WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia
IEEE CEC

Continuous Presentation for Multi-Objective Channel Selection in Brain-Computer Interfaces

Noura Al Moubayed
School of Computing
The Robert Gordon University
Aberdeen, UK
n.al-moubayed@rgu.ac.uk

Bashar Awwad Shiekh Hasan
Institute of Medical Sciences
University of Aberdeen
Aberdeen, UK
bashar.awwad@abdn.ac.uk

John Q. Gan
BCI Group
University of Essex
Colchester, UK
jqgan@essex.ac.uk

Andrei Petrovski
and John McCall
School of Computing
The Robert Gordon University
Aberdeen, UK

***Abstract*—A novel presentation for channel selection problem in Brain-Computer Interfaces (BCI) is introduced here. Continuous presentation in a projected two-dimensional space of the Electroencephalograph (EEG) cap is proposed. A multi-objective particle swarm optimization method (D^2MOPSO) is employed where particles move in the EEG cap space to locate the optimum set of solutions that minimize the number of selected channels and the classification error rate. This representation focuses on the local relationships among EEG channels as the physical location of the channels is explicitly represented in the search space avoiding picking up channels that are known to be uncorrelated with the mental task. In addition continuous presentation is a more natural way for problem solving in PSO framework. The method is validated on 10 subjects performing right-vs-left motor imagery BCI. The results are compared to these obtained using Sequential Floating Forward Search (SFFS) and shows significant enhancement in classification accuracy but most importantly in the distribution of the selected channels.**

***Index Terms*—Continuous Presentation, Channel Selection, Brain Computer Interfaces, EEG, Multi-Objective Problem, D^2MOPSO, Multi-Objective Particle Swarm Optimization, Dominance, Decomposition**

I. Introduction

Brain-Computer Interface (BCI) is a relatively new approach to communication between man and machine, which translates brain activity into commands for communication and control (i.e., computer cursor control, wheelchair control, robotic control, etc.) [1], [2], [3]. The user of a BCI system can perform several well-studied mental tasks to communicate and control [4], [5], [6]. The machine must be able to recognize these tasks from brain signals accordingly within a suitable time window for control. Motor imagery tasks are commonly used in BCI environment due to their good separability and the understanding of their neurological mechanisms [7].

Non-invasive BCI uses Electroencephalography (EEG) signals associated with predefined mental tasks. The number of channels used by an EEG system can vary according to experiment paradigm and hardware design. It usually ranges between 10 and 256 channels. For real-life BCI application it is important to select a smaller set of channels with as little sacrifice as possible in classification accuracy.

In order to avoid a large number of channels one can choose several electrode positions that are known from neurophysiological studies. Although this approach can be very useful, it ignores the fact that different subjects respond differently and the optimal positioning of the electrodes may vary. The other way to circumvent this problem is to use a large number of channels and use a method to reduce the dimensionality of the input features or to select the best set of channels for each subject.

Common Spatial Patterns (CSP) [8] is a well-known spatial filter that is widely used in BCI. CSP is useful for channel selection as it can be used to filter out the channels that provide less discriminate data. CSP requires the data from all the channels to be available online before the dimensionality is reduced. CSP depends on the estimation of the covariance matrices of multiple channel EEG data, which is usually very sensitive to noise.

The problem of channel selection is usually looked at as a search problem. The idea is to search the space of possible combinations of channels in order to find the optimal combination that produces the best classification accuracy. In [9], the author argues that feature selection is advantageous over dimensionality reduction in terms of interpretability. Feature selection (and similarly channel selection) using several search methods has been used frequently in the literature.

In [10] Digital Particle Swarm Optimization (DPSO) was used, where each particle contained a number of binary variables, which is equal to the number of channels, and cross validation results were used as the fitness function. In [11] a mixture of CSP and PSO based method was used for channel selection. In [12], [13] Sequential Floating Forward Search (SFFS) based methods were employed for channel/feature selection.

Most search-based solutions presented in the literature are single-objective methods. The classification accuracy is usually chosen as the only search criterion. The shortcoming of this approach is that the optimization process does not take into consideration the trade-off between the number of channels selected and the desirable classification accuracy. In theory, more channels would provide extra information that can help enhance the classification accuracy. In practice this might not be very accurate. It could be even desirable to sacrifice the accuracy in order to have fewer channels and hence a BCI system which can react to the user input within a more reasonable time window.

