Soft Comput (2013) 17:275–284
DOI 10.1007/s00500-012-0892-7

FOCUS

Memetic Pareto differential evolutionary neural network used to solve an unbalanced liver transplantation problem

M. Cruz-Ramírez · C. Hervás-Martínez · P. A. Gutiérrez · M. Pérez-Ortiz · J. Briceño · M. de la Mata

Published online: 24 July 2012

Abstract Donor–recipient matching constitutes a complex scenario difficult to model. The risk of subjectivity and the likelihood of falling into error must not be underestimated. Computational tools for the decision-making process in liver transplantation can be useful, despite the inherent complexity involved. Therefore, a multi-objective evolutionary algorithm and various techniques to select individuals from the Pareto front are used in this paper to obtain artificial neural network models to aid decision making. Moreover, a combination of two pre-processing methods has been applied to the dataset to offset the existing imbalance. One of them is a resampling method and the other is a outlier deletion method. The best model obtained with these procedures (with AUC = 0.66) give medical experts a probability of graft survival at 3 months after the operation. This probability can help medical experts to achieve the best possible decision without forgetting the principles of fairness, efficiency and equity.

Keywords Artificial neural networks · Liver transplantation · Multi-objective evolutionary algorithm · Organ allocations · Pre-processing

1 Introduction

Liver transplantation is an accepted treatment for patients suffering end-stage chronic liver disease. Numerous donor and recipient risk factors interact and influence the probability of survival 3 months after liver transplantation. It is critical to balance waiting-list mortality and post-transplant mortality. The objective is to devise a ranking system that predicts 3-month recipient survival following liver transplantation to complement the model for end-stage liver disease score (MELD) (Wiesner et al. 2003) in order to predict waiting-list mortality.

Most current organ allocation systems are based on the principle that the sickest patients should be treated first. The models thus developed to estimate the risk of death consider the underlying disease and urgency of the recipient, assuming that all donor livers imply the same risk of failure. This, however, is not the case: it has been shown in recent years that the risk of graft failure, and even patient death, after transplantation, differs from one recipient to another. While some patients may "tolerate" and overcome the initially poor functioning of a compromised donor organ, others may not have the same tolerance. Increasing awareness of the diversity in donor organ quality has

This paper is a significant extension of the work "Memetic Pareto differential evolutionary neural network for donor-recipient matching in liver transplantation" appearing in the International Work-Conference on Artificial Neural Networks 2011 (IWANN'11).

M. Cruz-Ramírez (✉) · C. Hervás-Martínez · P. A. Gutiérrez · M. Pérez-Ortiz
Department of Computer Science and Numerical Analysis, University of Córdoba, Córdoba, Spain
e-mail: mcruz@uco.es

C. Hervás-Martínez
e-mail: chervas@uco.es

P. A. Gutiérrez
e-mail: pagutierrez@uco.es

M. Pérez-Ortiz
e-mail: i82perom@uco.es

J. Briceño · M. de la Mata
Liver Transplantation Unit, Hospital Reina Sofía, CIBERehd, Córdoba, Spain
e-mail: javibriceno@hotmail.com

M. de la Mata
e-mail: mdelamatagarcia@gmail.com

stimulated the debate on matching specific recipient and donor factors to avoid futility, but also to avoid personal and institutional differences in organ acceptance. The insufficient supply of deceased donor livers for transplantation has motivated the expansion of acceptance criteria; such organs are covered by the terms marginal and expanded criteria livers. This context of aggressive liver utilisation motivated the donor risk index, a quantitative, objective, and continuous metric of liver quality based on factors that are known or knowable when an organ becomes available.

Thus, predicting the survival of liver transplant patients can potentially play a critical role in understanding and improving the procedure of matching the appropriate recipient with the graft. Although voluminous data related to the transplantation procedures is being collected and stored, only a small subset of the predictive factors has been used to model liver transplantation outcomes. Previous studies mainly focused on applying statistical techniques to a small set of factors selected by domain experts in order to reveal simple linear relationships between all factors and survival. Machine learning and soft computing methods offer significant advantages over conventional statistical techniques in dealing with the latter's limitations, such as normal assumptions derived from observation, independence of one type of observation from another, and linearity of the relationship between observation and output measure(s). Among these techniques, we will use artificial neural network models (ANNs) whose use in biomedicine as an alternative to other classification methods is based on different approaches: a Fisher transformation (Bishop 1995; Fisher 1936), due to its flexibility and high degree of accuracy in fitting biomedical data, generalised radial basis functions (Cruz-Ramírez et al. 2011), product unit neural networks and other types of basis functions (Haykin 1998). The networks are also used to assist in decision making for diagnosis (Farias et al. 2010) or to analyse protein sequence (Jarman et al. 2011). In the field of transplantations, ANNs have been designed to diagnose cytomegalovirus disease (Sheppard et al. 1999) and acute rejection using data obtained from post-transplantation renal biopsies after kidney transplantation (Furness et al. 1999). In addition, the use of ANNs was studied in the prediction of graft failure (Matis et al. 1995) as well as in the prediction of liver transplantation outcome (Dvorchik et al. 1996).

ANNs can be trained with evolutionary computation (EC) algorithms (Rivero et al. 2009). This methodology, widely used in the last few years to evolve neural-network architectures and weights, is known as evolutionary artificial neural networks (EANNs) and has been used in many applications (Kondo 2007; Ramasubramanian and Kannan 2006; Saxena and Saad 2007). EANNs provide a more successful platform for optimising network performance and architecture simultaneously (Gutiérrez et al. 2010).

