Hybrid Pareto Differential Evolutionary Artificial Neural Networks to Determined Growth Multi-classes in Predictive Microbiology

M. Cruz-Ramírez*, J. Sánchez-Monedero, F. Fernández-Navarro,
J.C. Fernández, and C. Hervás-Martínez

Department of Computer Science and Numerical Analysis, University of Córdoba,
Rabanales Campus, Albert Einstein building 3° floor, 14071, Córdoba, Spain

Abstract. The main objective of this work is to automatically design artificial neural network, ANN, models with sigmoid basis units for multiclassification tasks in predictive microbiology. The classifiers obtained achieve a double objective: high classification level in the dataset and high classification level for each class. For learning, the structure and weights of the ANN we present an Hybrid Pareto Differential Evolution Neural Network (HPDENN), a Differential Evolutionary approach based on the PDE multiobjective evolutionary algorithm . The PDE algorithm is augmented with a local search using the improved Resilient Backpropagation with backtraking–*IRprop*$^+$ algorithm. To analyze the robustness of this methodology, we have applied it to two complex problems of classification in predictive microbiology (*Staphylococcus Aureus* and *Shigella Flexneri*). The results obtained in Correct Classification Rate (C) and Minimum Sensitivity (S) for each class show that the generalization ability and the classification rate in each class can be more efficiently improved within this multiobjective algorithm.

Keywords: Neural Networks; Multiobjective; Accuracy; Sensitivity; Multiclassification; Memetic Algorithms; Differential Evolution; Predictive Microbiology.

1 Introduction

Growth/No Growth models have been arisen in the predictive microbiology field as an approach to determine the ability of growth of microorganisms. At this respect, many works have been published in recent years for both spoilage and pathogenic microorganisms. This fact is mainly due to the necessity of gaining knowledge, by using mathematical models, about the microbial behaviour in limiting conditions that prevent growth. Consequently, these mathematical models may lead to more realistic estimations of food safety risks and can provide

[*] Corresponding author at: E-mail address: i42crram@uco.es Phone: +34-957218349 This work has been partially subsidized by TIN 2008-06681-C06-03 (MICYT), FEDER funds and the P08-TIC-3745 project of the "Junta de Andalucía" (Spain).

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part III, LNAI 6098, pp. 646–655, 2010.

useful quantitative data for the development of processes which allow production of safer food products [1].

The main problems in modelling microbial interface are related to the abrupt transition, i.e. the large change in the value of growth probability (p) at a very narrow range of environmental factors, that is encountered between growth and no growth conditions. Then, to define more properly the interface, growth and no growth should only be considered if all replicas (or at least a very high percentage of them) grow and do not grow, respectively. Indeed, in this paper four observed microbial responses are obtained based in the probability of growth for a microorganism, ($p = 1$ (growth), G; $0.5 \leq p < 1$ (growth high probability), GHP; $0 < p < 0.5$ (growth low probability), GLP, and $p = 0$ (no growth), NG).

The importance of the use of ANNs in predictive microbiology as an alternative to regression techniques was stated by Basheer and Hajmeer [2] due to their flexibility and high degree of accuracy to fit to experimental data. In this work, we discuss learning and generalization improvement of classifiers designed using a Multi-Objective Evolutionary learning Algorithm (MOEA) [3] for the determination of growth limits for two pathogenic *Staphylococcus Aureus* and *Shigella Flexneri*. We specifically investigate the generation of neural network classifiers that achieve high classification level for each class. The methodology is based on two measures: the Correct Classification Rate, C, and the Sensitivity, S, as the minimum of the sensitivities of all classes.

The basic structure of our MOEA has been modified by introducing an additional step, where some individuals in the population have been enhanced by a local search method. For this purpose, a Hybrid Pareto Differential Evolution Neural Network (HPDENN) algorithm has been developed.

