

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/Metaheuristics>
<https://decsai.ugr.es>

Guión de Prácticas

Práctica 2.b:
Enfriamiento Simulado, Búsqueda Local Reiterada y
Evolución Diferencial para el
Problema del Aprendizaje de Pesos en Características

Curso 2016-2017

Tercer Curso del Grado en Ingeniería Informática

Práctica 2.b

Enfriamiento Simulado, Búsqueda Local Reiterada y Evolución Diferencial para el Problema del Aprendizaje de Pesos en Características

1. Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de los siguientes algoritmos: Enfriamiento Simulado (ES), Búsqueda Local Reiterada (ILS) y Evolución Diferencial (DE). Para ello, se requerirá que el estudiante adapte estos métodos para resolver el problema del aprendizaje de pesos en características (APC) descrito en las transparencias del Seminario 4.b y que compare los resultados obtenidos con los proporcionados por el clasificador 1-NN generado considerando todas las características igualmente ponderadas y con el clasificador 1-NN obtenido empleando los pesos aprendidos por el método *greedy* RELIEF en una serie de casos del problema.

La práctica se evalúa sobre un total de **3,5 puntos**, distribuidos de la siguiente forma: ES (1 punto), ILS (0,75 puntos) y DE (1,75 puntos).

La fecha límite de entrega será el **Viernes 2 de Junio de 2017** antes de las 23:59 horas. La entrega de la práctica se realizará por internet a través del acceso identificado de la web del departamento de CCIA (<https://decsai.ugr.es>).

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 (véase la Sección 3.2 de dicho guión de prácticas). El número de ejecuciones a realizar sobre ellos, el procedimiento de validación y los estadísticos de calidad (*Tasa_clas*, *Tasa_red*, *Agregado* y *Tiempo*) se pueden cambiar tal y como se explicará en la Sección 4 de este guión. Si el estudiante decide no hacerlo, éstos 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, empleado en la Práctica 1.b y explicado en las transparencias del seminario.
- *Función objetivo*: Será el porcentaje de acierto del clasificador 1-NN, modificando el valor de las características con el vector W y utilizando la técnica de validación *leave-one-out* explicada en las transparencias del seminario 2. 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 en ES e ILS*: 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 σ^2 . Su aplicación concreta dependerá del algoritmo específico.
- *Algoritmo de búsqueda local*: En ILS, se considerará la búsqueda local (BL) que sigue el enfoque del primer mejor vecino propuesta en la Práctica 1.b. Se detendrá la ejecución del algoritmo bien cuando no se encuentre mejora al generar un máximo número de vecinos o bien cuando se hayan evaluado 1000 soluciones distintas (en cualquier caso, en la BL, se parará también después de 1000 evaluaciones aunque siguiera habiendo soluciones mejores en el entorno).

A continuación veremos las particularidades de cada algoritmo.

3.1. Enfriamiento Simulado (ES)

Algoritmo

Se ha de emplear un algoritmo ES con las siguientes componentes:

- *Esquema de enfriamiento*: Se empleará el esquema de Cauchy modificado:

$$T_{k+1} = \frac{T_k}{1 + \beta \cdot T_k} \quad ; \quad \beta = \frac{T_0 - T_f}{M \cdot T_0 \cdot T_f}$$

donde M es el número de enfriamientos a realizar, T_0 es la temperatura inicial y T_f es la temperatura final que tendrá un valor cercano a cero¹.

- *Operador de Vecino y exploración del entorno para $L(T)$* : En cada iteración del bucle interno $L(T)$, se aplicará un único movimiento $Mov(W, \sigma)$ para generar una única solución vecina que será comparada con la solución actual. Se escogerá aleatoriamente la característica i a la que se le aplicará la perturbación.
- *Condición de enfriamiento $L(T)$* : Se enfriará la temperatura, finalizando la iteración actual, bien cuando se haya generado un número máximo de vecinos $máx_vecinos$ (independientemente de si han sido o no aceptados) o bien cuando se haya aceptado un número máximo de los vecinos generados $máx_éxitos$.
- *Condición de parada*: El algoritmo finalizará bien cuando haya alcanzado el número máximo de evaluaciones prefijado o bien cuando el número de éxitos en el enfriamiento actual sea igual a 0.

Valores de los parámetros y ejecuciones

La temperatura inicial se calculará en función de la siguiente fórmula:

$$T_0 = \frac{\mu \cdot C(S_0)}{-\ln(\phi)}$$

donde T_0 es la temperatura inicial, $C(S_0)$ es el coste de la solución inicial y $\phi \in [0,1]$ es la probabilidad de aceptar una solución un μ por 1 peor que la inicial. En las ejecuciones se considerará $\phi = \mu = 0,3$. La temperatura final T_f se fijará a 10^{-3} (*¡comprobando siempre que sea menor que la inicial!*).

