

# Selección de Canales en Sistemas BCI basados en Potenciales P300 mediante Inteligencia de Enjambre

V. Martínez-Cagigal<sup>a,\*</sup>, R. Hornero<sup>a,b,c</sup>

<sup>a</sup>Grupo de Ingeniería Biomédica, E.T.S.I. de Telecomunicación, Universidad de Valladolid, Valladolid, España.

<sup>b</sup>IMUVA, Instituto de Investigación en Matemáticas, Universidad de Valladolid, Valladolid, España.

<sup>c</sup>INCYL, Instituto de Neurociencias de Castilla y León, Salamanca, España.

## Resumen

Los sistemas Brain-Computer Interface (BCI) se definen como sistemas de comunicación que monitorizan la actividad cerebral y traducen determinadas características, correspondientes a las intenciones del usuario, en comandos de control de un dispositivo. La selección de canales en los sistemas BCI es fundamental para evitar el sobre-entrenamiento del clasificador, reducir la carga computacional y aumentar la comodidad del usuario. A pesar de que se han desarrollado varios algoritmos con anterioridad para tal fin, las metaheurísticas basadas en inteligencia de enjambre aún no han sido suficientemente explotadas en los sistemas BCI basados en potenciales P300. En este estudio se muestra una comparativa entre cinco métodos de enjambre, basados en el comportamiento de sistemas biológicos, aplicados con el objetivo de optimizar la selección de canales en este tipo de sistemas. Los métodos se han evaluado sobre la base de datos de la “III BCI Competition 2005”, reportando precisiones similares o, en algunos casos, incluso más altas que las obtenidas sin realizar ningún tipo de selección. Dado que los cinco métodos se han demostrado capaces de disminuir drásticamente los 64 canales originales a menos de la mitad sin comprometer el rendimiento del sistema, así como de superar el conjunto típico de 8 canales y el método *backward elimination*, se concluye que todos ellos son adecuados para su aplicación en la selección de canales en sistemas P300-BCI.

### Palabras Clave:

Interfaces Hombre-Máquina, aprendizaje automático, sistemas biomédicos, problemas de optimización, electrodos, sistemas de comunicación

## 1. Introducción

Los sistemas Brain-Computer Interface (BCI) permiten establecer un sistema de comunicación entre el cerebro y el medio capaz de traducir las intenciones del usuario en comandos de control de un dispositivo (Kübler et al., 2007; Kübler y Birbaumer, 2008; Wolpaw et al., 2000, 2002). Generalmente, se emplean con el objetivo de mejorar la calidad de vida de personas con grave discapacidad motora cuya capacidad de comunicación se encuentra limitada, por ejemplo, debido a enfermedades neurodegenerativas, síndrome de Guillain Barré, lesiones cerebrales, distrofias musculares u otras enfermedades que deterioran los caminos neuronales que controlan las funciones motoras (Kübler et al., 2007; Kübler y Birbaumer, 2008; Wolpaw et al., 2000, 2002). Aunque existen multitud de métodos para monitorizar la actividad cerebral, es común utilizar el electroencefalograma (EEG) debido a su bajo coste, facilidad de uso,

portabilidad, y naturaleza no invasiva. De esta manera, la actividad eléctrica del cerebro se registra mediante la colocación de una serie de electrodos sobre el cuero cabelludo del usuario (Wolpaw et al., 2000, 2002).

Los potenciales evocados P300 son una de las señales utilizadas en sistemas BCI para determinar la atención visual del usuario. Se corresponden con deflexiones positivas de voltaje producidas en la zona parietal del córtex como respuesta a un estímulo externo, aproximadamente 300 ms después de que éste se produzca (Wolpaw et al., 2002). La forma más habitual de generarlos es a través del paradigma *oddball*, consistente en presentar estímulos objetivo infrecuentes, que tienen que ser atendidos, camuflados entre estímulos irrelevantes que tienen que ser ignorados (Farwell y Donchin, 1988; Wolpaw et al., 2002).

Originalmente desarrollado por Farwell y Donchin (1988), el “P300 Speller” es una de las aplicaciones basadas en el paradigma *oddball* que permiten seleccionar comandos a través de una estimulación visual. Consiste en mostrar al usuario una matriz de comandos cuyas filas y columnas se iluminan aleatoriamente. El usuario únicamente debe prestar atención al co-

\* Autor en correspondencia

Correos electrónicos: victor.martinez@gib.tel.uva.es (V. Martínez-Cagigal), robhor@tel.uva.es (R. Hornero)

mando que desea seleccionar, provocando la generación de un potencial P300 cuando la fila o la columna que contienen dicho comando se iluminen. El comando que desea seleccionar el usuario, por tanto, se determina reconociendo dónde y cuándo se han producido tales potenciales (Farwell y Donchin, 1988; Wolpaw et al., 2002).

Debido a la alta variabilidad inter-sesión y a la baja relación señal a ruido, es necesario realizar un promediado de varias secuencias (i.e., repeticiones de las iluminaciones) para detectar un potencial P300 con fiabilidad. Esta necesidad provoca que el sistema tenga que trabajar a tiempo real con una gran cantidad de datos, lo cual puede producir un sobre-entrenamiento del clasificador (Cecotti et al., 2011; Perseh y Sharafat, 2012). Una reducción de los datos a procesar, ya sea mediante una selección de características o una selección de canales, por tanto, ayuda a evitar el sobre-entrenamiento y reducir la carga computacional del sistema. De hecho, el uso de un número óptimo de canales, pero a la vez reducido, no sólo es beneficioso en la etapa de clasificación debido a la eliminación de características irrelevantes (Yu et al., 2015), sino que supone una mayor comodidad para el usuario, puesto que el tiempo de preparación del sistema se reduce, así como la consumición de energía en los gorros inalámbricos (Cecotti et al., 2011; Rivet et al., 2010).

Sin embargo, la selección de los canales más relevantes no es un problema trivial: para una configuración de  $N$  canales, existen  $2^N$  subconjuntos de selección posibles. La búsqueda exhaustiva, por tanto, es muy compleja e intratable en la mayor parte de los casos (Cecotti et al., 2011). A pesar de que el problema de selección de canales se ha estudiado ampliamente para sistemas BCI basados en la imaginación motora, los estudios realizados sobre sistemas BCI basados en potenciales P300 (P300-BCI) son escasos (Colwell et al., 2014). En la mayor parte de estos estudios se han empleado técnicas de eliminación recursiva de canales con distintos criterios de selección: (i) los basados en la precisión de clasificadores, como un conjunto de máquinas vectoriales de soporte (Rakotomamonjy y Guigue, 2008) o un análisis discriminante lineal bayesiano (BLDA, *Bayesian Linear Discriminant Analysis*) después de aplicar un filtrado xDAWN para ensalzar los potenciales P300 (Rivet et al., 2012); y (ii) los basados en el ratio entre la señal buscada y la señal ruidosa (SSNR) (Cecotti et al., 2011). Otros han empleado modificaciones del algoritmo, tales como la eliminación por clústers basada en el índice de bloqueo de fase (Xu et al., 2013), o la selección salto-a-salto (Colwell et al., 2014). Recientemente, Yu et al. (2015) ha empleado una técnica basada en la determinación de relevancia automática agrupada. Los métodos deterministas funcionan adecuadamente cuando la función a optimizar es continua y unimodal. No obstante, en la práctica, la función de coste suele ser discontinua y presentar irregularidades, aumentando la probabilidad de que este tipo de métodos queden atrapados por un mínimo local (Yang, 2014).

En este aspecto, las metaheurísticas, entre las cuales se encuentra la inteligencia de enjambre (SI, *Swarm Intelligence*), han demostrado un buen rendimiento a la hora de resolver complejos problemas de optimización (Yang et al., 2013). La SI integra el comportamiento colectivo de sistemas descentralizados y auto-organizados, cuya población está formada por agentes

que interactúan entre ellos y con el entorno (Yang et al., 2013). El flujo de información que se transmite entre dichos agentes permite realizar una búsqueda local y global simultáneamente, lo cual aumenta la robustez del sistema al no depender de agentes individuales y, por tanto, evita la convergencia hacia un mínimo local (Brownlee, 2011; Yang et al., 2013). Diversas variantes de la optimización por enjambre de partículas (PSO, *Particle Swarm Optimization*) se han aplicado a los sistemas P300-BCI como métodos de selección de canales con anterioridad, reportando resultados adecuados (Jin et al., 2010; Perseh y Sharafat, 2012; Gonzalez et al., 2013). No obstante, la SI es un campo de investigación muy extenso que cada vez integra una mayor cantidad de métodos y, hasta donde tenemos conocimiento, no se ha evaluado el rendimiento de algoritmos alternativos a PSO.

El objetivo de este estudio es realizar una comparativa entre diversos métodos de SI orientados a la selección de canales en sistemas P300-BCI, con el fin de determinar cuál es el más adecuado para este tipo de problemas, así como evaluar la viabilidad de su aplicación. Concretamente, se han aplicado los siguientes métodos SI: (1) PSO, (2) algoritmo de abejas (BA, *Bees Algorithm*), (3) colonia artificial de abejas (ABC, *Artificial Bee Colony*), (4) sistema binario de hormigas (BAS, *Binary Ant System*), y (5) algoritmo de luciérnagas (FA, *Firefly Algorithm*).

