

**UNIVERSIDAD DE GRANADA**  
**E.T.S.I. INFORMÁTICA Y**  
**TELECOMUNICACIÓN**



**UNIVERSIDAD  
DE GRANADA**



Departamento de Ciencias de la  
Computación e Inteligencia Artificial

## **Metaheurísticas**

<http://sci2s.ugr.es/graduateCourses/Metaheuristicas>

<https://decsai.ugr.es>

## **Guión de Prácticas**

### **Práctica 2.b:**

**Técnicas de Búsqueda basadas en Poblaciones  
para el Problema del Aprendizaje de Pesos en  
Características**

Curso 2022-2023

Tercer Curso del Grado en Ingeniería Informática

# Práctica 2.b Técnicas de Búsqueda basadas en Poblaciones para el Problema del Aprendizaje de Pesos en Características

## 1. Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de las *Técnicas de Búsqueda basadas en Poblaciones* en la resolución del problema del aprendizaje de pesos en características (APC) descrito en las transparencias del Seminario 2. Para ello, se requerirá que el estudiante adapte las siguientes técnicas metaheurísticas al problema:

- Algoritmos Genéticos: Dos variantes generacionales elitistas (AGGs) y otras dos estacionarias (AGEs), descritas en el Seminario 3. Aparte del esquema de evolución, la única diferencia entre los dos modelos de AGGs y AGE será el operador de cruce empleado.
- Algoritmos Meméticos: Tres variantes de algoritmos meméticos (AMs) basadas en un AGG, descritas en el Seminario 3. La única diferencia entre las tres variantes de AMs serán los parámetros considerados para definir la aplicación de la búsqueda local. Para diseñar los AMs, se utilizará el método de Búsqueda Local (BL) desarrollado en la Práctica 1.b.

El estudiante deberá comparar los resultados obtenidos en una serie de casos del problema con los proporcionados por: el clasificador 1-NN generado considerando todas las características igualmente ponderadas (sin pesos) y con el clasificador 1-NN obtenido empleando los pesos aprendidos por el método *greedy* RELIEF, y con la propia BL. Todos ellos implementados en la Práctica 1.b.

La práctica se evalúa sobre un total de **2,5 puntos**, distribuidos de la siguiente forma:

- AGGs (0,75 puntos) y AGEs (1 punto).
- AMs (0,75 puntos).

La fecha límite de entrega será el **Domingo 21 de Mayo de 2023** antes de las 23:59 horas. La entrega de la práctica se realizará por medio de PRADO.

## 2. Trabajo a Realizar

El estudiante podrá desarrollar los algoritmos de la práctica siguiendo la modalidad que desee: trabajando con cualquiera de los *frameworks* de metaheurísticas estudiados en el Seminario 1, implementándolos a partir del código C proporcionado en la web de la asignatura o considerando cualquier código disponible en Internet.

Los métodos desarrollados serán ejecutados sobre una serie de casos del problema. Se realizará un estudio comparativo de los resultados obtenidos y se analizará el comportamiento de cada algoritmo en base a dichos resultados. **Este análisis influirá decisivamente en la calificación final de la práctica.**

En las secciones siguientes se describen los aspectos relacionados con cada algoritmo a desarrollar y las tablas de resultados a obtener. Los casos del problema serán los mismos que en la Práctica 1.b. De igual manera, el número de ejecuciones a realizar sobre ellos, el procedimiento de validación y los estadísticos de calidad (*Tasa\_clas*, *Tasa\_red*, *Agregado* y *Tiempo*) serán los mismos que en el guión de la Práctica 1.b. (véase la Sección 3 de dicho guión de prácticas).

## 3. Componentes de los Algoritmos

Los algoritmos de esta práctica tienen en común las siguientes componentes:

- *Esquema de representación*: Se seguirá la representación real basada en un vector  $W$  de tamaño  $n$  con valores en  $[0, 1]$  que indican el peso asociado a cada característica y la capacidad para eliminarla si su peso es menor que 0.1, explicado en las transparencias del seminario.
- *Función objetivo*: Será la combinación con pesos de las medidas de precisión (tasa de acierto sobre el conjunto de entrenamiento) y la complejidad (la tasa de reducción de características con respecto al conjunto original) del clasificador 1-NN diseñado empleando el vector  $W$ . Para calcular la tasa de acierto será necesario emplear la técnica de validación *leave-one-out* explicada en las transparencias del Seminario 2. El valor de  $\alpha$  considerado será  $\alpha=0.8$ . El objetivo será maximizar esta función.
- *Generación de la solución inicial*: La solución inicial se generará de forma aleatoria utilizando una distribución uniforme en  $[0, 1]$  en todos los casos.
- *Esquema de generación de vecinos*: Se empleará el movimiento de cambio por mutación normal  $Mov(W, \sigma)$  que altera el vector  $W$  sumándole otro vector  $Z$  generado a partir de una distribución normal de media 0 y varianza  $\sigma^2$ . Su aplicación concreta dependerá del algoritmo específico.
- *Criterio de aceptación*: Se considera una mejora cuando se aumenta el valor global de la función objetivo.