In [14], [15], we studied a novel multi-objective approach to the channel selection problem. The modeling of the channel selection problem was similar to that in single-objective case: the number of variables (usually binary) is equal to the number of channels with each channel can be either selected or not. The goal of the optimization method is to minimize two objectives: the number of channels and the error rate. The drawback of this modeling is that it does not count for the spatial relations among the channels making it prone to selecting outlier channels (i.e. channels that are known to have no correlation with the performed mental tasks).

This paper presents a new presentation of the channel selection problem based on the projection of the real channel positions in 3D into a two dimensional space. D^2MOPSO [16] is employed here as the multi-objective optimizer based on particle swarm optimization. D^2MOPSO utilizes the dominance concept [17] along with decomposition. It uses bounded crowding leaders' archive to store the non-dominated particles. The leader selection is then applied to the archive using the aggregation value as the selection criterion. The particle personal movement trajectory is updated using decomposition. All objectives are normalized in order to give them equal priorities when decomposition is applied. Towards the end of the optimisation process the size of the leaders' archive is substantially reduced to contain only non-dominated particles with the lowest crowding distance aiming at increasing the diversity and the coverage of low dense regions. For solving multi-objective channel selection problem, the particles in D^2MOPSO move continuously in the projected 2D space to locate the channels that achieve lowest error rate with fewer channels. This approach is tested on 10 participants performing right vs left motor-imagery tasks and the results are compared to SFFS based method.

II. Methods

A. Modeling Multi-Objective Channel Selection Problem

The modeling of the channel selection problem consists of defining the objective functions to be optimized and the representation of the problem in order for the optimization method to be able to solve it.

The channel selection problem is represented here by first setting up a maximum number of channels to select C. The number of decision variables would then be $2C$ as each channel is represented by its x and y coordinates in a projected 2D space of the channels 3D locations. The variables are real variables and can take any value within the space of the EEG cap. Each channel location is surrounded by an inclusion circle with radius R. A solution is defined as a set of tuples

$$s_i = \{(x_1, y_1), (x_2, y_2), \dots, (x_C, y_C)\} \tag{1}$$

Each tuple (x_i, y_i) is a point in the 2D cap space and is considered a selected channel if it falls within an inclusion circle. Should the point fall into two inclusion circles, the closest channel location is selected using Euclidean distance. Duplicated selected channels are ignored when calculating the objective values.

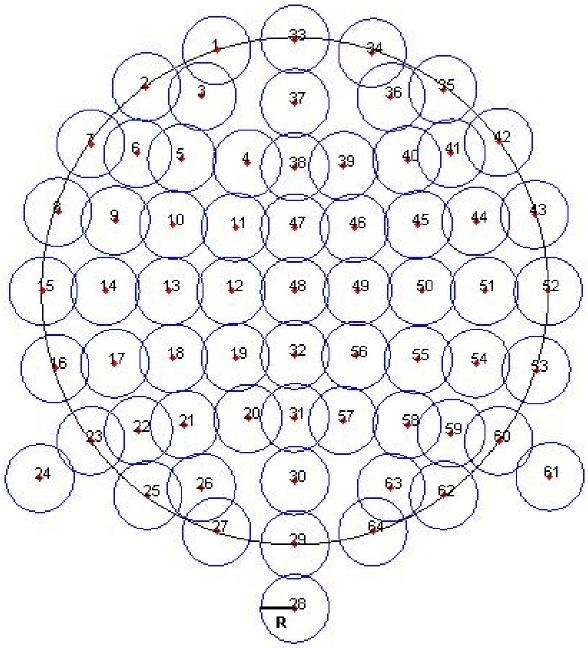

Fig. 1: Projected Biosemi 64+2 EEG channel locations. The numbering scheme follows the standard Biosemi numbering. Inclusion circles are drawn around each channel.

The first objective function is the classification error rate defined as $E = 1 - CV$ where CV is the cross-validation result. The second objective function is the number of selected channels C. The optimal solution(s) would have minimum number of channels with lowest error rate and hence minimize E and C.