The rest of the paper is organized as follows. In section 2 an explanation of Accuracy and Sensitivity is shown. Section 3 describes the HPDENN algorithm, followed by the experimental design in Section 4. Section 5 shows the obtained results and finally, the conclusions are drawn in Section 6.

2 Related Works

2.1 Accuracy and Sensitivity

In this section we present two measures to evaluate a classifier: the Correct Classification Rate or Accuracy, C, and the Sensitivity, S. To evaluate a classifier, the machine learning community has traditionally used C to measure its default performance. Actually, we simply have to realize that accuracy cannot capture all the different behavioural aspects found in two different classifiers in multiclassification problems. For that problems we consider two performance measures; the traditionally-used accuracy, $C = \frac{1}{N}\sum_{j=1}^{Q} n_{jj}$ (where Q is the number of classes, N is the number of patterns in training or testing and n_{jj} is the number of patterns from class *j-th* that are correctly classified), and the minimum of the sensitivities of all classes, S, 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, $S = \min\{S_i\}$. The Sensitivity versus

Accuracy pair (S, C) expresses two features associated with a classifier: global performance (C) and the rate of the worst classified class (S). The selection of S as a complementary measure of C can be justified upon considering that C is the weighted average of the sensitivities of each class.

One point in (S, C) space *dominates* another if it is above and to the right, i.e. it has more Accuracy and greater Sensitivity. Let C and S be respectively the accuracy and the sensitivity associated with a classifier g, then $S \leq C \leq 1 - (1 - S)p^*$, where p^* is the minimum of the estimated prior probabilities. Therefore, each classifier will be represented as a point in the shaded region in Fig. 1, hence the area outside of the triangle is marked as unfeasible.

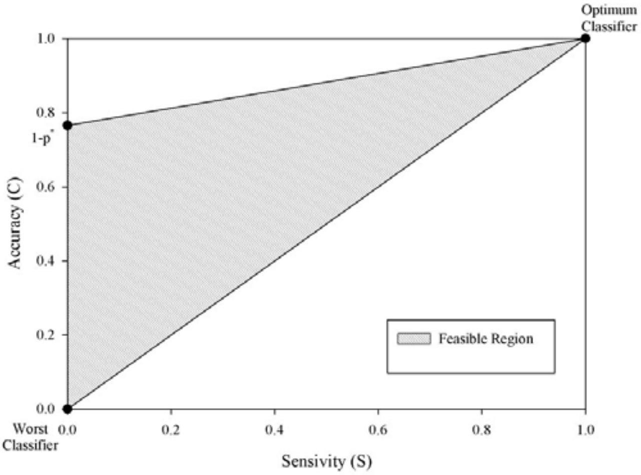

Fig. 1. Feasible region in the two dimensional (S, C) space

The area inside the triangle in Fig. 1 may be feasible (attainable), or may not be, depending upon the classifier and the difficulty of the problem. *A priori*, we can think that S and C objectives can be positively correlated, but while this may be true for small values of S and C, it is not for values close to 1 on both S and C. In this way competitive objectives are at the top right corner of the shaded region. This fact justifies the use of a MOEA.

3 Learning Methodology

At the beginning of this section describes the neural networks employed. Then, the proposed algorithm is shown and concludes with a description of the local search algorithm used.

3.1 Base Classifier Framework

In this paper we use the Base Classifier Framework described in [4].

3.2 Memetic Pareto Algorithm

We construct a MOEA with a local search algorithm, named as Hybrid Pareto Differential Evolutionary Neural Network (HPDENN), that tries to move the classifier population towards the optimum classifier located at the $(1, 1)$ point in the (S, C) space. The MOEA proposed is based on the PDE [5] and the local search algorithm is the Improved Resilient Backpropagation–*IRprop*$^+$ [6].

The Memetic Multiobjective Evolutionary Neural Network algorithm used in this work considers a fully specified ANN as an individual and it evolves architectures and connection weights simultaneously. The ANNs are represented using an object-oriented approach and the algorithm deals directly with the ANN phenotype. Each connection is specified by a binary value, which indicates whether the connection exists, and a real value representing its weight.

