Ensemble determination using the TOPSIS decision support system in multi-objective evolutionary neural network classifiers

M. Cruz-Ramírez, J.C. Fernández, J. Sánchez-Monedero,
F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez
Department of Computer Science and Numerical Analysis
University of Córdoba
Córdoba, Spain
Email: {mcruz,jcfernandez,i02samoj,
i22fenaf,chervas,pagutierrez}@uco.es

M. T. Lamata
Department of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
Email: mtl@decsai.ugr.es

***Abstract*—The selection of a particular neural network model belonging to the Pareto front is a problem that exists in all multi-objective algorithms. This paper proposes a novel solution to this problem based on a linear combination of the outputs of the two extremes in the Pareto front, which form an ensemble. The decision support TOPSIS method is used to determine which linear combination creates the best ensemble. This analysis selects the most representative individual that performs better in generalization than the extremes of the Pareto front do.**

***Keywords*-Ensembles, Evolutionary Algorithms, Multi-objective, Neural network, TOPSIS**

I. Introduction

Many techniques have been proposed for multiclassification tasks to improve the overall testing capability of the classifier designed (assuming, for example, the maximization of the correct classification rate), but very few methods maintain this capability in all classes (assuming, for example, maximization of the correct classification of each class). This latter objective is very important in some research areas to ensure the benefits of one classifier over another. Therefore it is necessary to perform a simultaneous optimization of the two conflicting objectives [1]. The solution to such problems (called "multi-objective") is different than a single-objective optimization. The main difference is that multi-objective optimization problems normally do not have one single solution, but a whole set of them which are all equally good. This set is called non-dominated solutions and form a Pareto front when two objectives are optimized. A non-dominated solution is one that is not dominated by any other solution, understanding the concept of Pareto dominance as follows: a solution dominates another if it better or equal in all objectives and at least one better [2].

Choosing one of the solutions of this set is a complex task that depends on the problem. One option is to select two solutions that correspond to the two extremes of the Pareto front in training [1]. These solutions represent the individuals with the best value for each one of the objective functions. The main problem with this methodology is that it does not guarantee that these individuals are the ones that give the best performance in generalization.

So ensembles are used to reduce this problem and to eliminate the task of selecting a solution. An ensemble is a compound model formed by the aggregation of several basic models, i.e., an ensemble prediction is a function of all the base models included [3].

There are many algorithms in the specialized literature to generate an ensemble. A broad review of current algorithms to create ensembles is provided in [4]. In addition to what is stated in [4], there are specific algorithms to form ensembles from the elements belonging to the Pareto front obtained through a multi-objective algorithm. These ensemble algorithms take into account the presence of multiple conflicting objectives. The problem is that individuals in the Pareto front may not be sufficiently diverse, which is essential for an evolutionary algorithm. It is therefore necessary for the multi-objective evolutionary process to lead to the optimal Pareto front while maintaining as diverse a distribution of solutions as possible [5], [6], [7], [8].

This paper proposes the creation of an ensemble through the linear combination of the outputs of the best models associated with the extremes in the Pareto front. Therefore, this ensemble will be generated from two basic models, which are diverse enough within the elements of the Pareto front. A method of decision support is employed to determine the best linear combination of the outputs of these models. There are several decision support methods, like the method proposed by Jiang in [9] or the ELECTRE method used in [10]. The decision support method that will be applied in this paper is the TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [11]. This method is based on the opinion of one or more experts, and allows us to opt for the best of several alternatives. The linear combination used for the application of this method is:

$$\lambda * O_{UE} + (1 - \lambda) * O_{LE},$$

978-1-4244-8136-1/10/$26.00 ©2010 IEEE

where O_{UE} and O_{LE} are the outputs of the models found in the upper and the lower extremes of the Pareto front respectively, with different values of λ between 0 and 1.

The paper is organised as follows: Section 2 shows an explanation of Accuracy and Minimum Sensitivity and the TOPSIS method, followed by the experimental design in Section 3. Section 4 details the application of the TOPSIS method and the conclusions are outlined in Section 5.

