CELL FORMATION USING SEQUENCE INFORMATION AND NEURAL NETWORKS

S. Lozano, D. Canca, F. Guerrero, J. Larrañeta and L. Onieva
Department of Industrial Management, University of Seville
Camino de los Descubrimientos, s/n, 41092 Seville, Spain

ABSTRACT: Most neural network approaches to the cell formation problem have been based on Competitive Learning-based algorithms such as ART (Adaptive Resonance Theory), Fuzzy Min-Max or Self-Organizing Feature Maps. These approaches do not use information on the sequence of operations on part types. They only use as input the binary part-machine incidence matrix. There are other neural network approaches such as the Hopfield model and Harmony Theory that have also been used to form manufacturing cells but again without considering the sequence of operations. In this paper we propose a sequence-based neural network approach for cell formation. The objective function considered is the minimization of transportation costs (including both intracellular and intercellular movements). Soft constraints on the minimum and maximum on the number of machines per cell can be imposed. The problem is formulated mathematically and shown to be equivalent to a quadratic programming integer program that uses symmetric, sequence-based similarity coefficients between each pair of machines. To solve such a problem two energy-based neural network approaches (Hopfield model and Potts Mean Field Annealing) are proposed.

INTRODUCTION

Cellular manufacturing consists in forming a number of manufacturing cells, each one dedicated to the processing of a family of similar part types. This can lead to a reduction of setup times, waiting times and work-in-process inventories (Wemmerlöv and Johnson 1997). There are many approaches to cell formation ranging from heuristics (e.g. Askin and Subramanian 1987) and metaheuristics (e.g. Lozano et al 1999a), to mathematical programming methods (e.g. Wang and Rose 1997) and conventional (e.g. McAuley 1972) as well as fuzzy (Dobado et al 1999) clustering methods.

There have also been a number of neural network approaches for cell formation. Most approaches involving neural networks use either competitive learning (Malave and Ramachandran 1991; Chu 1993), ART1 (Dagli and Huggahalli 1991; Kaparthi and Suresh 1992; Prasad and Rajan 1994), Fuzzy ART (Suresh and Kaparthi 1994, Kamal and Burke 1996), Fuzzy Min-Max (Lozano et al 1999b) or SOFM (Venugopal and Narendran 1994). The basic input data these methods use is a binary part-machine incidence matrix

$$a_{ij} = \begin{cases} 1 & \text{if machine } i \text{ is required for processing part type } j \\ 0 & \text{otherwise} \end{cases}$$

Other neural networks that aim at optimizing a certain energy function have also been proposed for cell formation (Lozano et al 1993, Canca et al 1999) but they too use the binary part-machine incidence matrix.

In this paper we propose a neural network approach that uses the information on the sequence of operations of the different part types. In the next section the problem is formulated and analyzed. The two solution approaches proposed are presented in section 3 and 4. Section 5 presents some computational experiences and in section 6 conclusions are drawn.

PROBLEM FORMULATION

Let

i machine index
j part type index
k machine cell index
l operation index
M Number of machines
P Number of part types
M_{min} Minimum number of machines per cell
M_{max} Maximum number of machines per cell
$C_{max} = \lfloor M/M_{min} \rfloor$ Maximum number of cells that can be formed
$C_{min} = \lceil M/M_{max} \rceil$ Minimum number of cells that can be formed
o_j Number of operations of part type j
m_{jl} Machine on which operation l of part type j is performed

$$\delta_{ii'jl} = \begin{cases} 1 & \text{if } m_{jl} = i \text{ and } m_{j,l+1} = i' \\ 0 & \text{otherwise} \end{cases}$$

D_j Demand of part type j

$n_{ii'} = \sum_{j=1}^{P} \sum_{l=1}^{o_j} D_j \, \delta_{ii'jl}$ Total number of movements from machine i to machine i'