In this report, learning and generalisation improvement of the classifiers designed using a multi-objective evolutionary learning algorithm (MOEA) (Coello Coello et al. 2007) are discussed to determine 3-month survival after liver transplantation. The data come from eleven hospitals where the generation of neural network classifiers is investigated to achieve high classification levels for each class. The methodology is based on two measures: the correct classification rate or accuracy (C), and minimum sensitivity (MS) as the minimum of the sensitivities of all classes. The main proposal of this paper is to determine which of the models obtained with the MOEA presented the best results and to use different methods for the selection of individuals from the Pareto front. The results obtained with these techniques are compared with those obtained by the ensemble methods (Löfström et al. 2009). All these methods are used by considering both the original dataset and a dataset obtained after applying resampling and outlier deletion techniques; these pre-processing techniques are applied due to the unbalanced nature of the original dataset.

The paper is organised as follows: Sect. 2 describes the liver transplant dataset; Sect. 3 shows a description of the algorithm and the selection methods; Sect. 4 explains the experimental design and the pre-processing process; Sect. 5 shows the results obtained, while the conclusions and the future work are outlined in Sect. 6.

2 Dataset description

A multi-centric retrospective analysis was conducted involving 11 Spanish units of liver transplantation, including all the consecutive liver transplants performed between January 1, 2007, and December 31, 2008. The dataset included all transplant recipients of 18 years of age or over. Recipient and donor characteristics were reported at the time of transplant. Patients excluded from the study included those undergoing partial, split or living donor liver transplantation and others receiving combined or multi-visceral transplants. All patients were followed from the date of transplant until either death, graft loss or completion of the first year after the liver transplant. Units of liver transplantation were homogeneously distributed throughout Spain.

Thus the dataset generated includes 1,001 patterns (donor–recipient pairs). For each donor–recipient pair, 16 recipient characteristics, 16 donor characteristics and 9 operative factors were reported. The characteristics of each pair can be seen in Fig. 1. The end-point variable for ANNs modeling was 3-month graft mortality. This is a binary variable equal to 0 when representing graft non-survival class and 1 when representing graft survival class. Thus, the problem refers to a binary classification problem. However, the relationship between the dependent variable and the independent/predictor variables is not known in advance.

3 Methods

This section endeavors to define conflicting evaluation measures of a classifier. These measures will guide the MOEA used. Finally, several methods to select individuals from the Pareto front are discussed, together with some ensembles techniques.

3.1 Accuracy and minimum sensitivity in classification problems

To evaluate a classifier, the machine learning community has traditionally used the correct classification rate or accuracy (C) to measure its default performance. However, C cannot capture all the different behavioural aspects found in two different classifiers. For example, if there are 100 patterns (95 of class 0 and 5 of class 1), two classifiers can obtain the same C value, both 95 %, but behave differently. One of them could classify all patterns in class 0, ignoring the class 1 (this is known as trivial classifier). On the other hand, other classifier could classify all patterns of class 1 correctly and 90 patterns of class 0. This classification does not ignore the class 1, therefore, the second classifier would be preferable to the first one. To solve this problem, two performance measures are considered: traditionally used C, as the number of patterns correctly classified, and the minimum of the sensitivities of all classes (MS), 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, $\mathrm{MS} = \min\{S_i\}$. This is, we assume the premise that a good classifier should combine a high classification rate level in the testing set with an acceptable level for each class.

In Fernández et al. (2010), C and MS are presented as objectives that could be positively correlated; however, 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, where the objectives are competitive and conflicting. This fact justifies the use of a MOEA for training ANNs to optimise both objectives.

3.2 Pareto differential evolution algorithm

This paper uses a MOEA based on differential evolution for training feedforward neural networks based on neurons with sigmoid activation function.

3.2.1 Base classifier framework

We employ in this paper the standard multilayer perceptron (MLP) containing one input layer with the independent variables or features, one hidden layer with sigmoidal hidden nodes and one output layer with one linear node.

Let a coded "1-of-J" outcome variable be $\mathbf{y}$ (i.e., the outcomes have the form $\mathbf{y} = (y^{(0)}, y^{(1)})$), where $y^{(j)} = 1$ if the pattern belongs to class j, and $y^{(j)} = 0$, in other cases);

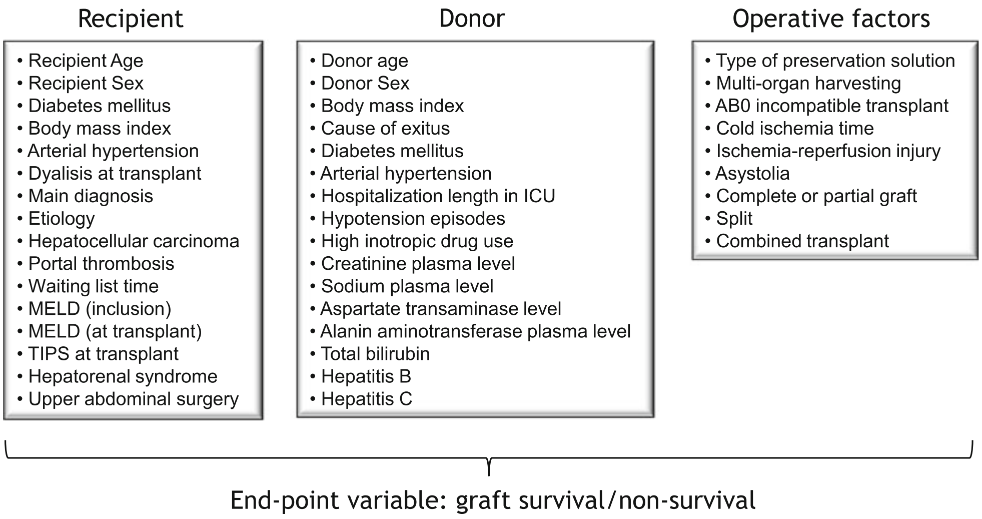

Fig. 1 Characteristics of a donor–recipient pair

and a vector $\mathbf{x} = (1, x_1, x_2, \ldots, x_K)$ of input variables, where K is the number of inputs (assuming that the vector of inputs includes the constant term to accommodate the intercept or bias). The model of an MLP can be described by the following equation:

