Multi-objective metaheuristics for preprocessing EEG data in brain–computer interfaces

Ricardo Aler[a]*, Alicia Vega,[b] Inés M. Galván[a] and Antonio J. Nebro[c]

[a]*Universidad Carlos III de Madrid, Spain;* [b]*Oracle Ibérica, Spain;* [c]*University of Málaga, Spain*

In the field of brain–computer interfaces, one of the main issues is to classify the electroencephalogram (EEG) accurately. EEG signals have a good temporal resolution, but a low spatial one. In this article, metaheuristics are used to compute spatial filters to improve the spatial resolution. Additionally, from a physiological point of view, not all frequency bands are equally relevant. Both spatial filters and relevant frequency bands are user-dependent. In this article a multi-objective formulation for spatial filter optimization and frequency-band selection is proposed. Several multi-objective metaheuristics have been tested for this purpose. The experimental results show, in general, that multi-objective algorithms are able to select a subset of the available frequency bands, while maintaining or improving the accuracy obtained with the whole set. Also, among the different metaheuristics tested, GDE3, which is based on differential evolution, is the most useful algorithm in this context.

Keywords: brain–computer interface; multi-objective optimization; EEG filter optimization

1. Introduction

Brain–Computer Interfaces (BCIs) are a new modality of man–machine interaction that allows users to use thoughts to control external devices (Kubler and Müller 2007). BCIs are often aimed at assisting disabled patients (suffering for instance from amyotrophic lateral sclerosis or the locked-in syndrome), although some other applications, like robotics (Millán *et al.* 2004) or videogames (Bos *et al.* 2010a,b), are also target applications. There are many varieties of BCIs. Depending on the degree of invasiveness, BCIs can be classified as invasive, partially invasive, and non-invasive. Only non-invasive BCIs can be used without performing a surgical operation on the patient to install a set of electrodes. The most common kind of non-invasive BCIs are those based on Electroencephalogram (EEG) signals. EEG-based BCIs capture the EEG signal by means of electrodes disposed on the subject's scalp. Other kinds of signals, like Magnetoencephalography (MEG) or functional Magnetic Resonance Imaging (fMRI), can also be employed for BCIs, but they are more costly and difficult to use. This work is focused on EEG-based BCIs.

*Corresponding author. Email: aler@inf.uc3m.es

Several features of the EEG are correlated with well-known brain processes. These features can be either voluntarily generated by the user, like slow cortical potentials or Sensorimotor Rhythms (SMRs) or elicited by visual or auditory stimulation (event-related potentials or steady-state evoked potentials). The data used in this article involve mainly SMRs, so they will be explained in more detail. SMRs can be detected on the somatosensory cortex as changes in amplitude around the frequency 10 Hz in the α band, often mixed with a β (around 20 Hz) and a γ (around 40 Hz) component (Kubler and Müller 2007). Users can control SMRs by voluntarily blocking or desynchronizing them (*i.e.* the signal amplitude decreases at those frequencies). Desynchronization occurs with movement or preparation for movement, but also with motor imagery (imagination of movements). Synchronization (increase in amplitude) returns with relaxation or after the (imagination of the) movement. These processes are called event-related desynchronization (ERD) and event-related synchronization (ERS) (Pfurtscheller and da Silva 1994). Of particular interest for BCIs is the modulation of SMRs with motor imagery because it can be used with patients who cannot move any of their body parts. Just the imagination of the movement of the left or the right hand is enough to desynchronize a particular region of the somatosensory cortex. Motor imagery of different body parts desynchronize different regions of the somatosensory cortex.

However, there is a high variability of EEG patterns among different subjects: frequency bands and scalp locations where ERD/ERS is detected are user dependent. This makes machine learning classification techniques a commonly used approach (Dornhege *et al.* 2007). The process involves training user-dependent classifiers from user-generated data by means of machine learning techniques (Dornhege *et al.* 2007). The classifier can then be used to detect user patterns on real-time EEG data.

EEG signals are basically time-series that can be captured from electrodes distributed on the user's scalp. Every electrode contributes a different time-series. However, it is known that the raw signals are not the most appropriate for training classifiers. Hence, preprocessing (or filtering) becomes a very important step for classification (Hammon and de Sa 2007). Basically, two kinds of transformation are commonly used (Dornhege *et al.* 2007): spectral and spatial filters.

Spectral filters are used to select frequency bands because it is known that the relevant information is located at some particular frequency bands. Band selection can be achieved by means of high-pass, low-pass, and band-pass filters. Another alternative for frequency filtering is using the Fourier transform to convert from the time domain to the frequency domain, where the appropriate frequency bands can be selected. Finally, the wavelet transform, which retains information from both the frequency and the time domain, can also be applied for this purpose (Bostanov 2004, Xu and Song 2008).

The second kind of transformation involves spatial filters. These are typically used in order to improve the spatial resolution of EEG signals. Some instances of simple spatial filters are bipolar and Laplace filtering (Dornhege *et al.* 2007). The first finds the difference between the signals from pairs of electrodes, while the second subtracts the average of the signals from surrounding electrodes from each electrode signal. Both try to emphasize the local activity while attenuating more distant contributions. More generally, spatial filters can be represented by linear transformations (*i.e.* multiplying the original data by a matrix). Matrices that represent spatial filters can be adjusted manually, but this requires some experience and a process of trial and error. There are also approaches to compute spatial filters automatically, Common Spatial Patterns (CSPs) being one of the most successful and well established methods in the BCI community (Koles *et al.* 1990, Dornhege *et al.* 2007, Blankertz *et al.* 2008). In the literature there are also some CSP variants that combine both spatial and spectral filters (Lemm *et al.* 2005, Blankertz *et al.* 2008). The CSP approach computes data projections, represented by linear transformations, that maximize the variance of signals of one class and minimize the variance of signals of the other class. This increases the discriminability of the different classes. However, CSPs do not directly optimize the accuracy of the classifier on the data.

In this article, metaheuristics (Blum and Roli 2003) are used to compute spatial filters represented as matrices, in order to optimize the classifier accuracy directly (*i.e.* to minimize the error-rate). Metaheuristics are able to optimize functions without making strong assumptions about them and therefore are specially appropriate for noisy problems like BCI classification. The first goal of this work is to test a number of metaheuristic techniques empirically to obtain an insight into their performance in the BCI domain.

Second, as explained before, ERD/ERS phenomena can be detected in the frequency domain, but not all frequency bands are equally relevant, and, additionally, the relevant frequency bands are user-dependent. Therefore, from a physiological point of view, it is important to select the most appropriate user-dependent frequency bands. Furthermore, in the context of machine learning, it is known that using many attributes may lead to poor generalization. Previous work has shown that a metaheuristic approach is suitable for this task (Aler *et al.* 2010). In the present work, the main motivation is to explore the use of multi-objective optimization techniques, which has been shown to be successful in the feature selection task (Emmanouilidis *et al.* 2000, Oliveira *et al.* 2003, Pappa *et al.* 2004). In the literature, multi-objective techniques have already been used in the context of BCI for electrode selection (Hasan *et al.* 2010, Moubayed *et al.* 2010). In those works, two objectives are optimized: training error and number of electrodes. The aim here is also to minimize the training error, but instead of the number of electrodes, the number of frequency bands is minimized in order to supply to the classifier only the most relevant features. Additionally in this work, both the selection of features and the spatial filter are optimized simultaneously.

