Nature-Inspired Algorithms for Selecting EEG Sources for Motor Imagery Based BCI

Sebastián Basterrech[1](✉), Pavel Bobrov[1,2], Alexander Frolov[2], and Dušan Húsek[3]

[1] VŠB–Technical University of Ostrava
Ostrava, Czech Republic
Sebastian.Basterrech.Tiscordio@vsb.cz

[2] Institute of Higher Nervous Activity and Neurophysiologgy,
RAS, Moscow, Russia
P-bobrov@yandex.ru, Aafrolov@mail.ru

[3] Institute of Computer Science, Academy of Sciences of the Czech Republic
Prague, Czech Republic
Dusan@cs.cas.cz

Abstract. In this article we examine the performance of two well-known metaheuristic techniques (Genetic Algorithm and Simulating Annealing) for selecting the input features of a classifier in a BCI system. An important problem of the EEG-based BCI system consists in designing the EEG pattern classifier. The selection of the EEG channels used for building that learning predictor has impact in the classifier performance. We present results of both metaheuristic techniques on real data set when the classifier is a Bayesian predictor. We statistically compare that performances with a random selection of the EEG channels. According our empirical results our approach significantly increases the accuracy of the learning predictor.

Keywords: Brain computer interface · EEG pattern selection · Bayesian classifier · Genetic algorithms · Simulating annealing

1 Introduction

A *Brain Computer Interface (BCI)* is a functional interaction between the brain and an external device. It can be useful means for assisting and repairing human cognitive and sensory-motor functions. A BCI basically consists of three components: a brain signal acquisition system, an information processing device, and an external device. In respect of first component, there are several kinds of signals that have been used for BCI. The most widespread signal is the *Electroencephalography (EEG)* that presents good advantages in respect to the other ones, such as: good temporal resolution, portability, and low set-up cost. The second BCI provides a parametric mapping between the brain signals and the mental states. This tool is used for discriminating EEG patterns related to different mental states and includes supervised learning methods. The third component is an external device committed to receive commands from the classifier

L. Rutkowski et al. (Eds.): ICAISC 2015, Part II, LNAI 9120, pp. 79–90, 2015.
DOI: 10.1007/978-3-319-19369-4_8

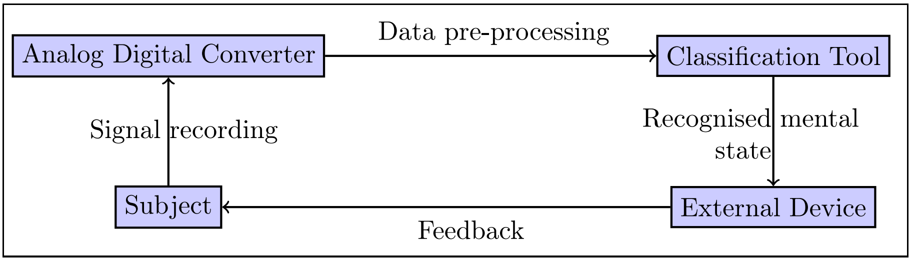

Fig. 1. General diagram of a signal based BCI system

and to provide feedbacks to the subject. Figure 1 is depicted a BCI architecture including information flows.

In the EEG-based BCI experiment the signals are recorded using N electrodes. Then, we dispose of a high-dimensional time series data, which is most often affected by sources of noise. In order to improve the classification efficiency in the BCI, a pre-training step is performed. This consists among others in selecting a subset of the EEG signals from the all-channel time series data. The problem of finding the best configuration of the EEG signals via *brute-force search* has factorial algorithmic time. Instead, we propose an alternative approach to find a *good* configuration of EGG channels as relevant sources for BCI.

The goal of this article is to study the efficiency of two well-known nature-inspired metaheuristic techniques, *Genetic Algorithm (GA)* and *Simulating Annealing (SA)*, for selecting the input information of the EEG pattern classifier. The SA is a probabilistic metaheuristic for solving optimisation problems, that was inspired from the annealing process in thermodynamics. The original motivation of GA was to simulate the natural selection process. In this article, we consider a *Bayesian Predictor* as the EEG pattern classifier, due to well performances of this learning tool in our previous works [1]. Therefore, the learning process has two phases. The first one consists of performing the metaheuristic techniques for selecting the EEG channels. The second one consists in training a Bayesian classifier in order to generate a mapping between a set of EEG signals and mental tasks. The signals that are not selected in the first phase by the nature-inspired algorithms are omitted in the second learning phase. We compute the accuracy of that technique using the κ function [2,3].