$$f(\mathbf{x}, \boldsymbol{\Theta}) = \beta_0 + \sum_{j=1}^{M} \beta_j \sigma_j \left(w_0^j + \sum_{i=1}^{K} w_i^j x_i \right),$$

where $\boldsymbol{\Theta} = \{\beta_0, \ldots, \beta_M, \mathbf{w}_1, \ldots, \mathbf{w}_M\}$ is the weights vector of the model, $\boldsymbol{\beta} = \{\beta_0, \ldots, \beta_M\}$ is the vector of the connection weights between the hidden layer and the output layer, M is the number of hidden nodes, $\mathbf{w}_j = \{w_0^j, \ldots, w_K^j\}, \text{for} j = 1, \ldots, M,$ is the vector of input weights of the hidden node j and $\sigma(\cdot)$ is the sigmoidal activation function.

In order to tackle this classification problem, the output of the model has been interpreted from the point of view of probability through the use of the softmax activation function (Richard and David 1989), which is given by:

$$p_1(\mathbf{x}, \boldsymbol{\Theta}) = \frac{\exp f(\mathbf{x}, \boldsymbol{\Theta})}{1 + \exp f(\mathbf{x}, \boldsymbol{\Theta})}, \qquad (1)$$

where $f(\mathbf{x}, \boldsymbol{\Theta})$ is the output of the model for pattern $\mathbf{x}$ and $p_1(\mathbf{x}, \boldsymbol{\Theta})$ is the probability that pattern $\mathbf{x}$ belongs to the survival class (the probability of non-survival class is $p_0(\mathbf{x}, \boldsymbol{\Theta}) = 1 - p_1(\mathbf{x}, \boldsymbol{\Theta})$, accordingly).

Using the softmax activation function presented in expression (1), the class predicted by the MLP corresponds to the largest probability. In this way, the optimum classification rule $C(\mathbf{x})$ is the following:

$$C(\mathbf{x}) = \widehat{l}, \quad \text{where } \widehat{l} = \arg\max_{l} \; p_l(\mathbf{x}, \boldsymbol{\Theta}), \quad \text{for } l = 0, 1.$$

The MOEA uses, as objective function, the negative log-likelihood, called entropy (E) (Bishop 2006), for N observations associated with the MLP model:

$$E(\boldsymbol{\Theta}) = \frac{1}{N} \sum_{n=1}^{N} \left[-y_n^{(1)} f(\mathbf{x}_n, \boldsymbol{\Theta}) + \ln \exp f(\mathbf{x}_n, \boldsymbol{\Theta}) \right], \qquad (2)$$

where $y_n^{(1)}$ is equal to 1 if pattern $\mathbf{x}_n$ belongs to the survival class and is equal to 0 otherwise. From a statistical point of view, this approach can be seen as nonlinear binary logistic regression, where log-likelihood is optimised using a MOEA.

3.2.2 *Fitness functions*

Given a training dataset $D = \{(\mathbf{x}_n, \mathbf{y}_n); n = 1, 2, \ldots, N\}$, where $\mathbf{x}_n = (x_{1,n}, \ldots, x_{K,n})$ is the random vector of measurements taking values in $\Omega \subset R^K$, and $\mathbf{y}_n$ is the class level of the *nth* individual, C is defined by:

$$C = (1/N) \sum_{n=1}^{N} (I(C(\mathbf{x}_n) = \mathbf{y}_n)),$$

where $I(\cdot)$ is the zero-one loss function, $\mathbf{y}_n$ is the desired output for pattern n and N is the total number of patterns in the dataset. A good classifier tries to achieve the highest possible C in a given problem. However, the C measure is a discontinuous function, which makes convergence more difficult in neural network optimisation.

Thus, instead of C, the continuous function given in expression (2) is considered. The advantage of using the error function $E(\boldsymbol{\Theta})$ instead of C is that this is a continuous function, which makes the convergence more robust.

As a first objective, a strictly decreasing transformation of the $E(\boldsymbol{\Theta})$ is proposed as the fitness measure to maximise:

$$A(\boldsymbol{\Theta}) = \frac{1}{1 + E(\boldsymbol{\Theta})}, \quad 0 < A(\boldsymbol{\Theta}) \leq 1.$$

The second objective to maximise is the MS of the classifier, i.e., maximising the lowest percentage of examples correctly predicted as belonging to each class with respect to the total number of examples in the corresponding class.

3.2.3 *Memetic Pareto differential evolution neural network*

In this paper, one of the most prominent MOEA in the bibliography is used: memetic Pareto differential evolution neural network (MPDENN) algorithm. This algorithm was developed by Storn and Price (1997), modified by Abbass et al. (2001) to train neural networks and adapted to C and MS by Fernández et al. (2009). The fundamental bases of this algorithm are differential evolution (DE) (Ahandani et al. 2011) and the concept of Pareto dominance.