Based on the two goals of the article and motivated by the application of a multi-objective approach, the following techniques were compared. NSGA-II (Deb *et al.* 2002, the reference algorithm in the field), a steady-state version of it (ssNSGA-II, Durillo *et al.* 2009), the PAES evolution strategy (Knowles and Corne 2000), and the differential evolution algorithm GDE3 (Kukkonen and Lampinen 2005) have been selected. The idea is also to determine how the multi-objective formulation compares to results obtained using the whole set of frequency bands. Thus, in order to obtain results with all the bands, four single-objective algorithms have been included in this study. They have been chosen as the algorithms that can be considered as the counterpart of those of the former class, namely a generational genetic algorithm (gGA), a steady-state GA (ssGA), an evolution strategy (ES), and a differential evolution (DE) technique. Finally, CSPs have also been included because it is one of the standard methods for computing spatial filters for EEG-based BCIs.

This article is structured as follows. Section 2 describes how spatial- and frequency-selection filters can be used to preprocess raw EEG data and how they can be evolved by metaheuristics. The single- and multi-objective metaheuristics used in this article are described in Section 4, including the particular solution encoding and operators that have been employed. Section 5 describes the experimental setup. The approach is empirically tested and compared in Section 5. Finally, Section 6 summarizes what has been achieved.

2. Evolutionary approach for BCI filter optimization

In order to train a user-dependent classifier, it is necessary to record EEG data from the user during an acquisition session, then adjust the spatial filter to determine the most appropriate frequency bands and, finally, learn the classifier. This process is carried out off-line, either manually or automatically. The filters and the classifier can later be used on-line by the subject. During on-line use, the EEG signal is first preprocessed, then classified (by the classifier learned off-line), and finally used to control the computer or some other device like a wheelchair. For instance, left-hand/right-hand motor imagery could be used to turn the wheelchair to the left or to the right respectively. This on-line process is continuous: the EEG is continuously fed to the preprocessing stage and then to the classifier, whose output controls the device, and so on.

In this work, preprocessing involves applying the spatial filter, then transforming the signal to the frequency domain (where SMR features can be detected), and finally, selecting the relevant frequency bands. Both spatial and frequency-band filters are evolved to be optimal for the classifier. The preprocessing (filtering) of EEG signals is explained in Subsection 2.1, where it is assumed that a spatial filter has already been adjusted and the set of relevant frequency bands has already been determined. The mechanism to adjust the filters automatically by means of metaheuristics (single-objective and multi-objective), so that the classification of EEG signals is optimized, is detailed in Subsection 2.2.

2.1. *Preprocessing EEG signals*

The EEG signal is recorded from c electrodes (or channels) located at different places on the scalp of the user. The continuous signal captured by the electrodes is discretized with sampling frequency f. Therefore, a time series of data points (or time instants) will be generated for each of the c channels.

Then, the EEG data has to be preprocessed. In this work, the preprocessing stage consists of the following three steps.

(1) A spatial filter L is applied to the time-domain data. As has been mentioned in the introduction, spatial filters can be represented by a linear transformation of the original data (Dornhege *et al.* 2007). Therefore, this spatial filter is an arbitrary matrix, denoted by L.
(2) The Fast Fourier Transform (**FFT**) is applied to the spatially filtered data. Thus, EEG data is transformed from the time domain to the frequency domain.
(3) Finally, the most relevant frequency bands are selected. Let this frequency-band selection be denoted by B.

At each time, only a portion or chunk of the EEG is considered. For instance, a chunk can be made of the last second of the EEG. Let the nth EEG chunk be called S_n. S_n is a $t \times c$ matrix, where $t = sf$, where s is the time-length of the chunk and f is the sampling frequency (*i.e.* if $s = 1\,\text{sec}$ and $f = 512\,\text{Hz}$ then $t = 512$). The first step of preprocessing (the spatial filter) can be represented by Equation (1). The spatial filter L is a $c \times c'$ matrix. If $c' = c$, then S'_n has the same number of columns (*i.e.* the number of channels) as the original matrix S_n. If $c' < c$, then the number of channels of S'_n is reduced. In some sense, the spatial filter L transforms c channels into c' channels.

$$S'_n = S_n \cdot L. \tag{1}$$

The next preprocessing step transforms to the frequency domain by means of **FFT**. Since **FFT** is applied to a finite length signal chunk S'_n, the Short Time Fast Fourier Transform is used. This is easily done by applying Equation (2):

$$S''_n = |\mathbf{FFT}(S'_n)|, \tag{2}$$

where $|\cdot|$ denotes the modulus.[1] Rows of S''_n from 1 to $t/2$ represent the wide frequency band $[0, f/2]\,\text{Hz}$ (where f is the sampling frequency and t is the window size), with a resolution of $\delta f = (f/t)\,\text{Hz}$. The frequency bands contained in the matrix are $[0, \delta f]$, $[\delta f, 2\delta f]$, etc. Only the frequency bands within the interval $[8, 30]\,\text{Hz}$ have been considered, because it has been shown that reasonable classification of EEG signals could be achieved with the information contained within this range (Millán 2004). Therefore, only the rows from $\lceil 8/\delta f \rceil + 1$ to $\lceil 30/\delta f \rceil + 1$ will be selected.[2] In order to simplify the notation, it will be assumed that S'_n already contains only the rows related to the $[8, 30]$ Hz frequency band. Thus, the number of rows remaining in S''_n is $t' = \lceil 8/\delta f \rceil - \lceil 30/\delta f \rceil + 1$.

The third preprocessing step selects a set of frequency bands, presumably the most relevant for a particular user. This step is represented by filter B in Equation (3). B is represented by a matrix with the same dimensions as S''_n but containing only binary values (0 or 1). If component $B(i,j) = 0$ the $S''_n(i,j)$ component will be removed from the set of available frequency bands. Only those components of $S''_n(i,j)$ with $B(i,j) = 1$ will remain:

$$S'''_n = S''_n \otimes B \tag{3}$$

Equation (4) summarizes the three-step preprocessing of signal chunk S_n by filters L and B. **vec** is the vectorization of the resulting matrix, so that all the components become concatenated in a list:

$$I_n = \mathbf{vec}(|(\mathbf{FFT}(S_n \cdot L))| \otimes B). \tag{4}$$

Equation (5) displays the component-by-component decomposition of I_n in terms of the selected frequency bands, where $S''_n(i,j)$ represents frequency band $i \in \{1 \ldots t/2\}$ at channel $j \in \{1 \ldots c'\}$:

$$I_n = (S''_n(1,1), \ldots, S''_n(t/2,1), S''_n(1,2), \ldots, S''_n(t/2,2), \ldots, S''_n(1,c'), \ldots, S''_n(t/2,c')). \tag{5}$$

2.2. *Evolution of spatial- and frequency-selection filters*

One of the aims of this article is to evolve highly accurate filters using only the relevant frequency bands. This has been formulated as a multi-objective problem, where multi-objective metaheuristics are used to minimize both classification error and the number of frequency bands. In order to compare with the no-feature-selection case, single-objective metaheuristics have been used to evolve highly accurate filters but using all frequency bands.