## 2. Señales

Los métodos aplicados en este estudio se han testeado con la base de datos de la “III BCI Competition 2005”, relativa al deletreo de palabras con el “P300 Speller”, proporcionada por el “BCI Laboratory” del “Wadsworth center” (New York Department of Health) (Blankertz et al., 2006). La base de datos la forman los registros EEG de dos sujetos durante cinco sesiones distintas. En cada sesión se pidió a los sujetos escribir nueve palabras con una matriz de  $6 \times 6$  (i.e., 36 caracteres). Los usuarios debían fijarse en los caracteres a seleccionar mientras las filas y las columnas de la matriz se iluminaban aleatoriamente con una frecuencia de 5,7 Hz. Las señales EEG se registraron con el *software* BCI2000 (Schalk et al., 2004), usando un gorro de 64 electrodos referenciados al lóbulo de la oreja, con una frecuencia de muestreo de 256 Hz. Como pre-procesado de la señal, se aplicó un filtro paso-banda entre 0,1–60 Hz y un filtro de ranura a 50 Hz para eliminar la interferencia de la red eléctrica. Asimismo, los conjuntos de 12 iluminaciones se repitieron durante 15 secuencias para cada carácter. Los subconjuntos de entrenamiento y test están formados por 85 y 100 caracteres, respectivamente. Se puede consultar información adicional en Blankertz et al. (2006).

## 3. Métodos

Por norma general, los métodos de SI se diseñan para problemas de optimización continuos. En este caso, dado que la decisión a tomar para cada canal del subconjunto es dicotómica: se rechaza o se selecciona; es necesario adaptarlos a un problema binario. Por lo tanto, denotamos una posible solución al

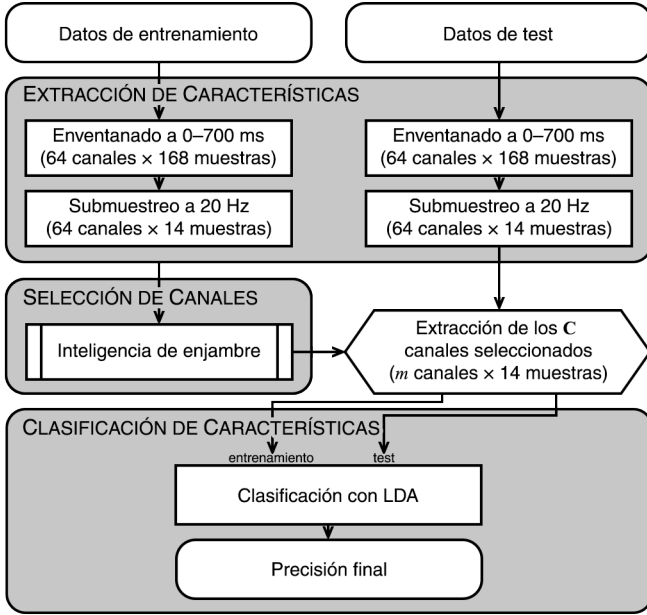


Figura 1: Metodología empleada a la hora de testear cada método, compuesta por tres etapas: (1) extracción de características, (2) selección de canales sobre los datos de entrenamiento y aplicación de la solución óptima, y (3) clasificación de características.

problema como  $\mathbf{c} = [c_1, c_2, \dots, c_{N_d}]$ , donde  $c_n = \{0, 1\}$  para  $n = 1, 2, \dots, N_d$ , con  $N_d$  siendo el número máximo de canales (i.e., en este caso,  $N_d = 64$ ). En los apartados siguientes se describe cada método, así como la adaptación binaria para su aplicación en los sistemas P300-BCI y la función de coste multi-objetivo empleada.

La figura 1 ilustra la metodología empleada para cada método. En primer lugar, se aplicó una etapa de extracción de características en la que: (i) se extrajeron los datos relativos a cada estímulo en una ventana entre 0 y 700 ms; y (ii) se realizó un submuestreo a  $f = 20$  Hz sobre dichos datos, manteniendo únicamente 14 muestras por estímulo en cada canal, correspondientes a la media de 13 segmentos equiespaciados (Krusienski et al., 2008). Posteriormente, se aplicó el método de SI deseado, evaluando la aptitud (*fitness*) de cada posible solución sobre el conjunto de datos de entrenamiento de acuerdo a una función de coste multi-objetivo que: (1) minimiza el número de canales a seleccionar, y (2) maximiza la precisión del clasificador. En este caso, la precisión del clasificador para cada posible solución se ha calculado usando un análisis discriminante lineal (*Linear Discriminant Analysis*, LDA) con una validación cruzada de  $K = 5$  iteraciones sobre los datos de entrenamiento. Finalmente, se evaluó la mejor solución obtenida para cada método sobre el conjunto de test. Este procedimiento se repitió 10 veces con el objetivo de evitar caer en mínimos locales a la hora de comparar cada uno de los métodos.

### 3.1. Particle Swarm Optimization

La optimización por enjambre de partículas (PSO, *Particle Swarm Optimization*), originalmente desarrollada por Kennedy y Eberhart (1995), constituyó una novedosa metaheurística basada en el comportamiento social de aves y bancos de peces que

provocó la aparición de lo que hoy se conoce como SI (Brownlee, 2011; Yang et al., 2013). El algoritmo busca optimizar una función objetivo mediante el ajuste de las trayectorias de agentes (i.e., partículas) individuales que se mueven a través del espacio de búsqueda de manera iterativa (Yang et al., 2013).

El movimiento de dichas partículas posee una componente determinista y una componente estocástica. Por un lado, cada partícula  $i = 1, 2, \dots, N_p$  se ve tanto atraída (i) hacia la mejor posición que ha encontrado anteriormente  $\mathbf{l}_i$ , como (ii) hacia la mejor posición encontrada por todo el enjambre  $\mathbf{g}$ . Por otro lado, posee también tendencia a moverse de manera aleatoria. En cada iteración, por tanto, se actualizan las velocidades  $\mathbf{v}$  y las posiciones  $\mathbf{x}$  de las partículas como sigue:

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \alpha_1 \varepsilon_1 \underbrace{[\mathbf{l}_i - \mathbf{x}_i(t)]}_{(i)} + \alpha_2 \varepsilon_2 \underbrace{[\mathbf{g} - \mathbf{x}_i(t)]}_{(ii)}, \quad (1)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t), \quad (2)$$

donde  $\varepsilon_1, \varepsilon_2 \sim U(0, 1)$  son componentes aleatorias y  $\alpha_1$  y  $\alpha_2$  son los coeficientes de confianza de la mejor posición individual y global, respectivamente (Brownlee, 2011; Yang et al., 2013).

Tal y como se puede observar en la expresión (1), la velocidad es un vector de componentes continuas. Debido a la naturaleza binaria del problema de este estudio, se ha discretizado la posición de cada partícula a través de una umbralización sobre una función sigmoide (i.e., logística) (Kennedy y Eberhart, 1997; Yang et al., 2013; Gonzalez et al., 2013), tal y como se muestra a continuación:

$$\mathbf{x}_i(t+1) = \begin{cases} 1 & \text{si } S(\mathbf{v}_i(t)) \geq L; \\ 0 & \text{si } S(\mathbf{v}_i(t)) < L, \end{cases} \quad (3)$$

donde  $S(u) = 1/(1 + e^{-u})$  es la función sigmoide empleada, y  $L$  es el umbral establecido.

### 3.2. Bees Algorithm

El algoritmo de abejas (BA, *Bees Algorithm*), desarrollado por Pham et al. (2006), se basa en el comportamiento que muestran las abejas a la hora de buscar comida. Para empezar, la colmena envía abejas exploradoras que se mueven de manera aleatoria en busca de flores con néctar o polen. Cuando éstas encuentran localizaciones prometedoras, regresan a la colmena a informar al resto sobre su descubrimiento, indicando: (1) la dirección a seguir, (2) la distancia a recorrer, y (3) la calidad del lugar. Dicha calidad, o aptitud, es directamente proporcional al número de abejas destinadas a explotarlo. Mientras las abejas que explotan las flores seleccionadas informan de la cantidad de néctar o polen que queda, un número reducido de abejas exploradoras continúa buscando nuevos lugares (Pham et al., 2006).

Basado en el comportamiento descrito, el BA se inicializa enviando  $N_b$  abejas exploradoras a posiciones aleatorias en un espacio de dimensión  $N_d$ . Una vez evaluada la calidad de tales posiciones, se ordenan según su aptitud de manera descendente y se explotan los  $m_s$  lugares más prometedores, de los cuales  $m_{es}$  se asignarán como lugares élite. Para explotarlos, se asignan  $N_o$  abejas exploradoras (o  $N_e$  si es un lugar élite) que evaluarán la calidad de la vecindad, variando ligeramente su posición en

**Pseudocódigo 1:** Método PSO binario empleado

---

**Entrada:**  $N_d, N_p, \alpha_1, \alpha_2, S, L$   
**Salida :**  $c$

```

1  $g \leftarrow$  Ceros  $(1, N_d)$ ;
2 for  $i = 1$  to  $N_p$  do
3    $v_i \leftarrow$  Ceros  $(1, N_d)$ ;
4    $x_i \leftarrow$  PosiciónAleatoria  $(1, N_d) \in \{0, 1\}$ ;
5    $l_i \leftarrow x_i$ ;
6   if Coste  $(l_i) <$  Coste  $(g)$  then
7      $g \leftarrow l_i$ ;
8   end
9 end
10 while  $\neg$ FinCondición do
11   foreach partícula $i$  do
12      $\varepsilon_1, \varepsilon_2 \leftarrow$  rand  $() \sim U(0, 1)$ ;
13      $v_i \leftarrow v_i + \alpha_1 \varepsilon_1 [l_i - x_i] - \alpha_2 \varepsilon_2 [g_i - x_i]$ ;
14      $x_i \leftarrow \begin{cases} 1 & \text{si } S(v_i(t)) \geq L; \\ 0 & \text{si } S(v_i(t)) < L; \end{cases}$ 
15     if Coste  $(x_i) <$  Coste  $(l_i)$  then
16        $l_i \leftarrow x_i$ ;
17       if Coste  $(l_i) <$  Coste  $(g)$  then
18          $g \leftarrow l_i$ ;
19       end
20     end
21   end
22 end
23 return  $c \leftarrow g$ ;

```

---

torno al lugar de explotación. La distancia máxima de variación la determina la variable  $\eta$ , que decrece de manera proporcional a la constante  $\lambda$  a medida que aumentan las iteraciones del algoritmo (Pham et al., 2006; Brownlee, 2011).