The HPDENN is based on the algorithm described in [4]. In HPDENN, local search does not apply to all childs to be added to the population. Instead, the childs most representative of the population are optimized in some generations. The pseudocode shown in Fig. 2.

```
Create a random initial population
while Stop condition is not met do
    Evaluate population
    Adjust the size of the population
    while The population is not complete do
        Select parents
        Cross parents
        Mutate the child
        Evaluate the child
        Add the child in the population according to dominance relationships with the main parent
    end while
    if $k\ mod(LS) = 0$ then
        if Number of individuals of the first Pareto front of $P_k < num$ then
            Apply iRprop$^+$ to the individuals of the first Pareto front
        else
            Generate $num$ cluster in the first Pareto front using K-means
            Apply iRprop$^+$ to the $num$ centers
        end if
    end if
end while
```

Fig. 2. HPDENN algorithm pseudocode

The algorithm starts generating a random population P_0 of size M. The population is sorted according to the non-domination concept explained in Section 2.1. Dominated individuals are removed from the population. Then the population is adjusted until its size is between 3 and half the maximum size by adding dominated individuals or deleting individuals according to their distance from the nearest neighbour respectively. After that, the population is completed with news child generated from three randomly selected individuals of the population. The child is generated by crossing the three parents. The resultant child is a perturbation of the main parent. This perturbation occurs with a probability P_c for each neuron. This perturbation may be structural, according to the expression (1), with which neurons are removed or added to the hidden layer;

or parametric, according to the expression (2) (for de hidden layer) or (3) (for the output layer), with which weight of the main parent is modified with the difference of the weights of secondary parents.

$$\rho_h^{child} \leftarrow \begin{cases} 1 \; if \, (\rho_h^{\alpha_1} +) \, N(0,1) \, (\rho_h^{\alpha_2} - \rho_h^{\alpha_3}) \geq 0.5 \\ 0 \; otherwise \end{cases} \tag{1}$$

$$w_{ih}^{child} \leftarrow w_{ih}^{\alpha_1} + N(0,1) \, (w_{ih}^{\alpha_2} - w_{ih}^{\alpha_3}) \tag{2}$$

$$w_{ho}^{child} \leftarrow w_{ho}^{\alpha_1} + N(0,1) \, (w_{ho}^{\alpha_2} - w_{ho}^{\alpha_3}) \tag{3}$$

Afterwards, the mutation operator is applied on the child. The mutation operator consists on adding or deleting neurons in the hidden layer depending on a P_m probability for each them. Taking into account the maximum number of hidden neurons that may exist in an individual in a specific problem, the probability will be used as many times as number of neuron has the classifier. If the neuron exists, is deleted, but if it does not exist, then it is created and the weights are established randomly, according to the expression (4).

$$\rho_h^{child} \leftarrow \begin{cases} 1 \; if \rho_h^{child} = 0 \\ 0 \; otherwise \end{cases} \tag{4}$$

Finally, the child is added to the population according to dominance relationships with the main parent. In some generations, depending on the size of the first Pareto front, local search is applied to all individuals in the first Pareto front or the most representative individuals in this front (obtained by the K-means algorithm [7]).

3.3 Local Search Algorithm

The combination of Evolutionary Algorithms, EA, and local procedures, EAs would carry out a global search inside the space of solutions, locating ANNs near the global optimum and the local procedure would arrive quick and efficiently to the best solution. This type of algorithms receives the name of Memetic or Hybrid Algorithms [8].

Many MOEAs use local optimizers to fine tune the ANNs weights. This is called "lifetime learning" and it consists in updating each individual regarding the approximation error. In addition, the weights modified during the lifetime learning are encoded back to the chromosome, which is known as the Lamarckian type of inheritance. This procedure has a high computational cost, something that we wanted to avoid. For this reason we propose the following.

