Meta-Learning Evolutionary Artificial Neural Network for Selecting Flexible Manufacturing Systems

Arijit Bhattacharya[1], Ajith Abraham[2], Crina Grosan[4], Pandian Vasant[3], Sang Yong Han[2]

[1]The Patent Office, Bouddhik Sampada Bhawan
CP-2, Sector V, Salt Lake , Kolkata 700 091, West Bengal, India.
arijit.bhattacharya2005@gmail.com
[2]School of Computer Science and Engineering, Chung-Ang University
221, Heukseok-dong, Dongjak-gu, Seoul 156-756, Korea
ajith.abraham@ieee.org, hansy@cau.ac.kr
[3]Universiti Teknologi Petronas, 31750 Tronoh, BSI, Perak DR, Malaysia.
pvasant@gmail.com
[4]Department of Computer Science, Babes-Bolyai University, Romania.
crina.grosan@gmail.com

Abstract. This paper proposes the application of Meta-Learning Evolutionary Artificial Neural Network (MLEANN) in selecting flexible manufacturing systems (FMS) from a group of candidate FMS's. First, multi-criteria decision-making (MCDM) methodology using an improved *S*-shaped membership function has been developed for finding out the "best candidate FMS alternative" from a set of candidate-FMSs. The MCDM model trade-offs among various parameters, viz., design parameters, economic considerations, etc., affecting the FMS selection process in multi-criteria decision-making environment. Genetic algorithm is used to evolve the architecture and weights of the proposed neural network method. Further, a back-propagation (BP) algorithm is used as the local search algorithm. All the randomly generated architecture of the initial population are trained by BP algorithm for a fixed number of epochs. The learning rate and momentum of the BP algorithm have been adapted suiting the generated data of the MCDM problem. The selection of FMS are made according to the error output of the results found from the MCDM model.

1. Introduction

Flexible manufacturing system (FMS) is a set of integrated computer controlled automated material handling equipments and numerical controlled machine tools capable of processing a variety of part types. Due to the competitive advantages like flexibility, speed of response, quality, reduction of lead-time, reduction of labour etc., FMSs are now-a-days gaining popularity in industries.

Today's manufacturing strategy is purely a choice of alternatives. The better the choice, more will be the productivity as well as the profit maintaining quality of product and responsiveness to customers. In this era of rapid globalisation the overall objective is to purchase a minimum amount of capacity (i.e., capital investment) and utilize it in the most effective way. Though FMS is an outgrowth of existing manufacturing technologies, its selection is not oft studied. It has been a focal point in manufacturing related research since early 1970s. FMS provides a low inventory environment with unbalanced operations unique to the conventional production environment. Process design of FMS consists of a set of crucial decisions that are to be made carefully. It requires decision-making, e.g., selection of CNC machine tool, material handling system, product mix, etc. The selection of a FMS thus requires trading-off among the various parameters of the FMS alternatives. The selection parameters are conflicting in nature. High quality management is not enough for dealing with the complex and ill-structured factors that are conflicting-in-nature [4]. Therefore, there is a need for sophisticated and applicable technique to help the decision-makers for selecting the proper FMS in a manufacturing organization.

AHP is widely used for tackling FMS selection problems due to the concept's simplicity and efficiency [9]. Ayag [3] uses the AHP technique for the evaluation of the hardware and software components for a targeted computer-aided system and uses a simulation generator integrated with the AHP in order to try the alternatives that are ranked by the AHP study, on a real-life product organization model of a company, until a model is found that provides the best performance values as determined by the company's management.

Triantaphyllou, and Mann [10] suggest that decision-maker should be very cautious while using AHP and MCDM to engineering problems. There is sufficient evidence to suggest that the recommendations made the AHP should not be taken literally. As a matter of fact, the closer the final priority values are with each other, the more careful the user should be.

In Abdi and Labib's [1] work AHP is employed for structuring the decision-making process for selection of a manufacturing system among feasible alternatives based on the "*Reconfigurable Manufacturing System*" (RMS) study. The AHP model highlights manufacturing responsiveness as a new economic objective along with classical objectives such as low cost and high quality. The forward-backward process is also proposed to direct and control the design strategy under uncertain conditions during its implementation period [1]. Expert Choice software is applied to examine the structure of the proposed model and achieve synthesise/graphical results considering inconsistency ratios. The results are examined by monitoring sensitivity analysis while changing the criteria priorities. Finally, to allocate available resources to the alternative solutions, a (0-1) knapsack formulation algorithm is represented by Abdi and Labib [1].