El algoritmo empleado en este estudio se muestra en el pseudocódigo 2. Dado que es necesario emplear un enfoque binario, la variación continua de la posición de las abejas en los lugares prometedores de acuerdo a  $\eta$  no tiene sentido. Por esta razón, en este estudio se ha optado por emplear el operador de mutación, común en los algoritmos genéticos (GA, *Genetic Algorithms*). De esta manera, las abejas que explotan cada lugar mutan  $\eta$  elementos aleatorios de su vector de posición en cada iteración, entendiendo por mutación un cambio de 0 a 1, y viceversa.

### 3.3. Artificial Bee Colony

La colonia artificial de abejas (ABC, *Artificial Bee Colony*) es un algoritmo diseñado por Karaboga (2005), que también se basa en el comportamiento de las abejas a la hora de buscar comida. En este caso, se diferencian tres tipos de abejas: (1) las abejas exploradoras (i.e., *scout bees*), que exploran el espacio de búsqueda de manera aleatoria; (2) las abejas empleadas (i.e., *employed bees*), que explotan la vecindad de las mejores localizaciones; y (3) las abejas observadoras (*onlooker bees*), que usan la aptitud de las posiciones previas para decidir qué lugares es necesario explotar en cada iteración (Karaboga, 2005; Yang et al., 2013; Karaboga et al., 2014; Kiran, 2015).

**Pseudocódigo 2:** Método BA binario empleado

---

**Entrada:**  $N_d, N_b, N_o, N_e, m_s, m_{es}, \eta, \lambda$   
**Salida :**  $c$

```

1  $X \leftarrow$  PosiciónAleatoria  $(N_b, N_d) \in \{0, 1\}$ ;
2 while  $\neg$ FinCondición do
3    $F \in \mathbb{R}^{N_b \times 1} \leftarrow$  EvaluarCostePoblación  $(X)$ ;
4    $g \leftarrow$  arg mín  $_X(F)$ ;
5   NuevaGen  $\leftarrow$  Ceros  $(N_b, N_d)$ ;
6    $\eta \leftarrow$  round  $(\lambda \cdot \eta)$ ;
7    $X_s \leftarrow$  OrdenarPosiciones  $(X, F)$ ;
8   Sitios  $\in \mathbb{Z}^{m_s \times N_d} \leftarrow$  MejoresPosiciones  $(X_s, m_s)$ ;
9   foreach Sitios $i$  do
10     if  $i < m_{es}$  then
11        $nr = N_e$ ;           % Sitio de élite;
12     else
13        $nr = N_o$ ;           % Sitio común;
14     end
15      $N \leftarrow$  Ceros  $(nr + 1, N_d)$ ;
16      $N_1 \leftarrow$  Sitios $i$ ;
17     for  $j = 2$  to  $nr + 1$  do
18        $N_j \leftarrow$  Sitios $i$ ;
19        $N_j \leftarrow$  MutarBitsAleatorios  $(N_j, \eta)$ ;
20     end
21      $F_n \in \mathbb{R}^{nr+1 \times 1} \leftarrow$  EvaluarCostePoblación  $(N)$ ;
22     NuevaGen $i$   $\leftarrow$  arg mín  $_N(F_n)$ ;
23   end
24   for  $r = 1$  to  $(N_b - m_s)$  do
25     NuevaGen $m_s+r$   $\leftarrow$  PosiciónAleatoria  $(1, N_d)$ ;
26   end
27    $X \leftarrow$  NuevaGen;
28 end
29 return  $c \leftarrow g$ ;

```

---

El pseudocódigo 3 detalla el algoritmo utilizado. Para empezar, se generan aleatoriamente  $N_s$  posiciones de comida y se les asigna un contador. En la fase de las abejas empleadas, se varía cada localización  $i$  de acuerdo a la siguiente expresión:

$$y_i = x_i + \phi(x_i - x_n), \quad (4)$$

donde  $\phi \sim U(-1, 1)$  y  $n$  indica un vecino seleccionado aleatoriamente. Si la nueva localización posee una mejor aptitud que la anterior, se actualiza la posición y se reinicia el contador. A continuación, cada una de las  $N_s$  abejas observadoras selecciona probabilísticamente una localización mediante una selección por ruleta (*roulette wheel*). En este método, la probabilidad de seleccionar una localización  $i$  es  $p_i = F_i / \sum F$ , con  $F$  siendo el vector de aptitudes de cada posición. Se evalúa la vecindad de las localizaciones seleccionadas mediante la expresión (4) y se actualizan los contadores. Para finalizar, se comprueba si algún contador supera el límite de abandono  $\varphi$  y, en tal caso, se crea una abeja exploradora que sustituye dicha localización generando una posición aleatoria (Kiran, 2015).

Cabe destacar que, debido a la naturaleza del problema bajo estudio, se ha empleado el enfoque de Kiran (2015), que

**Pseudocódigo 3:** Método ABC binario empleado

---

**Entrada:**  $N_d, N_s, \varphi$   
**Salida :**  $c$

```

1  $X \leftarrow$  PosiciónAleatoria( $N_s, N_d$ )  $\in \{0, 1\}$ ;
2 contador  $\leftarrow$  Ceros( $N_s, 1$ );
3  $F \in \mathbb{R}^{N_s \times 1} \leftarrow$  EvaluarCostePoblación( $X$ );
4  $g \leftarrow \arg \min_X(F)$ ;
5 while  $\neg$ FinCondición do
6   contador  $\leftarrow$  contador + 1;
7   for  $e = 1$  to  $N_s$  do
8      $n \leftarrow$  SelecciónVecinoAleatorio( $X, e$ );
9      $y \leftarrow \phi(X_e - X_n)$ ;
10    if Coste( $y$ ) <  $F_e$  then
11      contador $_e \leftarrow 0$ ;
12       $F_e \leftarrow$  Coste( $y$ );
13       $X_e \leftarrow y$ ;
14    end
15  end
16  for  $o = 1$  to  $N_s$  do
17    índice  $\leftarrow$  RouletteWheel( $X, F$ );
18     $n \leftarrow$  SelecciónVecinoAleatorio( $X, e$ );
19     $y \leftarrow \phi(X_{\text{índice}} - X_n)$ ;
20    if Coste( $y$ ) <  $F_i$  then
21      contador $_{\text{índice}} \leftarrow 0$ ;
22       $F_{\text{índice}} \leftarrow$  Coste( $y$ );
23       $X_{\text{índice}} \leftarrow y$ ;
24    else
25      contador $_{\text{índice}} \leftarrow$  contador $_{\text{índice}} + 1$ ;
26    end
27  end
28  for  $s = 1$  to  $N_s$  do
29    if contador $_s > \varphi$  then
30       $X_s \leftarrow$  PosiciónAleatoria( $1, N_d$ )  $\in \{0, 1\}$ ;
31       $F_s \leftarrow$  Coste( $X_s$ );
32      contador $_s \leftarrow 0$ ;
33    end
34  end
35   $F \in \mathbb{R}^{N_s \times 1} \leftarrow$  EvaluarCostePoblación( $X$ );
36  if  $\min(F) < \text{Coste}(g)$  then
37     $g \leftarrow \arg \min_X(F)$ ;
38  end
39 end
40 return  $c \leftarrow g$ ;
```

---

convierte las posiciones continuas  $y_i$  en posiciones binarias de acuerdo a la siguiente transformación:

$$y_i = \text{round}(|y_i \bmod 2|) \bmod 2. \quad (5)$$

### 3.4. Binary Ant System

El sistema binario de hormigas (BAS, *Binary Ant System*) es un algoritmo desarrollado por Kong et al. (2008) basado en

la optimización por colonia de hormigas (ACO, *Ant Colony Optimization*), que está orientado a resolver problemas de optimización binarios. Tanto BAS como ACO están inspirados por la comunicación indirecta que presentan las hormigas a la hora de buscar comida. Éstas inicialmente se mueven de manera aleatoria y, una vez encuentran comida, regresan a la colonia dejando un rastro de feromonas. Si otras hormigas encuentran dicho rastro, es probable que lo sigan y lo refuercen. Asimismo, las feromonas se evaporan con el paso del tiempo, permitiendo así el abandono de los caminos con poca comida o mayor recorrido y la convergencia hacia una solución óptima (Dorigo y Di Caro, 1999; Dorigo y Stützle, 2004).

El pseudocódigo 4 describe el algoritmo utilizado. Una vez inicializado cada elemento de las matrices de feromonas  $T^{(j)} = [\tau_1^{(j)}, \tau_2^{(j)}, \dots, \tau_{N_a}^{(j)}]^T \in \mathbb{R}^{N_a \times N_a}$ ,  $j = \{0, 1\}$  a la constante de inicialización  $\tau$ , con  $N_a$  siendo el número de hormigas de la colonia, se calcula la posición de cada hormiga. La probabilidad de que una hormiga  $i$  seleccione un conjunto de canales se muestra a continuación:

$$p_i^{(1)} = \frac{\tau_i^{(1)}}{\tau_i^{(0)} + \tau_i^{(1)}}, \quad \text{con } i = 1, 2, \dots, N_a, \quad (6)$$

por tanto, la probabilidad de rechazarlos será  $p_i^{(0)} = 1 - p_i^{(1)}$ . Posteriormente, se actualiza el factor de convergencia y se evalúa la población, actualizando: (1) la mejor solución encontrada en la iteración,  $s_{ib}$ ; (2) la mejor solución global encontrada,  $s_{gb}$ ; y (3) la mejor solución encontrada por cada hormiga,  $S_{rb}$ . Finalmente, se actualizan las feromonas en dos pasos: (i) evaporación (7,1), e (ii) intensificación (7,2).

$$\tau_i^{(j)} = \underbrace{(1 - \rho)\tau_i^{(j)}}_{(7,1)} + \underbrace{\rho(w_{ib}s_{ib} + w_{gb}s_{gb} + w_{rb}S_{rb})}_{(7,2)}, \quad (7)$$

donde  $\rho \in [0, 1]$  es la constante de evaporización; y  $w_{ib}$ ,  $w_{gb}$  y  $w_{rb}$  son los pesos que ponderan la experiencia local, global y personal de las hormigas, respectivamente, establecidos en Kong et al. (2008) según el valor de los límites del factor de convergencia  $c_{lim}$ .