For reducing the runtime, local search is applied only in three generations of evolution, the first to start the second half and the third at the end.

The local search algorithm is applied once the population is completed. Thus, local search is not applied to those children who are rejected. Local search does not apply to all individuals, but to the most representative. The process for selecting these individuals is as follows: If the number of individuals in the first

Pareto front is less than or equal to the desired number of clusters (*num*), local search is carried out without necessity of applying the K-means [7]. Instead, if the number of individuals of the first front is greater than *num*, the K-means is applied to the first front to get the most representative *num* individuals to which local search will be applied.

This local search will improve the obtained Pareto front in only one objective, specifically in the direction of the objective that tries to minimize the classification error.

As far as we are concerned, *Rprop* (resilient Backpropagation) algorithm [9] is used to be one of the best techniques in terms of convergence speed, accuracy and robustness.

4 Experiments

To analyze the robustness of the proposed methodology, in the experimental design we consider two complex problems on predictive microbiology for describing the behavior of pathogen and spoilage micro-organism under a given set of environmental conditions. The objective is to determine the conditions under which these microorganisms can be classified as G/GHP/GLP/NG, and to create a neural classifier for this purpose. Specifically, we have considered as problems the pathogen growth limits of *Staphylococcus Aureus* and *Shigella Flexneri.*

In all experiment, the population size for HPDENN is established to $M = 25$. The crossover probability is 0.8 and the mutation probability is 0.1. For IRprop$^+$, the adopted parameters are $\eta^+ = 1.2$, $\eta^- = 0.5$, $\Delta_0 = 0.0125$ (the initial value of the Δ_{ij}), $\Delta_{\min} = 0$, $\Delta_{\max} = 50$ and $Epochs = 25$, see [9] for IRprop$^+$ parameters description. The optimization process is applied 3 times during the execution (each 33.33% of generations, $LS = 33.33$) and use $num = 5$ cluster in the clustering algorithm. To start processing data, each one of the input variables was scaled in the ranks $[-1.0, 1.0]$ to avoid the saturation of the signal.

In Table 1 we can see the features for each dataset. For each database we had used the fractional factorial design present in different papers [10] for *Staphylococcus Aureus* and [11] for *Shighella Flexneri*) in order to find out the growth limits of each microorganism.

During the experiment, we train models using Entropy (E) and Sensitivity (S) as objective functions, but when validated using Accuracy (C) and Sensitivity (S). E is used instead of C in training because C is a discontinuous function, which makes convergence vary difficult in neural network optimization.

Once the Pareto front is built, two methodologies are considered in order to build a neural network model with the information of the models on it. These

Table 1. Characteristics for Datasets

Dataset	#Patterns	#Training patterns	#Test patterns	#Input variables	#Classes	#Patterns per class	p^*
S. Aureus	287	146	141	3	4	$(117, 45, 12, 113)$	0.0418
S. Flexneri	123	76	47	5	4	$(39, 8, 7, 69)$	0.0569

are called HPDENN-E and HPDENN-S. These methodologies provide us single models that can be compared with other classification methods existing in the literature. The process followed in these methodologies is the next one: once the first Pareto front is calculated using the patterns of the training set, the best individual belonging to the Pareto front on Entropy (EI) is chosen for HPDENN-E, and the best individual in terms of sensitivity (SI) is selected for HPDENN-S. Once this is done, the values of C and S are obtained by testing the EI and SI individuals. Therefore we will have an individual $EI_{testing} = (C_{testing}, S_{testing})$ and an individual $SI_{testing} = (C_{testing}, S_{testing})$. This is repeated 30 times and then the average and standard deviation obtained from the individuals is estimated, $\overline{EI}_{testing} = (\overline{C}_{testing}, \overline{S}_{testing})$ and $\overline{SI}_{testing} = (\overline{C}_{testing}, \overline{S}_{testing})$. The first expression is the average obtained taking entropy into account as the primary objective, and the second one is obtained by taking sensitivity into account as the primary objective. So, the opposite extremes of the Pareto front are taken in each of the executions. Hence, the first procedure is called HPDENN-E (Entropy) and the second HPDENN-S (Sensitivity).