But the above works and many other associated published works in the field of MCDM application to select best possible FMS alternative from a group of candidate-FMSs contain data with hidden errors. Thus, an attempt has been made in this paper using Meta-Learning Evolutionary Artificial Neural Network (MLEANN) [2] approache to select the best possible FMS from a group of candidate-FMSs. The selection is made trading off the errors of output data while using the fuzzy-MCDM approach based on AHP.

2. Evolutionary Artificial Neural Networks (EANN)

Artificial Neural Networks (ANN) are designed to mimic the characteristics of the biological neurons in the human brain and nervous system. Learning typically occurs by example through training, where the training algorithm iteratively adjusts the connection weights (synapses). Many of the conventional ANNs now being designed are statistically quite accurate but they still leave a bad taste with users who expect computers to solve their problems accurately. The important drawback is that the designer has to specify the number of neurons, their distribution over several layers and interconnection between them. Several methods have been proposed to automatically construct ANNs for reduction in network complexity that is to determine the appropriate number of hidden units, layers, etc. The interest in evolutionary search procedures for designing ANN architecture has been growing in recent years as they can evolve towards the optimal architecture without outside interference, thus eliminating the tedious trial and error work of manually finding an optimal network. The advantage of the automatic design over the manual design becomes clearer as the complexity of ANN increases.

We used the Meta-Learning Evolutionary Artificial Neural Network (MLEANN) framework [2] in this research. Figure 1 illustrates the general interaction mechanism with the learning mechanism of the EANN evolving at the highest level on the slowest time scale. The efficiency of evolutionary training can be improved significantly by incorporating a local search procedure into the evolution. Evolutionary algorithms are used to first locate a good region in the space and then a local search procedure is used to find a near optimal solution in this region. It is interesting to consider finding good initial weights as locating a good region in the space. Defining that the basin of attraction of a local minimum is composed of all the points, sets of weights in this case, which can converge to the local minimum through a local search algorithm, then a global minimum can easily be found by the local search algorithm if the evolutionary algorithm can locate any point, i.e., a set of initial weights, in the basin of attraction of the global minimum. In this research, back-propagation (BP) algorithm is used as the local search algorithm. All the randomly generated architecture of the initial population are trained by BP algorithm for a fixed number of epochs. The learning rate and momentum of the BP algorithm are adapted according to the problem. The basic algorithm of the proposed MLEANN framework is given below.

1. *Set t=0 and randomly generate an initial population of neural networks with architectures, node transfer functions and connection weights assigned at random.*
2. *Evaluate fitness of each ANN using BP algorithm*
3. *Based on fitness value, select parents for reproduction*
4. *Apply mutation to the parents and produce offspring (s) for next generation. Refill the population back to the defined size.*
5. *Repeat step 2*

6. *STOP when the required solution is found or number of iterations has reached the required limit.*

Architecture of the chromosome is depicted in Figure 2.

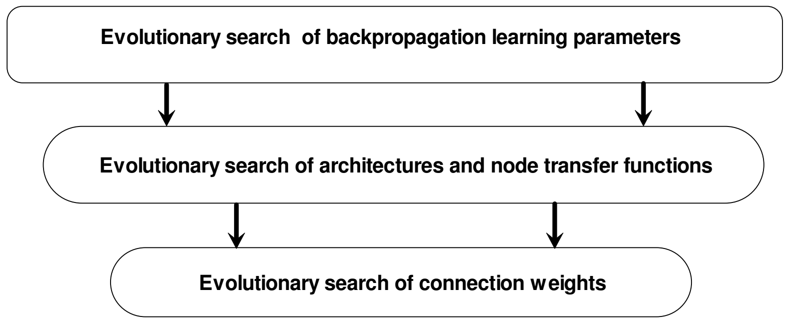

Figure 1. Interaction of various evolutionary search mechanisms in the MLEANN framework