Figure 2 shows the framework of the MPDENN algorithm. The main feature of the MPDENN algorithm is the inclusion of a crossover operator together with the mutation operator. The crossover operator is based on a random choice of three parents, where one of them (main parent) is modified using the weighted difference of the other two parents (secondary parents). The child generated by the crossover and mutation operator is included in the population if it dominates the main parent, if it has no relationship with him or if it is the best child of the rejected children. At the beginning of each generation, dominated individuals are eliminated from the population. A generation of the evolutionary process ends when the population has been completed. In three generations of the evolution (the first initially, the second in the middle and the third at the end), a local search algorithm is applied to the most representative individuals in the population. The local search algorithm used by the MPDENN algorithm is

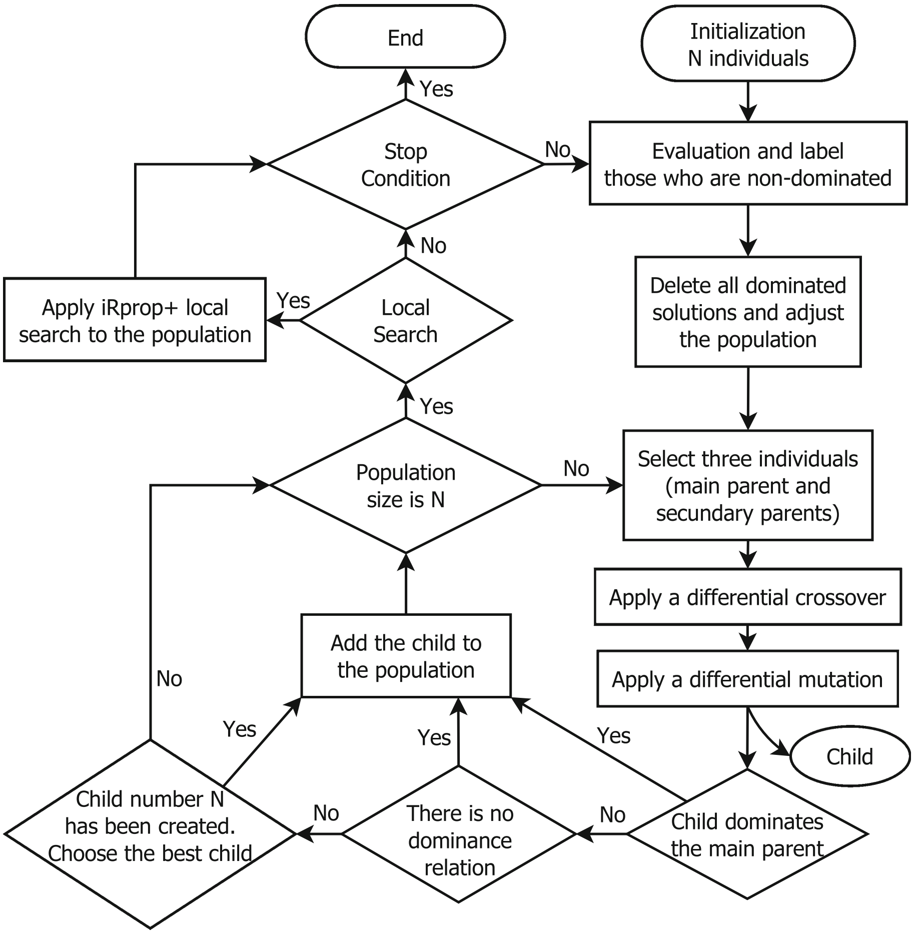

Fig. 2 Framework of MPDENN algorithm

iRprop $^{+}$ (Igel and Hüsken 2003) (more details of this algorithm in Cruz-Ramírez et al. 2010).

3.3 Selection methods

Once the execution of the MPDENN algorithm ends, various methods for the selection of individuals from the resulting Pareto front are used:

- *MPDENN-E* It consists of choosing the upper extreme of the Pareto front in training, i.e., the best individual in entropy, because one of the fitness functions of the MOEA is *E*. This method is described in Fernández et al. (2010).
- *MPDENN-MS* This technique is similar to the previous one, but selects the best individual in MS, i.e., the individual at the lower extreme of the Pareto front. This method is described in Fernández et al. (2010).
- *MPDENN-CE* This method selects all individuals from the first and second Pareto fronts obtained with the MPDENN algorithm. This group of individuals is divided into two subgroups by a two-means algorithm (because there are two objective functions, *E* and MS). The individual that is closest to the centroid of the upper cluster (cluster that takes the *E* measure into account) is selected.
- *MPDENN-CMS* This method works in a similar way to the MPDENN-CE method, but in this case, the individual that is closest to the centroid is selected, taking the MS measure into account (lower cluster). Models selected by MPDENN-CE and MPDENN-CMS are considered to be the most representative individuals in the population (the fact that these individuals do not have the greatest value in any objective using the training set does not mean that they do not generalise well). We decided to include the second Pareto front in the clustering process, in order to expand the number of individuals and to increase diversity. In addition, individuals belonging to this front may have a high classification ranking in generalisation because it is a way to avoid over-training. In the extreme case of there being only one individual in each of the fronts (there would be only two individuals), each of these individuals would be assigned to a cluster. Figure 3 shows a diagram about the selection process with the cluster.
- *MPDENN-MV* The *majority voting* (MV) is a well-known ensemble technique (Theodoridis and Koutroumbas 2006)

and, in our case, it is applied to all the individuals in the first Pareto front. With this technique, a pattern belongs to the class that has the highest number of votes, according to the independent classification of each of the elements that makes up the ensemble. Let us define a set of classifiers $D = \{D_1, \ldots, D_T\}$ and the decision of the tth classifier, D_t, for the class j and the pattern $\mathbf{x}$ as $d_{t,j}(\mathbf{x}) \in \{0,1\}, t = 1, \ldots, T$ and $j = 0, \ldots, J-1$, where T is the number of individuals in the Pareto front and J the number of classes. $d_{t,j}(\mathbf{x}) = 1$ if the tth classifier predicts the class j for the pattern $\mathbf{x}$ and zero otherwise. The MV decision can be defined as:

$$d_{\mathrm{MV}}(\mathbf{x}) = \arg\max_j \sum_{t=1}^{T} d_{t,j}(\mathbf{x}) \quad \text{for } j = 0, \ldots, J-1.$$