II. Materials and Methods

A. Accuracy and Minimum Sensitivity in Classification Problems

This section presents two measures to evaluate a classifier: the Correct Classification Rate or Accuracy, C, and the Minimum Sensitivity, MS. To evaluate a classifier, the machine learning community has traditionally used C to measure its default performance. Currently, it is simply necessary to realize that C cannot capture all the different behavioral aspects found in two different classifiers in multiclassification problems. For these problems, two performance measures are considered: traditionally-used C, $C = \frac{1}{N}\sum_{j=1}^{Q} n_{jj}$ (where Q is the number of classes, N is the number of patterns in training or testing and n_{jj} is the number of correctly classified patterns from class *j-th*); and the MS in all classes, that is, the lowest percentage of examples correctly predicted as belonging to each class, S_i, with respect to the total number of examples in the corresponding class, $MS = \min\{S_i\}$. For a more detailed description of these measures, see [1]. *A priori*, MS and C objectives could be considered to be positively correlated, but while this may be true for small values of MS and C, it is not so for values close to 1 on both MS and C. This fact justifies the use of a Multi-Objective Evolutionary Algorithm (MOEA).

B. The TOPSIS Method

The TOPSIS method (Technique for Order Preference by Similarity to Ideal Solution) is one of the most common methods in problems involving multi-criteria decisions. TOPSIS was first proposed by Hwang and Yoon in [12]. The underlying logic of TOPSIS is to define the positive ideal solution and negative ideal solution. The positive ideal solution is the one that maximizes the benefit criteria and minimizes the cost criteria; the negative ideal solution is the one that maximizes the cost criteria and minimizes the benefit criteria. The best alternative is the one that is closest to the positive ideal solution and farthest from the negative ideal solution. The alternatives are ranked by their value of "proximity to the ideal solution", and the alternative with the highest value is considered the best. An example application of TOPSIS for a multi-criteria decision problem can be seen in [13] and [14] where Li used the TOPSIS method, along with other decision support systems, to solve discrete multi-criteria decision making problems.

Table I
Characteristics for datasets

Dataset	#Patterns	#Input variables	#Classes	#Patterns per class	p^*
AustralianC	690	51	2	(307,383)	0.4449
Balance	625	4	3	(288,49,288)	0.0784
Gene	3175	120	3	(762,765,1648)	0.2400
Sonar	208	60	2	(97,111)	0.4663
Waveform	5000	40	3	(1692,1653,1655)	0.3306
Wine	178	13	3	(59,71,48)	0.2696

III. Experiments

Six datasets taken from the UCI repository are considered in the experimental design. This design was conducted using a stratified holdout procedure with 30 runs, where approximately 75% of the patterns were randomly selected for the training set and the remaining 25% for the test set.

Table I shows the features for each dataset. The total number of instances or patterns in each dataset appear, as well as the number of input variables, the number of classes, the total number of instances per class and the p^* value (the minimum of prior estimated probabilities).

The MOEA used is called Pareto Differential Evolution (PDE) and is described in [15]. This algorithm is based on Differential Evolution [16] and the work of H. Abbass [17].

During the experiment, models are trained using a fitness function based on Entropy, E (a detailed description of this function can be seen in [1]) and MS as objective functions. The validation is done by considering C and MS.

IV. Application of TOPSIS method

This section applies the TOPSIS method to determine the best linear combination of the outputs of the extreme models in the Pareto front in training, which is obtained by a multi-objective evolutionary algorithm that achieves a set of individuals which are sorted in Pareto fronts. The same procedure is followed to study the best linear combination of the outputs of these models in generalization, although only the results shown here in order not to repeat the whole process.

The application of the TOPSIS method uses the results for the six datasets from the UCI repository: AustralianC (AC), Balance (B), Gene (G), Sonar (S), Waveform (Wa) and Wine (Wi).

The first task to be done is to set the possible solutions to the problem (alternatives) and decision criteria to be used. The alternatives are selected from the following expression:

$$\lambda * O_{UE} + (1 - \lambda) * O_{LE},$$

where $\lambda \in \{0, 0.1, 0.2, ..., 1\}$ and O_{UE} and O_{LE} are the outputs of the models found in the upper and lower extremes of the Pareto front respectively. This gives 11 different alternatives, $A = \{A_1, ..., A_{11}\}$, where A_1 or C corresponds to the upper extreme of the Pareto front in training ($\lambda = 1$) and A_{11} or MS to the lower extreme of the front ($\lambda = 0$).