### 3.5. Firefly Algorithm

El algoritmo de luciérnagas (FA, *Firefly Algorithm*) es otro método de SI desarrollado originalmente por (Yang, 2009), basado en el comportamiento que muestran las luciérnagas a la hora de comunicarse entre ellas y buscar presas potenciales. Las luciérnagas macho iluminan su abdomen con patrones rítmicos mediante un mecanismo llamado bioluminiscencia. Si la hembra se muestra interesada por el macho, imita su patrón para ir a su encuentro.

FA se basa en tres reglas: (1) todas las luciérnagas se atraen entre sí; (2) dicha atracción es proporcional a la intensidad de la iluminación, que disminuye a medida que aumenta la distancia; y (3) el brillo de las iluminaciones se corresponde con la función de coste. El algoritmo se detalla en el pseudocódigo 5. Para empezar, se genera aleatoriamente la población de  $N_f$  luciérnagas y se evalúa su coste  $F$  (i.e., brillo). Para cada luciérnaga  $i$ , se acumula el atractivo  $\beta$  que otras luciérnagas  $j$  ejercen sobre

**Pseudocódigo 4:** Método BAS empleado

---

**Entrada:**  $N_d, N_a, \tau, \rho, c f_{lims}$   
**Salida :**  $c$

- 1  $T^{(0)}, T^{(1)} \in \mathbb{R}^{N_a \times N_d} \leftarrow \tau;$
- 2  $s_{ib}, s_{gb} \leftarrow \infty(1 \times N_d);$
- 3  $S_{rb} \leftarrow \infty(N_a \times N_d);$
- 4 **while**  $\neg$ FinCondición **do**
- 5      $X \leftarrow$  CalcularPosiciones( $N_a, N_d, T^{(0)}, T^{(1)}$ );
- 6     FactorConvergencia  $\leftarrow \frac{1}{N_d} \sum_{j=1}^{N_d} |T_j^{(0)} - T_j^{(1)}|;$
- 7      $F \in \mathbb{R}^{N_a \times 1} \leftarrow$  EvaluarCostePoblación( $X$ );
- 8      $s_{ib} \leftarrow \arg \min_X(F);$
- 9     **if**  $\min(F) < \text{Coste}(s_{gb})$  **then**
- 10          $s_{gb} \leftarrow s_{ib};$
- 11     **end**
- 12     **for** ant = 1 **to**  $N_a$  **do**
- 13         **if**  $\text{Coste}(F_{ant}) < \text{Coste}(S_{rbant})$  **then**
- 14              $S_{rbant} \leftarrow F_{ant};$
- 15         **end**
- 16     **end**
- 17      $T^{(0)} \leftarrow$  Feromonas( $T^{(0)}, \rho, c f_{lims}, s_{ib}, s_{gb}, S_{rb}$ );
- 18      $T^{(1)} \leftarrow$  Feromonas( $T^{(1)}, \rho, c f_{lims}, s_{ib}, s_{gb}, S_{rb}$ );
- 19 **end**
- 20 **return**  $c \leftarrow s_{gb};$

---

**Pseudocódigo 5:** Método FA binario empleado

---

**Entrada:**  $N_d, N_f, \beta_0, \gamma, \delta, \mu_0, S, L$   
**Salida :**  $c$

- 1  $X \leftarrow$  PosiciónAleatoria( $N_f, N_d$ )  $\in \{0, 1\};$
- 2 **while**  $\neg$ FinCondición **do**
- 3      $F \in \mathbb{R}^{N_f \times 1} \leftarrow$  EvaluarCostePoblación( $X$ );
- 4      $g \leftarrow \arg \min_X(F);$
- 5     **foreach** firefly $_i$  **do**
- 6          $\beta, mejora \leftarrow 0;$
- 7         **for**  $j = 1$  **to**  $N_f$  **with** ( $i \neq j$ ) **do**
- 8             **if**  $\text{Coste}(F_j) < \text{Coste}(\text{firefly}_i)$  **then**
- 9                  $\beta \leftarrow \beta + \beta_0 e^{-\gamma r_{ij}^\delta};$
- 10             **else**
- 11                  $mejora \leftarrow mejora + 1;$
- 12             **end**
- 13         **end**
- 14         **if**  $mejora \neq N_f$  **then**
- 15              $X_i \leftarrow X_i + \beta + \mu_0[\text{rand}() - 0,5];$
- 16              $X_i \leftarrow$  ActualizarPosición( $S, L, X_i$ );
- 17         **else**
- 18              $X_i \leftarrow$  PosiciónAleatoria( $1, N_d$ )  $\in \{0, 1\};$
- 19         **end**
- 20     **end**
- 21 **end**
- 22 **return**  $c \leftarrow g;$

---

ella. Si ninguna de las luciérnagas tiene mejor aptitud que ésta, se ignora el atractivo y se calcula una nueva posición aleatoria para  $i$ . El atractivo se computa como sigue:

$$\beta = \beta_0 e^{-\gamma r_{ij}^\delta}, \text{ con } \delta \geq 1, \quad (8)$$

donde  $\beta_0$  es la atracción inicial,  $\gamma$  y  $\delta$  son constantes que caracterizan el comportamiento exponencial de la atracción, y  $r_{ij}$  es la distancia entre las luciérnagas  $i$  y  $j$ . Posteriormente, se suma la atracción a la posición anterior y se le añade una componente aleatoria  $\mu \sim \mu_0 U(-0,5, 0,5)$  (Yang, 2009; Yang et al., 2013).

Dado que el algoritmo FA original esta orientado a un espacio continuo, se ha discretizado la actualización de la posición mediante la función sigmoide, tal y como se mostró en la expresión (3) (Yang et al., 2013). Debido a la naturaleza binaria del problema que nos atañe, una medida de distancia euclidiana solamente sería capaz de devolver distancias  $r \in \{0, 1\}$ . Por esta razón, se ha empleado el enfoque de Bhattacharjee y Sarmah (2015), que las sustituye por distancias de Hamming. Es decir, lo que realmente mide la distancia  $r_{i,j}$  es el porcentaje de bits que difieren entre los vectores binarios  $i$  y  $j$  (Bhattacharjee y Sarmah, 2015).

### 3.6. Backward Elimination

Con el fin enriquecer la comparativa de los algoritmos de inteligencia de enjambre, se ha implementado un método de eliminación recursiva de canales, comúnmente conocido con el nombre de *backward elimination* (BE) (Jobson, 1991). Inicialmente, el conjunto seleccionado se compone de todos los canales posibles. A continuación, se evalúa la precisión al quitar cada uno de los canales del conjunto de forma recursiva y se van

eliminando aquellos cuya relevancia sea menor que el resto. Se entiende como canal irrelevante aquel que apenas provoca una disminución de la precisión final cuando no se utiliza. En este caso, hemos eliminado dos canales en cada iteración del algoritmo (Cecotti et al., 2011). Una vez se han eliminado todos los canales, se determina la importancia de cada uno de ellos en base a su historial de eliminación: cuanto más tarde se haya eliminado, más importante se considera.

### 3.7. Función de coste multi-objetivo

Consideramos que un algoritmo de selección de canales no solamente debe (1) maximizar la precisión de la clasificación, sino también (2) minimizar el número de canales a emplear en la medida de lo posible. Por tanto, en este estudio proponemos la utilización de una función de coste multi-objetivo, que trata de optimizar ambos simultáneamente (Konak et al., 2006; Kee et al., 2015). Hasta donde tenemos conocimiento, los estudios realizados con anterioridad solamente consideran como objetivo la minimización del error en clasificación (Perseh y Sharafat, 2012; Gonzalez et al., 2013).

Para estimar la precisión hemos utilizado un algoritmo de validación cruzada con  $K = 5$  iteraciones sobre los datos de entrenamiento. Es decir, se subdividen los datos de entrenamiento en 5 subconjuntos del mismo tamaño. Se entrena el clasificador LDA con uno de los subconjuntos, y se testea con el resto. Se repite este procedimiento con el resto de subconjuntos, retornando así 5 precisiones distintas (i.e., porcentaje de caracteres correctamente clasificados). Finalmente, se calcula la precisión final  $f_\alpha$  haciendo un promedio de todas ellas.

	Método	Orig.	8-set	BE	PSO	BA	ABC	BAS	FA
A	Precisión (%)	96	85	87	95	96	94	96	94
	Canales	64	8	7	25	29	25	32	28
B	Precisión (%)	92	89	85	92	96	93	92	94
	Canales	64	8	7	17	19	24	19	24
Media	Precisión (%)	<b>94,0</b>	<b>87,0</b>	<b>86,0</b>	<b>93,5</b>	<b>96,0</b>	<b>93,5</b>	<b>94,0</b>	<b>94,0</b>
	Canales	<b>64,0</b>	<b>8,0</b>	<b>7,0</b>	<b>21,0</b>	<b>24,0</b>	<b>24,5</b>	<b>25,5</b>	<b>26,0</b>

Tabla 1: Mejores precisiones alcanzadas por cada método en fase de test.

La función de aptitud que evalúe la cantidad de canales empleados tiene que ser nula cuando solamente se emplea un canal (i.e., caso óptimo) y máxima cuando se empleen los  $N_d$  canales disponibles. Dado que un número muy bajo de canales suele estar asociado a grandes pérdidas de precisión, consideramos que una función polinómica, que favorezca en mayor medida la disminución de canales cuando la cantidad empleada es alta, es más adecuada para el problema que atañe a este estudio.