- *MPDENN-SA* The *simple averaging* (SA) is other ensemble technique (Theodoridis and Koutroumbas 2006) and we apply it to the first Pareto front to calculate the arithmetic mean of the probability for each of the J classes by using all the models in the ensemble. The assignment will take the class that has the highest averaged probability. Following the previous notation, the probability of the tth classifier for the class j and the pattern $\mathbf{x}$ is denoted by $p_{t,j}(\mathbf{x}) \in [0,1]$, $t = 1, \ldots, T$ and $j = 0, \ldots, J-1$. Thee decision of the SA ensemble can be defined as:

$$d_{\mathrm{SA}}(\mathbf{x}) = \arg\max_j \frac{1}{T} \sum_{t=1}^{T} p_{t,j}(\mathbf{x}) \quad \text{for } j = 0, \ldots, J-1.$$

  So this ensemble technique chooses the class with the highest probability.
- *MPDENN-WTA* In the *winner take all* (WTA) ensemble method (Theodoridis and Koutroumbas 2006), the probabilities used as the output of the ensemble are those of the model with the highest probability in one of the outputs. It is applied, again, to the individuals in the first Pareto front. The WTA decision for the pattern $\mathbf{x}$ is defined as:

$$d_{\mathrm{WTA}}(\mathbf{x}) = \arg\max_j [\max_t (p_{t,j}(\mathbf{x}))],$$

  for $t = 1, \ldots, T$ and $j = 0, \ldots, J-1$.

4 Experimental study

4.1 Experimental design

The experimental design was conducted using a stratified fourfold procedure with 10 runs per fold (40 runs in total). Training sets are composed approximately of 75 % of the patterns randomly selected, and generalisation sets are composed of the remaining 25 %. In this way, all patterns are used to train and to generalise. During the creation of these sets, a proportion of 75–25 % was also kept for the training-testing patterns of each of the participating hospitals. Table 1 shows the features of these folds.

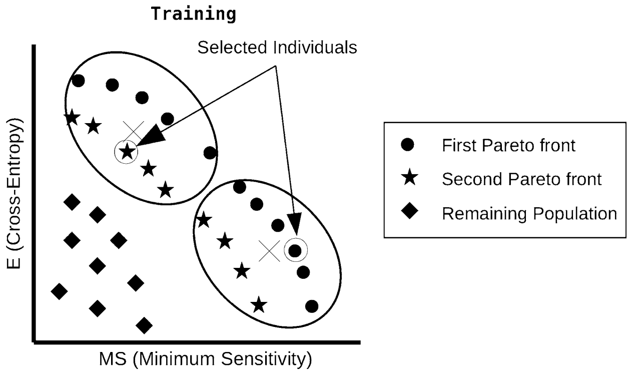

Fig. 3 Selection process using clusters of individuals

In order to make comparisons and check the goodness of the results, two common algorithms found in the literature are used. One of them is the C-support vector machine (SVM) (through the `libsvm`[1] implementation, an integrated software for support vector classification) (Chang and Lin 2011) and the other one is the Logistic Model Tree (LMT), which was run using WEKA[2] software (Witten and Frank 2005). `libsvm` is able to automatically adjust the hyper-parameters associated to this kind of models (i.e. the cost parameter, C, and the width of the Gaussian kernel, G) by using a nested fivefold cross-validation over the training set. LMT is an algorithm to deal with classification tasks that use a combination of a tree structure and logistic regression models, resulting in a single tree. By using logistic regression, explicit class probability estimates are produced rather than just crisp classifications (Landwehr et al. 2005). These algorithms are deterministic, so that a single execution is performed for each fold. Regarding their parameter values, `libsvm` algorithm has been cross-validated by accuracy, using the ranges $C = 10\hat{}\{-3, -2, \ldots, 3\}$ and $G = 10\hat{}\{-3, -2, \ldots, 3\}$. For the LMT method, the minimum number of instances at which a node is considered for splitting is 15.

4.2 Pre-processed data

Table 1 shows the characteristics of the original dataset, where its unbalanced nature can be appreciated. In order to deal with this unbalanced nature, a pre-processing method has been designed and applied to each one of the four

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[2] http://www.cs.waikato.ac.nz/ml/weka/.

Table 1 Features of the original and pre-processed datasets

Dataset	Fold	No. of training patterns	No. of generalisation patterns	No. of patterns per class in training	No. of patterns per class in generalisation
Original	1	750	251	(84–666)	(29–222)
Original	2	751	250	(85–666)	(28–222)
Original	3	751	250	(85–666)	(28–222)
Original	4	751	250	(85−666)	(28–222)
Pre-processed	1	739	251	(168–571)	(29–222)
Pre-processed	2	748	250	(170–578)	(28–222)
Pre-processed	3	740	250	(170–570)	(28–222)
Pre-processed	4	755	250	(170–585)	(28–222)

training sets. The procedure applied consists of the following two methods:

- Firstly, the synthetic minority over-sampling technique algorithm (SMOTE) (Chawla et al. 2002) is applied to the minority class (class 0) in such a way that its number of patterns is duplicated. The synthetic generated patterns are only used to train the model, not to test it, as they cannot be considered real data. These synthetic pattern were generated using information from the five nearest neighbors.
- Secondly, an outlier deletion method, the interquartile range method (Hodge and Austin 2004), has also been applied to the last class (the majority one). Thus, the number of training patterns associated with class 1 decreases since some outliers and extreme values are deleted. As the training patterns randomly changes in each fold, the final number of patterns associated to class 1 is quite different for each set. The factor used for determining the thresholds for extreme values and outliers are 6.0 and 3.0, respectively.

These methods have been configured and run using WEKA software (Hall et al. 2009). The final distribution of the dataset after applying this methodology can be seen in Table 1.

4.3 Algorithm parameters

In all the experiments, the population size for MPDENN is established as $M = 25$. The crossover probability is 0.8 and the mutation probability is 0.1. For *iRprop* $^+$ as local search algorithm, 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 $Epochs = 10$, see Igel and Hüsken (2003) for the *iRprop* $^+$ parameter description.

