

UNIVERSIDAD DE GRANADA

ETSIIT INFORMÁTICA Y TELECOMUNICACIÓN



UNIVERSIDAD
DE GRANADA

Departamento de Ciencias de la
Computación e Inteligencia Artificial

Metaheurísticas

Guión de Prácticas

Práctica 3.a

Técnicas de Búsqueda basadas en
Trayectorias de Poblaciones
para el Problema de Maximizar Influencia
En Redes Sociales (SNIMP)

Curso 2024-25

Tercero en Grado en Ingeniería Informática

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 Maximizar Influencia En Redes Sociales (SNIMP) descrito en las transparencias del Seminario 2, y planteados en el Seminario 4. Para ello, se requerirá que el estudiante adapte los siguientes algoritmos a dicho problema:

- Enfriamiento Simulado (ES), **1 punto**, que permite empeorar de forma controlada la mejor solución.
- Búsqueda Multiarranque Básica (BMB), **0.25 puntos**.
- Búsqueda Local Reiterada (ILS), **0.5 puntos**.
- Hibridación de ILS y ES (ILS-ES), **0.25 puntos**.
- GRASP tanto sin BL como con BL, **0.5 puntos**.

La práctica se evalúa sobre un total de **2.5 puntos**, distribuidos según la distribución anterior.

El estudiante deberá comparar los resultados obtenidos en la serie de casos del problema con los proporcionados por los algoritmos de las prácticas anteriores, con atención especial al método de Búsqueda Local, BL, de la práctica 1.

La fecha límite de entrega será el **el domingo 1 de Junio de 2025** 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 deberá de desarrollar los distintos algoritmos al problema planteado. 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.

3 COMPONENTES DE LOS ALGORITMOS

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

- **Esquema de representación:** Se seguirá la representación entera usada hasta ahora.
- **Función objetivo:** El ya explicado en los seminarios.
- **Generación de la solución inicial:** La solución inicial se hará al igual que hasta ahora creando aleatoriamente valores en cada posición dentro del rango, sin que se repita ningún valor.
- **Exploración de vecinos:** Todos los algoritmos usarán el mismo mecanismo de exploración de vecinos.

A continuación veremos las particularidades de cada algoritmo.

4 ALGORITMOS

A continuación detallamos los distintos algoritmos.

4.1 Enfriamiento Simulado (ES)

Implementaremos el algoritmo de Enfriamiento Simulado, ES, visto en clase.

Recuerdo que es un algoritmo inspirado en el calentamiento de metales realizado durante una forja. La idea desde un punto de vista algorítmico es similar a una BL, pero con la característica de que permite empeorar la mejor solución en las fases iniciales de la búsqueda, según un parámetro *temperatura* que va reduciéndose durante la ejecución del algoritmo.



Down at the forge (1) por Neil. Moralee bajo licencia CC BY-NC-ND 2.0.

4.1.1. Componentes del Algoritmo

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

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

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

en 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 paso de la exploración se elegirá un movimiento (posición a cambiar y nuevo valor) elegido de forma aleatoria. Para asegurarse de poder generar todo el entorno, a la hora de escoger el objeto a quitar se seleccionará un movimiento distinta sin repetición.
- **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.

4.1.2. Valores de los parámetros y ejecuciones

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

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

donde T_0 es la temperatura inicial, $Coste(S_0)$ es el coste de una solución inicial S_0 y $\varphi \in [0, 1]$ es la probabilidad de aceptar una solución un μ por 1 peor que la inicial. En las ejecuciones se considerará $\varphi = 0,3$ y $\mu = 0,2$. 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* = $5 \cdot m$ (tamaño de la solución) y *máx_éxitos* = $0,1 \cdot máx_vecinos$. El número máximo de evaluaciones será 1000. Por lo tanto, el número de iteraciones (enfriamientos) M del algoritmo ES será igual a $1000 / máx_vecinos$.

4.2 Búsqueda Multiarranque Básica (BMB)

4.2.1. 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 indicado. Se devolverá la mejor

solución encontrada en todo el proceso.

4.2.2. 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, 20 sin mejorar, o bien cuando se hayan realizado **100 evaluaciones**.

5 BÚSQUEDA LOCAL REITERADA (ILS)

5.1 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 mutar la mejor solución encontrada, se cambiarán el 20 % de los valores por otros tantos valores escogidos de entre los no elegidos (ninguno previamente elegido se podrá volver a coger).

5.1.1. Valores de los parámetros y ejecuciones

Se realizará una única ejecución del ILS sobre cada caso del problema. En dicha ejecución, se aplicarán **10 veces** el algoritmo de BL, la **primera vez sobre una solución aleatoria** inicial, y las **9 veces restantes** sobre soluciones mutadas.

Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno, 20 sin mejorar (no así en el ILS-ES), o bien cuando se hayan realizado **100 evaluaciones**.

6 GREEDY RANDOMIZE ADAPTATIVE SEARCH PROCEDURE (GRASP)

6.1 Algoritmo

Se recuerda que este algoritmo es un algoritmo que aplicará la BL pero en vez de sobre soluciones aleatorias como en el BMB, sobre soluciones iniciales guiada por una heurística.