B. D^2MOPSO: Multi-Objective Particle Swarm Optimizer based on Decomposition and Dominance

Decomposition transforms the MOP into a set of distinct aggregation problems. Each particle solves the corresponding problem by assigning a priority to each objective according to a weight vector (λ). This assists the optimisation process to find potential solutions that are evenly distributed along the Pareto Front (PF). By associating each particle with a distinct aggregation problem (i.e. λ value), the exploration activity of each particle is focused on a specific region in the objective space and aimed at reducing the distance to the reference point.

Substituting entirely the dominance approach in MOPSO with decomposition (i.e. using the aggregation value instead of dominance as the leaders' selection criterion) might lead to premature convergence as each particle is strictly directed to one destination. At some point during optimisation, the particles would be unable to update their positions and personal best memory as the global best and neighborhood information are not changing. In addition to this, solving a MOP with complicated PF raises a serious challenge as some λ vectors direct the related particles to unpromising areas. In this case, part of the swarm is waisting a large number of evaluations investigating undesirable regions. Another drawback of decomposition is that while solving MOP with high dimensional objective spaces, it fails to produce a sufficient number of non-dominated solutions that cover the entire PF as the space required to be covered by the swarm using λ vectors grows exponentially with the number of dimensions. To cope with this growth, decomposition based approaches need to use a large swarm to be able to offer a good PF coverage, which increases the number of function evaluations. The number of evaluations any evolutionary method needs to cover the PF is an important features as large number of evaluations counts as a big disadvantage for any EA and especially in real-life problem where evaluation can be very expensive.

To overcome these drawbacks, D^2MOPSO integrates both dominance and decomposition approaches. The bounded crowding leaders' archive [17], where the leaders of the swarm are selected from, is based on the dominance approach where only non-dominated particles are stored. When the archive is full, the non-dominated particles are only added at the low dense regions replacing those ones at the high dense regions. The personal best values are updated and the leaders are selected based on the decomposition's aggregation function.

Decomposition requires an aggregation function to decompose the MOP into several aggregation problems. Many functions have been proposed in the literature (e.g., weighted sum, Tchebycheff, weighted Tchebycheff and Penalty based boundary intersection (PBI)). Recently the weighted PBI method is reported to be of interest [18] and is used in this paper. PBI is originally proposed in [19] by modifying Normal Boundary Intersections (NBI). PBI uses a weighted vector λ and a penalty value θ for minimizing the distance to the utopia vector (i.e. a hypothetical vector between the reference point ($z^* = \min\{f_i(p)|p \in \Omega\}$) and the center of the PF) d_1 and the direction error to the weighted vector d_2 from the solution $F(p)$ in the objective space. PBI is then defined in [20]:

$$minimize \quad g(p|\lambda, z^*) = d_1 + \theta d_2 \tag{2}$$

where

$$\begin{aligned} d_1 &= \frac{\| (F(p) - z^*)^T \lambda \|}{\| \lambda \|} \\ d_2 &= \| (F(p) - z^*) - d_1 \frac{\lambda}{\| \lambda \|} \| \end{aligned} \tag{3}$$

p is used here to denote the position of the particle in the solution space rather than x to avoid confusion with the channel location in the EEG cap space.

D^2MOPSO uses the PBI approach to decompose the optimisation objective defined by Eq. 4 into N scalar optimisation problems, where N is the swarm's size. By changing the weights and using the reference point defined above, any Pareto optimal solution can be reached.

$$F(p) = \{f_1(p), f_2(p), \ldots, f_m(p)\} \tag{4}$$

where $p \in \Omega$, and m is the number of objectives.

In addition to combining dominance and decomposition, D^2MOPSO normalizes the MOP objectives. As the ranges of the objectives' values can differ considerably and are rarely known a priori for the majority of real life problems, objectives needed to be normalized before aggregation. This ensures equal priorities for all objectives, thereby preventing one objective from dominating the others when the aggregation is applied. The objective values are normalized using a sigmoid limiting transformation function defined in Eq.5. The Sigmoid limiting transformation is chosen as it does not need any prior knowledge of the objectives' ranges.

$$S(f_i(p)) = \frac{1}{(1 + e^{-f_i(p)})} \tag{5}$$

In Eq. 3 the normalized value of each objective is used instead of the objective values: $S(F(p)) = (S(f_1(p), f_2(p), \ldots, f_m(p)))$ instead of $F(p)$.

