clustering Algorithms in Cellular Manufacturing", *Journal of Manufacturing Systems*, 13, 1994, pp. 119-127.

[25] Sherald, M., Neural nets versus expert system, *PC AI Magazine*, 3, 10-15. 1989.

[26] Wann, M., Hediger, T.,and Greenbaun, N. N., "The influence of training sets on generalization in feed-forward neural networks", *Proceedings of IEEE International Conference on Neural Networks*, 3, 1990, pp. 137-142.

[27] Wu, H. L., Venugopal,R., and Barash, M. M., "Design of a cellular Manufacturing System: A syntactic pattern recognition approach", *Journal of Manufacturing Systems*, 5, 1986, pp. 81-87.

[28] Wu, N, and Salvendy, G., "A Modified Network Approach for the Design of Cellular Manufacturing Systems", *International Journal of Production Research*, 31, 1993, pp. 1409-1421.

[29] Xu, H., and Wang, H., "Part family formation for GT application based on fuzzy mathematics", *International Journal of Production Research*, 27, 1989, pp. 1637-1651.

97년 3월 최초 접수, 97년 11월 최종 수정