Table II
DECISION MATRIX X IN TRAINING

	$A_1 \equiv C$	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	$A_{11} \equiv MS$
$C_1 \equiv \widehat{\mu}_{C,AC}$	0.8506	0.8508	0.8509	0.8513	0.8528	0.8528	0.8538	0.8540	0.8549	0.8565	0.8593
$C_2 \equiv \widehat{\sigma}_{C,AC}$	0.0347	0.0349	0.0341	0.0331	0.0324	0.0331	0.0333	0.0336	0.0346	0.0352	0.0356
$C_3 \equiv \widehat{\mu}_{MS,AC}$	0.8091	0.8128	0.8155	0.8190	0.8245	0.8280	0.8333	0.8374	0.8428	0.8495	0.8575
$C_4 \equiv \widehat{\sigma}_{MS,AC}$	0.0515	0.0510	0.0488	0.0465	0.0428	0.0418	0.0397	0.0387	0.0390	0.0386	0.0361
$C_5 \equiv \widehat{\mu}_{C,B}$	0.9004	0.9050	0.9088	0.9122	0.9137	0.9151	0.9156	0.9161	0.9164	0.9156	0.9154
$C_6 \equiv \widehat{\sigma}_{C,B}$	0.0131	0.0115	0.0112	0.0098	0.0091	0.0072	0.0077	0.0064	0.0062	0.0063	0.0072
$C_7 \equiv \widehat{\mu}_{MS,B}$	0.3120	0.4060	0.5043	0.5932	0.6521	0.7248	0.7761	0.8197	0.8581	0.8825	0.9049
$C_8 \equiv \widehat{\sigma}_{MS,B}$	0.1890	0.1919	0.1979	0.1851	0.1708	0.1450	0.1201	0.0795	0.0470	0.0234	0.0099
$C_9 \equiv \widehat{\mu}_{C,G}$	0.7188	0.7194	0.7199	0.7196	0.7192	0.7201	0.7190	0.7180	0.7167	0.7138	0.7113
$C_{10} \equiv \widehat{\sigma}_{C,G}$	0.0374	0.0375	0.0375	0.0368	0.0372	0.0376	0.0380	0.0386	0.0389	0.0404	0.0420
$C_{11} \equiv \widehat{\mu}_{MS,G}$	0.5849	0.6011	0.6142	0.6253	0.6344	0.6470	0.6600	0.6709	0.6812	0.6910	0.7061
$C_{12} \equiv \widehat{\sigma}_{MS,G}$	0.1163	0.1062	0.0958	0.0838	0.0765	0.0723	0.0622	0.0566	0.0531	0.0463	0.0438
$C_{13} \equiv \widehat{\mu}_{C,S}$	0.8598	0.8620	0.8650	0.8656	0.8675	0.8692	0.8720	0.8750	0.8782	0.8816	0.8895
$C_{14} \equiv \widehat{\sigma}_{C,S}$	0.0590	0.0562	0.0554	0.0541	0.0537	0.0532	0.0507	0.0494	0.0469	0.0421	0.0380
$C_{15} \equiv \widehat{\mu}_{MS,S}$	0.8273	0.8314	0.8381	0.8404	0.8436	0.8473	0.8531	0.8580	0.8664	0.8728	0.8861
$C_{16} \equiv \widehat{\sigma}_{MS,S}$	0.1100	0.1009	0.0994	0.0957	0.0937	0.0907	0.0845	0.0792	0.0678	0.0523	0.0388
$C_{17} \equiv \widehat{\mu}_{C,Wa}$	0.7834	0.7838	0.7842	0.7846	0.7850	0.7854	0.7857	0.7860	0.7864	0.7869	0.7871
$C_{18} \equiv \widehat{\sigma}_{C,Wa}$	0.0159	0.0159	0.0159	0.0160	0.0159	0.0160	0.0161	0.0159	0.0159	0.0159	0.0160
$C_{19} \equiv \widehat{\mu}_{MS,Wa}$	0.7575	0.7598	0.7624	0.7651	0.7678	0.7706	0.7736	0.7766	0.7788	0.7812	0.7858
$C_{20} \equiv \widehat{\sigma}_{MS,Wa}$	0.0238	0.0233	0.0227	0.0217	0.0215	0.0201	0.0197	0.0187	0.0180	0.0177	0.0162
$C_{21} \equiv \widehat{\mu}_{C,Wi}$	0.9900	0.9898	0.9898	0.9898	0.9898	0.9903	0.9903	0.9905	0.9903	0.9905	0.9940
$C_{22} \equiv \widehat{\sigma}_{C,Wi}$	0.0221	0.0231	0.0231	0.0231	0.0231	0.0227	0.0227	0.0217	0.0227	0.0225	0.0201
$C_{23} \equiv \widehat{\mu}_{MS,Wi}$	0.9802	0.9802	0.9802	0.9802	0.9802	0.9802	0.9802	0.9808	0.9808	0.9814	0.9920
$C_{24} \equiv \widehat{\sigma}_{MS,Wi}$	0.0370	0.0370	0.0370	0.0370	0.0370	0.0370	0.0370	0.0348	0.0348	0.0343	0.0255