The particle's position is a solution to the channel selection problem. The position is presented as:

$$\begin{aligned} P &= \{x_1, y_1, x_2, y_2, \ldots, x_C, y_C\} \\ &= \{p_1, p_2, p_3, y_4, \ldots, p_{2*C-1}, p_{2*C}\} \end{aligned} \tag{6}$$

where C is the maximum number of selected channels, x_i, y_i are coordinates in the cap space as defined in Eq.1.

Every particle determines the next move by finding the new velocity and new position using Eq. 7 and Eq. 8. The new velocity, at iteration t, is calculated using $pbest$, $gbest$ (i.e. the information of a global leader selected from the leaders' archive), and the velocity of the particle in the previous iteration $v_{i_{t-1}}$.

$$\begin{aligned} v_{i_t} &= w * v_{i_{t-1}} + C_1 * r_1 * (p_{pbest_i} - p_{i_{t-1}}) \\ &+ C_2 * r_2 * (p_{gbest_i} - p_{i_{t-1}}) \end{aligned} \tag{7}$$

$$p_{i_t} = p_{i_{t-1}} + v_{i_t} \tag{8}$$

where $pbest_i$ is the personal best performance of $particle_i$, $gbest$ is the global best position of the leader selected from the archive, $r_1, r_2 \in [0, 1]$ are random values, $w \in [0.1, 0.5]$ is the inertia weight, and $C_1, C_2 \in [1.5, 2.0]$ are the learning factors that take uniformly distributed random values in their predefined ranges. The process of leader selection uses a uniformly distributed random variable $r \in [0, 1]$ to decide whether to select the leaders randomly or using their aggregation values (i.e. each particle selects the leader that gives the best aggregation value using the particle's λ) depending on a 0.5 threshold.

Towards the end of the optimisation process the swarm is likely to be converged but the particles are still evaluated till the termination of the algorithm. At the last $\alpha\%$ of the iterations, D^2MOPSO reduces the leaders archive size to $\beta\%$ of its original size. $100 - \beta\%$ of the particles in the original archive are removed according to their crowding distance leaving the particles with the lowest crowding distance in the archive (i.e. particles at the low dense regions in the objective space). The particles then select their leaders from the new archive. This directs all the particles at the end of the optimisation towards the low dense regions enhancing thereby the coverage and diversity. α and β are set based on the convergence of the leaders' archive, i.e. the leaders' archive is no more updated with new non-dominated particles.

The pseudo-code of D^2MOPSO is listed in Algorithm 1.

Algorithm 1 D^2MOPSO

```
Initialize the swarm with $N$ particles and $N$ $\lambda$ vectors
for $i = 1$ to $N$ do
    assign the particle $i$ to its closest $\lambda$ vector
    initialize velocities $V = \{v_1, \ldots, v_N\}$ and $pbest_i$
    Initialize leaders' archive, external archive and $z^*$
end for
Crowding(leader archive)
for $i = 1$ to $MaxIteration$ do
    if $i == (1 - \alpha\%) * MaxIteration$ then
        reduce leaders archive size
    end if
    for $j = 1$ to $N$ do
        update Velocity, $v_j(t+1)$
        update position, $p_j(t+1)$
        evaluate the new position
        normalize objectives and calculate aggregate function for $j$
        update $pbest_j$, leaders archive, external archive, and $z^*$
    end for
end for
Return the final result in the external archive
```

C. Sequential Forward Floating Search

SFFS is a comprehensive search single objective algorithm. It starts by selecting the single best channel (with CV as the search criterion). The algorithm continues by combining the selected channel with all the non-selected channels one by one and as a result selects the best two channels and so on. This is called the growing phase as opposed to the pruning phase in which a channel is removed from the selected set of channels and the criterion is checked again. If the criterion has a higher value with lower number of channels then the new channels set is adopted. The algorithm alternates between growing and pruning phases until a maximum number of channels N is selected or the maximum number of iteration is reached [13]. SFFS is applied here as a baseline to check the performance of the proposed algorithm.

III. Experiments and Results

A. Data Recording and Pre-processing

EEG data were recorded from 10 healthy subjects using a (64+2)-channel Biosemi system. Standard synchronous motor-imagery training was used [21] with two motor-imagery tasks: left hand, and right hand. No feedback sessions were recorded. Figure 2 shows the structure of the trials.