A previous study of EEG feature selection using metaheuristic was presented in [4]. In this work, the authors use *Support Vector Machine (SVM)* as classification technique over a specific data set, and the EEG feature selection was done applying GA. In [5], the authors use GA and SVM to search the features on a EEG-based BCI. The main differences of our work with these articles are: we use our own BCI experimental data, we study the performance of two feature selection tools (GA and SA), and we use our own Bayesian learning classifier [6].

The article is organised as follows. Section 2 presents the experimental procedure used for collecting the data set. This section also contains a description of a Bayesian classifier, and it presents the criteria used for measuring its performance assesment. Section 3 introduces two nature-inspired techniques

(SA and GA) employed as feature selection. Besides, we present an algorithm that shows how to use the SA and GA in the BCI context. The experimental results are presented in Section 4. Next, we go for final conclusions and future work.

2 Methodology

In this Section, we specify the methodology and the protocol used during the experiments. In addition, a background about the Bayesian Classifier is introduced in subsection 2.2, and the definition of the performance assessment is presented in subsection 2.3.

2.1 Experimental Procedure

In this Section, we describe in brief the protocol used for collecting the data set. More details about the experiments can be seen in [2,7]. The data was collected during the experimental sessions with 5 right-handed and healthy subjects aged from 25 to 50. The subjects performed instructions displayed on a screen. There are four instructions: to *relax*, to imagine the movement of the *right hand*, to imagine the movement of the *left hand*, and to imagine the movement of the *feet*. The movement that they were asked to imagine was a handgrip or feet pressure. An experiment for each subject consists of training and testing sessions. The training session was performed in order to train the BCI classifier. During the testing session, in real time we provide to the subjects the output of the BCI classifier in orden to enhance the subject efforts to imagine a movement.

The subject was sitting in a comfortable chair located one meter from a 17" monitor. The subject was instructed to fix a gaze on a motionless circle of 1 cm in diameter, in the middle of the screen. Each 10 seconds, one command instruction was displayed in the screen. Four gray markers were placed around the circle. A marker changes the colour into green signalled to the subject that mental task must be performed. Each clue was preceded by a 4-second warning when the marker color changed into blue. Green color in the left and right markers indicates to *left and right hand movement imagining*, respectively. The top marker indicates *relaxation*. The lower marker corresponds to *feet movement imagining*. Four such instructions presented in random order constituted a block. The training session is composed by one block and the testing session has nine blocks as is illustrated in Figure 2. Each subject received 10 blocks of instructions at each experimental day. The structure of the block is presented in Figure 3. During the testing sessions, the result of the predictor classification was presented to the subject. This was done using green color in the central circle when the estimation of the classifier predictor coincides with the instruction. Besides, we increase the brightness of the central circle showing the augmenting of the classifying confidence. During the instruction to *relax* the subject does not receive feedbacks from the screen.

The EEG signals were recorded using 48 active electrodes and *g.USBamp* and *g.USBamp API* for MATLAB (g-tec, Graz, Austria). The sampling frequency employed was 256 Hz. The EEG signal were filtered by notch filter in order to suppress supply noise. The position of the electrodes were: Fz, F3, F4, Fcz, Fc3, Fc4, F7, F8, Fcz, Fc3, Fc4, Fc5, Fc6, Fc7, Fc8, Cz, C1, C2, C3, C4, C5, C6, T7, T8, Cpz, Cp1, Cp2, Cp3, Cp4, Cp5, Cp6, Tp7, Tp8, Pz, P1, P2, P3, P4, P5, P6, P7, P8, Pose, Po3, Po4, Po7, Po8, Oz, O1, O2. The central frontal electrode (Afz) was taken as reference. All codes of data processing were carry out with Matlab (Mathworks Inc. Natick, Ma, USA). The subjects have provided written a participation consent. The experimental procedure was approved by the *Board of Ethics at the Institute for Higher Nervous Activity and Neurophysiology of the Russian Academy of Sciences* [8].