The decision criteria used will be the average of C, the standard deviation of C, the average of MS and the standard deviation of MS obtained in training after running the algorithm 30 times. Therefore, when working with 6 datasets and the same 4 criteria for each of them, this involves a 24 criteria decision, $C = \{C_1, ..., C_{24}\}$, where $C_1 = \widehat{\mu}_C$, $C_2 = \widehat{\sigma}_C$, $C_3 = \widehat{\mu}_{MS}$ and $C_4 = \widehat{\sigma}_{MS}$ are the statistics for the AustralianC dataset, the following four criteria correspond to the statistical values for the Balance dataset and so on, ending up with the statistical values of the Wine dataset.

Regarding these criteria, it is desirable to maximize the values of $\widehat{\mu}_C$ and $\widehat{\mu}_{MS}$ and minimize $\widehat{\sigma}_C$ and $\widehat{\sigma}_{MS}$ for all datasets.

After determining the alternatives and decision criteria, the decision matrix X is constructed, where the element X_{ij} is the value of decision criteria i for the alternative j (where $i = 1, ..., 24$ and $j = 1, ..., 11$).

Table II shows the decision matrix X of the classification problem for these 6 databases. The data shown in this table are those obtained with the training sets. For example, the AustralianC for the pair (C_1, A_2) would have the element X_{12} whose value is 0.8508.

Starting from the decision matrix X, the normalized decision matrix N will be built by dividing each element X_{ij} by the Euclidean norm:

$$N_{ij} = \frac{X_{ij}}{\sqrt{\sum_{j=1}^{11} (X_{ij})^2}}, i = 1, ..., 24, j = 1, ..., 11.$$

With this transformation the criteria lose their dimension, so we will now work with dimensionless criteria.

To assign weights to each criterion, an expert was consulted with respecto to the relative importance of some criteria over others, obtaining the results shown in Table III (Values are considered equal for all datasets).

Table III
BINARY COMPARISONS OF THE CRITERIA

	$\widehat{\mu}_C$	$\widehat{\sigma}_C$	$\widehat{\mu}_{MS}$	$\widehat{\sigma}_{MS}$
$\widehat{\mu}_C$	1	7	5	9
$\widehat{\sigma}_C$	1/7	1	1/3	5
$\widehat{\mu}_{MS}$	1/5	3	1	5
$\widehat{\sigma}_{MS}$	1/9	1/5	1/5	1

Since the comparison matrix should be positive and reciprocal with ones in the main diagonal, the expert needs only to provide the comparison values associated with the upper triangular matrix. To help the expert, the correspondence shown in Table IV was used.

Table IV
NUMERICAL VALUE OF THE COMPARISONS

Relation	Numerical value
A_i and A_j are equally important	1
A_i is moderately more important than A_j	3
A_i is more important than A_j	5
A_i is much more important than A_j	7
A_i is extremely more important than A_j	9

The scale values 2, 4, 6 and 8 represent compromises in the above values

Observing the values provided by the expert, it can be said that the criteria $\widehat{\mu}_C$ and $\widehat{\mu}_{MS}$ are more important than $\widehat{\sigma}_C$ and $\widehat{\sigma}_{MS}$. It is also noted that measures of C centralization and dispersion are more important than those of MS, as is usual in machine learning. This means that obtaining a good result in training for example, in $\widehat{\mu}_C$ is worth 5 times more than a good result in $\widehat{\mu}_{MS}$, is 7 times more value than a

Table V
VECTOR W, WEIGHTED NORMALIZED DECISION MATRIX V AND POSITIVE AND NEGATIVE IDEAL SOLUTION $(A+$ AND $A-)$ IN TRAINING