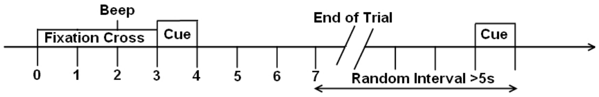

Fig. 2: The structure of the synchronous trials

Data were originally recorded at $256Hz$ but downsampled to $25Hz$ after feature extraction. Common reference was used in this study. Butterworth bandpass filter $(1 - 45Hz)$ is used to remove possible external interference. No artifact removal methods were applied but data were visually checked.

For each subject, data were recorded over 4 sessions with 10 minutes break in between. Every session consisted of 20 trials per class. In total 160 trials were recorded. The first 4 seconds of every trial are ignored as it does not contain any task-related information [21].

In order to get the channel locations a Biosemi 64+2 channels locations file provided by EEGLAB (http://sccn.ucsd.edu/eeglab/channellocation.html) is utilized. This provides a three-dimensional view of the channels which is projected onto a two-dimensional plane that goes through the assumed center of the brain (i.e. the origin used to define the location of the channels in the 3D space) and parallel to the XY plane.

B. Feature Extraction and Classification

$\mu(8 \sim 12Hz)$ and low $\beta(13 \sim 16Hz)$ rhythms are extracted from each channel by applying a bandpass FIR filter at the corresponding frequency band. The filtered data are squared and then averaged within consecutive time intervals [22].

If a channel is selected both its μ and β features are used. Extracted features from the selected channels are combined together to form one feature vector of maximum 20 features (the maximum number of selected channels is set to 10). Linear Discriminant Analysis (LDA) was used to classify the extracted features with an averaging window of 1 second to smooth the classifier output. 4-fold cross validation is applied, based on trial by trial classification, and is used to calculate the error rate.

C. D^2MOPSO parameter settings

D^2MOPSO employs a swarm of 100 particles that evolve through 100 generations. The inertia weight w is set to a random value in the range $[0.1, 0.5]$, C_1 and C_2 are set to 2, r_1, r_2 are set to random values in $[0, 1]$, and α and β are set to 10%. The algorithm was run 10 times per subject to avoid any bias due to the random initialization.

The maximum number of possible selected channels is set to 10 which means the solution space, in which the swarm evolves, has 20 dimensions (see Eq.6). Decision variables $(p_i : i \in [1, 2C])$ are bounded by an upper and lower limit to constrain these variables within the space of the EEG cap.

TABLE I: Results using SFFS

Subject	Cross Validation Accuracy%	Number of Channels
Subject-1	67.05	10
Subject-2	68.17	10
Subject-3	80.17	10
Subject-4	67.20	10
Subject-5	75.29	7
Subject-6	65.16	8
Subject-7	73.71	10
Subject-8	67.69	6
Subject-9	72.22	8
Subject-10	62.37	9
Average	69.90	8.8
Std	5.34	1.47