It is worth mentioning that a single-objective approach could have been applied by combining the two objectives, error and number of features, in a single scalar value. However, given that two goals have to be minimized simultaneously, a multi-objective formulation is a more natural way to deal with the problem. This way, instead of a single solution, a set (an approximation to the Pareto front) can be obtained, thus allowing a decision about the best trade-off solution.

In order to specify the problem to be solved with metaheuristics, both the representation of candidate solutions (the chromosome) and the function to be optimized (the fitness function), must be described. With respect to the first issue (encoding of candidate solutions), as single-objective metaheuristics have been used taking into account all frequency bands, only L is needed, and it can be encoded as a real matrix. In the case of the multi-objective approach, as the number of features is to be minimized as well, the chromosome has to include also a binary string for B.

About the fitness function, in order to evaluate the quality of filters L and B encoded in a chromosome, a classifier C is built on the set of training instances generated from the EEG data recorded during the user acquisition session. One training instance I_n is generated from every EEG chunk S_n, according to the preprocessing equation (4). The resulting instance I_n is a training instance made of the selected frequency bands, as expressed in Equation (5). In other words, every preprocessed signal chunk S_n is in fact a training instance I_n. The training set used to build the classifier C is made of many I_n's (each one resulting from a signal chunk S_n).

In this article, C has been chosen to be LDA (Linear Discriminant Analysis) a well-known fast linear classifier. LDA has been chosen because it has been shown to be an appropriate classifier for the BCI domain (Dornhege *et al.* 2007). As the fitness function is evaluated for every solution across many generations, its computation must be fast, and LDA satisfies this requirement.

In order to deal with multiclass problems, the one-versus-all approach has been applied to LDA. An N_c-class classification problem is transformed into N_c binary (2-class) problems where the goal is to separate class i from the rest. Thus, a linear classifier F_i is learned for each binary problem so that $F_i(x) > 0$ if instance x belongs to class i and $F_i(x) < 0$ if x belongs to the other classes. Every F_i is a hyperplane represented by $F_i(x) = w_i x + b_i$. The one-versus-all approach works by classifying instance x according to Equation (6):

$$F(x) = \arg\max_i (w_i x + b_i). \tag{6}$$

In the case of multi-objective metaheuristics, candidate solutions are evaluated according to two goals: the training error of the LDA classifier and the number of frequency bands selected. The training error is the error of the classifier learned from instances preprocessed by filters L and B, as explained before. The number of frequency bands selected is given by the number of components set to 1 in B. Both goals are to be minimized, so that accurate and simple (*i.e.* only the physiologically relevant frequency bands) filters are evolved. For single-objective metaheuristics, only the training error is considered.

The evolution of metaheuristics is driven by the minimization of the training error. It is known that this could result in overfitting. However, several experiments were carried out where the evolution of the training and test errors were displayed (the test error was computed on a dataset independent of the training data). It was not observed that the test error increased as the training error decreased. This is probably due to the classifier being used being linear, which is known to be less prone to overtraining.

3. Metaheuristics description

In this section, the algorithms used in this article are described first. After that, details about the encoding of solutions and the operators are given.

3.1. *Metaheuristics*

As commented in the introduction, NSGA-II, proposed by Deb *et al.* (2002), is by far the most well-known and widely used multi-objective metaheuristic. It is a generational genetic algorithm based on obtaining an auxiliary population from the original one by applying typical genetic operators (selection, crossover and mutation); then, the two populations are merged and the individuals are sorted according to their rank and the best solutions are selected to create a new population. If some individuals having the same rank have to be chosen, those belonging to the more dense regions (in objective space) should be avoided in order to promote diversity; NSGA-II applies a density estimator known as *crowding distance* for that purpose. As the single-objective counterpart of NSGA-II, the used gGA follows the typical generational scheme, in which the best two solutions of the current population are copied to the new one in order to keep the elitist behaviour.

A steady-state version of NSGA-II, called ssNSGA-II, was proposed in Durillo *et al.* (2009), where it was observed that, when solving a benchmark of continuous problems, it improved the original algorithm in terms of convergence and diversity in all cases, so it has been included in this work. Implementing ssNSGA-II is trivial, because it is in fact NSGA-II but using an auxiliary population of size one. As before, a classical single-objective steady-state GA (ssGA) is used, in which each new generated individual replaces the worst one in the population.

PAES is an algorithm proposed by Knowles and Corne (2000), and it is based on a simple $(1 + 1)$ evolution estrategy (ES). To promote diversity, PAES uses an external archive of non-dominated solutions, which is also used to decide about the replacement of old solutions by the new candidate ones. An adaptive grid is used as a density estimator in the archive. PAES is probably the simplest proposed multi-objective metaheuristic, and the fact that it does not require a crossover operator (only mutation is applied) makes it attractive to be used in complex problems such as the one considered in this article. A single-objective $(1 + 1)$ ES is really simple: produce a mutated solution from the current individual and take the better of the two as the new current one. However, this strategy can be prone to get stuck in local minima, so a $(1 + 10)$ ES has been adopted.

As will be seen in Section 3.2, the basic encoding of solutions is composed of real variables, and in this context differential evolution (DE) is currently one of the most powerful techniques (Storn and Price 1997). The choice of DE-based multi-objective algorithm has been made as GDE3 (Kukkonen and Lampinen 2005). It starts with a population of random solutions, which becomes the current population. At each generation, an offspring population is created using the differential evolution operators; then, the current population for the next generation is updated using the solutions of both the offspring and the current population. Before creating the next generation, the size of the population is reduced using non-dominated sorting and a pruning technique aimed at diversity preservation, in a similar way as NSGA-II. However, GDE3 modifies the crowding distance of NSGA-II in order to solve some of its drawbacks when dealing with problems having more than two objectives. Again, a standard single-objective DE algorithm is used, in which each solution is selected in turn, yielding a new offspring after applying the DE operators, which replaces the original one if better.

All the algorithms have been implemented in jMetal (Durillo and Nebro 2011), a Java framework aimed at multi-objective optimization with metaheuristics. jMetal is free software and can be downloaded from http://jmetal.sourceforge.net. By reusing the base classes of jMetal, all the techniques share the same basic core components (solution encodings, operators, etc.), which ensures a fair comparison of the algorithms considered.

3.2. *Solution encoding*

The encoding or representation of solutions strongly depends on the problem and determines the operations (*e.g.* recombination with other solutions, local search procedures, etc.) that can be applied. Thus, selecting a specific representation has a great impact on the behaviour of metaheuristics and, hence, on the results obtained.

jMetal has a flexible and extensible scheme allowing one to define whatever solution encoding is required if some of those already provided are not enough. As commented in Subsection 2.2, a candidate solution for the single-objective formulation of the BCI problem must contain the spatial filter L, which is represented by a matrix (vector) of real numbers, which is already included in the framework. In the case of the multi-objective formulation of the BCI problem, an encoding scheme representing a vector of reals plus a binary string has been defined. This scheme implies that the multi-objective metaheuristics have to include operators to deal with simultaneous real/binary encodings. A simpler design would be to represent the binary string as an array of real variables taking values in the range $[0 \ldots 1]$, but then the variation operators (*e.g.* crossover and mutation) would be less effective. For example, a polynomial mutation in a real-coded binary variable would produce no effect if the original and mutated values are both below or above 0.5; in the case of using a binary representation, a bit-flip mutation would guarantee a change in the binary value.