Por tanto, la función de coste multi-objetivo es la siguiente:

$$F(\mathbf{c}) = \omega_\alpha(1 - f_\alpha) + \omega_c \left( \frac{\sum \mathbf{c} - 1}{N_d - 1} \right)^3, \quad \text{con } \omega_\alpha + \omega_c = 1, \quad (9)$$

donde  $\omega_\alpha$  y  $\omega_c$  son constantes que ponderan la importancia de cada uno de los objetivos.

#### 4. Resultados

Los parámetros empleados en cada uno de los algoritmos se han establecido teniendo en cuenta las recomendaciones de sus publicaciones de referencia, las dimensiones del problema, su naturaleza binaria y las limitaciones computacionales del sistema que los ha ejecutado, siendo éste un Intel Core i7-4770 CPU @ 3.40 GHz, con 16 GB de RAM. Dado que la dimensionalidad del problema es de  $N_d = 64$ , se han empleado un total de 20 agentes (i.e., partículas, abejas, hormigas y luciérnagas, respectivamente) en PSO, ABC, BAS y FA; y un total de 17 en BA (Kennedy y Eberhart, 1997). Las constantes que determinan la ponderación de la función multi-objetivo se han fijado en  $\omega_\alpha = 0,7$  y  $\omega_c = 0,3$ , asumiendo que la importancia de asegurar una precisión alta es mayor que la de reducir drásticamente el número de canales. El umbral de la función sigmoide se ha fijado en  $L = 0,5$  tanto en PSO como en FA, dado que se desea que la probabilidad de seleccionar o rechazar un canal sea la misma. Asimismo, para facilitar la comparación entre los métodos, se han ejecutado 100 generaciones en todos ellos. Los parámetros específicos utilizados en cada método se pueden consultar en el Apéndice 1.

La convergencia promedio de todas las repeticiones de cada uno de los métodos para ambos sujetos se detalla en la Figura 2. Adicionalmente, se muestran los boxplots que recogen las precisiones obtenidas en la fase de test para cada una de las soluciones. Asimismo, las precisiones alcanzadas más altas se muestran en la Tabla 1, donde se observa que no sólo se igualan los resultados empleando menos de la mitad de los canales, sino que se mejoran en algunos casos. También se incluyen las precisiones obtenidas empleando los 8 canales típicos para los sistemas P300-BCI, indicados originalmente por Krusienski et al.

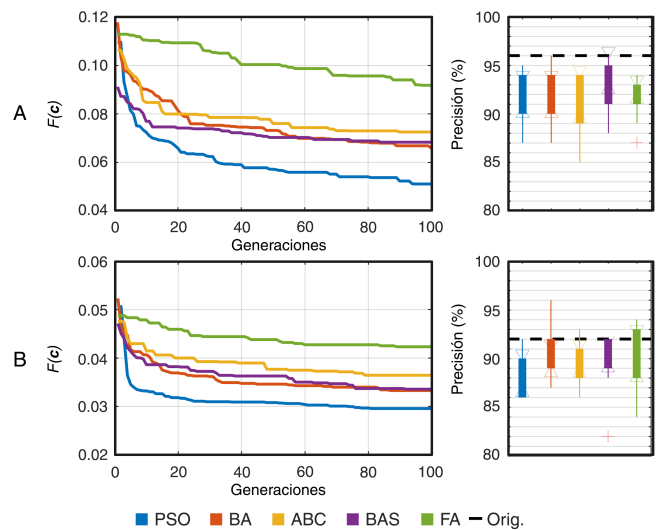


Figura 2: (Izquierda) Convergencia de la función de coste del promedio de las repeticiones de cada método a través de las generaciones. (Derecha) Boxplots de la precisiones alcanzadas en la fase de test para cada iteración comparadas con la precisión obtenida utilizando los 64 canales (línea negra discontinua).

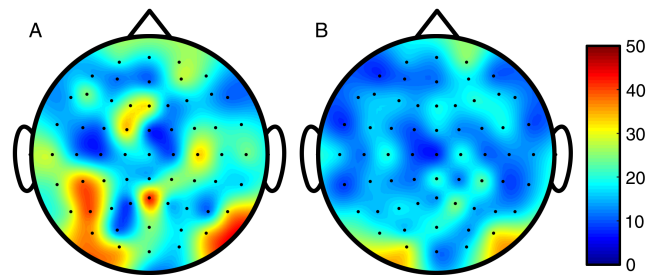


Figura 3: Acumulación de los canales seleccionados en las 10 repeticiones de cada método para el sujeto A (izquierda) y para el sujeto B (derecha).

(2008), en las posiciones Fz, Cz, Pz, P3, P4, P07, PO8 y Oz, de acuerdo al Sistema Internacional 10–20 (Salvaris y Sepulveda, 2009; Martínez-Cagigal et al., 2016).

La Figura 3 muestra la cantidad de veces que se ha seleccionado cada canal en las repeticiones de los métodos. Se observa una clara acumulación en ciertos canales de la zona parietal, sobre todo en el sujeto A, lo cual indica que poseen una alta relevancia a la hora de caracterizar los P300 y favorecer la clasificación.

En la Figura 4 se ilustra una comparativa entre los resultados alcanzados por los métodos de inteligencia de enjambre, el conjunto de los 8 canales típicos y el método de eliminación recursiva de canales. Tal y como se puede observar, ninguno es capaz de mejorar las precisiones alcanzadas por los 5 métodos de inteligencia de enjambre empleados.

Finalmente, la Figura 5 muestra las mejores soluciones encontradas por cada método en función del número de secuencias empleado y el número de canales seleccionados. Tal y como se puede observar, en general, la disminución drástica del número de canales no compromete la precisión obtenida, independientemente del número de secuencias empleado.



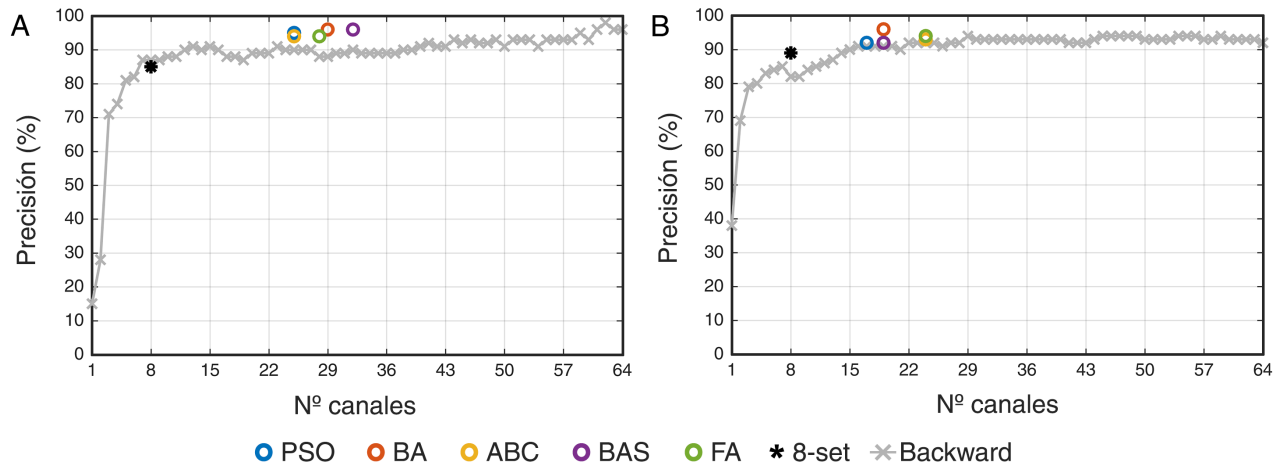


Figura 4: Comparación de las precisiones obtenidas en fase de test empleando la mejor solución alcanzada por cada método en función del número de canales seleccionados con respecto al set de 8 canales de Krusienski et al. (2008) (negro) y el método de eliminación recursiva (gris).

## 5. Discusión

En la Figura 2 se observa una clara convergencia de la función de coste multi-objetivo para todos los métodos, la cual se muestra más pronunciada para los casos de PSO y BA; seguida por ABC y BAS; y, finalmente, por FA. La leve variación que experimentan las curvas de convergencia para los algoritmos ABC, BAS y FA indica que, o bien (i) se encuentra una solución óptima en pocas generaciones, o (ii) la variación de las posiciones de los agentes está más influenciada por el comportamiento del enjambre que en los casos de PSO y BA.

En PSO se establece un compromiso entre la confianza individual de cada partícula y la confianza del enjambre. Típicamente se ponderan con el mismo valor (i.e.,  $\alpha_1 = \alpha_2 = 2$ ), favoreciendo una búsqueda local y global simultáneamente con la misma probabilidad (Kennedy y Eberhart, 1995; Brownlee, 2011). En el caso de BA, la aplicación del operador de mutación propuesto favorece enormemente la búsqueda global en las primeras generaciones, que se ve disminuido por un factor  $\lambda$  a medida que avanza el algoritmo (Brownlee, 2011), tal y como se ha reflejado en la Figura 2. De esta manera, el objetivo de las últimas generaciones es realizar una búsqueda local. En contraposición a PSO y ABC, a pesar de que se ha establecido un límite de abandono relativamente bajo (i.e.,  $\varphi = 3$ ), el algoritmo ABC se centra principalmente en explotar las vecindades de las soluciones propuestas (Kiran, 2015), generando posiciones aleatorias con menos frecuencia que los algoritmos PSO y BA. Una manera de aumentar la búsqueda global del algoritmo sería asegurar la presencia de una abeja exploradora que ignore los lugares de explotación (Karaboga, 2005). Además de la baja importancia que se asigna a la búsqueda global en ABC, podemos observar cómo su evolución ha convergido a la solución óptima en pocas iteraciones. De manera similar, las luciérnagas del algoritmo FA se atraen entre ellas, favoreciendo una búsqueda local. La búsqueda global se controla mediante el peso de la componente aleatoria  $\mu_0$  y mediante elitismo, siendo la mejor solución la única que puede reinicializarse aleatoria-

mente (Yang, 2009). Finalmente, cabe destacar que la estrategia del algoritmo BAS difiere ligeramente del resto, puesto que varía progresivamente la confianza individual, local y global de las hormigas de acuerdo al nivel de consolidación de una solución a lo largo del tiempo, es decir, en función del factor de convergencia (Kong et al., 2008). El algoritmo adapta el tipo de búsqueda según el estado de la evolución, mejorando la función de coste en pequeños pasos que siguen una tendencia lineal, a diferencia de PSO y BA (ver Figura 2). Por lo tanto, la selección de un método u otro depende de las características del problema a optimización a resolver, según la estrategia de búsqueda que se desee seguir en cada generación.