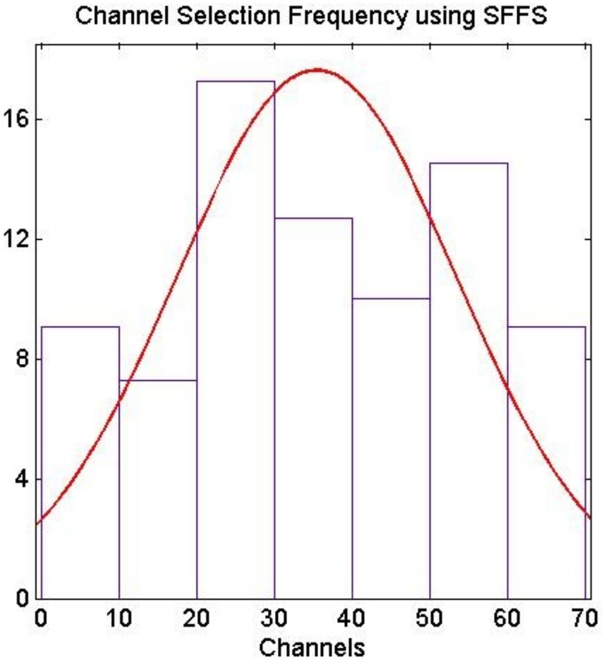

Fig. 3: Frequency of Channels selected via SFFS

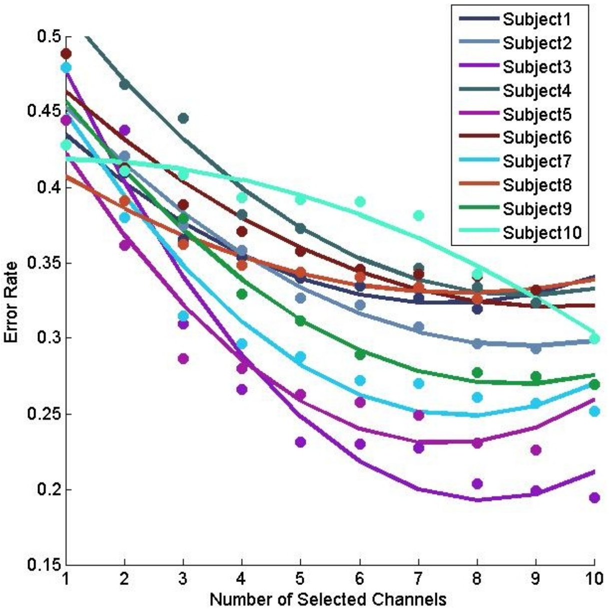

Fig. 4: Results using D^2MOPSO. Results of each subject are plotted with a polynominal fit of degree 2 to show the approximated Pareto Front.

TABLE II: Maximum Results using D^2MOPSO

Subject	Cross Validation Accuracy%	Number of Channels
Subject-1	68.03	10
Subject-2	70.70	9
Subject-3	80.11	9
Subject-4	66.78	9
Subject-5	77.39	10
Subject-6	70.03	10
Subject-7	74.81	10
Subject-8	67.39	9
Subject-9	73.07	10
Subject-10	70.03	10
Average	71.83	9.6
Std	4.43	0.52

D. Results

First the results using SFFS are presented as a baseline to check the validity of the results obtained using D^2MOPSO. Table I lists the results achieved on the 10 subjects. Figure 3 shows a histogram of the selected channels over all 10 subjects with a Gaussian fit of the distribution of the selected channels.

Figure 4 plots the results obtained using D^2MOPSO, where the results of each subject is plotted in distinct color with a ploynominal fit of degree 2 of the result to show the approximated Pareto Front for each subject.

To compare with SFFS in terms of classification accuracy only, Table II shows the maximum accuracy achieved for each of the 10 subjects. A two-sided t-test shows a significant enhancement ($p = 0.04 < 0.05$) of classification accuracy (taking the max accuracy for D^2MOPSO) which is supported by Figure 5 (the majority of points are under the unity line). The number of channels is harder to compare as D^2MOPSO provides a wide range of solutions as shown in Fig. 4, but comparing with the highest accuracy obtained by D^2MOPSO shows insignificant difference ($p = 0.1527 > 0.05$) suggesting that D^2MOPSO is capable of achieving higher classification accuracy for the same number of channels.

Figure 6 illustrates the distribution of channels selected for 10 subjects and using all the solutions obtained in the 10 runs per subject (after removing any dominated solutions), showing a wider distribution than that in Fig. 3. This can be interpreted as D^2MOPSO is able to widen its search space and provide more diverse solutions which reflects the goal of the continuous problem presentation in the first place. Looking at Fig. 1 it is clear that channels in the motor cortex area have diverse channel numbers which explains why the wider distribution of channels in Fig. 6 achieves higher classification accuracy.

IV. Conclusion and Discussion

A continuous presentation of the channel selection problem in Brain-Computer Interfaces was introduced within the