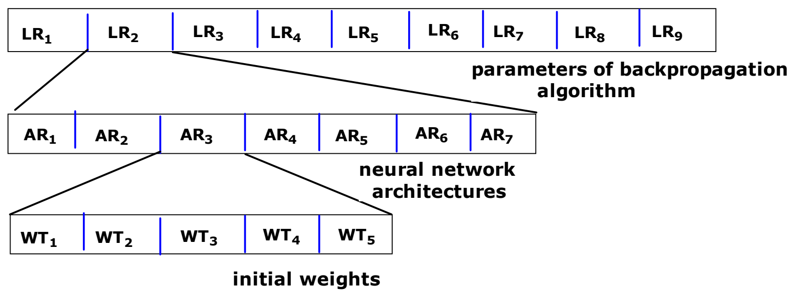

Figure 2. Chromosome representation of MLEANN

We used a special mutation operator, which decreases the mutation rate as the algorithm greedily proceeds in the search space [7]. If the allelic value x_i of the i-th gene ranges over the domain a_i and b_i the mutated gene $x_i^{'}$ is drawn randomly uniformly from the interval $[a_i , b_i]$.

$$x_i^{'} = \begin{cases} x_i + \Delta(t, b_i - x_i), if\ \omega = 0 \\ x_i + \Delta(t, x_i - a_i), if\ \omega = 1 \end{cases} \tag{1}$$

where ω represents an unbiased coin flip $p(\omega = 0) = p(\omega = 1) = 0.5$, and

$$\Delta(t,x) = x\left(1-\gamma^{\left(1-\frac{t}{t_{\max}}\right)^{b}}\right) \qquad (2)$$

defines the mutation step, where γ is the random number from the interval [*0,1*] and t is the current generation and t_{max} is the maximum number of generations. The function Δ computes a value in the range [*0,x*] such that the probability of returning a number close to zero increases as the algorithm proceeds with the search. The parameter b determines the impact of time on the probability distribution Δ over [*0,x*]. Large values of b decrease the likelihood of large mutations in a small number of generations.

2.1 Genetic Programming

MLEANN performance is compared with two Genetic Programming (GP) models to learn the different decision regions. Linear Genetic Programming (LGP) and Multi Expression Programming (MEP) are explored in this paper.

Linear Genetic Programming (LGP)

Linear genetic programming is a variant of the GP technique that acts on linear genomes [4]. Its main characteristics in comparison to tree-based GP lies in that the evolvable units are not the expressions of a functional programming language (like LISP), but the programs of an imperative language (like C/C++). The basic unit of evolution here is a native machine code instruction that runs on the floating-point processor unit (FPU). Since different instructions may have different sizes, here instructions are clubbed up together to form instruction blocks of 32 bits each. The instruction blocks hold one or more native machine code instructions, depending on the sizes of the instructions.

Multi Expression Programming (MEP)

MEP genes are represented by sub-strings of a variable length [8]. The number of genes per chromosome is constant. This number defines the length of the chromosome. Each gene encodes a terminal or a function symbol. A gene that encodes a function includes pointers towards the function arguments. Function arguments always have indices of lower values than the position of the function itself in the chromosome. The proposed representation ensures that no cycle arises while the chromosome is decoded. According to the proposed representation scheme, the first symbol of the chromosome must be a terminal symbol. In this way, only syntactically correct programs (MEP individuals) are obtained.

3. FMS Selection Problem

Nomenclature used in the MCDM model for FMS selection problem

α: Level of satisfaction of DM
OFM: Objective Factor Measure
SFM: Subjective Factor Measure
OFC: Objective Factor Cost
SI: Selection Index.
β: Fuzzy parameter which measures the degree of vagueness, $\beta = 0$ indicates crisp.

As a first step in testing the MCDM model, six different types of objective cost components have been identified for the selection problem. The total costs of each alternative are nothing but the Objective Factor Costs (OFCs) of the FMSs (refer to Table 1). The task is to select best candidate-FMS among five candidate-FMSs.

Table 1. Cost factor components (in US \$ x 10^5)

FMS → / OFCs ↓	S_1	S_2	S_3	S_4	S_5
1. Cost of Acquisition	1.500	0.800	1.300	1.000	0.900
2. Cost of Installation	0.075	0.061	0.063	0.053	0.067
3. Cost of Commissioning	0.063	0.052	0.055	0.050	0.061
4. Cost of Training	0.041	0.043	0.046	0.042	0.040
5. Cost of Operation	0.500	0.405	0.420	0.470	0.430
6. Cost of Maintenance	0.060	0.070	0.065	0.054	0.052
Total Cost (OFC)	2.239	1.431	1.949	1.669	1.550
Objective Factor Measure (OFM_i)	0.154	0.241	0.177	0.206	0.222

Table 2. Attributes influencing the FMS selection problem

Factor I	Factor II	Factor III	Factor IV	Factor V
Flexibility in pick-up and delivery	Flexibility in conveying system	Flexibility in automated storage and retrieval system	Life expectancy / pay back period	Tool magazine changing time

The subjective attributes influencing the selection of FMS are shown in Table 2. The study consists of five different attributes, viz., flexibility in pick-up and delivery, flexibility in conveying system, flexibility in automated storage and retrieval system, life expectancy / pay back period and tool magazine changing time. One may consider other attributes appropriate to selection of FMS.