En general, la mejor solución en fase de test no es necesariamente la que produce un coste  $F(c)$  menor. Aún así, cabe destacar que la variación de la precisión en test para las 10 repeticiones de cada algoritmo no es muy alta, produciendo una desviación estándar del 2–3 %. Este comportamiento es natural en los problemas de aprendizaje automático, dado que los conjuntos de entrenamiento y test no son similares y, por tanto, un buen rendimiento en uno de ellos no necesariamente asegura una buena clasificación en el restante (Witten y Frank, 2011).

La convergencia de los algoritmos basados en SI depende en gran medida de los parámetros utilizados. Aunque existen ciertas directrices y recomendaciones generales a la hora de fijarlos (Brownlee, 2011), su optimización depende de la experiencia del usuario y de las características del problema a resolver. En este aspecto, algoritmos como ABC o PSO en sus formas más básicas, que solamente emplean 2 o 3 parámetros, podrían ser optimizados mediante técnicas de validación. Sin embargo, la cantidad de parámetros que requieren BA, BAS y FA hace que su optimización exhaustiva sea demasiado costosa y, por tanto, que sea común fijarlos de manera empírica o basándose en ciertas directrices de forma teórica (Yang et al., 2013).

No obstante, tal y como muestra la Tabla 1, la convergencia no supone un problema en la solución final. Todos los métodos seleccionan una media de  $27,80 \pm 2,95$  canales para el sujeto A y de  $20,60 \pm 3,21$  canales para el sujeto B sin comprometer la



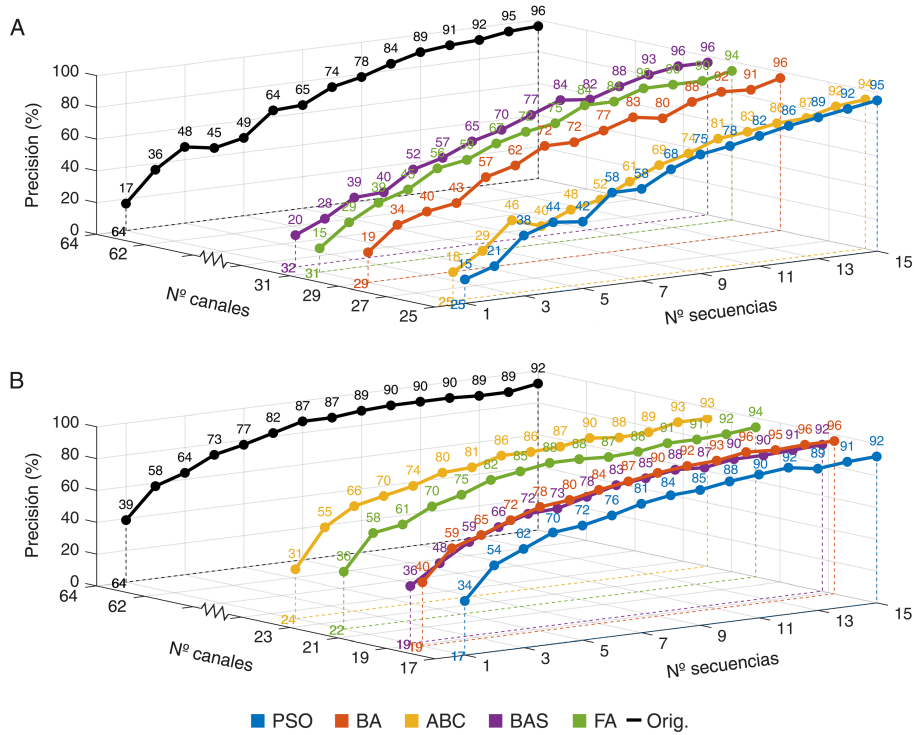


Figura 5: Precisiones obtenidas en la fase de test empleando la mejor solución alcanzada por cada método en función del número de secuencias y del número de canales seleccionados.

precisión final obtenida. El sujeto A alcanza un 96 % empleando los 64 canales, precisión que es igualada por las soluciones propuestas por los algoritmos BA y BAS empleando menos de la mitad de los canales disponibles. En este caso, las soluciones de PSO, ABC y FA comprometen ligeramente la precisión, obteniendo un máximo de 95 %, 94 % y 94 %, respectivamente. Para el sujeto B se obtiene una precisión del 92 % empleando los 64 canales disponibles, igualada por las soluciones de PSO y BAS, y superada por el resto. De hecho, BA obtiene una precisión del 96 %, seguido por FA con 94 % y ABC con 93 %. También es de utilidad comparar los resultados con los obtenidos empleando la selección típica de canales en sistemas P300-BCI, originalmente propuesta por Krusienski et al. (2008), consistente en el set de 8 canales: Fz, Cz, Pz, P3, P4, PO7, PO8 y Oz. En la Tabla 1 se puede comprobar que se alcanzan precisiones del 85 % y 89 % para los sujetos A y B, respectivamente. Asimismo, dado que BE propone una solución para cada número de canales posible, es necesario determinar un único subconjunto de canales. En este caso, hemos indicado como óptimo el punto más cercano a (1,100), correspondiente a seleccionar un único canal y obtener una precisión del 100 % (ver Figura 4). Dichos subconjuntos han seleccionado 7 canales para ambos sujetos, alcanzando precisiones del 87 % para el sujeto A y del 85 % para el sujeto B. Es decir, el rendimiento del sistema se ve disminuido drásticamente en comparación con el obtenido para cualquiera de los métodos SI empleados. Estos resultados ponen de manifiesto que todos los métodos empleados son útiles a la hora de seleccionar los canales más relevantes para la clasificación en sistemas P300-BCI y son capaces, no sólo de igualar

la precisión obtenida sin realizar una selección de canales, sino de superarla en algunos casos.

Tal y como muestra la Figura 3, los métodos de SI empleados seleccionan canales de manera recurrente en prácticamente todas sus repeticiones. Se observa una acumulación en la zona parietal y occipital del córtex para ambos usuarios, especialmente para el sujeto A. Esta localización de los canales más relevantes, por tanto, reafirma el estudio de (Krusiensi et al., 2008), que sostiene que los potenciales evocados P300 se producen en la zona parietal del córtex y recomienda una distribución de canales repartidos por la zona parietal y occipital, comúnmente empleada desde entonces en los estudios que involucran el uso de sistemas P300-BCI.

La Figura 4 pone de manifiesto que el método de eliminación recursiva de canales no es capaz de mejorar las soluciones encontradas por los métodos de inteligencia de enjambre empleados. Tampoco lo hace la solución de 8 canales propuesta por Krusienski et al. (2008), tal y como se ha discutido anteriormente. No obstante, a diferencia de PSO, BA, ABC, BAS y FA, el algoritmo BE propone una solución para cada número de canales posible. De esta manera, el usuario puede elegir el número de canales que más le convenga, lo cual, aunque haya alcanzado precisiones inferiores a los métodos de enjambre, puede ser útil en ciertos casos.

Como se puede observar en la Figura 5, no sólo se obtienen precisiones altas en comparación a las originales para 15 secuencias, sino que las obtenidas para un número de secuencias inferior muestran el mismo comportamiento. Se observan precisiones similares a las obtenidas para 64 canales con todos los

métodos de SI independientemente del número de secuencias utilizado. Asimismo, la variación de las precisiones para 1–15 secuencias es similar a la original incluso requiriendo menos de la mitad de los canales disponibles.

De hecho, PSO es el método que ha seleccionado menos canales. No obstante, no ha sido capaz de superar las precisiones originales, a diferencia del resto de métodos, probablemente debido al compromiso canales-precisión que se refleja en la función de coste de la expresión (9). En este aspecto, ABC, BAS y FA sí han conseguido superar la precisión original en uno de los dos sujetos. No obstante, BA ha superado las precisiones originales para 15 secuencias en ambos, seleccionando 29 y 19 canales para los sujetos A y B, respectivamente. Teniendo en cuenta la convergencia del método, las precisiones obtenidas, y el número de canales seleccionado, se puede afirmar que el algoritmo BA modificado ha alcanzado los mejores resultados en la presente comparativa.

A pesar del buen rendimiento que demuestran los métodos de SI a la hora de resolver complejos problemas de optimización (Yang et al., 2013), poseen ciertos inconvenientes. Tal y como se ha mencionado anteriormente, aunque existen excepciones, la mayor parte de los métodos de SI dependen de un gran número de parámetros cuya optimización recae principalmente en la experiencia del usuario (Yang et al., 2013). Adicionalmente, la convergencia de los algoritmos más conocidos, tales como PSO, ACO o ABC, se ha demostrado matemáticamente (Clerc y Kennedy, 2002; Yang et al., 2013; Dorigo y Stützle, 2004; Krüger et al., 2016) con anterioridad; pero aún existe una gran brecha entre el aspecto teórico y práctico de los algoritmos más recientes y de sus múltiples variaciones. También cabe destacar que las metaheurísticas no garantizan una solución óptima global, sino que buscan una solución adecuada en un tiempo razonable que, normalmente, es un óptimo local (Yang, 2014). Asimismo, dicha solución no se halla de forma inmediata, sino que requiere una evolución del algoritmo sobre un proceso iterativo, de una manera similar a los GA (Kennedy et al., 2001; Yang et al., 2013; Yang, 2014).