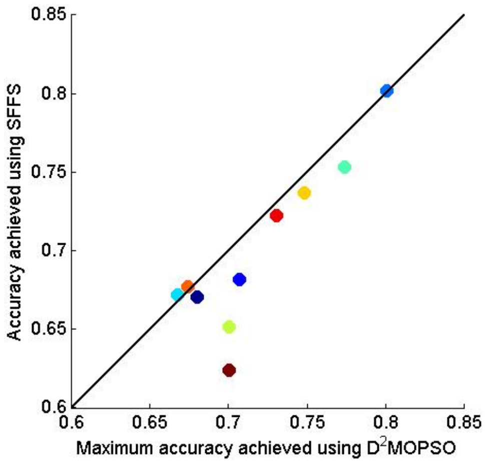

Fig. 5: Comparison between accuracy results obtained using SFFS and D^2MOPSO

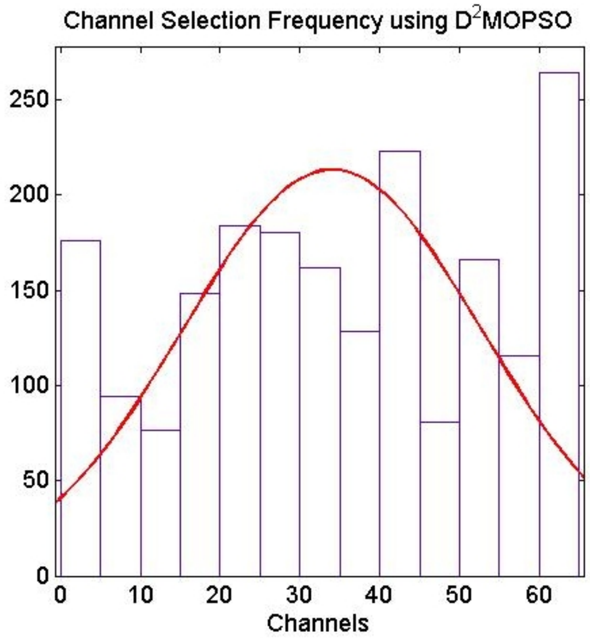

Fig. 6: Frequency of Channels selected via D^2MOPSO

framework of D^2MOPSO to offer a more natural way of solving the problem aiming at higher classification accuracy with lower number of selected channels. The usage of this presentation is not limited to D^2MOPSO but can easily be adopted in any other multi-objective (or even single objective) optimizer capable of solving continuous problems (e.g. GA).

The continuous presentation has an advantage of better exploring the search space by exploiting the spatial relationships among channels rather than looking at the channel selection problem as a discrete problem where channels are considered spatially independent, which is demonstrated in Figure 6 and Figure 3.

In a typical scenario, a particle (assumed to be looking for one channel for simplicity) will start from a random location within the 2D cap space and then navigate through the channels to find the one channel that achieves the highest accuracy. When more channels are needed the particle will use this navigability mechanism to maintain the spatial relationship among channels resulting in a much homogeneous set of channels for classification.

In the presented experiments, the particles were allowed to start from random locations in the space of the brain. Restricting the initial positions to regions that are known to contain task-related information might be advantageous but was not used here because we are also interested in finding channels outside the commonly known areas to account for individual differences. In general, however, the selected channels fall within the motor cortex area (i.e. around C_z) for most subjects.

The results presented in Fig. 4 show predicted Pareto fronts (i.e. error rate goes down with the increase of number of channels but only to a certain limit where the error rate can increase again) for most subjects with some exceptions (e.g. Subject10 which showed an increase in error rate with the increase in the number of channels).

The effectiveness of the multi-objective approach is minimizing the number of channels that can be used with insignificant sacrifice in accuracy is not discussed here in detail as this was previously studied and demonstrated in [14], [15]. However, Figure 4 details the different solutions obtained for each subject and it shows for most subjects similar results to these found in [14], [15].

The results presented here are based on synchronous BCI design (i.e. the timing is controlled by the system). We have not tested this approach on self-paced design (i.e. the timing is controlled by the subject) and hence we cannot make any claim on how it might perform, but we can comfortably predict similar results mainly due to the problem presentation proposed.

We did not compare the results of D^2MOPSO here with other multi-objective algorithms (e.g. [14]) as the goal of this work is to demonstrate the effectiveness of the proposed presentation rather than the optimizer itself.

References

[1] N. Birbaumer, T. Hinterberger, A. Kubler, and N. Neumann, "The thought-translation device (ttd): Neurobehavioral mechanisms and clinical outcome," *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 11, pp. 120–123, 2003.

[2] C. Tsui, J. Gan, and S. Roberts, "A self-paced brain–computer interface for controlling a robot simulator: An online event labelling paradigm and an extended Kalman filter based algorithm for online training," *Medical and Biological Engineering and Computing*, vol. 47, pp. 257–265, 2009.

[3] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan, "Brain-computer interfaces for communication and control," *Clinical neurophysiology*, vol. 113, no. 6, pp. 767–791, 2002.