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. Additionally, categorical variables have been transformed into as many binary variables as possible categories.

5 Results

C and AUC represent two of the most commonly used metrics in classification (Caruana and Niculescu-Mizil 2004). Our paper uses these two metrics together with the MS and the sensitivities or accuracies separately obtained by using the patterns of each class (S^0 and S^1). C, MS, S^0 and S^1 represent threshold metrics and AUC is a probabilistic metric. Table 2 presents the values of mean and standard deviation for C, MS, AUC, S^0 and S^1 in generalisation for the 40 runs of all the experiments performed.

The analysis of the results shows that the original dataset leads to the best results in all metrics except in AUC_G. However, these classifiers are not useful to medical experts, since they might be considered trivial or random classifiers, as some of them classified all the patterns (or most) in the majority class (class 1). This happens with the models obtained by MPDENN-E, MPDENN-CE, MPDENN-MV, MPDENN-SA, SVM and LMT. For example, the mean accuracy of MPDENN-E is 1.96 % for class 0 (S_0), i.e., it classifies almost all patterns in class 1. Other of these classifiers are also random because they obtain AUC_G values very close to 0.50 (or lower). For example, the MPDNN-MS model obtains values of accuracies per class of $S_0 = 46.17$ and $S_1 = 57.51$, very close to the 50 % ideally achieved by random guessing.

Analysing the results obtained from the dataset with preprocessing, better values are observed in the AUC_G metric. These AUC_G values are not too high (but higher than those obtained with the original dataset). These values mean that the models are better, even if they do not have the best values in C_G, MS_G, S^0_G or S^1_G. The complexity of the dataset can be seen from them, since it is difficult to obtain classifiers with AUC_G values greater than 0.55. This

Table 2 Statistical results for different methods in generalisation

Method	Mean_{SD}				
	C_{G} (%)	MS_{G} (%)	AUC_{G}	S^0_{G} (%)	S^1_{G} (%)
Original dataset					
MPDENN-E	$88.07_{0.93}$	$1.96_{3.61}$	$0.5251_{0.0671}$	$1.96_{3.61}$	$99.22_{1.07}$
MPDENN-MS	$54.36_{5.60}$	$\mathbf{44.74_{8.78}}$	$0.5156_{0.0586}$	$\mathbf{46.17_{10.41}}$	$57.51_{6.22}$
MPDENN-CE	$85.59_{3.64}$	$4.70_{5.73}$	$0.5128_{0.0560}$	$4.70_{5.73}$	$96.31_{4.13}$
MPDENN-CMS	$54.34_{9.23}$	$37.26_{8.76}$	$0.4896_{0.0493}$	$40.75_{12.66}$	$57.97_{10.07}$
MPDENN-MV	$86.77_{2.77}$	$4.17_{5.95}$	$0.5279_{0.0597}$	$4.17_{5.95}$	$97.52_{3.55}$
MPDENN-SA	$86.83_{2.44}$	$3.90_{4.74}$	$0.5282_{0.0464}$	$3.90_{4.74}$	$97.42_{3.23}$
MPDENN-WTA	$79.32_{10.38}$	$14.74_{16.24}$	$0.5223_{0.0593}$	$14.74_{16.24}$	$88.99_{11.79}$
SVM	$\mathbf{88.71_{0.17}}$	$0.00_{0.00}$	$0.5000_{0.0000}$	$0.00_{0.00}$	$\mathbf{100.00_{0.00}}$
LMT	$88.41_{0.56}$	$0.00_{0.00}$	$0.4983_{0.0033}$	$0.00_{0.00}$	$99.66_{0.67}$
Pre-processed dataset					
MPDENN-E	$76.63_{4.24}$	$15.78_{9.14}$	$\mathbf{0.5314_{0.0538}}$	$15.78_{9.14}$	$85.56_{4.37}$
MPDENN-MS	$62.79_{4.53}$	$37.07_{10.35}$	$0.5301_{0.0597}$	$37.07_{10.35}$	$68.14_{4.38}$
MPDENN-CE	$72.47_{6.55}$	$21.29_{11.57}$	$0.5225_{0.0590}$	$21.29_{11.57}$	$81.16_{7.21}$
MPDENN-CMS	$68.79_{5.45}$	$26.16_{9.67}$	$0.5245_{0.0547}$	$26.16_{9.67}$	$76.75_{5.76}$
MPDENN-MV	$74.76_{3.93}$	$19.58_{11.32}$	$0.5238_{0.0544}$	$19.58_{11.32}$	$83.01_{5.49}$
MPDENN-SA	$74.88_{4.08}$	$19.85_{10.67}$	$0.5296_{0.0549}$	$19.85_{10.67}$	$82.64_{5.65}$
MPDENN-WTA	$70.60_{6.71}$	$26.44_{11.05}$	$0.5306_{0.0603}$	$26.44_{11.05}$	$77.87_{7.31}$
SVM	$86.11_{2.05}$	$3.57_{7.14}$	$0.5009_{0.0353}$	$3.57_{7.14}$	$96.62_{2.34}$
LMT	$80.81_{2.18}$	$9.75_{3.48}$	$0.4981_{0.0166}$	$9.75_{3.48}$	$89.86_{2.71}$

The best result is in bold face and the second best result in italics

makes sense due to the existence of other characteristics difficult to be consider, besides donor, recipient and transplant features, which determine the 3-month survival after liver transplantation. This version of the dataset improves the classification of the minority class (S^0_{G}), which is the most interesting class in this problem (non-survival of the graft within 3 months), while maintaining an acceptable classification rate of the majority class (S^1_{G}). For this reason, the models obtained with the pre-processed dataset are the most useful ones for medical experts' decision making.