Sin embargo, el éxito de los métodos basados en SI en la optimización compleja radica en las siguientes características: (1) estocasticidad, (2) auto-organización, (3) flexibilidad, (4) robustez y (5) escalabilidad. Para empezar, su naturaleza estocástica favorece su rendimiento en comparación con los algoritmos deterministas clásicos basados en gradiente cuando la función a optimizar es multi-modal y posee discontinuidades, puesto que actúa sobre los valores de la función, sin necesidad de derivarla previamente (Yang, 2014). Asimismo, el flujo de información de los agentes que forman parte del sistema causan que éste tienda constantemente a auto-organizarse, evitando así la necesidad de supervisión constante y evolucionando hacia una solución más óptima de manera iterativa (Kennedy et al., 2001). Como consecuencia de dicha auto-organización, el sistema es flexible, adaptándose rápidamente al problema en un ámbito cuyas condiciones son variables (Bonabeau et al., 1999; Yang et al., 2013). Además, dado que la dependencia del sistema en los agentes es pequeña, el fallo de uno de los individuos no tiene impacto alguno en el rendimiento global del sistema, asegurando su robustez (Brownlee, 2011; Yang et al.,

2013). Finalmente, cabe destacar que, con una implementación adecuada, los algoritmos basados en SI son escalables, puesto que la misma arquitectura puede aplicarse tanto a unos pocos agentes como a miles de ellos (Brownlee, 2011; Yang et al., 2013).

Para finalizar, conviene mencionar que el proceso de selección de canales en este artículo se ha enfocado como un método *wrapper* (Guyon y Elisseeff, 2003). Es decir, se evalúa la utilidad de una posible solución en función del resultado del clasificador, tratando a éste como una caja negra. Dado que la selección depende del clasificador empleado, se asegura un buen rendimiento del sistema en la fase de test. No obstante, los procesos de validación cruzada, entrenamiento y test con el clasificador LDA son necesarios cada vez que se evalúa una solución, provocando un alto coste computacional (Guyon y Elisseeff, 2003).

Cabe destacar que todos los estudios realizados con anterioridad han empleado PSO, lo cual reafirma el hecho de que este tipo de metaheurísticas no han sido suficientemente explotadas por el momento. Jin et al. (2010) emplearon un enfoque *wrapper* que integraba PSO y BLDA sobre una base de datos propia. Dado que el objetivo del estudio estaba enfocado en crear un “P300 Speller” con caracteres chinos, la selección de canales no aparece detallada. No obstante, sí se indica que los subconjuntos seleccionados por PSO mejoraron el rendimiento de los conjuntos de 8 y 16 canales tradicionales. Asimismo, Perseh y Sharafat (2012) consiguieron alcanzar una media del 97,5 % de precisión en la base de datos de la “III BCI Competition 2005” mediante una combinación de transformada wavelet discreta (DWT, *Discrete Wavelet Transform*) y una versión modificada de PSO con distancias de Bhattacharyya. Para terminar, Gonzalez et al. (2013) fueron capaces de alcanzar una media del 67,5 % de precisión en el mismo dataset con solo 5 secuencias usando su método WASP, compuesto por: lifting DWT, PSO y SVM. Todos estos estudios han empleado un enfoque *wrapper* que combina PSO con sofisticados métodos de extracción y clasificación de características. Sin embargo, en este estudio hemos tratado de simplificar ambas etapas con soluciones del estado del arte: sub-muestreo como extracción, y LDA como clasificación; con el objetivo de enfocar el estudio en los métodos de selección de canales propuestos. Basándonos en los resultados obtenidos, podemos especular que la combinación de cualquiera de éstos con métodos de extracción y clasificación más sofisticados podría alcanzar precisiones más altas.

A pesar de que los resultados del estudio sugieren que todos los métodos de SI empleados son adecuados para su aplicación en la selección de canales de los sistemas P300-BCI, podemos destacar varias limitaciones. Para empezar, a excepción de PSO y ABC, el resto de métodos empleados requieren fijar una alta cantidad de parámetros, lo cual hace intratable computacionalmente su optimización mediante procesos de validación (Yang et al., 2013). Aunque existen ciertas directrices generales a la hora de fijarlos (Brownlee, 2011), la decisión depende en gran medida de la experiencia del usuario. Otra limitación radica en la evaluación de la función de coste. Como ya se ha mencionado, los métodos *wrapper* no sólo dependen del clasificador empleado, sino que aumentan considerablemente el coste computacional del sistema (Guyon y Elisseeff, 2003). Para finalizar,

cabe destacar que la base de datos de la “III BCI Competition 2005” solamente posee registros de dos usuarios, limitando la generalización de los resultados del estudio (Blankertz et al., 2006). Para superar estas limitaciones, planteamos las siguientes líneas futuras: (i) optimizar mediante métodos de validación los parámetros de PSO y ABC, tratar de simplificar el resto de algoritmos y buscar nuevos métodos de SI para enriquecer la comparativa; (ii) eliminar la dependencia del clasificador empleando un enfoque *filter* (Guyon y Elisseeff, 2003), tratando de encontrar métricas alternativas que evalúen la capacidad predictiva de cada solución, por ejemplo, las basadas en la SSNR (Cecotti et al., 2011); y (iii) testear los métodos con otras bases de datos.

## 6. Conclusiones

La alta variabilidad inter-sesión y la baja relación señal a ruido que caracterizan a los sistemas BCI basados en potenciales evocados P300 provocan la necesidad de analizar una gran cantidad de datos en tiempo real. La selección de canales se revela fundamental para evitar el sobre-entrenamiento del clasificador, reducir la carga computacional y, en algunos casos, mejorar el rendimiento del sistema. En este aspecto, las metaheurísticas basadas en sistemas biológicos, que han demostrado una gran capacidad para resolver complejos problemas de optimización, no han sido aplicadas en este tipo de problemas. En este estudio, por tanto, se han aplicado cinco métodos de enjambre (PSO, BA, ABC, BAS y FA) a la selección de canales en sistemas P300-BCI, con el fin de determinar su utilidad y su viabilidad. Dichos métodos se han evaluado con la base de datos de la “III BCI Competition 2005”. Los resultados indican que los cinco métodos reducen el número de canales a menos de la mitad de los disponibles sin comprometer la precisión del sistema. La media de precisión utilizando los 64 canales originales se sitúa en el 94 %, superada por BA (96 %, 24 canales), e igualada por FA (94 %, 26 canales) y BAS (94 %, 25,5 canales). Aunque todos ellos han demostrado ser adecuados para su aplicación en la selección de canales de los sistemas P300-BCI, dada su rápida convergencia y la alta precisión que ha alcanzado, se concluye que el algoritmo BA ha arrojado los mejores resultados de la comparativa.

## English Summary

### P300-Based Brain-Computer Interface Channel Selection using Swarm Intelligence

#### Abstract

Brain-Computer Interfaces (BCI) are direct communication pathways between the brain and the environment that translate certain features, which correspond to users' intentions, into device control commands. Channel selection in BCI systems is essential to avoid over-fitting, to reduce the computational cost and to increase the users' comfort. Although several algorithms have previously developed for that purpose, metaheuristics based on swarm intelligence have not been exploited yet in

P300-based BCI systems. In this study, a comparative among five different swarm methods, based on the behavior of biological systems, is shown. Those methods have been applied in order to optimize the channel selection procedure in this kind of systems, and have been tested with the 'III BCI Competition 2005' database II. Results show that the five methods can achieve similar or even higher accuracies than that obtained without performing any channel selection procedure. Owing to the fact that all the applied methods are able to drastically reduce the required number of channels without compromising the system performance, as well as to overcome the common 8-channel set and the backward elimination algorithm, we conclude that all of them are suitable for use in the P300-BCI systems channel selection procedure.

#### Keywords:

Human-machine interface, machine learning, biomedical systems, optimization problems, electrodes, communication systems.

## Agradecimientos

Este estudio se ha financiado parcialmente mediante el proyecto TEC2014-53196-R del Ministerio de Economía y Competitividad (MINECO) y FEDER, y el proyecto VA037U16 de la Consejería de Educación de la Junta de Castilla y León. V. Martínez-Cagigal se encuentra financiado por un contrato de “Promoción de Empleo Joven e Implantación de la Garantía Juvenil” del MINECO y la Universidad de Valladolid.