5 Results

In Table 2 we present the values of the average and the standard deviation for C, S and runtime in 30 runs of all experiments performed. It can be seen that de HPDENN algorithm produces good results with respect to C, S and runtime. In fact, from a purely descriptive point of view, HPDENN algorithm obtains the best result in C in a dataset (equal that MPANN), the best result in S in *Staphylococcus Aureus* and the best result in runtime in *Shighella Flexneri* (with times well below those of MPANN).

In Fig. 3, we can see the graphical results obtained for HPDENN algorithm for the datasets *Staphylococcus Aureus* and *Shigella Flexneri* in the training (S, E) and the test (S, C) spaces. For the (S, E) space we select the Pareto front for one specific run output of the 30 realized for each dataset, concretely the execution that presents the best individual on Entropy in training, where Entropy and Sensitivity are the objectives that guide HPDENN. On the (S, C)

Table 2. Statistical Resuls for PDE, MPANN and HPDENN

Dataset	Method	$C(\%)$	$S(\%)$	runtime(s)
S. Aureus	PDE-E	$71,27 \pm 2,50$	0 ± 0	$49,34 \pm 5,24$
	PDE-S	$53,52 \pm 6,57$	$20,59 \pm 10,98$	$49,34 \pm 5,24$
	MPANN-E	$74,44 \pm 1,41$	0 ± 0	$309,14 \pm 11,51$
	MPANN-S	$66,71 \pm 8,87$	$9,36 \pm 10,82$	$309,14 \pm 11,51$
	HPDENN-E	$73,00 \pm 1,59$	0 ± 0	$48,72 \pm 3,70$
	HPDENN-S	$53,64 \pm 7,24$	$21,81 \pm 14,02$	$48,72 \pm 3,70$
S. Flexneri	PDE-E	$83,75 \pm 1,04$	0 ± 0	$234,90 \pm 17,93$
	PDE-S	$84,04 \pm 3,77$	$9,77 \pm 14,82$	$234,90 \pm 17,93$
	MPANN-E	$87,02 \pm 0,85$	0 ± 0	$1956,93 \pm 45,28$
	MPANN-S	$87,02 \pm 0,85$	0 ± 0	$1956,93 \pm 45,28$
	HPDENN-E	$87,02 \pm 1,16$	0 ± 0	$249,12 \pm 13,99$
	HPDENN-S	$86,59 \pm 2,80$	$2,22 \pm 8,45$	$249,12 \pm 13,99$