$\eta_{ii'} = n_{ii'} + n_{i'i}$ Total number of movements between machines i and i'

h^{intra} Unit cost of intracellular movements

h^{inter} Unit cost of intercellular movements

$q = \dfrac{h^{intra}}{h^{inter}}$ Ratio of unit cost of intracellular movements to unit cost of intercellular movements

The decision variables are

$$x_{ik} = \begin{cases} 1 & \text{if machine i is assigned to cell k} \\ 0 & \text{otherwise} \end{cases}$$

Total transportation costs can be computed as

$$Cost = \sum_{i=1}^{M} \sum_{i'=1}^{M} \left[h^{intra} \, n_{ii'} \sum_{k=1}^{C_{max}} x_{ik} \, x_{i'k} + h^{inter} \, n_{ii'} \sum_{k=1}^{C_{max}} x_{ik} \, (1 - x_{i'k}) \right]$$

or equivalently

$$Cost = h^{\text{inter}}\left[\sum_{i=1}^{M}\sum_{i'>i}\eta_{ii'} - (1-q)\sum_{i=1}^{M}\sum_{i'>i}\sum_{k=1}^{C_{max}}\eta_{ii'}x_{ik}\,x_{i'k}\right]$$

Since the first term within the bracket is a constant and assuming q≤1 the minimization of total transportation costs is equivalent to maximizing the following objective function

$$Objective\ function = \sum_{i=1}^{M}\sum_{i'>i}\sum_{k=1}^{C_{max}} S_{ii'}x_{ik}\,x_{i'k}$$

where the similarity coefficient between machines i and i' is defined as $S_{ii'} = \eta_{ii'}$.

This objective function represents the sum of all pair-wise similarity coefficients between machines in the same cell. It also represents the total number of intracellular movements. Note that the value of q (or equivalently of h^{intra}) does not influence the optimal solution, as long as $h^{\text{intra}} \le h^{\text{inter}}$. The reason is that since the total number of movements in the system is fixed, as long as intercellular movements are more costly than intracellular ones but irrespective of their absolute values, total transportation costs are minimized when intercellular movements are minimized or equivalently intracellular movements are maximized. This means that there would have been no generality loss should we have assumed that instead of total transportation costs only intercellular (respectively intracellular) movements costs are minimized (respectively maximized).
Although in principle the number of cells to form is not fixed but can vary between C_{min} and C_{max}, in practice, since the objective function increases as the number of cells are decreased, the optimal solution will very likely involve the minimum feasible number of cells C_{min}. Therefore we can solve the problem assuming that the number of cells to form is fixed to C_{min}.

Thus, the complete model that results is

Maximize $$\sum_{i=1}^{M}\sum_{i'>i}\sum_{k=1}^{C_{min}} S_{ii'}x_{ik}\,x_{i'k}$$

subject to

$$\sum_{k=1}^{C_{min}} x_{ik} = 1 \quad \forall i$$

$$M_{min} \le \sum_{i=1}^{M} x_{ik} \le M_{max} \quad \forall k$$

$$x_{ik} = \{0,1\} \quad i = 1,2,...,M \quad k = 1,2,...,C_{min}$$

This is a quadratic programming problem with a combinatorial structure. To solve it we propose two energy function-based neural networks: the Hopfield model and Mean Field Annealing.

HOPFIELD MODEL APPROACH

The Hopfield model (Hopfield and Tank 1985) is a recurrent neural network with symmetric, bidirectional connections between every pair of units. Each unit has an activation level which is a sigmoid function of its net input. The net input of a neuron evolves according to an equation of motion that depends on the activation levels of the other neurons and on the weights of its connections with them. Hopfield (1984) shows that this dynamic system has an energy function and that the stable states to which the activation levels converge are local minimum of such Lyapunov function.