## Referencias

- Bhattacharjee, K. K., Sarmah, S. P., 2015. A binary firefly algorithm for knapsack problems. En: 2015 Int. Conf. Ind. Eng. Eng. Manag. pp. 73–77. DOI: 10.1109/IEEM.2015.7385611
- Blankertz, B., Müller, K.-R., Krusienski, D. J., Schalk, G., Wolpaw, J. R., Schlögl, A., Pfurtscheller, G., Millán, J. D. R., Schröder, M., Birbaumer, N., 2006. The BCI competition III: Validating alternative approaches to actual BCI problems. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14 (2), 153–159. DOI: 10.1109/TNSRE.2006.875642
- Bonabeau, E., Dorigo, M., Theraulaz, G., 1999. *Swarm intelligence: from natural to artificial systems*. Oxford University Press. DOI: 10.1007/s13398-014-0173-7.2
- Brownlee, J., 2011. *Clever Algorithms: Nature-Inspired Programming Recipes*, 2nd Edition. DOI: 10.1017/CB09781107415324.004
- Cecotti, H., Rivet, B., Congedo, M., Jutten, C., Bertrand, O., Maby, E., Mattout, J., 2011. A robust sensor-selection method for P300 brain-computer interfaces. *J. Neural Eng.* 8 (1), 016001. DOI: 10.1088/1741-2560/8/1/016001
- Clerc, M., Kennedy, J., 2002. The Particle Swarm–Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. Evol. Comput.* 6 (1), 58–73. DOI: 10.1109/4235.985692
- Colwell, K. A., Ryan, D. B., Throckmorton, C. S., Sellers, E. W., Collins, L. M., 2014. Channel selection methods for the P300 Speller. *J. Neurosci. Methods* 232, 6–15. DOI: 10.1016/j.jneumeth.2014.04.009
- Dorigo, M., Di Caro, G., 1999. The Ant Colony Optimization Meta-Heuristic. *New Ideas Optim.* 2, 11–32. DOI: 10.1109/CEC.1999.782657
- Dorigo, M., Stützle, T., 2004. *Ant Colony Optimization*. The MIT press.
- Farwell, L. A., Donchin, E., 1988. Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials. *Electroencephalogr. Clin. Neurophysiol.* 70 (6), 510–523. DOI: 10.1016/0013-4694(88)90149-6

- Gonzalez, A., Nambu, I., Hokari, H., Iwahashi, M., Wada, Y., 2013. Towards the classification of single-trial event-related potentials using adapted wavelets and particle swarm optimization. Proc. - 2013 IEEE Int. Conf. Syst. Man, Cybern. SMC 2013, 3089–3094. DOI: 10.1109/SMC.2013.527
- Guyon, I., Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 3 (3), 1157–1182. DOI: 10.1016/j.aca.2011.07.027
- Jin, J., Allison, B. Z., Brunner, C., Wang, B., Wang, X., Zhang, J., Neuper, C., Pfuerscheller, G., 2010. P300 Chinese input system based on Bayesian LDA. Biomed. Tech. 55 (1), 5–18. DOI: 10.1515/BMT.2010.003
- Jobson, J. D., 1991. Applied multivariate data analysis. Volume I: Regression and Experimental Design, 4th Edition. Vol. 1. Springer.
- Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. Tech. rep., Erciyes University.
- Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N., 2014. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. Artif. Intell. Rev. 42 (1), 21–57. DOI: 10.1007/s10462-012-9328-0
- Kee, C.-Y., Ponnambalam, S., Loo, C.-K., 2015. Multi-objective genetic algorithm as channel selection method for P300 and motor imagery data set. Neurocomputing 161, 120–131. DOI: 10.1016/j.neucom.2015.02.057
- Kennedy, J., Eberhart, R., 1995. Particle swarm optimization. Neural Networks, 1995. Proceedings., IEEE Int. Conf. 4, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968
- Kennedy, J., Eberhart, R., 1997. A Discrete Binary Version of the Particle Swarm Algorithm. 1997 IEEE Int. Conf. Syst. Man, Cybern. Comput. Cybern. Simul. 5, 4–8. DOI: 10.1109/ICSMC.1997.637339
- Kennedy, J., Eberhart, R. C., Shi, Y., 2001. Swarm Intelligence. Vol. 2. Academic Press. DOI: 10.4249/scholarpedia.1462
- Kiran, M. S., 2015. The continuous artificial bee colony algorithm for binary optimization. Appl. Soft Comput. J. 33, 15–23. DOI: 10.1016/j.asoc.2015.04.007
- Konak, A., Coit, D. W., Smith, A. E., 2006. Multi-objective optimization using genetic algorithms: A tutorial. Reliab. Eng. Syst. Saf. 91 (9), 992–1007. DOI: 10.1016/j.ress.2005.11.018
- Kong, M., Tian, P., Kao, Y., 2008. A new ant colony optimization algorithm for the multidimensional Knapsack problem. Comput. Oper. Res. 35 (8), 2672–2683. DOI: 10.1016/j.cor.2006.12.029
- Krüger, T. J., Davidović, T., Teodorović, D., Šelmić, M., 2016. The bee colony optimization algorithm and its convergence. Int. J. Bio-Inspired Comput. 8 (5), 340–354.
- Krusiński, D., Sellers, E., McFarland, D., Vaughan, T., Wolpaw, J., 2008. Toward enhanced P300 speller performance. J. Neurosci. Methods 167 (1), 15–21. DOI: 10.1016/j.jneumeth.2007.07.017
- Kübler, A., Birbaumer, N., 2008. Brain-computer interfaces and communication in paralysis: Extinction of goal directed thinking in completely paralysed patients? Clin. Neurophysiol. 119 (11), 2658–2666. DOI: 10.1016/j.clinph.2008.06.019
- Kübler, A., Nijboer, F., Birbaumer, N., 2007. Brain-Computer Interfaces for communication and motor control – perspectives on clinical application. En: Toward Brain-Computer Interfacing, 1st Edition. MA: The MIT Press, pp. 373–391.
- Martínez-Cagigal, V., Gomez-Pilar, J., Álvarez, D., Hornero, R., 2016. An asynchronous P300-based brain-computer interface web browser for severely disabled people. IEEE Transactions on Neural Systems and Rehabilitation Engineering (Aceptado). DOI: 10.1109/TNSRE.2016.2623381
- Perseh, B., Sharafat, A. R., jun 2012. An Efficient P300-based BCI Using Wavelet Features and IBPSO-based Channel Selection. J. Med. Signals Sens. 2 (3), 128–143.
- Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M., 2006. The Bees Algorithm - A Novel Tool for Complex Optimisation Problems. Intell. Prod. Mach. Syst. - 2nd I\*PROMS Virtual Int. Conf., 454–459. DOI: 10.1016/B978-008045157-2/50081-X
- Rakotomamonjy, A., Guigue, V., 2008. BCI Competition III : Dataset II - Ensemble of SVMs for BCI P300 Speller. IEEE Trans. Biomed. Eng. 55 (3), 1147–1154.
- Rivet, B., Cecotti, H., Maby, E., Mattout, J., 2012. Impact of spatial filters during sensor selection in a visual P300 brain-computer interface. Brain Topogr. 25 (1), 55–63. DOI: 10.1007/s10548-011-0193-y
- Rivet, B., Cecotti, H., Phlypo, R., Bertrand, O., Maby, E., Mattout, J., 2010. EEG sensor selection by sparse spatial filtering in P300 speller Brain-Computer Interface. 2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC'10, 5379–5382. DOI: 10.1109/IEMBS.2010.5626485
- Salvaris, M., Sepulveda, F., 2009. Visual modifications on the p300 speller bci paradigm. Journal of neural engineering 6 (4), 046011.
- Schalk, G., McFarland, D. J., Hinterberger, T., Birbaumer, N., Wolpaw, J. R., 2004. BCI2000: A general-purpose brain-computer interface (BCI) system. IEEE Trans. Biomed. Eng. 51 (6), 1034–1043. DOI: 10.1109/TBME.2004.827072
- Witten, I. H., Frank, E., 2011. Data Mining: Practical Machine Learning Tools and Techniques, 3rd Edition. Morgan Kaufmann.
- Wolpaw, J. R., Birbaumer, N., Heetderks, W. J., McFarland, D. J., Peckham, P. H., Schalk, G., Donchin, E., Quatrano, L. A., Robinson, C. J., Vaughan, T. M., 2000. Brain-computer interface technology: a review of the first international meeting. IEEE Trans. Rehabil. Eng. 8 (2), 164–173. DOI: 10.1109/TRE.2000.847807
- Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfuerscheller, G., Vaughan, T. M., 2002. Brain-computer interfaces for communication and control. Clin. Neurophysiol. 113 (6), 767–91. DOI: 10.1016/S1388-2457(02)00057-3
- Xu, M., Qi, H., Ma, L., Sun, C., Zhang, L., Wan, B., Yin, T., Ming, D., 2013. Channel Selection Based on Phase Measurement in P300-Based Brain-Computer Interface. PLoS One 8 (4), 1–9. DOI: 10.1371/journal.pone.0060608
- Yang, X. S., 2009. Firefly Algorithms for Multimodal Optimization. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 5792 LNCS, 169–178. DOI: 10.1007/978-3-642-04944-6\_14
- Yang, X.-S., 2014. Nature-Inspired Optimization Algorithms, 1st Edition. Elsevier Inc.
- Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., Karamanoglu, M., 2013. Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, 1st Edition. Elsevier Inc. DOI: 10.1016/B978-0-12-405163-8.00020-X
- Yu, T., Yu, Z., Gu, Z., Li, Y., 2015. Grouped Automatic Relevance Determination and Its Application in Channel Selection for P300 BCIs. IEEE Trans. Neural Syst. Rehabil. Eng. 23 (6), 1068–1077. DOI: 10.1109/TNSRE.2015.2413943

## Apéndice A. Parámetros empleados

Los parámetros específicos empleados en cada método de SI incluido en la comparativa se detallan a continuación.

	Parámetros específicos de cada método		Valor
PSO	$N_p$	Nº de partículas	20
	$\alpha_1$	Constante de confianza individual	2
	$\alpha_2$	Constante de confianza grupal	2
BA	$N_b$	Nº de abejas exploradoras	10
	$N_o$	Nº de abejas observadoras	2
	$N_e$	Nº de abejas de élite	5
	$m_s$	Nº de lugares	5
	$m_{es}$	Nº de lugares de élite	1
ABC	$\eta$	Nº de bits de mutación inicial	32
	$\lambda$	Factor de decrecimiento	0,95
	$N_s$	Nº de abejas	20
	$\varphi$	Límite de abandono	3
BAS	$N_a$	Nº de hormigas	20
	$\tau$	Cantidad de feromonas inicial	0,5
	$\rho$	Constante de evaporación	0,3
	$c f_{lims}$	Límites del factor de convergencia	[0,3, 0,5, 0,7, 0,9, 0,95]
FA	$N_f$	Nº de luciérnagas	20
	$\beta_0$	Atracción inicial	1
	$\gamma$	Constante de atracción	0,2
	$\delta$	Constante de distancia	2
	$\mu_0$	Peso de la componente aleatoria	0,2

Tabla A.2: Parámetros específicos empleados en cada método.