	W	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	$A+$	$A-$
$C_1 \equiv \widehat{\mu}_{C.AC}$	0.1081	0.0325	0.0325	0.0325	0.0325	0.0326	0.0326	0.0326	0.0326	0.0326	0.0327	0.0328	0.0328	0.0325
$C_2 \equiv \widehat{\sigma}_{C.AC}$	0.0183	0.0056	0.0057	0.0055	0.0054	0.0053	0.0054	0.0054	0.0054	0.0056	0.0057	0.0058	0.0053	0.0058
$C_3 \equiv \widehat{\mu}_{MS.AC}$	0.0335	0.0098	0.0099	0.0099	0.0100	0.0100	0.0101	0.0101	0.0102	0.0103	0.0103	0.0104	0.0104	0.0098
$C_4 \equiv \widehat{\sigma}_{MS.AC}$	0.0068	0.0024	0.0024	0.0023	0.0022	0.0020	0.0020	0.0019	0.0018	0.0018	0.0018	0.0017	0.0017	0.0024
$C_5 \equiv \widehat{\mu}_{C.B}$	0.1081	0.0322	0.0323	0.0325	0.0326	0.0326	0.0327	0.0327	0.0327	0.0327	0.0327	0.0327	0.0327	0.0322
$C_6 \equiv \widehat{\sigma}_{C.B}$	0.0183	0.0081	0.0070	0.0068	0.0060	0.0056	0.0044	0.0047	0.0039	0.0038	0.0039	0.0044	0.0038	0.0081
$C_7 \equiv \widehat{\mu}_{MS.B}$	0.0335	0.0045	0.0058	0.0073	0.0085	0.0094	0.0104	0.0112	0.0118	0.0123	0.0127	0.0130	0.0130	0.0045
$C_8 \equiv \widehat{\sigma}_{MS.B}$	0.0068	0.0027	0.0028	0.0029	0.0027	0.0025	0.0021	0.0017	0.0012	0.0007	0.0003	0.0001	0.0001	0.0029
$C_9 \equiv \widehat{\mu}_{C.G}$	0.1081	0.0326	0.0327	0.0327	0.0327	0.0327	0.0327	0.0326	0.0326	0.0325	0.0324	0.0323	0.0327	0.0323
$C_{10} \equiv \widehat{\sigma}_{C.G}$	0.0183	0.0054	0.0054	0.0054	0.0053	0.0053	0.0054	0.0055	0.0055	0.0056	0.0058	0.0060	0.0053	0.0060
$C_{11} \equiv \widehat{\mu}_{MS.G}$	0.0335	0.0091	0.0094	0.0096	0.0097	0.0099	0.0101	0.0103	0.0105	0.0106	0.0108	0.0110	0.0110	0.0091
$C_{12} \equiv \widehat{\sigma}_{MS.G}$	0.0068	0.0031	0.0028	0.0025	0.0022	0.0020	0.0019	0.0016	0.0015	0.0014	0.0012	0.0012	0.0012	0.0031
$C_{13} \equiv \widehat{\mu}_{C.S}$	0.1081	0.0322	0.0322	0.0324	0.0324	0.0324	0.0325	0.0326	0.0327	0.0328	0.0330	0.0333	0.0333	0.0322
$C_{14} \equiv \widehat{\sigma}_{C.S}$	0.0183	0.0064	0.0061	0.0060	0.0058	0.0058	0.0057	0.0055	0.0053	0.0051	0.0045	0.0041	0.0041	0.0064
$C_{15} \equiv \widehat{\mu}_{MS.S}$	0.0335	0.0098	0.0099	0.0099	0.0100	0.0100	0.0101	0.0101	0.0102	0.0103	0.0104	0.0105	0.0105	0.0098
$C_{16} \equiv \widehat{\sigma}_{MS.S}$	0.0068	0.0026	0.0024	0.0024	0.0023	0.0022	0.0022	0.0020	0.0019	0.0016	0.0013	0.0009	0.0009	0.0026
$C_{17} \equiv \widehat{\mu}_{C.Wa}$	0.1081	0.0325	0.0325	0.0325	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0327	0.0327	0.0327	0.0325
$C_{18} \equiv \widehat{\sigma}_{C.Wa}$	0.0183	0.0055	0.0055	0.0055	0.0055	0.0055	0.0055	0.0056	0.0055	0.0055	0.0055	0.0055	0.0055	0.0056
$C_{19} \equiv \widehat{\mu}_{MS.Wa}$	0.0335	0.0099	0.0100	0.0100	0.0100	0.0101	0.0101	0.0101	0.0102	0.0102	0.0102	0.0103	0.0103	0.0099
$C_{20} \equiv \widehat{\sigma}_{MS.Wa}$	0.0068	0.0024	0.0023	0.0023	0.0022	0.0022	0.0020	0.0020	0.0019	0.0018	0.0018	0.0016	0.0016	0.0024
$C_{21} \equiv \widehat{\mu}_{C.Wi}$	0.1081	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0326	0.0327	0.0327	0.0326
$C_{22} \equiv \widehat{\sigma}_{C.Wi}$	0.0183	0.0054	0.0057	0.0057	0.0057	0.0057	0.0056	0.0056	0.0053	0.0056	0.0055	0.0049	0.0049	0.0057
$C_{23} \equiv \widehat{\mu}_{MS.Wi}$	0.0335	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0101	0.0102	0.0102	0.0101
$C_{24} \equiv \widehat{\sigma}_{MS.Wi}$	0.0068	0.0021	0.0021	0.0021	0.0021	0.0021	0.0021	0.0021	0.0020	0.0020	0.0020	0.0015	0.0015	0.0021