Los parámetros que definen el bucle interno $L(T)$ tomarán valor $máx_vecinos = 10 \cdot n$ (tamaño del caso del problema) y $máx_éxitos = 0,1 \cdot máx_vecinos$. **El número máximo de evaluaciones será 15000**. Por lo tanto, el número de iteraciones (enfriamientos) M del algoritmo ES será igual a $15000 / máx_vecinos$ ².

3.2. Búsqueda Local Reiterada (ILS)

¹ **NOTA:** Si el estudiante observa que este esquema de enfriamiento enfría demasiado rápido, puede sustituirlo por un esquema proporcional: $T_{k+1} = \leftarrow \alpha \cdot T_k$ con $\alpha \in [0,9, 0,99]$.

² **NOTA 1:** Es posible que se realicen menos de 15000 evaluaciones durante la ejecución debido a la condición de enfriamiento cuando se alcanzan $máx_éxitos$.

NOTA 2: Puede que con $máx_vecinos = 10 \cdot n$ se generen demasiados vecinos para cada enfriamiento. El estudiante puede probar también con $máx_vecinos = 5 \cdot n$ o directamente n .

Algoritmo

El algoritmo ILS consistirá en generar una solución inicial aleatoria y aplicar el algoritmo de BL sobre ella. Una vez obtenida la solución optimizada, se estudiará si es mejor que la mejor solución encontrada hasta el momento y se realizará una mutación sobre la mejor de estas dos, volviendo a aplicar el algoritmo de BL sobre esta solución mutada. Este proceso se repite un determinado número de veces, devolviéndose la mejor solución encontrada en todo el proceso. Por tanto, se sigue el *criterio del mejor* como criterio de aceptación de la ILS.

Tal y como se describe en las transparencias del Seminario 4.b, el operador de mutación de ILS estará basado en un operador de vecino que provoque un cambio más brusco en la solución actual que el considerado en la BL. Para ello, usaremos un operador que cambia conjuntamente el peso de t características *escogidas aleatoriamente* en la solución.

Valores de los parámetros y ejecuciones

En cada ejecución del algoritmo se realizarán 15 iteraciones, es decir, se aplicará 15 veces el algoritmo de BL, la primera vez sobre una solución inicial aleatoria y las 14 restantes sobre soluciones mutadas. Se usará un valor $t=0.1 \cdot n$ en el operador de mutación, es decir, se cambiará el valor del peso en un 10% de las características. Además, para esta mutación se considerará un $\sigma=0.4$. El número máximo de evaluaciones de cada BL será de 1000 evaluaciones.

3.3. Evolución Diferencial (DE)

Algoritmos

La DE es un modelo evolutivo propuesto para optimización con parámetros reales que enfatiza la mutación y utiliza un operador de cruce/recombinación a posteriori. Consideraremos dos algoritmos de DE:

- *DE/Rand/1*: La fórmula para obtener el vector con mutación es la siguiente.

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}).$$

Los índices $r1$, $r2$ y $r3$ de cada individuo i en cada generación G serán escogidos aleatoriamente de forma mutuamente excluyente (incluyendo el vector i -ésimo).

- *DE/current-to-best/1*: La fórmula para obtener el vector con mutación es la siguiente.

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r1,G} - X_{r2,G}).$$

Los índices $r1$ y $r2$ de cada individuo i en cada generación G serán escogidos aleatoriamente de forma mutuamente excluyentes (incluyendo el vector i -ésimo). X_{best} denota el mejor vector en la generación G .

La recombinación utilizada en ambos algoritmos de DE será la binominal (ver diapositiva 7 del Seminario 4b). El reemplazamiento en ambos casos es *one-to-one*, es decir, selección del mejor entre padre e hijo. En esta versión, las componentes del vector solución se mantienen dentro de su rango permitido para no permitir soluciones inválidas.

Valores de los parámetros y ejecuciones

El tamaño de la población será de 50 vectores en ambos modelos (*DE/Rand/1* y *DE/current-to-best/1*). La probabilidad de cruce *CR* será 0,5 y *F* valdrá 0,5. El criterio de parada en las dos versiones de DE consistirá en realizar **15000 evaluaciones de la función objetivo**.