For the cell formation problem formulated above, the corresponding Hopfield model has $M \cdot C_{min}$ nodes, i.e. one for each possible machine to cell assignment. Each node has an associated activation level x_{ik}. The weight of the symmetric, bidirectional connection between each pair of nodes i and i' is the similarity coefficient $S_{ii'}$. The energy function of the network is

$$E = -\frac{1}{2}\sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{k=1}^{C_{min}} S_{ii'}x_{ik}x_{i'k} + \frac{\alpha}{2}\sum_{i=1}^{M}\sum_{k=1}^{C_{min}} x_{ik}(1-x_{ik}) + \frac{\beta}{2}\sum_{i=1}^{M}\left(\sum_{k=1}^{C_{min}} x_{ik} - 1\right)^2 +$$

$$+\frac{\gamma}{2}\sum_{k=1}^{C_{min}}\left[\sum_{i=1}^{M} x_{ik} - \frac{M_{min}+M_{max}}{2}\right]^2$$

The first term in the energy function corresponds to the quadratic objective function of the problem. The second term is a penalty aimed at imposing the integrality constraints on variables x_{ik}. The third term is also a penalty aimed at imposing the constraints that guarantee that each machine is assigned to one and only one cell. The fourth term is an approximate method of imposing the cell size limits. The need for these penalty terms and their corresponding penalty coefficients is one the major drawbacks of using the Hopfield model for constrained optimization problems since they distort the original objective function so that the network will eventually trade it off with the penalty terms trying to find a compromise: a feasible (or nearly feasible) solution with a good value of the objective function.

The activation level of a neuron ik is computed as $x_{ik}=g_T(u_{ik})$ where u_{ik} is the net input to neuron ik and the transfer function $g_T(u) = 1/1+e^{-u/T}$ is the logistic function with gain $1/T$. The equation of motion describing the time evolution of the model is

$$\frac{du_{ik}}{dt} = -\frac{\partial E}{\partial x_{ik}} = \sum_{i'=1}^{M} S_{ii'}x_{i'k} - \alpha(1-2x_{ik}) - \beta\left(\sum_{k'=1}^{C_{min}} x_{ik'} - 1\right) - \gamma\left[\sum_{i'=1}^{M} x_{i'k} - \frac{M_{min}+M_{max}}{2}\right]$$

The interpretation of the second term is that if x_{ik} is less than 0.5 then the net input should decrease so that the activation level is decreased trying to approach 0. On the contrary, if x_{ik} is greater than 0.5 then the net input should increase so that the activation level is also increased trying to approach 1. Note, however, that when x_{ik}=0.5 the second term does not contribute to changing the net input.

The interpretation of the other terms of the equation is easier if we assume that except x_{ik} the other activation levels are already binary. Then,

- the first term means that the higher the similarities between machine i and the other machines in cell k, the higher the increase in the net input u_{ik} in order to move x_{ik} towards 1 and join that cell
- the third term means that if machine i is already partially assigned to more than one cell, the net input u_{ik} (and consequently the activation x_{ik}) should decrease and if it is under-assigned both should increase. Note that if the equality holds, no change is made. Note also that this term does not depend on k so that the grade of membership of machine i to all cells are changed by the same amount.
- the fourth term means that if the total number of machines already in cell k is above (respectively below) the target size $(M_{min}+M_{max})/2$ then the net input u_{ik} and consequently the activation x_{ik} should decrease (respectively decrease). Note that this term does not depend on i so that the grade of membership of all machines assigned to cell k are changed by the same amount.

The solution provided by the Hopfield neural network depends on the initial value of the activations x_{ik} and of the penalty coefficients α,β and γ. Activations are initialized randomly in the vicinity of the center of the unit hipercube, i.e. $x_{ik}\approx 0.5\ \forall ik$. The strategy followed to determine the values of the penalty coefficients consists in augmenting gradually the value of α or β if any of its associated

constraints does not hold. Such a process continues both constraints hold, i.e., the activations are binary and each machine is assigned to one and only one cell. Since the target size (M_{min}+M_{max})/2 is a soft constraint the value of γ is not augmented gradually but stays fixed. After convergence is obtained cell sizes are computed and in case the maximum cell size is exceeded a greedy heuristic is used to guarantee feasibility.