Once the needed solution encodings are available, it is time to decide about the operators. As a new representation has been defined, it is necessary to implement specific operators to manipulate the individuals of that new encoding. This issue is discussed next.

3.3. *Operators*

NSGA-II, when it is used in continuous optimization, applies the simulated binary (SBX) crossover and polynomial mutation operators (Deb *et al.* 2002), so they will be the basis of all the techniques but the differential evolution algorithm. Concretely in the gGA, ssGA and ES (polynomial mutation only), they can be used directly, but in the multi-objective techniques it is necessary to define new operators to cope also with the binary part of the solutions. This also affects the GDE3 algorithm: while the single-objective DE will use the classical rand/1/bin scheme, to apply GDE3 the binary part has to be properly managed.

According to the adopted approach, three new operators have been defined, which are based on the aforementioned to be applied to the real part of the chromosome, while the binary part will be treated with binary operators. In particular, single point crossover and bit-flip mutation have been considered. As a result, the following operators have been included in jMetal.

- SBX + single point crossover: given two solutions, SBX and single point crossovers are used in their real and binary components, respectively, leading to two new individuals. Each crossover is applied according to a given probability.
- Polynomial + bit-flip mutation: similarly to the crossover, this operator takes a solution which is modified by using polynomial and bit-flip mutations in the real and binary parts, respectively. Each mutation is applied with a probability $1.0/l$, where l is the number of real variables in the case of polynomial mutation, and the number of bits of the binary part in the case of bit-flip.
- Differential evolution + bit-flip mutation: in DE, three solutions are necessary to produce a new one (both GDE3 and DE apply the rand/1/bin scheme, as indicated in Table 1). In the proposed operator, this is done in the same way by considering the real part of the solutions; however, DE cannot be used with a binary representation, so the adopted approach is that every bit of the new individual is selected randomly from the three parents and, after that, the result is mutated with bit-flip.

Table 1. Parameter settings of the algorithms used. Number of real variables: 64; number of binary variables: 22. The termination condition is to compute 7000 function evaluations.

Algorithm(s)	Parameter	Setting
		skip
NSGA-II and ssNSGA-II	Population size	30 individuals
	Selection of parents	Binary tournament
	Crossover	SBX ($p_c^{\text{real}} = 0.9$, $\eta_c = 20$) + Single point ($p_c^{\text{bin}} = 0.9$)
	Mutation	Polynomial ($p_m^{\text{real}} = 1/64$) + Bit-flip ($p_m^{\text{bin}} = 1/22$)
PAES	Archive size	30
	Bisections	4
	Mutation	Polynomial ($p_m^{\text{real}} = 1/64$) + Bit-flip ($p_m^{\text{bin}} = 1/22$)
GDE3	Population size	30 individuals
	Selection	Three randomly chosen parents
	DE/mutation	rand/1/bin ($CR = 0.5$, $F = 0.5$) + Bit-flip ($p_m^{\text{bin}} = 1.0/22.0$)
gGA and ssGA	Population size	30 individuals
	Selection of parents	Binary tournament
	Crossover	SBX ($p_c^{\text{real}} = 0.9$, $\eta_c = 20$)
	Mutation	Polynomial ($p_m^{\text{real}} = 1/64$)
ES	$\mu + \lambda$	$1 + 10$
	Mutation	Polynomial ($p_m^{\text{real}} = 1/64$)
DE	Population size	30 individuals
	Selection	Three randomly chosen parents
	Mutation/crossover	Differential evolution, $CR = 0.5$, $F = 0.5$
	DE variant	rand/1/bin

4. Experimental setup

The aim of this section is to provide details about the context of experiments. Subsection 4.1 describes the datasets used to compute the fitness and evaluate the generality of the filters evolved. Subsections 4.2 and 4.3 give details about general parameters of the system and the parameter setting for the metaheuristics, respectively.

4.1. *Data sets description*

In this article, the data will be used that was acquired by the IDIAP Research Institute from three subjects (Subject 1, Subject 2 and Subject 3) (Millán 2004). These datasets were also used as dataset V in the BCI III competition that took place in 2005 (Blankertz *et al.* 2006). For each subject four non-feedback sessions were carried out. Therefore, for each subject there are four datasets: session 1, 2, 3 and 4. In each session, 32 electrodes were located on the subjects's scalp. The subject was instructed to perform three mental tasks. From a classification point of view, this entails the following three-class classification problem.

- Class 1: imagination of repetitive self-paced left hand movements.
- Class 2: imagination of repetitive self-paced right hand movements.
- Class 3: generation of words beginning with the same random letter.

All four sessions of a given subject were acquired on the same day, each lasting 4 minutes with 5–10 minutes breaks in between them. The subject performed a given task for about 15 sec and then switched randomly to another task at the operator's request. The EEG data is not split in the trials since the subjects are continuously performing any one of the mental tasks.

In the present work, and following the BCI III competition rules (Blankertz *et al.* 2006), the first three sessions have been used for training and the fourth one for testing. This means that filters have been evolved using the training data (Sessions 1, 2 and 3) and their generality has been tested on the last session (Session 4).

4.2. *General parameters*

As explained before, there are four sessions for each one of the three subjects. The three first sessions were used for training, while the last one was used for testing. The parameters of the system used have been adjusted as recommended by the suppliers of the data (Blankertz *et al.* 2006). These are as follows.

- Sampling frequency $f = 512\,\text{Hz}$.
- 1 sec of data is used to construct every training instance, therefore the size of temporal windows is $t = 512$ (*i.e.* the signal chunk length).
- Training instances I_n are sampled 16 times per second (this means that $512 \div 16 = 32$ signal chunks are processed per second).
- The number of electrodes is $c = 32$.

Frequency bands outside the $[8, 30]$ Hz range have been removed because they contain no relevant information for this work, and this is how this dataset was intended to be used (Blankertz *et al.* 2006). Also, the frequency-band width considered by the frequency-selection filter B has been set to 2 Hz. This means that the frequency bands are $[8, 10]$ Hz, $[10, 12]$ Hz, …, $[28, 30]$ Hz. Therefore, B is made of 11 binary values for each channel. The spatial filter L is a $(c \times c')$ matrix: it transforms c channels into c' channels. It is known that imagination of left (right) hand movements is related to a certain part of the right (left) hemisphere (Dornhege *et al.* 2007). If there were only these two classes, it would be reasonable to set $c' = 2$. In this problem there is a third class (imagination of random words), therefore $c' = 3$ was also tested, finding similar results. Hence, c' has been set to 2 for the rest of the experiments. Therefore, given that there are 11 frequency bands per channel and that the spatial filter reduces to 2 channels ($c' = 2$), B is a binary vector with 22 components.