good result in $\widehat{\sigma}_C$ and is worth 9 times more than a good result in $\widehat{\sigma}_{MS}$.

The weight vector associated with the criteria (W) is calculated from the values in Table III. The elements of vector W are obtained by calculating the eigenvector of the matrix of binary comparisons of the criteria. After these calculations, the following vector is obtained: $\{0.6486, 0.1098, 0.2010, 0.0408\}$. This vector corresponds to a single dataset; therefore, for the 6 datasets, a 24 element vector is needed. To do this, each of the elements is divided by 6, giving each one of the criteria ($\widehat{\mu}_C, \widehat{\sigma}_C, \widehat{\mu}_{MS}, \widehat{\sigma}_{MS}$) the value that corresponds in function of the eigenvector obtained from Table III. This is so that the sum of all the elements in vector W add up to 1 and, therefore, the weights are standardized. The values of vector W can be seen in Table V.

Once the values of W are calculated, the weighted normalized decision matrix is built. To do this, each element N_{ij} is multiplied by the weight of the *i-th* criterion.

$$V_{ij} = W_i N_{ij},\ i = 1, ..., 24, j = 1, ..., 11.$$

At this point, the positive ideal solution A^+ can be determined as well as the negative ideal solution A^-. For the positive ideal solution $A^+ = [A_1^+, ..., A_{24}^+]$, where A_i^+ is the maximum value of V_{ij}, for $j = 1, ..., 11$, in the case of the criteria that represent desirable attributes ($\widehat{\mu}_C$ and $\widehat{\mu}_{MS}$) or has a minimum value of V_{ij} for criteria that represent undesirable attributes ($\widehat{\sigma}_C$ and $\widehat{\sigma}_{MS}$). For the negative ideal solution, minimum values of V_{ij} are selected, and criteria that represent desirable attributes or maximum values of V_{ij} in the case of the *i-th* criterion represent undesirable attributes.

$$A^+ = \{V_1^+, ..., V_{24}^+\} =$$

$$= \left\{ \left(\max_j V_{ij}, i \in I \right), \left(\min_j V_{ij}, i \in I' \right) \right\}, j = 1, ..., 11,$$

$$A^- = \{V_1^-, ..., V_{24}^-\} =$$

$$= \left\{ \left(\min_j V_{ij}, i \in I \right), \left(\max_j V_{ij}, i \in I' \right) \right\}, j = 1, ..., 11,$$

where I is associated with the criteria that represent desirable attributes and I' is associated with the criteria that represent undesirable attributes.

Table V shows the values of the vector W, the weighted normalized decision matrix V and the positive and negative ideal solution (A^+ and A^-).