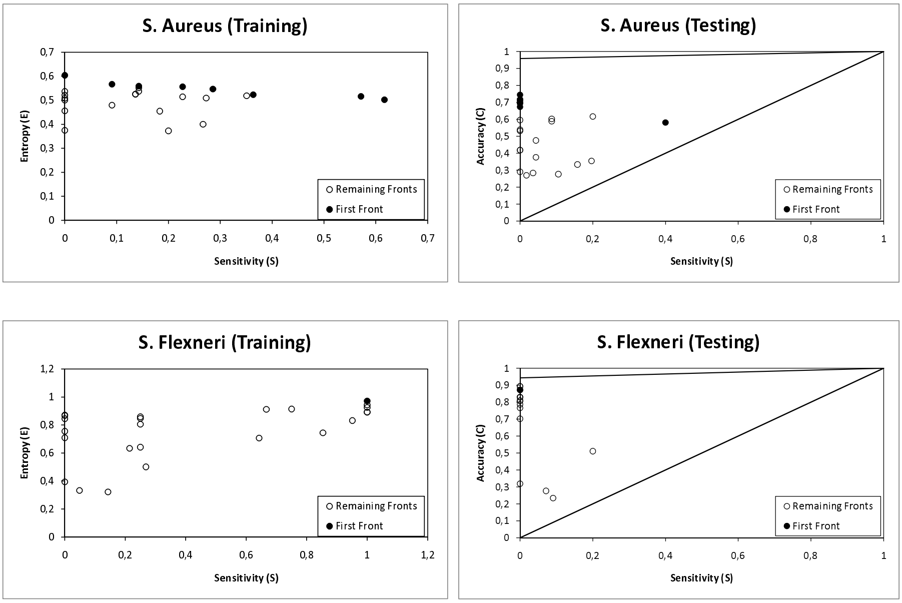

Fig. 3. Pareto front in training (S, E) and (S, C) associated values in testing in one specific run of the 30 runs carried out

testing graphics we show the S (Sensitivity) and C (Accuracy) values over the testing set for the individuals who are reflected in the (S, E) training graphics. Observe that the (S, C) values do not form Pareto fronts in testing, and the individuals that in the training graphics were in the first Pareto front, now can be located within space in a worst region. In general the structure of a Pareto front in training is not maintained in testing. Sometimes it is very difficult to obtain classifiers with a high percentage of classification and a high percentage of sensitivity, for this reason some fronts have few individuals.

In order to determine the best methodology for training MLP neural networks (in the sense of its influence on the Accuracy, C, and Sensitivity, S, in the test dataset), and minimum runtime, R, in the training procedure; an ANalysis Of the VAriance of one factor (ANOVA I) statistical method and the non parametric Kruskal-Wallis (K-W) tests were used. The tests were used, depending on the satisfaction of the normality hypothesis of C and R values. The factor F_i analyzes the effect over C (or S or R) of the *i-th* level of this factor, where F_i represents the different methodologies used in the algorithm, with levels: $i = 1$ to 6, (PDE-E (PE), PDE-S (PS), MPANN-E (ME), MPANN-S (MS), HPDENN-E (HE) and HPDENN-S (HS)). It is not possible to perform these test for S, because some methodologies in the two data sets present cero values. For R, the resuls are the same for -E and -S, and then there only exist three populations. The results of the ANOVA I analysis for test C values show that for the two datasets the six training methodologies effect is statistically significant at a 5% level of

Table 3. p-values of the Snedecor's F ANOVA I or K-W test and ranking of averages of the Tamhane Statistical multiple comparison tests and M-W pair test

	S. Aureus		S. Flexneri	
	$C(\%)$	$R(s)$	$C(\%)$	$R(s)$
F or K-W test	0.000(*)	0.000(*)	0.000(*)(°)	0.000(*)
Ranking of averages	$\boldsymbol{\mu_{ME}} \geq \mu_{HE} \geq \mu_{PE} \geq \mu_{MS} \geq \mu_{HS} > \mu_{PS}$	$\mu_{MPANN} > \mu_{PDE} \geq \boldsymbol{\mu_{HPDENN}}$	$\boldsymbol{\mu_{HE}} \geq \mu_{MS}$; $\boldsymbol{\mu_{HE}} \geq \mu_{ME}$; $\boldsymbol{\mu_{HE}} \geq \mu_{HS}$; $\boldsymbol{\mu_{HE}} > \mu_{PS}$; $\boldsymbol{\mu_{HE}} > \mu_{PE}$	$\mu_{MPANN} > \mu_{HPDENN} \geq \boldsymbol{\mu_{PDE}}$

(*)*The average difference is significant with p-value=0.05, $\mu_A \geq \mu_B$: the fitness function A yields better results in mean than the fitness function B, but the difference are not significant; and $\mu_A > \mu_B$: the fitness function A yields better results in mean than the fitness function B with significant differences. The binary relation $\geq$ is not transitive.*
(°)*Kruskal-Walis Test and Mann-Whitney pairs test.*

significance. The results of the ANOVA I analysis for R show that for the two datasets the three training methodologies effect is statistically significant at a 5% level of significance. Table 3 shows the results obtained (in the second and fourth columns for C and in the third and fifth columns for R).