POTTS MEAN FIELD ANNEALING APPROACH

One way of reducing the number of penalty terms in the energy function (which distort the original objective function) is using Mean Field Annealing. The Potts Mean Field Annealing (PMFA) algorithm (Peterson and Söderberg 1989, Van den Bout and Miller 1989, 1990, Gislen et al 1992) is a neural network approach with the same topology that the Hopfield model. However, in this approach:

- Neurons are considered as binary valued units with stochastic transfer functions $P(x_{ik}=1)=g_T(u_{ik})$ and $P(x_{ik}=0)=1-P(x_{ik}=1)$ while the operating variables are their mean values $\langle x_{ik} \rangle$
- The gain *1/T* of the logistic function is gradually increased or equivalently the Temperature parameter *T* is gradually decreased from a large initial value until it reaches a value close to zero that leads to the saturation of the activation levels, i.e. the activation levels are approximately binary. This is the way in which the integrality constraints are imposed and therefore do not need to be included in the energy function
- The mean activation levels $\langle x_{ik} \rangle$ are not computed independently for each pair ik but according to

$$\left\langle x_{ik} \right\rangle = \frac{e^{-\frac{h_{ik}}{T}}}{\sum_{k'=1}^{C_{min}} e^{-\frac{h_{ik'}}{T}}}$$

where

$$h_{ik} = \frac{\partial \langle E \rangle}{\partial \langle x_{ik} \rangle} = \langle E \rangle_{x_{ik}=1 \quad x_{ik'}=0 \ \forall k' \neq k} - \langle E \rangle_{x_{ik'}=0 \ \forall k'}$$

is called the mean internal field experienced by neuron ik. This form of computing the mean activation levels guarantee that

$$\sum_{k=1}^{C_{min}} \langle x_{ik} \rangle = 1 \quad \forall i$$

At each temperature T the PMFA algorithm iterates the activation levels computation equations until convergence. Then T is decreased and corresponding activation levels are computed again. This process is repeated until, for a sufficently low value of T the saturation condition holds

$$\frac{1}{M \cdot C_{min}} \sum_{i=1}^{M} \sum_{k=1}^{C_{min}} \langle x_{ik} \rangle (1 - \langle x_{ik} \rangle) \leq 0.01$$

- The energy function is equal to the original objective function plus the penalty terms corresponding to the cell size constraints. Its mean value is

$$\langle E(\overline{X}) \rangle = \left\langle -\frac{1}{2} \sum_{i=1}^{M} \sum_{i'=1}^{M} \sum_{k=1}^{C_{min}} S_{ii'} x_{ik} x_{i'k} + \frac{\gamma}{2} \sum_{k=1}^{C_{min}} \left[\sum_{i=1}^{M} x_{ik} - \frac{M_{min} + M_{max}}{2} \right]^2 \right\rangle =$$

$$= -\frac{1}{2}\left\langle \sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{k=1}^{C_{min}} S_{ii'}\, x_{ik} x_{i'k} \right\rangle + \frac{\gamma}{2}\left\langle \sum_{k=1}^{C_{min}} \left[\sum_{i=1}^{M} x_{ik} - \frac{M_{min}+M_{max}}{2} \right]^2 \right\rangle =$$

$$= -\frac{1}{2}\sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{k=1}^{C_{min}} S_{ii'}\left\langle x_{ik} x_{i'k} \right\rangle + \frac{\gamma}{2}\sum_{k=1}^{C_{min}} \left\langle \left[\sum_{i=1}^{M} x_{ik} - \frac{M_{min}+M_{max}}{2} \right]^2 \right\rangle =$$