A continuación veremos las particularidades de cada algoritmo.

### 3.1. Algoritmos Genéticos

Los AGs de esta práctica presentarán las siguientes componentes:

- *Esquema de evolución*: Como se ha comentado, se considerarán dos versiones, una basada en el esquema generacional con elitismo (AGG) y otra basada en el esquema estacionario (AGE). En el primero se seleccionará una población de

padres del mismo tamaño que la población genética mientras que en el segundo se seleccionarán únicamente dos padres.

- *Operador de selección:* Se usará el torneo binario, consistente en elegir aleatoriamente dos individuos de la población y seleccionar el mejor de ellos. En el esquema generacional, se aplicarán tantos torneos como individuos existan en la población genética, incluyendo los individuos ganadores en la población de padres. En el esquema estacionario, se aplicará dos veces el torneo para elegir los dos padres que serán posteriormente recombinados (cruzados).
- *Esquema de reemplazo:* En el esquema generacional, la población de hijos sustituye automáticamente a la actual. Para conservar el elitismo, si la mejor solución de la generación anterior no sobrevive, sustituye directamente la peor solución de la nueva población. En el estacionario, los dos descendientes generados tras el cruce y la mutación (esta última aplicada según una determinada probabilidad) sustituyen a los dos peores de la población actual, en caso de ser mejores que ellos. Es decir, de las 4 soluciones, las dos actuales peores y las dos nuevas, se mantienen en la población las 2 con mejor *fitness*.
- *Operador de cruce:* Se emplearán dos operadores de cruce para representación real, explicados en las transparencias de teoría del tema de Algoritmos Genéticos y del Seminario 3. Uno de ellos será el operador BLX- $\alpha$ , con  $\alpha=0.3$ . El otro será el cruce aritmético. **Esto resultará en el desarrollo de cuatro AGs distintos, dos generacionales (AGG-BLX y AGG-CA) y dos estacionarios (AGE-BLX y AGE-CA).**
- *Operador de mutación:* Se considerará el operador de mutación normal para representación real, explicado en las transparencias del Seminario 2 y de teoría. Para aplicarlo, se considerará un valor  $\sigma=0,3$ . Coincidirá por tanto con el operador de generación de vecinos de la BL.

### Valores de los parámetros y ejecuciones

El tamaño de la población será de 50 cromosomas. La probabilidad de cruce será 0,7 en el AGG y 1 en el AGE (siempre se cruzan los dos padres). La probabilidad de mutación (**por individuo**) será de 0,1 en ambos casos (la mutación por gen será de 0,1/genes\_individuo). El criterio de parada en las dos versiones del AG consistirá en realizar **15000 evaluaciones de la función objetivo**.

## 3.2. Algoritmos Meméticos

### Algoritmos

El AM consistirá en hibridar el algoritmo genético generacional (AGG) que mejor resultado haya proporcionado con la BL desarrollada en la Práctica 1.b. Se estudiarán las tres posibilidades de hibridación siguientes:

1. AM-(10,1.0): Cada 10 generaciones, se aplica la BL sobre todos los cromosomas de la población.

2. AM-(10,0.1): Cada 10 generaciones, se aplica la BL sobre un subconjunto de cromosomas de la población seleccionado aleatoriamente con probabilidad  $p_{LS}$  igual a 0.1 para cada cromosoma.
3. AM-(10,0.1mej): Cada 10 generaciones, aplicar la BL sobre los  $0.1 \cdot N$  mejores cromosomas de la población actual ( $N$  es el tamaño de ésta).

### Valores de los parámetros y ejecuciones

El tamaño de la población del AM será también de 50 cromosomas. Las probabilidades de cruce y mutación serán 0,7 (por cromosoma) y 0,1 (**por individuo**) en ambos casos. Se detendrá la ejecución de la BL aplicada sobre un cromosoma **cuando se hayan evaluado  $2 \cdot n$  vecinos distintos en la ejecución**, siendo  $n$  el tamaño del cromosoma. El criterio de parada del AM consistirá en realizar **15000 evaluaciones de la función objetivo**, incluidas por supuesto las de la BL.