Daily experimental sessions	
Training	Testing
1 block	9 blocks

Fig. 2. The sequence of sessions in the experimental protocol

Block							
Relaxation		Left hand MI		Right hand MI		Foot MI	
4	10	4	10	4	10	4	10

Fig. 3. Structure of the experimental block. Each instruction was presented only once and using random selection. The light blue areas of the block represent time of the instruction for warnings. The rest of the blue areas of the block represent time of the instruction for performance.

2.2 Bayesian Classifier Description

In this article, the mental tasks were classified using a *Bayesian Classifier (BC)* [1, 6]. Let L be the number of mental tasks to be classified and let N be the number of active electrodes used for recording the EEG signals. We denote by $X_n(t)$ the EEG signal recorded by the electrode n at time t. We assume that $X_n(t)$ has Gaussian distribution with zero mean for all n and t. We denote by $C_i(t)$ the covariance matrix of the EEG signal corresponding to the i task with $i = 1, \ldots, L$. Given a signal $X(t)$, for determining the class that the signal X(t) is associated, we compute the values of $Prob(X(t) \mid i)$, for all i. We assign to $X(t)$ the class such that occurs the maximum value of $Prob(X(t) \mid i)$. Due to the distribution of $X(t)$ is Gaussian, the probability to obtain $X(t)$ under the condition that it corresponds to performing the mental tasks i is given by

$$Prob(X(t) \mid i) \propto \exp\left(\frac{-V_i(t)}{2}\right), \qquad (1)$$

where $V_i(t)$ is defined as $V_i(t) = X^T(t)C_i(t)X(t) + \ln(det(C_i(t)))$ and $\det(\cdot)$ is the determinant function of a matrix. Note that, $C_i(t)$ must be a nonsingular matrix in order to compute the $\ln(det(C_i(t)))$.

The class that maximises the expression (1) is such that minimises the value $V_i(t)$. As a consequence, it is enough to find the minimal values of $V_i(t)$ at each time for all i. The value of $V_i(t)$ can be unstable in the time, therefore we split the signal in epochs. Let u be an epoch of duration Δt. We compute the data covariance matrix at each epoch u as

$$C(u) = \langle X(u)X^T(u)\rangle, \tag{2}$$

then we compute the average $V_i(u)$ by

$$\langle V_i(u)\rangle = \mathrm{trace}(\mathrm{C(u)C_i^{-1}(u)}) + \mathrm{ln(det(C_i(u)))}, \tag{3}$$

where trace$(\cdot)$ is the trace function of a matrix.

The training phase of the Bayesian predictor consists in computing the covariance matrices C_i for all i. The predictor was tested computing $C(u)$ and the $\langle V_i(u)\rangle$ values.

2.3 Accuracy of the Estimator

To evaluate the accuracy of the classifier and its generalisation capability we proceed as follows. We split the signals in epochs of a Δt duration. We randomly divide the learning set in 10 blocks. We randomly chose 7 blocks of them for computing the covariance matrices C_i for all mental tasks (training phase). The rest part of the learning set is used for testing the predictor. We repeat M times these classification trials. Next, we generate a confusion matrix P of dimensions $L \times L$ that contains the averages over all M classification trials. The P matrix has at the position (i, j) the probability $p_{i,j}$, that is the probability to recognise the $i-$th mental state in case that the instruction $j-$th mental task is performed. Note that, the better learning predictor performs the P is closer to the identity.

We chose the Cohen's κ function as indice of classification efficiency. The κ function is defined as follows:

$$\kappa = \frac{\frac{1}{L}\sum_{i=1}^{L} p_{i,i} - \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_{i,j}}{1 - \frac{1}{L^2}\sum_{i=1}^{L}\sum_{j=1}^{L} p_{i,j}}. \tag{4}$$

The value κ belongs to the $[0, 1]$ interval, closer is κ to 1 closer is the accuracy of the predictor, on the other hand a κ value closes to 0 indicates a deficient classifier.

3 Nature-Inspired Algorithm Description

This section introduces our main contribution, that is the algorithm that uses metaheuristic techniques for doing the EEG channels selection on the context of BCI based on motor imagery experiments. The section starts with a description of the nature-inspired algorithms used in this article: the *Simulating Annealing (SA)* and the *Genetic Algorithm (GA)*. Next, we specify our approach that mixes the metaheuristic techniques and the bayesian classifier.

