UNIVERSIDAD DE GRANADA

E.T.S.I. INFORMÁTICA Y TELECOMUNICACIÓN

Departamento de Ciencias de la Computación e Inteligencia Artificial

Metaheurísticas

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

https://decsai.ugr.es

Guión de Prácticas

Práctica 3.b:
Técnicas de Búsqueda basadas en Trayectorias para el Problema del Aprendizaje de Pesos en Características

Curso 2023-2024

Tercer Curso del Grado en Ingeniería Informática

Práctica 3.b

Técnicas de Búsqueda basadas en Trayectorias 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 Trayectorias* (tanto simples como múltiples) en la resolución del problema de Problema de 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 a dicho problema:

- Búsqueda Multiarranque Básica (BMB).
- Enfriamiento Simulado (ES).
- Greedy Aleatorio con Búsqueda Local (GRASP)
- Búsqueda Local Reiterada (ILS).
- Hibridación de ILS y ES (ILS-ES).

El estudiante deberá comparar los resultados obtenidos con las estimaciones existentes para el valor de los óptimos de una serie de casos del problema, así como con el algoritmo greedy básico y la búsqueda local (BL) descritos en el Seminario 2 y desarrollados en la Práctica 1.a.

La práctica se evalúa sobre un total de **2,0 puntos**, distribuidos de la siguiente forma: ES (1 punto), BMB (0,25 puntos), ILS (0,5 puntos), ILS-ES (0,25 puntos).

La fecha límite de entrega será el **Domingo 2 de Junio de 2024** antes de las 23:59 horas. La entrega de la práctica se realizará por internet a través del espacio de la asignatura en 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, Fitness* y *Tiempo*) serán los mismos que en el guión de la Práctica 1.b. (véase la Sección 3 de dicho guion 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.

- *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 α considerado será α=0.8, dándole más importancia al porcentaje de clasificación. 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. Para el caso de la DE se generarán todos los individuos de la población inicial de forma aleatoria.

- *Esquema de generación de vecinos en ES e ILS*: Se empleará el movimiento de cambio por mutación normal *Mov(W,σ)* 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 de cada iteración de la BL cuando se hayan evaluado 750 soluciones distintas.

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:

$$T_{k+1} = \frac{T_k}{1+\beta \cdot \mathrm{T}_k} \qquad ; \qquad \beta = \frac{T_0 - T_f}{\mathrm{M} \cdot \mathrm{T}_0 \cdot T_f}$$

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 cero.

- *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,σ)* 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:

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

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 ϕ=0,3 y μ=0,1. 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·*n* (tamaño del caso del problema) y *máx_éxitos=0.1*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* [1].

[1] **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·*n* se generen demasiados vecinos para cada enfriamiento. El/La estudiante puede probar también con *máx_vecinos*=5·*n* o directamente *n*.

3.2. Búsqueda Multiarranque Básica (BMB)

Algoritmo

El algoritmo BMB consistirá simplemente en generar un determinado número de soluciones aleatorias iniciales y optimizar cada una de ellas con el algoritmo de BL considerado. Se devolverá la mejor solución encontrada en todo el proceso.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución de la BMB sobre cada caso del problema. En dicha ejecución, se realizarán **20 iteraciones**, es decir, se generarán 20 soluciones iniciales aleatorias y se aplicará la BL sobre cada una de ellas. Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan realizado **750 evaluaciones**.

3.3. Búsqueda Local Reiterada (ILS)

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, 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 el peso de t características *elegidas aleatoriamente* en la solución sumándole un valor aleatorio (para cada peso) entre -0.25 y 0.25 según una distribución uniforme.

Valores de los parámetros y ejecuciones

En cada ejecución del algoritmo se realizarán **20 iteraciones**, es decir, se aplicará 20 veces el algoritmo de BL, la primera vez sobre una solución inicial aleatoria y las 19 restantes sobre soluciones mutadas. Se usará un valor $t=0.20 \cdot n$ en el operador de mutación, es decir, se reiniciará el valor del peso en un 20% de las características, siempre que se cambie al menos 3. El número máximo de evaluaciones de cada BL será de **750 evaluaciones**. En la hibridación con el ES, se aplicarán los mismos parámetros que con la BL.

4. Tablas de Resultados a Obtener

Se diseñará una tabla para cada algoritmo (BL, ES, ILS, y ILS-ES) 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 BL 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 4.1: Resultados globales en el problema del APC

	Ecoli				**Parkinsons**				**breast-cancer**			
	%_clas	***%_red***	***Fit.***	**T**	***%_clas***	***%_red***	***Fit.***	**T**	***%_clas***	***%_red***	***Fit.***	**T**
BL	x	x	x	x	x	x	x	x	x	x	x	x
Mejor P2	x	x	x	x	x	x	x	x	x	x	x	x
BMB	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
ILS-ES	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, BL, y el mejor de la práctica 2.

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***. 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, la BL. En esta práctica, se pueden comparar las técnicas en función de su tipología: basadas en trayectorias simples (ES), y trayectorias múltiples (ILS) en sus dos versiones.

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. Descripción en pseudocódigo del proceso de generación de soluciones aleatorias.
4. 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 e), se incluirá la siguiente información:

1. Descripción en pseudocódigo del esquema de búsqueda seguido por cada algoritmo (ES, BMB, ILS).
2. Además se detallarán, al menos, las siguientes componentes particulares de cada algoritmo:
    a. Para el algoritmo ES, descripción del cálculo de la temperatura inicial y del esquema de enfriamiento.
    b. Para el algoritmo de BL, descripción en pseudocódigo del método de creación de la lista de candidatos, el de exploración del entorno, el operador de generación de vecino.
    c. Para el algoritmo ILS, descripción en pseudocódigo del operador de mutación empleado.

Como recomendación, el apartado d) debería describirse en un máximo de dos 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 una página para ILS, BMB, y ILS.

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:

- **BMB**: El algoritmo de multiarranque básico.
- **ES:** El algoritmo de Enfriamiento Simulado.
- **ILS**: El algoritmo Iterativo de Búsqueda Local usando la BL del guión 1.
- **ILS-ES:** El algoritmo Iterativo de Búsqueda Local usando el algoritmo de Enfriamiento Simulado.