4.3. *Parameterization of metaheuristics*

Evaluating a solution of the BCI problem is time consuming, so instead of trying to find the best parameter settings, the adopted approach is to consider the typical parameterization used when solving benchmark problems (see, *e.g.*, Durillo *et al.* 2009). However, with regard to the number of evaluations, preliminary experiments showed that the typical value (25,000) was too large because it is observed that the fitness function became stable by about 7000 evaluations. This number of evaluations with population sizes of 100 could lead to poor convergence, so after some experiments a value of 30 was chosen in all the techniques but (1+10) ES.

A summary of the parameter settings of the algorithms is included in Table 1. In the table, 64 is the number of real variables in the chromosome (this is the spatial filter made by $32 * 2$ components), and 22 is the number of binary variables of B. The distribution indices for both SBX crossover and polynomial mutation operators are $\eta_c = 20$ and $\eta_m = 20$, respectively.

Thirty independent runs were made per configuration in order to obtain statistically significant conclusions. In the case of single-objective, each run returns the solution with the best fitness. In the case of multi-objective, each run returns a front of several solutions, from where one solution is selected according to the criterion that will be described in Subsection 5.2. The Wilcoxon rank sum test will be used to determine whether differences are statistically significant. Therefore, the Wilcoxon test is used to compare pairs of metaheuristics by taking into account two sets of 30 solutions each.

5. Experimental results and discussion

In this section, the evaluation of the classifiers with the test data (session 4) is presented. A discussion of results is also included. Subsection 5.1 presents the single-objective metaheuristic results, Subsection 5.2 discusses the results for the multi-objective algorithms, and Subsection 5.3 includes a comparative analysis of different results. In that final subsection, the results yielded are also compared with those obtained with CSPs, one of the most commonly used methods for computing spatial filters for BCIs.

5.1. *Single-objective metaheuristics*

Although the main interest was to study the benefits of using a multi-objective approach for frequency-band selection, for the sake of simplifying the presentation of the results, first the single-objective techniques were used to evolve spatial filters using all frequency bands. In turn, the filters were used to transform the data and build an LDA classifier. Each algorithm was run 30 times, starting with different random seeds and obtaining 30 different spatial filters. It is important to remark that those filters were evolved by optimizing the fitness function that was in turn based on the training data (sessions 1, 2 and 3). In order to determine if the filters generalize well on unseen data, they were tested on session 4. Table 2 displays the median test results for each of the three subjects and each of the four single-objective metaheuristics. Both the median accuracy (of LDA) and the interquartile range (Q3−Q1) are displayed. Single-objective metaheuristics use the whole set of 22 frequency bands.

Table 2. Median test error and interquartile range (with session 4) for the single-objective metaheuristics. The whole set of 22 frequency bands is used.

	Algorithm	Median [%] (Q3−Q1)
Subject 1	DE	**26.63 (1.31)**
	ES	31.17 (8.18)
	gGA	27.97 (3.12)
	ssGA	28.21 (2.58)
Subject 2	DE	**37.58 (4.13)**
	ES	47.71 (5.76)
	gGA	37.62 (3.90)
	ssGA	39.29 (3.61)
Subject 3	DE	**38.62 (2.23)**
	ES	46.93 (3.40)
	gGA	41.04 (4.12)
	ssGA	44.38 (4.51)

Table 3. Summary of the Wilcoxon rank sum test applied to the results of the single-objective algorithms. In the cells, each of the three symbols represents Subjects 1, 2 and 3, respectively. ▲ means the row results are significantly better than the column results, ▽ means the row results are significantly worse than the column results, and – means there is no significant difference.

	ES	gGA	ssGA
DE	▲ ▲ ▲	▲ – ▲	▲ ▲ ▲
ES		▽ ▽ ▽	▽ ▽ ▽
gGA			– ▲ ▲

Table 4. p-values of the results of the single-objective algorithms. In the cells, each of the three numbers represents Subjects 1, 2 and 3, respectively. Values with $p-\text{value} < 0.05$ indicate that the differences are statistically significant.

	ES	gGA	ssGA
DE	3.0e−03 1.8e−10 2.6e−10	1.6e−02 5.2e−01 1.2e−04	1.1e−02 1.7e−02 1.1e−05
ES		5.4e−03 1.3e−10 1.4e−06	9.9e−03 5.8e−08 4.6e−03
gGA			9.5e−01 6.7e−03 3.9e−02

From Table 2 it is clear that DE is systematically the most accurate algorithm. Also, in all cases, gGA is the second best, ssGA the third, and ES is always the last one. Table 3 shows the results of the Wilcoxon test for all subjects (the p-values are included in Table 4). All differences are significant, except in the case of gGA-ssGA for Subject 1 and DE-gGA for Subject 2.

5.2. *Multiple-objective metaheuristics*

In this case, the multi-objective algorithms were used to evolve both spatial filters and feature selection masks. Before analysing the results on the test data (session 4), the quality of the Pareto front approximations evolved by the multi-objective techniques will be compared. For that purpose, the hypervolume (Zitzler and Thiele 1999) quality indicator is adopted. The hypervolume provides a measure comprising both convergence and diversity in a single value, and it is considered as a very reliable indicator in the multi-objective research community.

The hypervolume computes the volume (in objective function space) covered by members of a non-dominated set of solutions Q for problems in which all objectives are to be minimized. Mathematically, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. The reference point can simply be found by constructing a vector with the worst objective function values. Thereafter, the union of all hypercubes is found and its hypervolume (HV) is calculated:

$$\text{HV} = \text{volume}\left(\bigcup_{i=1}^{|Q|} v_i\right). \tag{7}$$

Higher values of the HV performance measure imply more desirable solutions. A property of this quality indicator is that it measures both convergence to the Pareto front and diversity of the obtained fronts.

To apply the hypervolume, the optimal Pareto front or an approximation to it is needed. In the case of the BCI problem, no optimal front is available, so the hypervolume is calculated by using a reference front, which has been computed by calculating the non-dominated set from all the fronts obtained by all the algorithms.

Figure 1 displays the hypervolume boxplots for the three subjects. It can be observed that GDE3 has the highest median hypervolume (especially for Subjects 1 and 3) and the smallest dispersion (also especially for Subjects 1 and 3). According to the Wilcoxon test, differences between GDE3 and the rest of algorithms with regard to hypervolume are always statistically significant for Subjects 1 and 3. For Subject 2, the difference GDE3−NSGAII is not statistically significant. Based on these results, GDE3 gives the best quality fronts with respect to the two goals (1) training-error and (2) number of features. GDE3, therefore, is appropriately optimizing the multi-objective fitness function based on the training data (sessions 1, 2 and 3), at least in comparison with the other algorithms. Now, an analysis about whether the filters evolved by the multi-objective metaheuristics generalize well on the testing data is carried out.