## 4. Tablas de Resultados a Obtener

Se diseñará una tabla para cada algoritmo (1-NN, RELIEF, BL, AGG-BLX, AGG-CA, AGE-BLX, AGE-CA, AM-(10,1.0), AM-(10,0.1) y AM-(10,0.1mej)) donde se recojan los resultados de la ejecución de dicho algoritmo en los conjuntos de datos considerados. Tendrá la misma estructura que la Tabla 5.1. de la Práctica 1.b. De igual manera, los resultados para 1-NN, RELIEF y BL también serán los obtenidos en la Práctica 1.b.

Finalmente, se construirá una tabla de resultados global que recoja los resultados medios de calidad y tiempo para todos los algoritmos considerados, tal como se muestra en la Tabla 4.1. Para rellenar esta tabla se hará uso de los resultados medios mostrados en las tablas parciales. Aunque en la tabla que sirve de ejemplo se han incluido todos los algoritmos considerados en esta práctica, naturalmente sólo se incluirán los que se hayan desarrollado.

Tabla 5.2: Resultados globales en el problema del APC

	Diabetes				Ozone				Spectf-heart			
	%_clas	%_red	Agr.	T	%_clas	%_red	Agr.	T	%_clas	%_red	Agr.	T
1-NN	x	0	x	x	x	0	x	x	x	0	x	x
RELIEF	x	0	x	x	x	0	x	x	x	0	x	x
BL	x	X	x	x	x	x	x	x	x	x	x	x
AGG-BLX	x	X	x	x	x	x	x	x	x	x	x	x
AGG-CA	x	X	x	x	x	x	x	x	x	x	x	x
AGE-BLX	x	X	x	x	x	x	x	x	x	x	x	x
AGE-CA	x	X	x	x	x	x	x	x	x	x	x	x
AM-(10,1.0)	x	X	x	x	x	x	x	x	x	x	x	x
AM-(10,0.1)	x	X	x	x	x	x	x	x	x	x	x	x
AM-(10,0.1mej)	x	X	x	x	x	x	x	x	x	x	x	x

A partir de los datos mostrados en estas tablas, el estudiante realizará un análisis de los resultados, **que influirá significativamente en la calificación de la práctica**. Se deben comparar los distintos algoritmos en términos de las tasas de clasificación obtenidas (capacidad del algoritmo para obtener soluciones de calidad) y el tiempo requerido para obtenerlas (rapidez del algoritmo). Se comparará el rendimiento de las metaheurísticas entre sí, así como con respecto a los algoritmos de referencia, 1-NN original y RELIEF.

## 5. Documentación y Ficheros a Entregar

Además de la documentación detallada en la Sección 7 del guión de la Práctica 1.b, en lo referente al punto d) se incluirá, al menos, la siguiente información:

1. Esquema de representación de soluciones empleado.
2. Descripción en pseudocódigo de la función objetivo.
3. Pseudocódigo del proceso de generación de soluciones aleatorias.
4. Pseudocódigo de la selección de los AGs y los operadores de cruce y mutación.

En lo que respecta al punto e), se incluirá la siguiente información:

1. Para los AGs, el esquema de evolución y de reemplazamiento considerados.
2. Para los AMs, el esquema de búsqueda seguido por cada algoritmo en lo que respecta a la integración de la BL dentro del AG.

Como recomendación, el apartado d) debería describirse en un máximo de cuatro páginas. En el apartado e), el número total de páginas para describir cada algoritmo (incluyendo el pseudocódigo del esquema de búsqueda y de las componentes particulares) sería de dos páginas.

Aunque lo esencial es el contenido, también debe cuidarse la presentación y la redacción. Se recuerda que **la documentación nunca deberá incluir listado total o parcial del código fuente en caso de haberlo implementado**. En lo referente al **desarrollo de la práctica**, se seguirán los mismos criterios descritos en la Sección 7 del guión de la Práctica 1.b. El **método de evaluación** será el de la Sección 8 de ese guión.

Dentro del código fuente deberá de tener los siguientes métodos/funciones que recibiendo de alguna forma el dataset devuelva la mejor solución obtenida por el respectivo algoritmo, y su fitness correspondiente:

- **AGG**: Puede crearse un único método/función implementando ambas, o puede implementar un método/función para cada operador de cruce: AGG\_BLX y AGG\_Arit.
- **AGE**: Puede implementar tanto el AGE\_BLX y AGE\_Arit, o puede crearse un único método/función implementando ambas.
- **AM-All**: El algoritmo de Algoritmo Memético a todos los individuos.
- **AM-Rand**: El algoritmo de Algoritmo Memético a una solución aleatoria.
- **AM-Best**: El algoritmo de Algoritmo Memético al mejor individuo.