The most important task for a decision-maker is the selection of the factors. Thorough representation of the problem indicating the overall goal, criteria, sub-criteria (if any) and alternatives in all levels maintaining the sensitivity to change in the elements is a vital issue. The number of criteria or alternatives in the proposed methodology should be reasonably small to allow consistent pair-wise comparisons.

Matrix 1 is the decision matrix based on the judgemental values from different judges. Matrices 2 to 6 show comparisons of the weightages for each of the attribute. Matrix 7 consolidates the results of the earlier tables in arriving at the composite weights, i.e., SFM_i values, of each of the alternatives.

$$D = \begin{bmatrix} 1 & 5 & 3 & 4 & 5 \\ \frac{1}{5} & 1 & \frac{1}{3} & \frac{1}{2} & 1 \\ \frac{1}{3} & 3 & 1 & 3 & 5 \\ \frac{1}{4} & 2 & \frac{1}{3} & 1 & 3 \\ \frac{1}{5} & 1 & \frac{1}{5} & \frac{1}{3} & 1 \end{bmatrix}$$

Matrix 1. Decision matrix

$$A_1 = \begin{bmatrix} 1 & 3 & 2 & 5 & 4 \\ \frac{1}{3} & 1 & \frac{1}{3} & 5 & 2 \\ \frac{1}{2} & 3 & 1 & 4 & 3 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{4} & 1 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{3} & 3 & 1 \end{bmatrix}$$

Matrix 2. Comparison matrix for 'F I'

$$A_2 = \begin{bmatrix} 1 & 7 & 3 & 5 & 6 \\ \frac{1}{7} & 1 & \frac{1}{4} & \frac{1}{3} & \frac{1}{2} \\ \frac{1}{3} & 4 & 1 & 3 & 4 \\ \frac{1}{5} & 3 & \frac{1}{3} & 1 & 2 \\ \frac{1}{6} & 2 & \frac{1}{4} & \frac{1}{2} & 1 \end{bmatrix}$$

Matrix 3. Comparison matrix for 'F II'

$$A_3 = \begin{bmatrix} 1 & 4 & 1 & 3 & 7 \\ \frac{1}{4} & 1 & \frac{1}{4} & \frac{1}{2} & 5 \\ 1 & 4 & 1 & 2 & 7 \\ \frac{1}{3} & 2 & \frac{1}{2} & 1 & 3 \\ \frac{1}{7} & \frac{1}{5} & \frac{1}{7} & \frac{1}{3} & 1 \end{bmatrix}$$

Matrix 4. Comparison matrix for 'F III'

$$A_4 = \begin{bmatrix} 1 & \frac{1}{3} & 5 & 3 & 6 \\ 3 & 1 & 5 & 7 & 6 \\ \frac{1}{5} & \frac{1}{5} & 1 & 2 & 3 \\ \frac{1}{3} & \frac{1}{7} & \frac{1}{2} & 1 & 2 \\ \frac{1}{6} & \frac{1}{6} & \frac{1}{3} & \frac{1}{2} & 1 \end{bmatrix}$$

Matrix 5. Comparison matrix for 'F IV'

$$A_5 = \begin{bmatrix} 1 & \frac{1}{3} & 5 & 7 & 4 \\ 3 & 1 & 5 & 6 & 4 \\ \frac{1}{5} & \frac{1}{5} & 1 & 2 & \frac{1}{2} \\ \frac{1}{7} & \frac{1}{6} & \frac{1}{2} & 1 & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & 2 & 3 & 1 \end{bmatrix}$$