With respect to the results on the test data (session 4 dataset), contrary to the single-objective techniques, the end result of a run is not a single solution but a front of non-dominated solutions, from which one solution has to be selected. The front represents the trade-off between the number of features and the training error. The aim of the approach is both to obtain high accuracy spatial filters and to select the user-dependent physiologically relevant frequency bands. Therefore, a solution from the front has to be selected that is an appropriate trade-off between training error and number of features. In this article, feature reduction is performed, by not increasing the number of features when the gain in training accuracy is too small (according to some threshold). The criterion for selecting the frequency bands is based on the gain in error per feature. This criterion is motivated by considering that adding new features to the solution may not be worthwhile if the gain in training accuracy is too small. According to this criterion, the first solution i in the set of solutions yielded by an algorithm that fulfils Equation (8) is selected:

$$\frac{E_B - E_i}{A_i - A_B} \leq T, \tag{8}$$

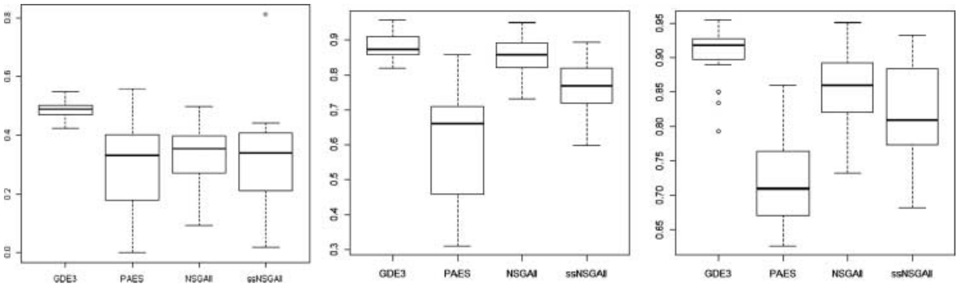

Figure 1. Box plots for the hypervolume of the multi-objective metaheuristics for Subjects 1 (left), 2 (centre) and 3 (right).

Table 5. Median test error and interquartile range (with session 4) for the multi-objective metaheuristics with the gain per feature criterion $(E_B - E_i)(A_i - A_B)^{-1} \leq 1$.

	Algorithm	Median error [%] (Q3−Q1)	No. features
Subject 1	GDE3	**26.66 (1.76)**	7.5
	PAES	27.68 (1.99)	8.5
	NSGAII	27.32 (5.50)	10
	ssNSGAII	28.95 (3.38)	9.9
Subject 2	GDE3	**33.26 (1.98)**	8.2
	PAES	36.83 (4.35)	8.3
	NSGAII	38.40 (9.73)	9.3
	ssNSGAII	38.18 (3.14)	8.9
Subject 3	GDE3	**46.42 (7.02)**	7.7
	PAES	46.83 (4.70)	5.9
	NSGAII	46.59 (4.24)	7.3
	ssNSGAII	48.06 (5.01)	6.9

where E_B is the error of the solution with maximum error in the obtained front. A_B is the number of features of that solution. E_i is the training error of the ith solution in the Pareto front (sorted by training error in descending order). Finally, A_i is the number of features of the ith solution in the Pareto front. This criterion means that a solution is selected when the error reduction achieved by adding a new feature is smaller than a given threshold. The threshold T has been set to 1% for the three subjects. This considers that gains in error reduction per feature smaller than 1% are not considered relevant.

The gain per feature was used to select the best solutions from the fronts of solutions. Table 5 displays the multi-objective results for this criterion. As 30 runs were executed for every metaheuristic, the median test error and the interquartile range (Q3−Q1) are computed from the 30 solutions selected from each of the 30 fronts. The median number of features selected is also displayed.

According to Table 5, the best performer is always GDE3. This is the algorithm related to DE, the best single-objective technique in the conducted study (see Subsection 5.1). The second, third and last algorithms are NSGAII, ssNSGAII and PAES, respectively, for Subjects 1 and 2. They all perform similarly for Subject 3. According to Tables 6 and 7, where statistical significance differences are displayed (Wilcoxon test and p-values, respectively), GDE3 is significantly better than the rest of algorithms for Subjects 1 and 2. For Subject 3, no algorithm outperforms the others significantly.

The use of multiobjective metaheuristics with the selection criterion manages to reduce the number of frequency bands used by the classifier with respect to the total set of frequency bands. This occurs for all the subjects and metaheuristics. In particular, the best algorithm (GDE3) reduces the number of frequency bands by 66, 63 and 65% for Subjects 1, 2 and 3, respectively.

Table 6. Summary of the Wilcoxon rank sum test applied to the results of the multiobject-objective algorithms. In the cells, each of the three symbols represents Subjects 1, 2 and 3, respectively. ▲ means the row results are significantly better than the column results, ▽ means row results are significantly worse than the column results, and – means there is no significant difference.

	NSGAII	PAES	ssNSGAII
GDE3	▲ ▲ –	▲ ▲ –	▲ ▲ –
NSGAII		– ▲ –	– – –
PAES			– – –

Table 7. p-values of the results of the multi-objective algorithms. In the cells, each of the three numbers represents Subjects 1, 2 and 3, respectively. Values with p-value < 0.05 indicate that the differences are statistically significant.

	NSGAII	PAES	ssNSGAII
GDE3	2.5e−03 4.8e−07 7.3e−01	2.4e−02 9.3e−10 7.4e−01	6.6e−05 2.1e−07 5.7e−01
NSGAII		7.1e−01 1.4e−02 8.5e−01	5.0e−02 6.1e−01 7.8e−01
PAES			5.4e−01 5.4e−02 5.7e−01

Figure 2 displays the shapes of representative fronts returned by GDE3 for each subject. The solid line represents the front obtained after the evolution of GDE3. Every point of the solid line represents a model. The X-axis is the number of features (number of frequency bands) of the model and the Y-axis its training error. The circles are the test errors for every model, and the big circle encloses the test error of the model selected by the criterion previously defined. It can be observed that solutions with very few features have a large training error. This error decreases as the number of features is increased. However, when the model includes many features, the decrease in error becomes almost stable. It is in the stable region where the criterion selects one of the models.

It can also be seen in Figure 2 that, for Subjects 2 and 3, the test errors are very different from the training errors. This is due to the known fact that some BCI users do not always show consistency across sessions. This is the case with Subject 2 and specially Subject 3, as had already pointed out by other researchers using this same dataset (Galán *et al.* 2007).

5.3. *Comparative analysis*

In this subsection, the feature selection approach formulated as a multi-objective problem will be compared with no feature selection (implemented with single-objective metaheuristics that use all frequency bands). First, from the experimental results shown in the previous subsections, it can be concluded that the best single-objective and multi-objective metaheuristics are, respectively, DE and GDE3. They perform systematically better for all subject datasets. Therefore, the following comparison will be restricted to DE and GDE3.

By comparing DE and GDE3 in Tables 2 and 5, it can be seen that, using frequency-band selection, there is a large test-error improvement for Subject 2 (from 37.58 to 33.26%) and it is maintained for Subject 1 (26.63 versus 26.66%). For Subject 3, the multi-objective formulation does not improve the test-error returned by using the whole set of features. For Subject 3, it can be observed that all frequency bands are required to classify the data accurately. The 30 fronts returned by GDE3 have been analysed, and it is observed that only in a few runs are solutions with 22 features returned. This is the reason the interquartile range is larger in the case of Subject 3 compared with Subjects 1 and 2 (Q3−Q1 is 7.02, compared with 1.76 and 1.98%). According to a Wilcoxon test, the differences are statistically significant for Subjects 2 and 3.