3.1 Simulating Annealing Description

The *Simulating Annealing (SA)* method is an optimisation technique particularly interesting for solving problems of large scale. It has been applied for solving both combinatorial and continuous optimisation problems. The technique is mainly useful when the goal is to find a global extremum that is hidden among several local ones [9]. The goal is to minimise/maximise an objective function that in the SA context is often referred as *energy function*. The algorithm tries random steps following some criteria that arises from physical phenomena. The method is an analogy with the thermodynamical process that liquids freeze and crystallise or metals cool and anneal. Following this analogy, the method has a parameter called *temperature* (T), and a constant called *Boltzmann's constant* relates the temperature with the energy of the current system state. The technique is iterative, at each iteration we replace a current solution $\mathbf{s}^{\text{curr}}$ by a random *nearby solution* $\mathbf{s}^{\text{new}}$ that is chosen with a probability p. We consider a *nearby solution* such that its Hamming distance with the current solution is less than or equal to 1. In other words, the strings $\mathbf{s}^{\text{curr}}$ and $\mathbf{s}^{\text{new}}$ differ only in one bit. The temperature T decreases at each iteration until is reached some arbitrary value T^{end}. The probability of selecting a new solution is given by

$$p = \min\{\exp\left(-(E(\mathbf{s}^{\text{new}}) - E(\mathbf{s}^{\text{curr}}))/kT\right), 1\}, \tag{5}$$

where k is the Boltzmann's constant. The probability brings the capacity to jump from a local optimum to another part of the searching space. This exploration criteria usually takes a downhill step while sometimes takes an uphill step is popularly known under the name of *Metropolis Algorithm*. The algorithm starts from an initial solution with an initial temperature T, and a sequence of solutions are proposed, and the temperature decreases its value until it reaches a *frozen* condition.

3.2 Genetic Algorithm Description

The *Genetic Algorithm (GA)* family started in the 60's [10]. At the beginning, the technique was motivated by a biological analogy with the selective breeding of the plants and animals [10]. In the last 20 years, the GA trend has become increasingly popular for solving optimisation problems. A GA is an iterative procedure. At each iteration, points in the searching space are analysed as possible solutions, and they are combined according some rules. Following the biological analogy, the points collection is named *population*, each individual point is called *chromosome*, and the coordinates in a particular point are named *genes*. Each chromosome is evaluated by a fitness function $E(\cdot)$ that is the function to be optimised. The algorithm consists in modifying the population applying the following three evolutionary operations:

- Selection: there are several selection schema presented in the literature In this article, we study the selection following the *Baker's stochastic universal selection*, a single value is used for sample all of the solutions by choosing them at evenly spaced intervals.
- Crossover: It is a function that takes two chromosomes (often referred as *parents*) and generates two new ones (often referred as *offspring*). The operation replaces some genes of one parent by the corresponding genes of the other one. In general, the selection of which genes to replace is random. In this article, we follow the criteria of *one-point crossover*. Given two parents A and B, this operation consists in random selecting a cutting-point, it means a random position in the chromosome. Next, to generate a new two chromosomes. One of them in its first part (until the cutting point) has genes from parent A and the second part (from the cutting point till the end) has genes from the another parent B. Another chromosome has in its first part the genes from B and in its second part has the genes from A.
- Mutation: In this operation a randomly selected group of genes is changed. In our problem, the genes are binaries, then the gene mutation is the binary complement operation.

3.3 Applying the Nature-Inspired Algorithms for Feature Selection

Without loss of generality we enumerate the EEG channels by $\{1, \ldots, N\}$, where N is the number of electrodes sources of the EEG signals. The searching space of our problem is $\{0,1\}^N$. Possible solutions have the form $\mathbf{s} = [s_1, s_2, \ldots, s_N]$ where $s_i = 0$ represents that the signal captured by the electrode i is omitted as source of the classification tool, and $s_i = 1$ represents that the signal measured by the electrode i is an input of the classification tool. Besides, we consider as accuracy of our model the *kappa* function given by (4), that has domain in $[0,1]$. The problem is to find $\mathbf{s} \in \{0,1\}^N$ such that the kappa function is maximized when the BC is used for the mental class estimation. Note that a larger kappa value implies a better model accuracy. For this reason we have a maximisation problem instead of a minimisation one.