Each alternative ($A_j (j = 1, ..., 11)$, A^+ and A^-) can be represented geometrically as a point in a space of 24 dimensions, where the *i-th* axis measures the weighted normalized performance of this alternative according to the criterion C_i $(i = 1, ..., 24)$. Therefore, we can calculate the Euclidean distance (d_j^+ and d_j^-) of each alternative A_j for the positive ideal solution (A^+) and negative ideal solution (A^-), respectively:

$$d_j^+ = \left\{ \sum_{i=1}^{24} (V_{ij} - V_i^+)\, 2 \right\}^{\frac{1}{2}}, j = 1, ..., 11,$$

$$d_j^- = \left\{ \sum_{i=1}^{24} (V_{ij} - V_i^-)\, 2 \right\}^{\frac{1}{2}}, j = 1, ..., 11.$$

From the distances of all alternatives to the positive ideal solution and negative ideal solution, the separation index of each of the alternatives is calculated with:

$$R_j = \frac{d_j^-}{d_j^+ + d_j^-},\ j = 1, ..., 11.$$

The value of R_j always belong to the interval $[0, 1]$. If $R_j = 1$ $(d_j^+ = 0)$, then $A_j = A^+$ (ideal solution). If $R_j = 0$ $(d_j^- = 0)$, then $A_j = A^-$. That is, the closer to 1 that the separation index of an alternative is, the better the alternative is. Table VI shows the distances of each alternative to the positive and negative ideal solution and the separation index in training.

Table VI
DISTANCE AND SEPARATION INDEX IN TRAINING

	$d+$	$d-$	R
A_1	0.0108	0.0008	0.0687
A_2	0.0092	0.0019	0.1743
A_3	0.0079	0.0033	0.2915
A_4	0.0065	0.0048	0.4275
A_5	0.0056	0.0058	0.5116
A_6	0.0043	0.0073	0.6284
A_7	0.0035	0.0080	0.6927
A_8	0.0026	0.0090	0.7785
A_9	0.0019	0.0097	0.8354
A_{10}	0.0014	0.0103	0.8828
A_{11}	0.0012	0.0106	**0.9017**

Finally, all alternatives are ranked according to their separation index and select the alternative with the highest index as the best of all (The symbol ">" means "better than").

$$A_{11} > A_{10} > A_9 > A_8 > A_7 >$$
$$> A_6 > A_5 > A_4 > A_3 > A_2 > A_1$$

Since alternative A_{11} gets a higher separation index, it is considered the best alternative. This alternative corresponds to a value of $\lambda = 0.0$, i.e. the lower extreme model of the Pareto front, the model with the highest value in MS.

Table VII shows the distances to the positive and negative ideal solution and the separation index obtained by repeating the same procedure, but with the result obtained in generalization.

When sorting the alternatives according to their separation index (recall that the symbol ">" means "better than"), the best alternative is seen to be A_{10}^G. Therefore, we can say that the best ensemble in generalization formed by the outputs of the models in the extremes of the Pareto front is obtained with a value of of $\lambda = 0.1$.

$$A_{10}^G > A_{11}^G > A_8^G > A_9^G > A_7^G >$$
$$> A_6^G > A_5^G > A_4^G > A_3^G > A_2^G > A_1^G$$

Table VII
DISTANCE AND SEPARATION INDEX IN GENERALIZATION

	$d+$	$d-$	R
A_1^G	0.0102	0.0014	0.1193
A_2^G	0.0089	0.0019	0.1747
A_3^G	0.0077	0.0030	0.2834
A_4^G	0.0062	0.0046	0.4246
A_5^G	0.0046	0.0060	0.5644
A_6^G	0.0035	0.0073	0.6723
A_7^G	0.0024	0.0084	0.7812
A_8^G	0.0018	0.0090	0.8341
A_9^G	0.0020	0.0093	0.8190
A_{10}^G	0.0010	0.0101	**0.9123**
A_{11}^G	0.0012	0.0103	0.8948

V. CONCLUSIONS

The selection of a particular neural network model belonging to the Pareto front is a problem for multi-objective algorithms. That is why we propose the use of decision support methods to solve this problem.

This paper applies the TOPSIS method to determine the best aggregation of the models found at the extremes of the Pareto front to form the ensemble, because we believe that a combination of the two models will get better generalization results.

