

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>

Guion de Prácticas

Práctica 3.b:
Búsquedas por Trayectorias para el
Problema del Agrupamiento con Restricciones

Curso 2019-2020

Tercer Curso del Grado en Ingeniería Informática

Práctica 3.b

Búsquedas por Trayectorias para el Problema del Agrupamiento con Restricciones

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 del agrupamiento con restricciones (PAR) 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).
- 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.b.

La práctica se evalúa sobre un total de **3 puntos**, distribuidos de la siguiente forma: ES (1,25 puntos), BMB (0,5 puntos), ILS (0,75 puntos) e ILS-ES (0,5 puntos).

La fecha límite de entrega será el **lunes 25 de mayo de 2020** antes de las 23:55 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 **pero se ha añadido un conjunto de datos adicional** ya descrito en la Práctica 2.b. De igual manera, el número de ejecuciones a realizar sobre ellos, el procedimiento de validación y los estadísticos de calidad (*Tasa_C*, *Tasa_inf*, *Agregado* y *Tiempo*) serán los mismos que en el guion 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 entera basada en un vector S de tamaño n (número de instancias del conjunto de datos) con valores en $\{1, \dots, k\}$ (siendo k el número de agrupamientos, que coincidirá con el de las clases del conjunto de datos real) que indican el cluster asociado a cada instancia, explicado en las transparencias del seminario.
- *Función objetivo*: Será la combinación mediante el peso λ de las medidas de \sqrt{C} (desviación general de la partición) e *infeasability* resultantes del agrupamiento obtenido por la solución representada en el vector S . El valor de λ está explicado en las transparencias del Seminario 2. El objetivo será minimizar esta función.
- *Generación de la solución inicial*: La solución inicial se generará de forma aleatoria escogiendo un valor en $\{1, \dots, k\}$ en cada posición del vector S , **comprobando que cada cluster tiene al menos una instancia asignada.**
- *Esquema de generación de vecinos empleado en los algoritmos de búsqueda por trayectorias simples (ES y BL)*: Se empleará el movimiento de cambio de cluster que altera el vector S sustituyendo el valor s_i en una posición i -ésima por otro valor $l \in \{1, \dots, k\}$ y $l \neq s_i$, **comprobando que no deja ningún cluster vacío.** Será obligatorio emplear la **factorización** de la función objetivo en todos los casos.
- *Algoritmo de búsqueda por trayectorias*: En ILS, se considerarán dos variantes. La primera usará la búsqueda local (BL) que sigue el enfoque del primer mejor vecino propuesta en la Práctica 1.b. La segunda, el ES diseñado en esta práctica.

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 *Cambio_Cluster(S, i, l)* para generar una única solución vecina que será comparada con la solución actual (*es necesario verificar que se cumplen las restricciones del problema del clustering*). Se escogerá aleatoriamente el elemento i a la que se le aplicará el movimiento.
- *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* = $0,1 \cdot \text{máx_vecinos}$. **El número máximo de evaluaciones será 100000**. Por lo tanto, el número de iteraciones (enfriamientos) M del algoritmo ES será igual a $100000 / \text{máx_vecinos}$ ².

¹ **NOTA**: Si el/la 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 100000 evaluaciones durante la ejecución debido a la condición de enfriamiento cuando se alcanzan *máx_éxitos*.

3.2. Búsqueda Multiarranque Básica

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 10 iteraciones, es decir, se generarán 10 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 **10000 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 repetirá un determinado número de veces, devolviéndose la mejor solución encontrada en toda la ejecución. Por tanto, se seguirá 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 empleamos el operador de mutación por segmento, basado en seleccionar un segmento de longitud fija de la solución a mutar y aplicarle un cambio fuerte. El cambio consiste en reasignar de forma aleatoria las etiquetas de las instancias asociadas a las posiciones contenidas en el segmento.

Valores de los parámetros y ejecuciones

En cada ejecución del algoritmo se realizarán 10 iteraciones, es decir, se aplicará 10 veces el algoritmo de BL, la primera vez sobre una solución inicial aleatoria y las 9 restantes sobre soluciones mutadas. Se usará un valor $v=0.1 \cdot n$ en el operador de mutación, es decir, se reasignarán aleatoriamente de clúster el 10% de los elementos. El número máximo de evaluaciones de cada BL será de **10000 evaluaciones**.

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

3.4. Algoritmo Híbrido ILS-ES

Algoritmo

El algoritmo ILS-ES tendrá la misma composición que el ILS estándar, con la única diferencia que el algoritmo de búsqueda por trayectorias simples considerado para refinar las soluciones iniciales será el ES de la Sección 3.1 en lugar de la BL empleada hasta ahora.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución del algoritmo ILS-ES sobre cada caso del problema. En dicha ejecución, se realizarán **10 iteraciones**, igual que en la ILS estándar y se optimizarán usando el algoritmo ES con los parámetros indicados en la Sección 3.1 pero con un **número máximo de 10000 evaluaciones**.

4. Tablas de Resultados a Obtener

Se diseñará una tabla para cada algoritmo (COPKM, BL, ES, BMB, ILS, ILS-ES) y cada porcentaje de restricciones donde se recojan los resultados de la ejecución de dicho algoritmo en los conjuntos de datos considerados. Tendrá la misma estructura que la Tabla 6.1 de la Práctica 1.b. Los resultados para COPKM y BL también serán los obtenidos en la Práctica 1.b en los conjuntos de datos previos utilizados en la Práctica 1.

Finalmente, se construirán dos tablas de resultados globales (una por porcentaje de restricciones) que recojan los resultados medios de calidad y tiempo para todos los algoritmos considerados, tal como se muestra en la Tabla 4.1. Para rellenar estas tablas 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 PAR con XX% de restricciones

	Iris				Ecoli				Rand				Newthyroid			
	Tasa_C	Tasa_inf	Agr.	T	Tasa_C	Tasa_inf	Agr.	T	Tasa_C	Tasa_inf	Agr.	T	Tasa_C	Tasa_inf	Agr.	T
COPKM	X	X	X	X	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	X	X	X	X
ES	X	X	X	X	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	X	X	X	X
ILS	X	X	X	X	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	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*. En dicho análisis se deben comparar los distintos algoritmos en términos de las tasas de desviación general de la partición obtenidas (capacidad del algoritmo para obtener soluciones de calidad), número de restricciones no satisfechas y tiempos requeridos para obtener las soluciones (rapidez del algoritmo). En este caso concreto, se pueden

comparar las técnicas basadas en trayectorias simples (BL y ES) con las de trayectorias múltiples (BMB e ILS), teniendo en cuenta que las segundas tienen ventaja al ejecutar el optimizador de trayectoria simple varias veces. Por otro lado, se puede analizar también el comportamiento de los algoritmos en algunos de los casos individuales que presenten un comportamiento más destacado.

5. Documentación y Ficheros a Entregar

Además de la documentación detallada en la Sección 7 del guion 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 (usado en la BMB y en la ILS).

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 e ILS).
2. Además, se detallarán, al menos, las siguientes componentes particulares de cada algoritmo:
 - a. Para el algoritmo ES, descripción en pseudocódigo 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.

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 6 del guion de la Práctica 1.b. El **método de evaluación** será el de la Sección 7 de ese guion.