4. Nuevo Método de Evaluación de Soluciones

4.1. Nueva función objetivo

Con motivo de la falta de diversidad en los resultados obtenidos en la práctica anterior, se propone un cambio en la evaluación de soluciones dentro del problema del Aprendizaje de Pesos en Características. Como ya hemos mencionado, el problema del APC consiste en optimizar el rendimiento de un clasificador basado en vecinos cercanos a partir de la inclusión de pesos asociados a las características del problema que modifican su valor en el momento de calcular las distancias entre ejemplos. En nuestro caso, el clasificador considerado será el 1-NN (k-NN, k vecinos más cercanos, con k=1 vecino). También se mencionó que existen distintos criterios para determinar cuándo el clasificador generado es mejor. **Ahora consideraremos un nuevo criterio basado en la simplicidad del clasificador diseñado basado en la minimización del número de características a utilizar en el clasificador final.** Así, el problema del APC se puede reformular como:

Maximizar $función_objetivo = tasa_clas \cdot \alpha + tasa_red \cdot (1 - \alpha)$

Donde $tasa_clas = 100 \cdot \frac{n^\circ \text{ instancias bien clasificadas de } T}{n^\circ \text{ instancias en } T}$,

$$tasa_red = 100 \cdot \frac{n^\circ \text{ valores } w_i < 0.1}{n^\circ \text{ características}}$$

sujeto a que:

- $w_i = [0, 1], 1 \leq i \leq n$

donde:

- $W = (w_1, \dots, w_n)$ es una solución al problema que consiste en un vector de números reales de tamaño n que define el peso que pondera a cada una de las características f_i .
- T es el conjunto de datos sobre el que se evalúa el clasificador.
- La tasa de clasificación es la misma que se usaba en la Práctica 1.b y mide el porcentaje de acierto con el clasificador 1-NN utilizando la técnica de validación leave-one-out y los pesos en W que se asocian a las n características.
- La tasa de reducción mide el número de características que no se van a utilizar durante el proceso de clasificación en relación al número total de características. Tal y como define la fórmula, se considerará que una característica no interviene en la clasificación cuando su peso $w_i < 0.1$.
- El valor de α pondera la importancia entre el acierto y la reducción de la solución encontrada. Nosotros consideraremos $\alpha=0.5$, dándole la misma importancia a ambos criterios.
- El objetivo es obtener el mejor W que maximiza esta función; es decir, un conjunto de pesos que maximice el acierto del clasificador 1-NN y, a la vez, que considere el menor número de características posible.

4.2. Mecanismo de validación

En esta práctica, consideraremos el método de validación **5-fold cross validation (5fcv)**. Para ello, el conjunto de datos se divide en 5 particiones disjuntas al 20%. Aprenderemos un clasificador utilizando hasta un total del 80% de los datos disponibles (4 particiones de las 5) y validaremos con el 20% restante (la partición restante), teniendo por tanto 5 particiones posibles al 80-20%. Así, obtendremos hasta un total de cinco valores de porcentaje de clasificación en el conjunto de prueba, uno para cada partición que ha sido parte del conjunto de validación.

La medida de validación será la tasa de acierto del clasificador 1-NN sobre el conjunto de prueba. El resultado final será la media de los 5 valores obtenidos, uno para cada ejecución del algoritmo.

Para facilitar la comparación de algoritmos en las prácticas del APC se considerarán cuatro estadísticos distintos denominados *Tasa_clas*, *Tasa_red*, *Agregado* y *Tiempo*:

- *Tasa_clas* se calcula como la media de los porcentajes de acierto (las tasas de clasificación) obtenidos por cada método en cada partición del conjunto de datos (es el valor final resultante del 5fcv).
- *Tasa_red* corresponde al porcentaje de reducción obtenido en la selección del subconjunto de características respecto al total. El cálculo se realiza con la expresión mostrada en la sección anterior. El valor global se obtiene como la media de los porcentajes de reducción obtenidos por cada método en cada partición del conjunto de datos.
- *Agregado* corresponde al valor de la función objetivo que está optimizando el algoritmo; es decir, al valor a maximizar proveniente de la fórmula

$función_objetivo = tasa_clas \cdot \alpha + tasa_red \cdot (1 - \alpha)$. Igualmente, se indicará el valor medio de las 5 ejecuciones del 5fcv.

- *Tiempo* se calcula como la media del tiempo de ejecución empleado por el algoritmo para resolver cada caso del problema (cada conjunto de datos). Es decir, en nuestro caso es la media del tiempo empleado por las 5 ejecuciones.