A solution with all 22 features is one of the extremes of the Pareto front. It is difficult for the multi-objective metaheuristics to find them because solutions with many features are harder to optimize: in order for solutions with many features to appear in the Pareto front, their training error must be smaller than the solutions with fewer features, but obtaining very small training errors is a difficult optimization problem, specially if there are many features. Therefore, it is not easy for multi-objective techniques always to include these solutions in the Pareto front.

In order to determine the significance of results returned by metaheuristics in the BCI context, they are compared with CSPs, one of the established methods for computing spatial filters in the BCI field. In order to do so, the results have been compared to those obtained by previous researchers in the BCI III competition, who used CSPs (Blankertz *et al.* 2005).

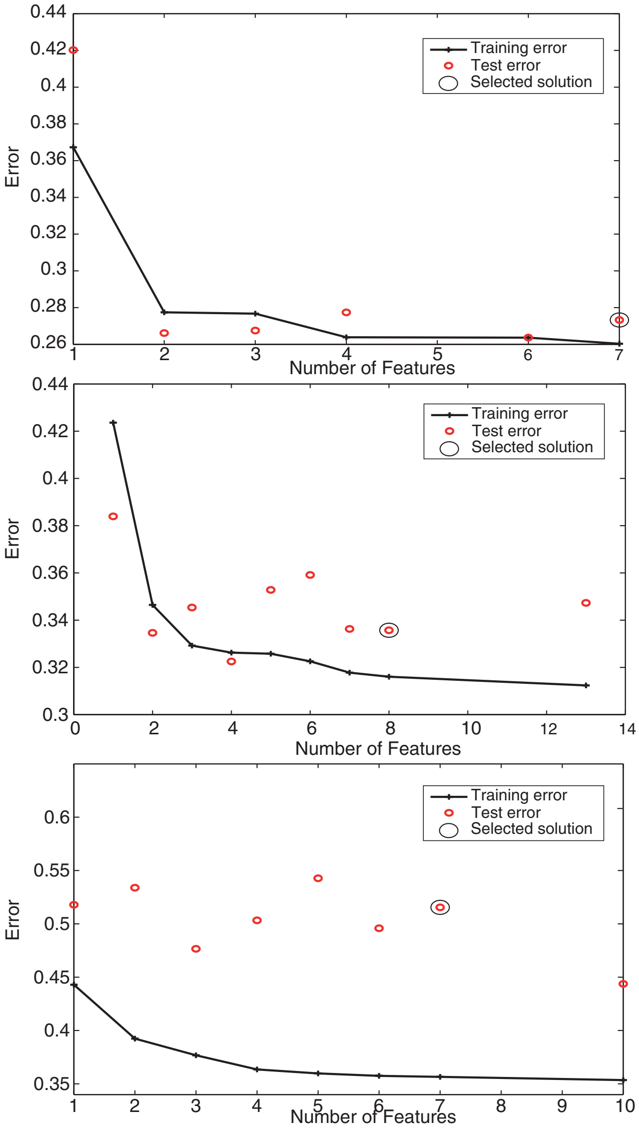

Figure 2. Pareto front approximations returned by GDE3 for Subjects 1, 2 and 3.

However, the results shown in the previous tables (Tables 2 and 5) cannot be directly compared with the competition results. One of the requirements of the competition was to get a response every 0.5 sec. Since input vectors were computed 16 times per second (*i.e.* eight times every half a second), obtaining a response every 0.5 sec implies averaging eight consecutive samples (Blankertz *et al.* 2006). This smoothing process tends to improve results because averaging cancels randomness out. This new way has been used to compute the error of DE and GDE3, so that the results can be compared with those of the competition. The comparison is shown in Table 8. The first two rows of Table 8 show the (median) testing errors for DE and GDE3, respectively. The third row displays the error obtained by the best participant in the competition that used the raw data directly (Sun *et al.* 2005). The results have been compared with theirs because they worked with the raw original data (as in the present work) and they applied CSP. The last row

Table 8. DE, GDE3 and competition results.

	Subject 1	Subject 2	Subject 3
DE	25.31	35.43	40.23
GDE3	24.89	32.90	41.8
BCI III competition with CSP on raw data	25.69	37.68	48.01
BCI III competition winner with hand preprocessed data	20.40	29.69	43.98

of Table 8 shows the winner of the competition (Blankertz *et al.* 2005, Galán *et al.* 2007). The winner's results have been included although they cannot be fairly compared with the results reported in this article. The reason is that the winner of the competition did not use the raw data, but a transformed dataset preprocessed manually that was provided by the competition organizer. This preprocessing applies a manually tuned surface Laplacian filter on a selected subset of eight channels (out of the 32), using a priori neurological knowledge.

Comparing the results of spatial filters evolved by metaheuristics (DE and GDE3) with CSP, it can be seen that, in general, the error is lower. For Subject 1, the errors are similar (the difference is smaller than 1%). For Subjects 2 and 3, DE and GDE3 provide better results than CSP: for Subject 2 the improvement is around 4% and for Subject 3 around 6%. Probably, this difference is due to the fact that the approach presented in this article directly optimizes the error, while CSPs optimize the variance of one of the classes (and minimize the variance of the other classes).

6. Conclusions

In the field of Brain–Computer Interfaces (BCIs) one of the most important issues is to classify the EEG signal accurately. The success in classification depends largely on the preprocessing or filtering of the EEG time series. Two kinds of filters are important in this context: spatial filters and the selection of the relevant frequency bands. Spatial filters are able to improve classification rates by locating the sources of signals in the brain more precisely. However, classifiers and spatial filters are subject dependent and it is important that they are appropriately tuned for each different subject. It is also known that the important information is located on several frequency bands, but not all of them are equally relevant. The relevant frequency bands are also subject dependent.

The main aim of the article is to optimize spatial- and frequency-band selection filters by means of metaheuristics. More specifically, it is intended to evolve filters that achieve a small classification error with the minimum required frequency bands. Given that two goals have to be optimized, a multi-objective formulation has been proposed and several multi-objective metaheuristics have been tested. In order to compare with results with all frequency bands, single-objective metaheuristics have been used to optimize the accuracy of the spatial filter only (with no frequency-band selection). In both the multi- and single-objective cases, several metaheuristics were tested, since it was also of interest to determine whether some of them performed systematically better on the datasets used, or whether they were also subject dependent. Experiments have been carried out on three datasets with raw EEG signals obtained from three different subjects.

With respect to the single-objective results, four algorithms were tested: DE, ES, gGA and ssGA. DE has obtained statistically significant higher classification accuracies than the rest of algorithms by about 1% better for the first two subjects and 2.4% better for Subject 3. Therefore, it is advisable to use DE in preference to the other metaheuristics tested, independently of the subject. Additionally, DE results are significant in the BCI context, since it improves on previous results obtained by commonly used CSPs on the same datasets. Results are similar for Subject 1, but around 2% better for Subject 2 and 8% better for Subject 3.