Because there exists a significant difference in mean for C and R using the Snedecor's F or the K-W test; we perform, in the first case, under the normality hypothesis, a post hoc multiple comparison test of the mean of C and R obtained with the different levels of each factor. We perform a Tamhane test under normality and a pair-wise Mann-Whitney test in other case. Table 3 shows the results obtained (in the second and fourth columns for C and in the third and fifth columns for R). If we analyze the test results for Accuracy C, we can observe that HPDENN-E methodology obtains results that are, in mean, similar to the obtained with MPANN-E (the second best methodology) in the two datasets, but with a statistical significant less runtime. On the other hand, the results of average Sensitivity S show that the HPDENN-S methodology obtains a performance that is better than the rest of methodologies for S in *S. Aureus* database and the second best for *S. Flexneri.* It can be noticed than the -E methodologies classify the two databases but they leave a class with no well classified pattern.

6 Conclusions

In this paper we study the application of a memetic algorithm based on differential evolution in the resolution of multiclass classification problems in redictive microbiology. With this algorithm, we intend to obtain good results in Sensitivity and Accuracy (S, C), but also decrease the computational cost to reduce the runtime. We have proposed to apply local search to the most representative individuals of the population, selected through clustering techniques, to optimize the most promising individuals. This memetic algorithm has obtained similar results in C and better in S with respect to MPANN. In addition, HPDENN runs much faster. That's why we recommend using those HPDENN to address multiclass problems in which good results are desired in a reduced runtime.

References

1. Garcia, D., Ramos, A.J., Sanchis, V., Marin, S.: Predicting mycotoxins in foods: A review. Food Microbiology 26, 757–769 (2009)
2. Basheer, I.A., Hajmeer, M.N.: Artificial neural networks: fundamentals, computation, design and application. Journal of Microbiological Methods 43, 3–31 (2000)
3. Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, Heidelberg (2007)
4. Fernández, J.C., Hervás-Martínez, C., Martínez-Estudillo, F.J., Gutiérrez, P.A., Cruz-Ramírez, M.: Memetic Pareto Differential Evolution for designing Artificial Neural Networks in Multiclassification Problems using Cross-Entropy versus Sensitivity. In: 10th International Work-Conference on Artificial Neural Networks (IWANN 2009), Salamanca, Spain, pp. 433–441 (2009)
5. Abbass, H.A.: A Memetic Pareto Evolutionary Approach to Artificial Neural Networks. In: Stumptner, M., Corbett, D.R., Brooks, M. (eds.) Canadian AI 2001. LNCS (LNAI), vol. 2256, pp. 1–12. Springer, Heidelberg (2001)
6. Igel, C., Hüsken, M.: Improving the Rprop Learning Algorithm. In: Proc. Proceedings of the Second International ICSC Symposium on Neural Computation (NC 2000), pp. 115–121. ICSC Academic Press (2000)
7. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
8. Cotta, C., Moscato, P.: A Gentle Introduction To Memetic Algorithms. In: Glover, F., Kochenberger, G.A. (eds.) Handbook on Metaheuristics. Kluwer Academics, Dordrecht (2001)
9. Igel, C., Hüsken, M.: Empirical evaluation of the improved Rprop learning algorithms. Neurocomputing 50(6), 105–123 (2003)
10. Valero, A., Pérez-Rodríguez, F., Carrasco, E., Fuentes-Alventosa, J.M., García-Gimeno, R.M., Zurera, G.: Modelling the growth boundaries of Staphylococcus aureus: Effect of temperature, pH and water activity. International Journal Food Microbiology 133, 186–194 (2009)
11. Zaika, L.L., Moulden, E., Weimer, L., Phillips, J.G., Buchanan, R.L.: Model for the combined effects of temperature, initial pH, sodium chloride and sodium nitrite concentrations on anaerobic growth of Shigella Flexneri. International Journal of Food Microbiology 23, 345–358 (1994)