In the SA method, given a current solution $\mathbf{s}^{\text{curr}}$ we must select a *nearby* solution of $\mathbf{s}^{\text{curr}}$ that we denote by $\mathbf{s}^{\text{new}}$. In this step, we randomly select a value i in $[1, N]$. Next, we define the nearby solution as $s_j^{\text{new}} = s_i^{\text{curr}}$ for all $j \neq i$ and $s_j^{\text{new}} = s_i^{\text{curr}} + 1 \mod 2$, where $\mod$ is the module function. The procedure for generating the classification tool in the BCI using SA is presented in Algorithm (1). The algorithm has the following input parameters: an initial temperature $T^{(0)}$ and the stop condition T^{end}. Besides, it must be defined the a cooling schedule for decreasing the temperature. Algorithm (2) presents the method for generating the feature selection using GA.

Algorithm 1. Procedure for generating the classification tool in the BCI using Simulated Annealing.

```
Define an initial population $\{\mathbf{s}^{(1)}, \dots \mathbf{s}^{(K)}\}$;
for ($k = 1$ to $K$) do
    Generate the time-series data with the electrodes channels that verifies $s_i^{(k)} = 1$;
    Train the classification tool;
    Compute the kappa function;
$T = T^{(0)}$;
$\mathbf{s}^{\mathrm{curr}} = \mathbf{s}^{(k)}$;
while ($T \geq T^{\mathrm{end}}$) do
    Select a random nearby solution $\mathbf{s}^{\mathrm{new}}$;
    if ($kappa(\mathbf{s}^{\mathrm{new}}) \leq kappa(\mathbf{s}^{\mathrm{curr}})$) then
        $\mathbf{s}^{\mathrm{curr}} = \mathbf{s}^{\mathrm{new}}$;
    else
        Compute $p$ using expression (1);
        if ($rand(0,1) < p$) then
            $\mathbf{s}^{\mathrm{curr}} = \mathbf{s}^{\mathrm{new}}$;
    $i = i + 1$;
    Decrease temperature $T$;
Return $\mathbf{s}^{\mathrm{curr}}$;
```

4 Experimental Results

We begin by specifying the notation. we use the following abbreviations: the Bayesian Classifier without using metaheuristic is denoted by BC, Bayesian Classifier with feature selection using Simulating Annealing is denoted by BC-SA, and Bayesian Classifier with feature selection using Genetic Algorithms is denoted by BC-GA. A tradeoff between time resolution and accuracy is presented in the expression (3), wherein must be defined the epoch length criteria. We follow the same criteria that in [1] where the authors used epochs of 1 second length. The setting of the GA method was done as follows. We perform 1500 generations, each generation has 100 chromosomes, we use the Baker selection for select the parent chromosomes, and the mutation factor is 1/48. In the SA technique the cooling schedule consists in decrease the temperature in one unit at each algorithm iteration. In order to compare performance between SA and GA, both algorithms are performed during the same time. In order to have reference values about the accuracy of the BC without the feature selection using the metaheuristics, we perform 50 times the BC using random selection of the EEG channels. Then, we compute the kappa value reached by the BC predictor for each one of the 50 trials.

Table 1 shows the accuracy obtained by the BC, BC-SA and the BC-GA procedures. First column shows an identificador of the studied subject. The columns

Algorithm 2. Procedure for generating the classification tool in the BCI using Genetic Algorithm.

```
Define an initial population $\{\mathbf{s}^{(1)}, \dots \mathbf{s}^{(K)}\}$;
for $(k = 1$ to $K)$ do
    Generate the time-series data with the electrodes channels that
    verifies $s_i^{(k)} = 1$;
    Train the classification tool;
    Compute the kappa function;
while (cond is not satisfied) do
    repeat
        Select parent chromosomes;
        Choose a cutting point;
        Perform crossover;
        Choose mutation points;
        Perform mutation;
        Evaluate fitness of the offspring;
    until (New generation has enough offsprings);
Return the classification tool and the best combination of EEG channels;
```

Table 1. Classification accuracy using the kappa function. The first column presents the experiment identification. The second column presents the results when a Bayesian Classifier (BC) was performed using the all EEG channels. The third column shows the results of to use SA as feature selection and then to use BC. The last column shows the accuracy reached when GA is used for selecting the channels and BC is performed. 50 iteraciones