Matrix 6. Comparison matrix for 'F V'

$$G = \begin{bmatrix} 0.471 & 0.076 & 0.259 & 0.131 & 0.063 \\ 0.408 & 0.512 & 0.366 & 0.273 & 0.305 \\ 0.159 & 0.051 & 0.104 & 0.501 & 0.458 \\ 0.279 & 0.246 & 0.338 & 0.103 & 0.074 \\ 0.050 & 0.117 & 0.151 & 0.075 & 0.047 \\ 0.103 & 0.075 & 0.040 & 0.047 & 0.116 \end{bmatrix}$$

Matrix 7. Final matrix to find out Global Priority

In the proposed methodology, the unit of Objective Factor Cost (OFC) is US $, whereas Objective Factor Measure (OFM) is a non-dimensional quantity. Correspondingly, the SI is also a non-dimensional quantity. Higher the SI values, the better would be the selection. The value of objective factor decision weight (α) lies between 0 and 1. For $\alpha = 0$, SI = SFM, i.e., selection is solely dependent on subjective factor measure values found from AHP and SFM values dominate over OFM values. There is no significance of considering the cost factor components for $\alpha = 0$. For $\alpha=1$, SI = OFM, i.e., OFM values dominate over the SFM values and the FMS selection is dependent on OFM values only. For α =1, the cost factors get priority than the other factors. Keeping this in mind, the values of α are taken in between 0 and 1.

The basic fuzzified equation governing the selection process is given in equation (3). It is to be remembered that the equation (3) uses the membership function (MF) as depicted by equation (6) [5].

$$\tilde{SI}_i \Big|_{\alpha=\alpha_{SFM_i}} = SI_L + \left(\frac{SI_U - SI_L}{\gamma} \right) \ln \frac{1}{C} \left(\frac{A}{\alpha_{LSI_i}} - 1 \right) \tag{3}$$

where, $$OFM_i = \frac{1}{[OFC_i \times \sum_{i=1}^{n} OFC^{-1}]} \tag{4}$$

and, non-fuzzy $$SI_i = [(\alpha \times SFM_i) + (1 - \alpha) \times OFM_i] \tag{5}$$

$$\mu(x) = \begin{cases} 1 & x < x^a \\ 0.999 & x = x^a \\ \dfrac{B}{1 + Ce^{\beta x}} & x^a < x < x^b \\ 0.001 & x = x^b \\ 0 & x > x^b \end{cases} \tag{6}$$

It is found from all the FMSs, FMS_1 has the highest SI value when objective factor decision weight lies between 0.33 and 1.00. However, FMS_2 would be preferred to other FMS candidate-alternatives when the value of level of satisfaction lies between 0.00 and 0.33.

The appropriate value of the level of satisfaction (α) is to be selected cautiously. The reason behind this is as following. The higher the α value, the dominance of the SFM_i values will be higher. The lower the α value more will be the dominance of cost factor components and subsequently, the intangible factors will get less priority.

The selection of the best candidate-FMS alternative is based on the error output of the results found from this MCDM model. The MCDM model is not described in detail for limitation of page numbers herein. One may refer to [6] for detailed description of the model and its analysis. The output data of MCDM is treated as input to MLEANN. Below are the results of using MLEANN process.

4. Experiment Results

We have applied the MLEANN framework for evaluating the candidate-FMS alternatives as discussed in earlier. For performance comparison, we used the same set of training and test data that were used for experimentations with conventional design of neural networks. For performance evaluation, the parameters used in our experiments were set to be the same for all the problems. Fitness value is calculated based on the RMSE achieved on the test set. In this experiment, we have considered the best-evolved neural network as the best individual of the last generation. All the genotypes were represented using binary coding and the initial populations were randomly generated based on the parameters shown in Table 3. Parameters used By LGP and MEP are illustrated in Tables 4 and 5 respectively. The MLEANN learning (convergence) showing the best fitness values is illustrated in Figure 3. Due to page restrictions, we have illustrated only one performance comparison for $\alpha = 0.1$. Empirical results (Root Mean Squared Error – RMSE, and Correlation Coefficient – CC) using the three methods and a direct back-propagation approach are illustrated in Table 7.