$$= -\frac{1}{2}\sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{k=1}^{C_{min}} S_{ii'}\left\langle x_{ik} x_{i'k} \right\rangle + \frac{\gamma}{2}\sum_{k=1}^{C_{min}}\sum_{i=1}^{M} \left\langle x_{ik}\, x_{ik} \right\rangle + \frac{\gamma}{2} C_{min}\left[\frac{M_{min}+M_{max}}{2} \right]^2 +$$

$$-\frac{\gamma}{2}(M_{min}+M_{max})\sum_{k=1}^{C_{min}}\sum_{i=1}^{M}\left\langle x_{ik} \right\rangle + \frac{\gamma}{2}\sum_{k=1}^{C_{min}}\sum_{i=1}^{M}\sum_{i'=i} \left\langle x_{ik}\, x_{i'k} \right\rangle \cong$$

$$\cong -\frac{1}{2}\sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{k=1}^{C_{min}} S_{ii'}\left\langle x_{ik} \right\rangle\left\langle x_{i'k} \right\rangle + \frac{\gamma}{2}\sum_{k=1}^{C_{min}}\sum_{i=1}^{M} \left\langle x_{ik} \right\rangle\left\langle x_{ik} \right\rangle + \frac{\gamma}{2} C_{min}\left[\frac{M_{min}+M_{max}}{2} \right]^2 +$$

$$-\frac{\gamma}{2}(M_{min}+M_{max})\sum_{k=1}^{C_{min}}\sum_{i=1}^{M}\left\langle x_{ik} \right\rangle + \frac{\gamma}{2}\sum_{k=1}^{C_{min}}\sum_{i=1}^{M}\sum_{i'=i} \left\langle x_{ik} \right\rangle\left\langle x_{i'k} \right\rangle =$$

$$= h^{ext} + \sum_{i=1}^{M}\sum_{k=1}^{C_{min}} \left\langle x_{ik} \right\rangle h_{ik}$$

where

$$h^{ext} = \frac{\gamma}{2} C_{min}\left[\frac{M_{min}+M_{max}}{2} \right]^2$$

is a constant, external field and

$$h_{ik} = \frac{\partial\left\langle E \right\rangle}{\partial\left\langle x_{ik} \right\rangle} = -\frac{1}{2}\sum_{i'=i}(S_{ii'}-\gamma)\left\langle x_{i'k} \right\rangle + \frac{\gamma}{2}\left\langle x_{ik} \right\rangle - \frac{\gamma}{2}(M_{min}+M_{max})$$

is the mean internal field experienced by neuron ik.

COMPUTATIONAL EXPERIENCES

In order to test the performance of Hopfield and Potts MFA we have carried out a set of experiments with 120 random instances corresponding to four groups of problems. The number of machines and part types in each group of problems is shown in table 1.

Table 1

Number of Machines and Part Types

Problem group	**M**	**P**
I	20	40
II	30	60
III	40	80
IV	50	100

The route for each part type was generated randomly with a length between 2 and 10 operations. The demand for each part was generated randomly between 100-δ and 100+δ. Three values of δ were

used (δ=10, δ=25 and δ=50). For each problem group and each value of δ 10 different problems were generated. A value of M_{max}=7 was used for all problems.

As a comparison, the best solution found by a Tabu Search algorithm has also been computed. Tabu Search (TS) is known to be a very good (and fast) metaheuristic [Lozano et al, 1999]. Since neural network approaches are not expected to outperform a well designed TS algorithm, we use it just as a benchmark. Table 2 shows for each problem category the average ratio of the objective function found by each method and the best value of the objective function found by any method. Recall the problem is a maximization one so the closer the ratio to unity, the better performing the method. Table 3 shows the average of the same ratios for each problem group. Note that, as expected, Tabu Search gives almost always the best solution, followed by PMFA with an average deviation of 6% and Hopfield in third place with an average deviation of 10%.