Experiment	50 random selections				SA-BC	GA-BC
Id	Mean	Std	95%	Max		
A	0.1424	0.0412	0.1538	0.2433	0.2522	**0.2631**
B	0.3398	0.0653	0.3579	0.4867	0.4836	**0.4972**
C	0.3076	0.1062	0.3370	0.4696	0.5360	**0.5372**
D	0.1152	0.0585	0.1314	0.2264	0.2478	**0.2544**
E	0.2028	0.0439	0.2150	0.2753	**0.2922**	0.2882

2 to 5 show results reached using the BC. In column 2, we can see the average value of kappa among the kappa values reached on the 50 trials. Column 3 presents the standard deviation of these set of kappa values, column 4 shows the upper endpoint of a 95% confidence interval, and the column 5 presents the maximum kappa value reached among the 50 trials. Column 6 shows the best kappa value reached by the BC-SA, and column 7 presents the best kappa value reached by the BC-GA. We can see that in all experiments the upper endpoint of the 95% confidence interval is lower than the kappa value reached used meta-heuristics. Even the maximum kappa value reached among the set of experiment

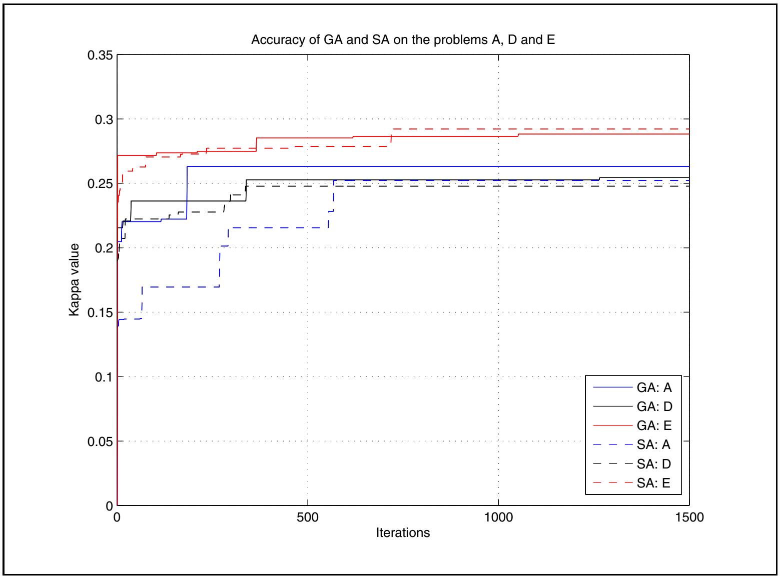

Fig. 4. Example of the evolution of the best kappa function for the experiments *A*, *D* and *E*. The dashed lines correspond to values obtained using BC-SA and continuous lines refer to the values obtained withe the BC-GA.

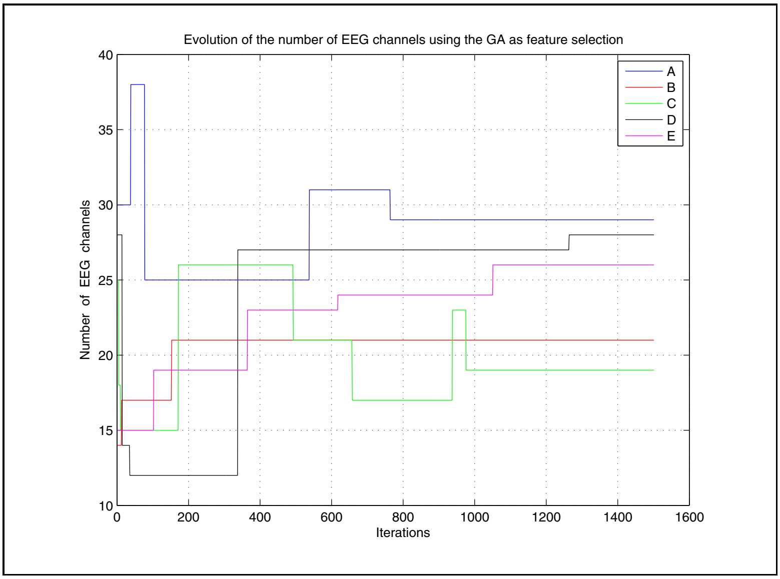

Fig. 5. Example of the evolution of the number of EEG channels used in the best solution when the BC-GA is applied

is less than the kappa value computed using BC-SA and BC-GA. In the 80% of the experiments the BC-GA reaches better kappa value than GA-SA. Figure 4 illustrates the evolution of the kappa value for the experiments A, B and C for both procedures BC-SA and BC-GA. Figure 5 presents an example of the evolution of the number of EEG channels used for computed the best kappa value at each generation of the BC-GA method. The figure has the evolution of number of electrodes used for the 5 experiments.