In order to ascertain the statistical significance of the differences observed in the different methods and determine which selection method is the best for the pre-processed dataset, statistical tests for S^0_{G} and S^1_{G} are applied. First of all, a Kolmogorov–Smirnov's test (KS-test) with a significance level $\alpha = 0.05$ was used to evaluate if the different performance metrics in all the methods followed a normal distribution. Any method obtains a p value lower than the critical level for the S^0_{G} and S^1_{G} measures. Thereby a normal distribution cannot be assumed in any of the cases. As a consequence, a non-parametric Friedman's test for independent samples was selected in order to check if the method applied significantly affects the results obtained. The test concludes that these differences are significant (with a p value $= 0.00$ for S^0_{G} and S^1_{G}). So, the statistical analysis ends by applying the Wilcoxon's signed-rank test for all pairs of algorithms and the results are shown in Table 3. These results include, for each method, the number of algorithms statistically outperformed (wins, W), the number of draws (non-significant differences, D) and the number of losses (number of algorithms that outperform the method, L). Results of the MPDENN-E and MPDENN-MS methods indicate that these methods focus in a trivial manner only on one class of the problem. This behaviour is unwanted due to the nature of the problem, so these methods should not be considered when determining the best selection method. Therefore, MPDENN-CMS and MPDENN-WTA methods are preferable for S^0_{G} metric and MPDENN-MV and MPDENN-SA for S^1_{G} measures. But, in general, selecting the best method will depend on the opinion of medical experts.

Additionally, the AUC_{G} results have also been studied to ascertain possible statistically significant differences. Based on the normality KS-test for values of AUC_{G} in the pre-processed dataset, an ANalysis Of VAriance of one factor (ANOVA I) test is applied. The results of the ANOVA I test show that, in average, the methodologies' results are not significantly different. However, a Student's t test shows that there are significant differences in average (p value $= 0.072$) and in standard deviation (p value $= 0.035$) when comparing the best and the worst methods (MPDENN-E and LMT, respectively).

One possibility to obtain a final classifier may be to combine the results of one of the methods that obtain a good performance in the S_{G}^{0} metric with another method that obtain a good efficiency in the S_{G}^{1} metric. This combination could provide a useful tool for the problem of donor–recipient assignment. This combination could be a rule-based system or a weighted aggregation of the outputs of the two models, although in our opinion, the rules-based system would provide a more understandable and comprehensive tool for experts. The system would receive a set of potential recipients as input and it would form a donor–recipient pair based on this donor/organ data. These pairs would be the input for the neural network models. With the results provided by these models and using a simple set of rules, the system would determine which of the recipients should receive the organ.

Table 4 shows the results obtained by the best models of the MPDENN-CMS, MPDENN-WTA, MPDENN-MV and MPDENN-SA methods. These generalization results (obtained from a 25 % of the total number of patterns) are given by the best model from the 40 models found by the different algorithms, trained with the remaining 75 % of patterns. For the best models obtained with the methods that have the best performance in the S_{G}^{0} metric, both models obtain a good value on $\mathrm{AUC_G}$ and acceptable values in medical terms in the other metrics. Both models have similar performance, although the best MPDENN-WTA model is slightly preferable. For the other models, the best MPDENN-MV model is better than the best MPDENN-SA model because it has higher value in the C_{G} measure.

Table 3 Number of wins (*W*), draws (*D*) and losses (*L*) when comparing the different methods using the Wilcoxon's signed-rank test with $\alpha = 0.05$

	S_{G}^{0}	S_{G}^{1}
	W/D/L	*W/D/L*
MPDENN-E	0/0/6	6/0/0
MPDENN-MS	6/0/0	0/0/6
MPDENN-CE	1/2/3	2/3/1
MPDENN-CMS	4/1/1	1/1/4
MPDENN-MV	1/2/3	3/2/1
MPDENN-SA	1/2/3	3/2/1
MPDENN-WTA	4/1/1	1/2/3

Table 4 Statistical results for the best models

Best model	C_{G} (%)	$\mathrm{MS_G}$ (%)	$\mathrm{AUC_G}$	S_{G}^{0} (%)	S_{G}^{1} (%)
Best performance in the S_{G}^{0} metric					
MPDENN-CMS	66.80	42.86	0.6099	42.86	75.23
MPDENN-WTA	68.13	48.28	0.6605	48.28	70.72
Best performance in the S_{G}^{1} metric					
MPDENN-MV	82.07	3.44	0.5505	3.45	92.79
MPDENN-SA	81.27	3.45	0.5505	3.45	91.89

6 Conclusions

In this paper, the problem of donor–recipient matching when treating liver transplantation has been analysed and studied in order to help medical experts on the complex decision of donor–recipient allocation. To do so, different selection methods are used to obtain artificial neural network models from a multi-objective evolutionary algorithm that can help medical experts in donor–recipient allocation. These models are obtained from the Pareto front built through a multi-objective evolutionary algorithm where accuracy and the minimum sensitivity are the measures considered for evaluating model performance. Minimum sensitivity is used to avoid the design of models characterized by high global performance in general, but bad performance when considering the classification rate for each class (survival or non-survival). In addition, the sensitivity of each class and the AUC metric have been used to detect trivial and random classifiers, respectively.

A pre-processing procedure has been applied to the original dataset. This pre-processing removes the patterns with extreme values in the majority class and duplicates the patterns of the minority class using the well-known synthetic minority over-sampling technique algorithm. The results obtained show that the algorithm performs more consistently when using the pre-processed dataset.

With two of the best models (one with the best performance in the S_{G}^{0} metric and the other in the S_{G}^{1} measure) obtained using the pre-processed dataset, a rule-based system could be used to perform donor–recipient matching. This rule-based system should be generated by medical experts and machine learning experts, to uphold the principles of fairness, efficiency and equity. Current allocation systems are based on the risk of dying while on the waiting-list, and do not recognise distinctions in "donor organ quality". With the rule-based system, "donor organ quality" would be taken into account to improve allocation and ensure the survival of recipients.