Table 2

Objective function ratios (TS=Tabu Search, HOP=Hopfield, PMFA=Potts MFA)

Problem group	δ	**O.F. TS/ Best O.F.**	**O.F. HOP/ Best O.F.**	**O.F. PMFA/ Best O.F.**
I	10	1.000	0.955	0.940
I	25	1.000	0.937	0.960
I	50	1.000	0.951	0.944
II	10	0.998	0.901	0.971
II	25	0.999	0.900	0.959
II	50	0.988	0.876	0.956
III	10	1.000	0.882	0.912
III	25	1.000	0.884	0.919
III	50	1.000	0.867	0.907
IV	10	1.000	0.880	0.932
IV	25	1.000	0.855	0.940
IV	50	1.000	0.863	0.925

Table 3

Objective function ratios by problem group (TS=Tabu Search, HOP=Hopfield, PMFA=Potts MFA)

Problem group	**O.F. TS/ Best O.F.**	**O.F. HOP/ Best O.F.**	**O.F. PMFA/ Best O.F.**
I	1.000	0.948	0.948
II	0.995	0.892	0.962
III	1.000	0.877	0.913
IV	1.000	0.866	0.932
Average	0.999	0.896	0.939

As for efficiency, Tables 4 and 5 show the average CPU times (in an Intel Pentium II 450 MHz) for each method. It can be noted that for all methods running times are correlated with problem size, i.e.

the larger the number of machines the longer it takes to explore the solution space. Of the three methods tested, Hopfield is slowest and PMFA is fastest. Tabu Search is in between.

CONCLUSIONS

In this paper a new approach to cell formation is proposed. It uses similarity coefficients between pairs of machines. These similarity coefficients are computed using the information on the sequence of operations along the routes of the part types. Although two neural networks have been proposed to solve the machine grouping problem, the one based on Potts MFA seems best suited because it requires only one penalty coefficient. Compared with a Tabu Search algorithm PMFA gives an average of 6% deviation but its computational requirements are much lower.

Table 4

CPU times (TS=Tabu Search, HOP=Hopfield, PMFA=Potts MFA)

Problem group	δ	**CPU time TS**	**CPU time HOP**	**CPU time PFMA**
I	10	0.126	0.774	0.045
I	25	0.110	0.582	0.046
I	50	0.104	0.676	0.050
II	10	0.390	2.748	0.081
II	25	0.357	2.724	0.077
II	50	0.396	2.285	0.068
III	10	1.302	5.108	0.125
III	25	1.297	5.668	0.159
III	50	1.319	6.359	0.149
IV	10	1.676	11.649	0.187
IV	25	1.769	16.490	0.187
IV	50	1.857	13.153	0.195

Table 5

CPU times (TS=Tabu Search, HOP=Hopfield, PMFA=Potts MFA)

Problem group	**CPU time TS**	**CPU time HOP**	**CPU time PFMA**
I	0.114	0.677	0.047
II	0.381	2.586	0.075
III	1.306	5.712	0.144
IV	1.767	13.764	0.190
Average	0.892	5.685	0.114

REFERENCES

Askin, R.G. and Subramanian, S.P., A cost-based heuristic for group technology configuration, *Int. J. Prod. Res.*, **25**, 1, 101-113 (1987)

Canca, D., Onieva, L, Guerrero, F. and Racero, J., Application of the Hopfield-Tank model to Part-Machine Grouping, in *Flexible Automation and Intelligent Manufacturing 1999*, J. Ashayeri et al (eds.) Begell House, New York, 489-499 (1999)

Chu, C. Manufacturing cell formation by competitive learning, *Int. J. Prod. Res.* **31**, 829-843 (1993)

Dagli, C. and Huggahalli, R., Neural network approach to group technology. In: *Knowledge-Based Systems and Neural Networks: Techniques and Applications*, R. Sharda et al (eds.), Elsevier, 213-218 (1991)

Del Valle, A.G., Balarezo, S. and Tejero, J., A heuristic workload-based model to form cells by minimizing intercellular movements. *International Journal of Production Research* 32, 2275-2285 (1994)