5 Conclusions and Future Work

An important task of a BCI system development consists in designing the EGG pattern classifier. The analysis based on EEG signals presents significant difficulties, due to the presence of noise. As a consequence, the selection of the EEG channels used for building a learning predictor can impact in the predictor performance. In general, in the EEG-based BCI experiments several EEG channels are used for collecting the data. For instance, in our experiments we are using 48 channels. Therefore, the selection of a best combination of EEG channels can not be done using a *brute-force* strategy.

In this article, we propose a solution for this problem that is based on two well-known metaheuristic techniques: *Simulating Annealing (SA) and Genetic Algorithms (GA)*. We analyse the performance of both techniques for selecting the EEG channels when we are using a Bayesian Classifier in the BCI system. We compare the performance of these techniques with a EEG random selection strategy. Besides, we present statistical results for that comparisons. We can affirm that the use of both metaheuristic procedures significantly improve the accuracy of the EEG pattern predictor. In particular, in the 80% of the experiments the higher accuracy is reached when the selection is done using the GA. As a for future work, we are interested in applying the same approach for EEG-based BCI visual imagery. Additionally, we have plans to compare the performance reached by SA and GA with other nature-inspired techniques.

Acknowledgement. This work was supported within the framework of the IT4Innovations Centre of Excellence project, reg. no. CZ.1.05/1.1.00/02.0070 supported by Operational Programme 'Research and Development for Innovations' funded by Structural Funds of the European Union and state budget of the Czech Republic and this article has been elaborated in the framework of the project New creative teams in priorities of scientific research, reg. no. CZ.1.07/2.3.00/30.0055. The authors also thank for their work support from the long-term strategic development financing budget of the Institute of Computer Science (RVO:67985807).

References

1. Frolov, A.A., Husek, D., Bobrov, P.: Comparison of four classification methods for brain computer interface. Neural Network World 21(2), 101–115 (2011)
2. Frolov, A.A., Husek, D., Bobrov, P., Mokienko, O., Tintera, J.: Sources of electrical brain activity most relevant to performance of brain-computer interface based on motor imagery. In: Brain-Computer Interface Systems - Recent Progress and Future Prospects, pp. 175–193. InTech (2013)
3. Bobrov, P., Frolov, A.A., Cantor, C., Fedulova, I., Bakhnyan, M., Zhavoronkov, A.: Brain-computer interface based on generation of visual images. PLOS ONE 6(6), 1–12 (2011)
4. Schröder, M., Bogdan, M., Hinterberger, T., Birbaumer, N.: Automated EEG feature selection for brain computer interfaces. In: First International IEEE EMBS Conference on Neural Engineering, pp. 626–629 (March 2003)
5. Peterson, D.A., Knight, J.N., Kirby, M.J., Anderson, C.W., Thaut, M.H.: Feature selection and blind source separation in an EEG-based brain-computer interface. EURASIP J. Appl. Signal Process. 2005, 3128–3140 (2005)
6. Bobrov, P.D., Korshakov, A.V., Roshchin, V.I., Frolov, A.A.: Bayesian Classifier for Brain-computer Interface B]ased on Mental Representation of Movements. Zh Vyssh Nerv Deiat Im I P Pavlova 62(1), 89–99 (2012)
7. Frolov, A.A., Husek, D., Bobrov, P., Korshakov, A., Chernikova, L., Konovalov, R., Mokienko, O.: Sources of EEG activity most relevant to performance of brain-computer interface based on motor imagery. Neural Network World 22(1), 21–37 (2012)
8. Institute of Higher Nervous Activity and Neurophysiology of RAS (IHNA & NPh RAS), Moscow, Russia, http://www.ihna.ru/en/
9. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C++: The Art of Scientific Computing. Cambridge University Press (February 2002)
10. Reeves, C.R.: Genetic Algorithms for the Operations Research. INFORMS Journal of Computing 9(3), 231–250 (1997)