Acknowledgments This work was supported in part by the Spanish Inter-Ministerial Commission of Science and Technology under Project TIN2011-22794, the European Regional Development fund, and the "Junta de Andalucía" (Spain), under Project P2011-TIC-7508.

M. Cruz-Ramírez's research has been subsidized by the FPU Predoctoral Program (Spanish Ministry of Education and Science) with Grant reference AP2009-0487. Finally, we would like to thank Astellas Pharma Company for their partial support and the Editor and the Reviewers for their helpful suggestions for the paper.

References

Abbass HA, Sarker R, Newton C (2001) PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. In: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, vol 2

Ahandani M, Shirjoposh N, Banimahd R (2011) Three modified versions of differential evolution algorithm for continuous optimization. Soft Comput 15(4):803–830

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin

Caruana R, Niculescu-Mizil A (2004) Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: KDD-2004—Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, pp 69–78

Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2:27:1–27:27

Chawla N, Bowyer K, Hall L, Kegelmeyer W (2002) SMOTE: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357

Coello Coello C, Lamont G, Veldhuizen D (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, Berlin

Cruz-Ramírez M, Sánchez-Monedero J, Fernández-Navarro F, Fernández J, Hervás-Martínez C (2010) Memetic pareto differential evolutionary artificial neural networks to determine growth multi-classes in predictive microbiology. Evol Intell 3(3–4):187–199

Cruz-Ramírez M, Fernández J, Fernández-Navarro F, Briceño J, de la Mata M, Hervás-Martínez C (2011) Memetic evolutionary multiobjective neural network classifier to predict graft survival in liver transplant patients. In: Genetic and evolutionary computation conference (GECCO2011), pp 479–486

Dvorchik I, Subotin M, Marsh W, McMichael J, Fung J (1996) Performance of multi-layer feedforward neural networks to predict liver transplantation outcome. Methods Inf Med 35:12–18

Farias G, Santos M, López V (2010) Making decisions on brain tumor diagnosis by soft computing techniques. Soft Comput 14(12): 1287–1296

Fernández JC, Hervás C, Martínez FJ, Gutiérrez PA, Cruz M (2009) 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, pp 433–441

Fernández JC, Martínez-Estudillo FJ, Hervás-Martínez C, Gutiérrez PA (2010) Sensitivity versus accuracy in multiclass problems using memetic Pareto evolutionary neural networks. IEEE Trans Neural Netw 21(5):750–770

Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7(7):179–188

Furness P, Levesley J, Luo Z, Taub N, Kazi J, Bates W, Nicholson M (1999) A neural network approach to the biopsy diagnosis of early acute renal transplant rejection. Histopathology 35(5):461–467

Gutiérrez PA, Hervás C, Lozano M (2010) Designing multilayer perceptrons using a guided saw-tooth evolutionary programming algorithm. Soft Comput 14(6):599–613

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11:10–18

Haykin S (1998) Neural networks: a comprehensive foundation, 2nd edn. Prentice Hall, Upper Saddle River

Hodge VJ, Austin J (2004) A survey of outlier detection methodologies. Artif Intell Rev 22:2004

Igel C, Hüsken M (2003) Empirical evaluation of the improved rprop learning algorithms. Neurocomputing 50(6):105–123

Jarman I, Etchells T, Bacciu D, Garibaldi J, Ellis I, Lisboa P (2011) Clustering of protein expression data: a benchmark of statistical and neural approaches. Soft Comput 15(8):1459–1469

Kondo T (2007) Evolutionary design and behavior analysis of neuromodulatory neural networks for mobile robots control. Appl Soft Comput 7:189–202

Landwehr N, Hall M, Frank E (2005) Logistic model trees. Mach Learn 59(1–2):161–205

Löfström T, Johansson U, Boström H (2009) Ensemble member selection using multi-objective optimization. In: IEEE symposium on computational intelligence and data mining, pp 245–251

Matis S, Doyle H, Marino I, Mural R, Uberbacher E (1995) Use of neural networks for prediction of graft failure following liver transplantation. IEEE symposium on computer-based medical systems, pp 133–140

Ramasubramanian P, Kannan A (2006) A genetic-algorithm based neural network short-term forecasting framework for database intrusion prediction system. Soft Comput 10(8):699–714

Richard D, David ER (1989) Product units: a computationally powerful and biologically plausible extension to backpropagation networks. Neural Comput 1(1):133–142

Rivero D, Dorado J, Rabuñal J, Pazos A (2009) Modifying genetic programming for artificial neural network development for data mining. Soft Comput 13(3):291–305

Saxena A, Saad A (2007) Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems. Appl Soft Comput 7:441–454

Sheppard D, McPhee D, Darke C, Shrethra B, Moore R, Jurewitz A, Gray A (1999) Predicting cytomegalovirus disease after renal transplantation: an artificial neural network approach. Int J Med Inf 54(1):55–76

Storn R, Price K (1997) Differential evolution. A fast and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359

Theodoridis S, Koutroumbas K (2006) Pattern Recognit. Academic Press, Elsevier

Wiesner R, Edwards E, Freeman R, Harper A, Kim R, Kamath P, Kremers W, Lake J, Howard T, Merion R, Wolfe R, Krom R, Colombani P, Cottingham P, Dunn S, Fung J, Hanto D, McDiarmid S, Rabkin J, Teperman L, Turcotte J, Wegman L (2003) Model for end-stage liver disease (MELD) and allocation of donor livers. Gastroenterology 124(1):91–96

Witten IH, Frank E (2005) Data mining: practical machine learning tools and techniques. In: Data management systems, 2nd edn. Morgan Kaufmann (Elsevier), New York