Cuanto mayor es el valor de *Tasa_clas* para un algoritmo, mejor calidad tiene dicho algoritmo, porque obtiene clasificadores más precisos en media. Del mismo modo, cuanto mayor es el valor de *Tasa_red*, mejor calidad tiene también el algoritmo, porque obtiene clasificadores más simples en media. La elección entre un clasificador de mejor rendimiento o de mayor simplicidad depende de los requisitos de la aplicación concreta. Finalmente, si dos métodos obtienen soluciones con la misma calidad (tienen valores de *Tasa_clas* y *Tasa_red* similares), uno será mejor que el otro si emplea menos tiempo en media.

5. Tablas de Resultados a Obtener

Aparte de los algoritmos considerados en esta práctica, se añadirán los resultados de tres algoritmos de la práctica anterior que consideren el nuevo método de evaluación de soluciones utilizando un 5fcv: BL, AGG-BLX y AM-(10,0.1mej), **siempre que se hayan realizado dichos algoritmos en la primera práctica.**

Se diseñará una tabla para cada algoritmo (1-NN, RELIEF, ES, ILS, DE/rand/1, DE/current-to-best/1, BL, AGG-BLX 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.

Tabla 5.1: Resultados obtenidos por el algoritmo X en el problema del APC

	Sonar				Wdbc				Spambase			
	% clas	% red	Agr.	T	% clas	% red	Agr.	T	% clas	% red	Agr.	T
Partición 1	x	x	X	x	x	x	x	x	x	X	x	x
Partición 2	x	x	X	x	x	x	x	x	x	x	x	x
Partición 3	x	x	x	x	x	x	x	x	x	x	x	x
Partición 4	x	x	x	x	x	x	x	x	x	x	x	x
Partición 5	x	x	x	x	x	x	x	x	x	x	x	x
Media	x	x	x	x	x	x	x	x	x	x	x	x

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 5.2. 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

	Sonar				Wdbc				Spambase			
	% clas	% red	Agr.	T	% clas	% red	Agr.	T	% clas	% red	Agr.	T
1-NN	x	x	x	x	x	x	x	x	x	x	x	x
RELIEF	x	x	x	x	x	x	x	x	x	x	x	x
ES	x	x	x	x	x	x	x	x	x	x	x	x
ILS	x	x	x	x	x	x	x	x	x	x	x	x
DE/rand/1	x	x	x	x	x	x	x	x	x	x	x	x
DE/current-to-best/1	x	x	x	x	x	x	x	x	x	x	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
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 obtenidos, que influirá significativamente en la calificación de la práctica. En dicho análisis se deben comparar los distintos algoritmos en términos de las tasas de clasificación obtenidas (capacidad del algoritmo para obtener soluciones de calidad), las tasas de reducción y el tiempo requerido para obtener las soluciones (rapidez del algoritmo). Se comparará el rendimiento de las metaheurísticas entre sí, así como con respecto a los algoritmos de referencia, el 1-NN original y el RELIEF. En esta práctica, se pueden comparar las técnicas en función de su tipología: basadas en trayectorias simples (ES), trayectorias múltiples (ILS) y búsqueda evolutiva (DE).

6. Documentación y Ficheros a Entregar

Además de la documentación detallada en la Sección 6 del guión de la Práctica 1.b, en lo referente al punto 4 se incluirá, al menos, la siguiente información:

- Esquema de representación de soluciones empleado.
- Descripción en pseudocódigo de la función objetivo.
- Descripción en pseudocódigo del proceso de generación de soluciones aleatorias.
- Descripción en pseudocódigo del algoritmo de BL empleado, incluyendo el método de exploración del entorno y el operador de generación de vecino.

En lo que respecta al punto 5, se incluirá la siguiente información:

- Descripción en pseudocódigo del esquema de búsqueda seguido por cada algoritmo (ES, ILS y DE).
- Además se detallarán, al menos, las siguientes componentes particulares de cada algoritmo:
 - Para el algoritmo ES, descripción del cálculo de la temperatura inicial y del esquema de enfriamiento.
 - Para el algoritmo ILS, descripción en pseudocódigo del operador de mutación empleado.

- c. Para los algoritmos de DE, descripción en pseudocódigo del proceso de mutación seguido por cada uno de los dos modelos considerados, así como la recombinación y reemplazamiento empleados.

En el punto 6, se incluirá una breve descripción del algoritmo de comparación RELIEF.

Como recomendación, el apartado 4 debería describirse en un máximo de dos páginas. En el apartado 5, el número total de páginas para describir cada algoritmo (incluyendo el pseudocódigo del esquema de búsqueda y de las componentes particulares) sería de una página para ILS, y de dos páginas para ES y DE.

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 6 del guión de la Práctica 1.b. El **método de evaluación** será el descrito en la Sección 7 de dicho guión.