With regard to the multi-objective approach for the optimization of spatial- and frequency-band selection filters, four multi-objective generalizations of the four single-objective metaheuristis have been taken into consideration: GDE3, NSGAII, PAES and ssNSGAII. In order to select a solution from the Pareto front approximations returned by these metaheuristics, a selection criterion has been proposed based on the trade-off between accuracy and number of features. Empirical results show that GDE3, the multi-objective version of DE, also gives the best classification accuracy for all subjects (similar for Subjects 1 and 3, but about 3% better for Subject 2). Differences are statistically significant for Subjects 1 and 2, therefore it is also advisable to use GDE3 in preference to the others. Similarly to DE, GDE3 also obtains better accuracies than the standard CSP approach (similar for Subject 1, 4% better for Subject 2, and 6% better for Subject 3).

But the aim of the multi-objective approach was also to reduce the number of frequency bands used, in addition to maximizing accuracy. In this regard, GDE3 is able to select a small subset of about 8 out of 22 appropriate frequency bands for Subjects 1 and 2, while at the same time maintaining and improving by 1% (respectively) the classification test rates for both subjects. For Subject 3, where all frequency bands seem to be relevant, GDE3 does not provide the most accurate solution. In cases where the best solution is one of the extremes of the Pareto front, it is advisable to use the solution provided by the single-objective techniques, which include all frequency bands.

Acknowledgement

This work has been funded by the Spanish Ministry of Science under contract TIN2008-06491-C04-03 (MSTAR project).

Notes

1. **FFT** returns complex numbers, with phase and modulus. SMR phenomena can be easily detected by means of the amplitude, so the phase will be ignored in this article (Dornhege *et al.* 2007).
2. $\lceil x \rceil$ rounds x to the nearest integer greater than or equal to x.

References

Aler, R., Galván, I., and Valls, J., 2010. Evolving spatial and frequency selection filters for Brain–Computer Interfaces. *In*: *IEEE congress on evolutionary computation*, 18–20 July, Barcelona. Piscataway, NJ: IEEE Press, 1–7.

Blankertz, B., *et al.*, 2005. *Results of the BCI Competition III*. Available from: http://www.bbci.de/competition/iii/results/bci_competition_iii_results_list.pdf [Accessed 1 May 2010].

Blankertz, B., *et al.*, 2006. The BCI competition III: validating alternative approaches to actual BCI problems. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14 (2), 153–159.

Blankertz, B., *et al.*, 2008. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, 25 (1), 41–56.

Blum, C. and Roli, A., 2003. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Computing Surveys*, 35 (3), 268–308.

Bos, D., *et al.*, 2010a. Brain-computer interfacing and games. *In*: Desney S. Tan and Anton Nijholt, eds. *Brain-Computer Interfaces*. New York: Springer, 149–178.

Bos, D., *et al.*, 2010b. Human–computer interaction for BCI games: usability and user experience. *In*: *International conference on cyberworlds (CW)*, 20–22 October, Singapore, 277–281.

Bostanov, V., 2004. BCI competition 2003 – data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the *t*-value scalogram. *IEEE Transactions on Biomedical Engineering*, 51 (6), 1057–1061.

Deb, K., *et al.*, 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2), 182–197.

Dornhege, G., *et al.*, 2007. General signal processing and machine learning tools for BCI analysis. *In*: *Towards brain–computer interfacing*, Cambridge, MA: MIT Press, 207–234.

Durillo, J.J. and Nebro, A.J., 2011. jMetal: a Java framework for multi-objective optimization. *Advances in Engineering Software*, 42 (10), 760–771.

Durillo, J., *et al.*, 2009. On the effect of the steady-state selection scheme in multi-objective genetic algorithms. *In*: *5th international conference (EMO 2009)*, 7–10 April, Nantes, France. Lecture notes in computer science Vol. 5467. Berlin: Springer-Verlag, 183–197.

Emmanouilidis, C., Hunter, A., and MacIntyre, J., 2000. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. *In*: *Proceedings of the congress on evolutionary computation*, 16–19 July, La Jolla, CA, USA. Piscataway, NJ: IEEE Press.

Galán, F., Oliva, F., and Guardia, J., 2007. Using mental tasks transitions detection to improve spontaneous mental activity classification. *Medical and Biological Engineering and Computing*, 45 (6), 603–609.

Hammon, P.S. and de Sa, V.R., 2007. Preprocessing and meta-classification for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 54 (3), 518–525.

Hasan, B.A.S., Gan, J.Q., and Zhang, Q., 2010. Multi-objective evolutionary methods for channel selection in brain–computer interfaces: some preliminary experimental results. *In*: *IEEE congress on evolutionary computation*, 18–23 July, Barcelona. Piscataway, NJ: IEEE Press, 1–6.

Knowles, J.D. and Corne, D.W., 2000. Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8 (2), 149–172.

Koles, Z.J., Lazar, M.S., and Zhou, S.Z., 1990. Spatial patterns underlying population differences in the background EEG. *Brain Topography*, 2 (4), 275–284.

Kubler, A. and Müller, K.R., 2007. An introduction to brain–computer interfacing. *In*: *Toward brain–computer interfacing*, Cambridge, MA: MIT Press, 1–26.

Kukkonen, S. and Lampinen, J., 2005. GDE3: the third evolution step of generalized differential evolution. *In*: *IEEE congress on evolutionary computation (CEC'2005)*, 2–4 September, Edinburgh, UK. Piscataway, NJ: IEEE Press, 443–450.

Lemm, S., *et al.*, 2005. Spatio-spectral filters for improved classification of single trial EEG. *IEEE Transactions on Biomedical Engineering*, 52 (9), 1541–1548.

Millán, J., 2004. On the need for on-line learning in brain–computer interfaces. *In*: *Proceedings of the international joint conference on neural networks*, 25–29 July, Budapest, Hungary. IDIAP-RR 03-30.

Millán, J., *et al.*, 2004. Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, 51 (6), 1026–1033.

Moubayed, N.A., *et al.*, 2010. Binary-SDMOPSO and its application in channel selection for brain–computer interfaces. *In*: *10th annual workshop on computational intelligence (UKCI 2010)*, 8–10 September, Colchester, UK. Piscataway, NJ: IEEE Press.

Oliveira, L.S., *et al.*, 2003. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17 (6), 903–929.

Pappa, G.L., Freitas, A.A., and Kaestner, C.A.A., 2004. Multi-objective algorithms for attribute selection in data mining. *In*: C.A.C. Coello and G. Lamont, eds. *Applications of multi-objective evolutionary algorithms*. Singapore: World Scientific, 603–626.

Pfurtscheller, G. and da Silva, F.H.L., 1994. Event-related synchronization of mu rhythm in the EEG over the cortical hand area in man. *NeuroScience Letters*, 174 (1), 93–96.

Storn, R. and Price, K., 1997. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.

Sun, S., Zhang, C., and Pan, J., 2005. *Algorithm submitted to the BCI Competition III*. Available from: http://www.bbci.de/competition/iii/results/martigny/ShiliangSun_desc.txt [Accessed 1 February 2010].

Xu, B.G. and Song, A.G., 2008. Pattern recognition of motor imagery EEG using wavelet transform. *Journal of Biomedical Science and Engineering*, 1 (1), 64–67.

Zitzler, E. and Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3 (4), 257–271.