Population size	30
Maximum no of generations	25
Number of hidden nodes	5-9 hidden nodes
Activation functions	tanh *(T)*, logistic *(L)*, sigmoidal *(S)*, tanh-sigmoidal *(T*)*, log-sigmoidal *(L*)*
Output neuron	linear
Training epochs	500
Initialization of weights	+/- 0.1
Ranked based selection	0.50
Learning rate	0.15-0.01
Momentum	0.15-0.01
Elitism	5 %
Initial mutation rate	0.70

Table 3. Parameters used for evolutionary design of artificial neural networks

Parameter		Value
Population size		100
Mutation frequency		50%
Crossover frequency		95%
Number of demes		10
Program size	Initial	80
	maximum	1000

Table 4. Parameters used by LGP

Parameter	Value
Population size	50
Number of mutations per chromosome	3
Crossover probability	0.8
Code length	30
Number of generations	30
Tournament size	4

Table 5. Parameters used by MEP

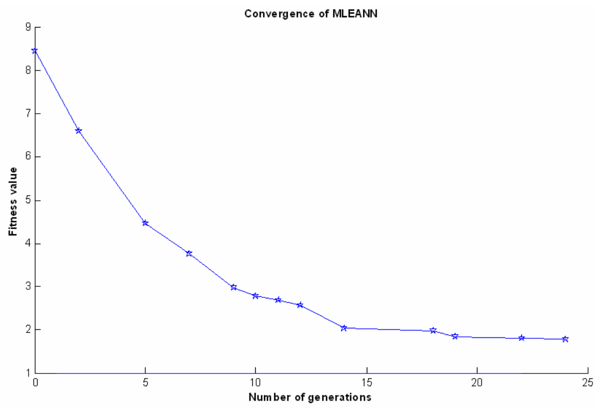

Figure 3. Convergence of MLEANN algorithm for FMS_1

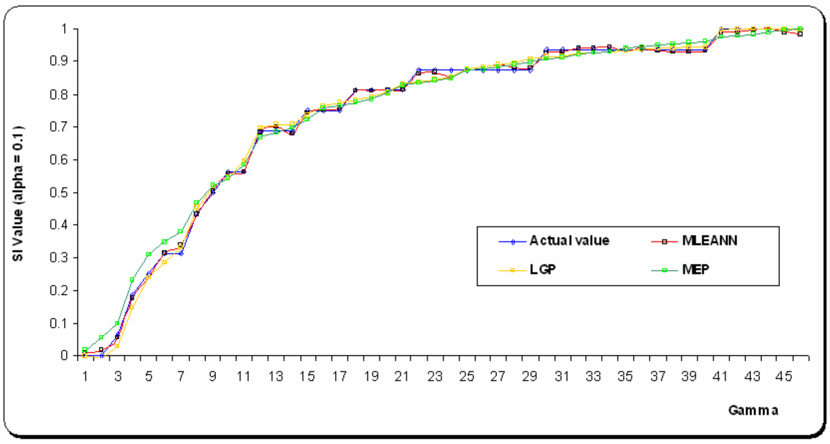

Figure 4. Performance comparison for FMS_1 ($\alpha = 0.1$)

Table 6. FMS ranking

Candidate-FMS	SI_i values	Rank #
FMS_1	0.249	#1
FMS_2	0.224	#2
FMS_3	0.210	#3
FMS_4	0.155	#5
FMS_5	0.162	#4

	FMS_1	FMS_2	FMS_3	FMS_4	FMS_5
			MLEANN		
RMSE (α=0.1)	0.0082	0.0065	0.0067	0.0084	0.0045
RMSE $\alpha = 0.5$	0.0065	0.0075	0.0056	0.0054	0.0063
RMSE $\alpha = 0.9$	0.0056	0.0087	0.0067	0.0056	0.0056
CC $\alpha = 0.1$	0.999	0.998	0.999	0.998	0.999
CC $\alpha = 0.5$	0.999	0.999	0.998	0.999	0.998
CC $\alpha = 0.9$	0.998	0.999	0.998	0.998	0.999
			ANN		
RMSE $\alpha = 0.1$	0.022	0.0365	0.0267	0.0284	0.0245
RMSE $\alpha = 0.5$	0.0265	0.0275	0.0256	0.0254	0.0263
RMSE $\alpha = 0.9$	0.0256	0.0287	0.0267	0.0256	0.0256
CC $\alpha = 0.1$	0.997	0.996	0.997	0.998	0.996
CC $\alpha = 0.5$	0.997	0.996	0.997	0.998	0.998
CC $\alpha = 0.9$	0.998	0.997	0.996	0.999	0.996
			MEP		
RMSE $\alpha = 0.1$	0.0263	0.0196	0.0201	0.0154	0.0175
RMSE $\alpha = 0.5$	0.0168	0.0199	0.0223	0.0185	0.019
RMSE $\alpha = 0.9$	0.0236	0.0287	0.0176	0.0164	0.0177
CC $\alpha = 0.1$	0.998	0.998	0.998	0.997	0.999
CC $\alpha = 0.5$	0.996	0.998	0.999	0.998	0.998
CC $\alpha = 0.9$	0.997	0.997	0.998	0.999	0.998
			LGP		
RMSE $\alpha = 0.1$	0.1820	0.1965	0.1767	0.1840	0.1745
RMSE $\alpha = 0.5$	0.0987	0.0295	0.0324	0.0354	0.02863
RMSE $\alpha = 0.9$	0.0216	0.0248	0.0257	0.0216	0.0246
CC $\alpha = 0.1$	0.998	0.995	0.996	0.996	0.998
CC $\alpha = 0.5$	0.996	0.998	0.998	0.999	0.997
CC $\alpha = 0.9$	0.998	0.998	0.996	0.999	0.996