[4] G. Pfurtscheller, C. Neuper, and N. Birbaumer, *Human brain–computer interface (BCI). In A. Riehle and E. Vaadia (Eds.), Motor Cortex in Voluntary Movements. A Distributed System For Distributed Functions*. Boca Raton, FL: CRC Press, 2005.

[5] G. Pfurtscheller and F. L. da Silva, "Event-related EEG/MEG synchronization and desynchronization: basic principles," *Clinical Neurophysiology*, vol. 110, no. 11, pp. 1842–1857, 1999.

[6] C. Neuper and G. Pfurtsheller, “Motor imagery and ERD,” *In: Pfurtscheller G, Lopes da Silva FH. Handbook of Electroencephalography and Clinical Neurophysiology- Event-related Desynchronization*, 1999.

[7] B. A. S. Hasan and J. Gan, “Temporal modeling of eeg during self-paced hand movement and its application in onset detection,” *Journal of Neural Engineering*, vol. 8, p. 056015, 2011.

[8] G. Dornhege, B. Blankertz, M. Krauledat, and et al, “Combined optimization of spatial and temporal filters for improving brain-computer interfacing,” *IEEE Transactions on Biomedical Engineering*, pp. 2274–2281, 2006.

[9] J. Gan, “Feature dimensionality reduction by manifold learning in brain computer interfaces,” in *3rd International Workshop on Brain Computer Interfaces*, Graz, Austria, 2006, pp. 28–29.

[10] J. Jin, X. Wang, and J. Zhang, “Optimal selection of eeg electrodes via dpso algorithm,” in *In Proc. of the 7th World Congress on Intelligent Control and Automation*, 2008.

[11] L. Jun and L. Meichun, “Common spatial pattern and particle swarm optimization for channel selection in bci,” in *In Proc. 3rd International Conference on Innovative Computing Information and Control. ICICIC 08*, 2008, pp. 457–457.

[12] M. Dyson, F. Sepulveda, J. Gan, and S.J.Roberts, “Sequential classification of mental tasks vs. idle,” in *In International IEEE EMBS Conf. on Neural Engineering*, 2009, pp. 351–354.

[13] J. Q. Gan, B. A. S. Hasan, and C. Tsui, “A hybrid approach to feature subset selection for brain-computer interface design,” *Intelligent Data Engineering and Automated Learning-IDEAL 2011*, pp. 279–286, 2011.

[14] B. A. S. Hasan, J. Gan, and Q. Zhang, “Multi-objective evolutionary methods for channel selection in brain-computer interfaces: some preliminary experimental results,” in *WCCI*. Barcelona, Spain: IEEE, July 2010, pp. 3339–3344.

[15] N. A. Moubayed, B. A. S. Hasan, J. Gan, A. Petrovski, and J. McCall, “Binary-SDMOPSO and its application in channel selection for brain-computer interfaces,” in *Computational Intelligence (UKCI), 2010 UK Workshop on*. IEEE, 2010, pp. 1–6.

[16] N. A. Moubayed, A. Petrovski, and J. McCall, “D^2MOPSO: Multi-objective particle swarm optimizer based on decomposition and dominance,” in *The 12th European Conference on Evolutionary Computation in Combinatorial Optimisation*. IEEE, 2012.

[17] M. Reyes-Sierra and C. Coello, *Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance*, ser. Lecture Notes in Computer Science. Springer, 2005, vol. 3410.

[18] S. Martínez and C. Coello, “A multi-objective particle swarm optimizer based on decomposition,” in *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, ser. GECCO ’11. ACM, 2011.

[19] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of moea/d on cec09 unconstrained mop test instances,” in *CEC’09: Proceedings of the Eleventh conference on Congress on Evolutionary Computation*. IEEE, 2009, pp. 203–208.

[20] Q. Zhang and H. Li, “MOEA/D: A multi-objective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] G. Pfurtscheller and C. Neuper, “Motor imagery and direct brain–computer communication,” *Proceedings of the IEEE*, vol. 89, pp. 1123–1134, 2001.

[22] W. Kilmesch, “Event–related band power changes and memory performance,” *Handbook of Electroencephalography and Clinical Neurophysiology Revised Series*, vol. 6, pp. 161–178, 1999.