Dobado, D., Lozano, S., Eguia, I. and Larrañeta, J., Parallel fuzzy clustering of parts and machines for cellular manufacturing, in *Flexible Automation and Intelligent Manufacturing 1999*, J. Ashayeri et al (eds.) Begell House, New York, 473-487 (1999)

Gislén, L. Peterson, C. and Söderberg, B., Complex Scheduling with Potts Neural Networks. *Neural Computation* **4**, 805-831 (1992)

Hopfield, J.J., Neural networks with graded response have collective computational properties like those of two-state neurons, *Proceedings of the National Academy of Science USA*, **81**, 3088-3092 (1984)

Hopfield, J.J. and Tank, D.W., "Neural" Computation of Decisions in Optimization Problems. *Biol. Cybern.* **52**, 141-152 (1985)

Kamal, S. and Burke, L.I., FACT: A new neural network-based clustering algorithm for group technology, *Int. J. Prod. Res.*, **34**, 4, 919-946 (1996)

Kaparthi, S. and Suresh, N.C., Machine Component Cell Formation in Group Technology: A Neural Network Approach,. *Int. J. Prod. Res.* **30**, 6, 1353-1367 (1992)

Lozano, S., Onieva, L,. Larrañeta, J. and Teba, J., A Neural Network Approach to Part-Machine Grouping in GT Manufacturing, in *Robotics, Mechatronics and Manufacturing Systems*, T. Takamori and K. Tsuchiya (eds.), North-Holland, Amsterdam, 619-624 (1993)

Lozano, S., Guerrero, F., Eguia, I., Canca, D. and Smith, K., Cell Formation Using Two Neural Networks in Series, in *Intelligent Engineering Systems Through Artificial Neural Networks. Volume 8*, C.H. Dagli et al (eds.) ASME Press, New York, 341-346 (1998)

Lozano, S., Adenso-Diaz, B., Eguia, I. and Onieva, L., A one-step tabu search algorithm for manufacturing cell design, *Journal of the Operational Research Society*, 50, 509-516 (1999a)

Lozano, S, Dobado, D., Eguia, I., Guerrero, F. and Canca, D., Fuzzy neural network for part family formation, *Intelligent Engineering Through Artificial Neural Networks, vol. 9*, C.H. Dagli et al (eds.), ASME Press, 1141-1146 (1999b)

Malave, C.O. and Ramachandran, S., Neural network-based design of cellular manufacturing systems, *J. Intel. Manuf.* **2**, 305-314 (1991)

McAuley, J., Machine Grouping for Efficient Production, *The Production Engineer*, 51, 2, 53-57 (1972)

Peterson, C. and Söderberg, B., A new method for mapping optimization problems onto neural networks. *Int. J. Neural Syst.* **1**, 3-22 (1989)

Prasad, R. and Rajan, V.N., Group Technology Cell Formation Using the ART Neural Network Paradigm, *Intelligent Engineering Through Artificial Neural Networks, vol. 4*, C.H. Dagli et al (eds.), ASME Press, 1085-1090 (1994)

Suresh, N.C. and Kaparthi, S., Performance of Fuzzy ART neural network for group technology formation, *Int. J. Prod. Res.* **32**, 1693-1714 (1994)

Van den Bout, D.E. and Miller, T.K., Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing. *Biological Cybernetics*, **62**, 129-139 (1989)

Van den Bout, D.E. and Miller, T.K., Graph Partitioning Using Annealed Neural Networks. *IEEE Trans. Neural Networks* **1**, 192-203 (1990)

Venugopal, V. and Narendran, T.T., Machine-cell formation through neural network models. *Int. J. Prod. Res.*, **32**, 2105-2116 (1994)

Wang, J. and Roze, C., Formation of machine cells and part families: a modified p-median model and a comparative study, *Int. J. Prod. Res.*, **35**, 5 1259-1286 (1997)