Table 7. RMSE and CC values for the different FMS using 4 different algorithms

5. Conclusions

It is seen from the MCDM model combining both cardinal and ordinal factors for selecting FMS that at lower level-of-satisfaction (α) the chances of getting involved higher degree of fuzziness (β) increase. Therefore, a decision maker's (DM) level-of-satisfaction should be at least moderate in order to avoid higher degree of fuzziness while making any kind of decision using the MCDM model.

One underlying assumption of the MCDM methodology was that the selection is made under certainty of the information data. In reality, the information available is highly uncertain and sometimes may be under risk also. The fuzzy *S*-curve MF helps in reducing the level of uncertainty as validated further by introducing the MLEANN framework shown in Table 6. The RMSE and CC as compared and a trade off is made to select the error levels in the said MCDM model's decision. It is found that using the MLEANN framework the following decision depicted in Table 6 can be consolidated with DM's α value of $\alpha = 0.42$

Acknowledgements

This research was supported by the International Joint Research Grant of the IITA Institute of Information Technology Assessment) foreign professor invitation program of the MIC (Ministry of Information and Communication), Korea.

References

[1] Abdi, M.R. and Labib, A.W., A design strategy for reconfigurable manufacturing systems (RMSs) using analytical hierarchical process (AHP): a case study. International Journal of Production Research, 2003, 41(10), 2273–2299.

[2] Abraham A., Meta-Learning Evolutionary Artificial Neural Networks, Neurocomputing Journal, Elsevier Science, Netherlands, Vol. 56c, pp. 1-38, 2004.

[3] Ayag, Z., An analytic-hierarchy-process based simulation model for implementation and analysis of computer-aided systems. International Journal of Production Research, 2002, 40(13), 3053–3073.

[4] Banzhaf. W., Nordin. P., Keller. E. R., Francone F. D. Genetic Programming: An Introduction on The Automatic Evolution of Computer Programs and its Applications, Morgan Kaufmann Publishers, Inc., 1998.

[5] Bhattacharya, A., Abraham, A., and Vasant, P., FMS selection under disparate level-of-satisfaction of decision maker using intelligent fuzzy-MCDM model, In: C. Kahraman (ed.) *Fuzzy Multi-Criteria Decision-Making Theory and Applications with Recent Developments*, Kluwer Academic Publishers, 2006.

[6] Bhattacharya, A., Abraham, A., and Vasant, P., Measurement of level-of-satisfaction of decision maker in intelligent fuzzy-MCDM theory: A generalised approach, In: C. Kahraman (ed.) Fuzzy Multi-Criteria Decision-Making Theory and Applications with Recent Developments, Kluwer Academic Publishers, 2006.

[7] Cordón O., Herrera F., Hoffmann F. and Magdalena L. (2001), Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases, World Scientific Publishing Company, Singapore.
[8] Oltean, M., Grosan, C. Evolving Evolutionary Algorithms using Multi Expression Programming, Proceedings of the 7th European Conference on Artificial Life, Dortmund, Germany, pp. 651-658, 2003.
[9] Saaty, T.L., How to make a decision: the analytic hierarchy process. European Journal of Operational Research, 1990, 48(1) 9–26.
[10] Triantaphyllou, E., Mann, S.H., Using the analytic hierarchy process for decision making in engineering applications: some challenges, International Journal of Industrial Engineering: Applications and Practice, 1995, 2(1), 35–44.