After applying the TOPSIS method in training, we conclude that the best alternative of all those considered is the A_{11}, i.e, the model with the highest value in MS. This alternative was obtained by using a value of $\lambda = 0.0$. By applying the TOPSIS method in generalization, the best alternative obtained is A_{10}^G using a value of $\lambda = 0.1$. By using this factor, the lower extreme model of the Pareto front has a weight of 90% and the model of the upper extreme has an importance of 10%. Therefore, the lower extreme model in the front is shown to be more important as it provides greater accuracy and minimum sensitivity in the generalization sets of the databases proposed.

This study suggests several lines of future research. First, it could be interesting to analyze how the decisions made by the expert in the problem affect the methodology. There could be a study on the changes produced in the results when varying the values of the preferences given by the expert. Another possibility would be to consult experts and weigh the value of their judgments according to their experience. Finally, one could use the C and MS coefficient of variation instead of the means and standard deviations as the criteria to decide which alternative produces better results.

ACKNOWLEDGEMENTS

This work has been partially subsidized by TIN 2008-06681-C06-03 project of the Spanish Ministerial Commission of Science and Technology (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain). The research of Manuel Cruz Ramírez has been funded by the "Junta de Andalucía", grant reference P08-TIC-3745.

References

[1] J. C. Fernández, F. J. Martínez, C. Hervás, and P. A. Gutiérrez, "Sensitivity versus accuracy in multi-class problems using memetic Pareto evolutionary neural networks," *IEEE Transactions Neural Networks*, vol. 21, no. 5, pp. 750–770, 2010.

[2] J. E. Friedlsend and S. Singh, "Pareto evolutionary neural networks," *IEEE Transactions Neural Networks*, vol. 16, no. 2, pp. 338–354, 2005.

[3] T. Löfström, U. Johansson, and H. Boström, "Ensemble member selection using multi-objective optimization," in *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009*, 2009, pp. 245–251.

[4] L. Rokach, "Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography," *Computational Statistics and Data Analysis*, vol. 53, no. 12, pp. 4046–4072, 2009.

[5] H. A. Abbass, "Pareto Neuro-Evolution: Constructing Ensemble of Neural Networks Using Multi-objective Optimization," in *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 3. Canberra, Australia: IEEE Press, 2003, pp. 2074–2080.

[6] A. Chandra and X. Yao, "DIVACE: Diverse and accurate ensemble learning algorithm," in *Proceedings of the Fifth International Conference on intelligent Data Engineering and Automated learning*, vol. 3177. Exeter, UK: Lectures Notes and Computer Science, Springer, Berlin, 2004, pp. 619–625.

[7] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: A survey and categorisation," *Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

[8] A. Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 417–445, 2006.

[9] Q. Jiang and C.-H. Chen, "A multi-dimensional fuzzy decision support strategy," *Decision Support Systems*, vol. 38, no. 4, pp. 591–598, 2005.

[10] L. Kiss, J.-M. Martel, and R. Nadeau, "Eleccalc - an interactive software for modelling the decision maker's preferences," *Decision Support Systems*, vol. 12, no. 4-5, pp. 311–326, 1994.

[11] C. Hwang and K. Yoon, "Multiple attribute decision making: Methods and application," *Springer-Verlag*, 1981.

[12] Y. Hwang and S. Bang, "An efficient method lo construct radial basis function neural network classifier," *Neural Networks*, vol. 10, no. 8, pp. 1495–1503, 1997.

[13] M. García Cascales and M. Lamata, "Multi-criteria analysis for a maintenance management problem in an engine factory: Rational choice," *Journal of Intelligent Manufacturing. Doi: 10.1007/s10845-009-0290-x*, 2009.

[14] H.-L. Li, "Solving discrete multicriteria decision problems based on logic-based decision support systems," *Decision Support Systems*, vol. 3, no. 2, pp. 101–119, 1987.

[15] J. C. Fernández, C. Hervás, F. J. Martínez, P. A. Gutiérrez, and M. Cruz, "Memetic Pareto differential evolution for designing artificial neural networks in multiclassification problems using cross-entropy versus sensitivity," in *Hybrid Artificial Intelligence Systems*, vol. 5572. Springer Berlin / Heidelberg, 2009, pp. 433–441.

[16] R. Storn and K. Price, "Differential evolution. a fast and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.

[17] H. A. Abbass, "A memetic Pareto evolutionary approach to artificial neural networks," in *AI2001*, M. Brooks, D. Corbet, and M. Stumptner, Eds. LNAI 2256, Springer-Verlag, 2001, pp. 1–12.