Tal y como se ha comentado en el Seminario 4, se construye la solución de forma heurística. En cada paso se utiliza un vector de *LRC* nuevos valores ordenados por heurística. De esos valores, se construye un subconjunto *LRC* con los $|LRC|$ mejores valores. Finalmente, se selecciona aleatoriamente de entre dicho conjunto *LRC* y se añade a la solución (y se actualiza *LC* adecuadamente). Este proceso iterativamente permite construir una solución completa.

El primero elemento se escogerá de forma aleatoria. El tamaño $|LRC|$ es dinámico, se introducirán todos aquellos valores cuyo valor heurístico sea mayor o igual a:

$heur_{max} - rand() \cdot (heur_{max} - heur_{min})$, en donde $rand()$ es un valor aleatorio entre 0 y 1.

Una vez aplicada la construcción de la solución, se aplicará dos versiones:

- GRASP-NOBL: Simplemente se construyen las distintas soluciones aleatorias, sin aplicar la BL, y se devuelve la mejor.
- GRASP-SIBL: Tras construir cada solución aleatoria se aplica la BL al igual que se hace con el BMB.

6.1.1. Valores de los parámetros y ejecuciones del GRASP-NOBL

Se generarán **10 soluciones** según el criterio greedy aleatorio, y se devuelve la mejor de ellas.

6.1.2. Valores de los parámetros y ejecuciones del GRASP-SIBL

Se realizará una única ejecución de la BL sobre cada solución generada. Por tanto se generarán **10 soluciones** según el criterio greedy aleatorio, y posteriormente la BL sobre cada una de dichas soluciones, por tanto se aplicará la BL **unas restantes** sobre soluciones mutadas.

Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno, 20 sin mejorar, o bien cuando se hayan realizado **100 evaluaciones**.

7 TABLAS DE RESULTADOS

Se diseñará una tabla para cada algoritmo (BL, ES, ILS, ILS-ES, GRASP) donde se recojan los resultados de la ejecución de dicho algoritmo en los conjuntos de datos considerados. Tendrá la misma estructura que la Tabla 5.2 de la Práctica 1.b.

Debido al gran número de ficheros en vez de pedir una tabla para cada algoritmo con los resultados de la ejecución de dicho algoritmo para cada caso del problema, se pide dar únicamente las siguientes tablas incluyendo únicamente los resultados del *Greedy* y la BL de la práctica 1, así como los resultados del mejor algoritmo de la práctica 2.

Tabla 1: Formato de resultados para el conjunto XXX

Algoritmo	Posición	Fitness	Tiempo (segs)	Evaluaciones
Random	x	x	x	1000
Greedy	x	x	x	1
BL	x	x	x	x
Mejor Práctica 2	x	x	x	x
BMB	x	x	x	x
ILS	x	x	x	x
ES	x	x	x	x
ILS-ES	x	x	x	x
GRASP-NOBL	x	x	x	10
GRASP-SIBL	x	x	x	x

Finalmente, se usará la tabla 2, indicando por cada algoritmo la posición promedio, el tiempo promedio y el promedio de evaluaciones.

Tabla 2: Tabla final de resultados

Algoritmo	Posición Promedia	Tiempo Promedio (segs)	Total evaluaciones
Random	x	x	1000
Greedy	x	x	1
BL	x	x	x
Mejor Práctica 2	x	x	x
BMB	x	x	x
ILS	x	x	x
ES	x	x	x
ILS-ES	x	x	x
GRASP-NOBL	x	x	10
GRASP-SIBL	x	x	x

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. Los resultados del Greedy y de la BL corresponden a los de la Práctica 1. También habría que plantear el resultado del mejor algoritmo de la práctica 2, indicando su valor.

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 calidad de las soluciones y tiempo requerido para producirlas. Por otro lado, se puede analizar también el comportamiento de los algoritmos en algunos de los casos individuales que presenten un comportamiento más destacado.

8 DOCUMENTACIÓN Y FICHEROS A ENTREGAR

En general, la **documentación** de ésta y de cualquier otra práctica será un fichero pdf que deberá incluir, al menos, el siguiente contenido:

- a) Portada con el número y título de la práctica (con el nombre del problema), el curso académico, el nombre, DNI y dirección e-mail del estudiante, y su horario de prácticas.
- b) Índice del contenido de la documentación con la numeración de las páginas.
- c) **Breve descripción/formulación del problema (máximo 1 página)**. Podrá incluirse el mismo contenido repetido en todas las prácticas presentadas por el estudiante.
- d) Breve descripción de la aplicación de los algoritmos empleados al problema (**máximo 4 páginas por algoritmo**): Todas las consideraciones comunes a los distintos algoritmos se describirán en este apartado, que será previo a la descripción de los algoritmos específicos. Incluirá por ejemplo la descripción del esquema de representación de soluciones y la **descripción en pseudocódigo (no código)** de la función objetivo, la generación de solución aleatoria, y la exploración del vecindario.
- e) Descripción en **pseudocódigo** de la **estructura del método de búsqueda** y de todas aquellas **operaciones relevantes** de cada algoritmo. Este contenido, específico a cada algoritmo se detallará en los correspondientes guiones de prácticas. El pseudocódigo **deberá forzosamente reflejar la implementación/ el desarrollo realizados** y no ser una descripción genérica extraída de las transparencias de clase o de cualquier otra fuente. La descripción de cada algoritmo no deberá ocupar más de **2 páginas**.
- f) Breve explicación de la **estructura del código de la práctica**, incluyendo un pequeño **manual de usuario describiendo el proceso para que el profesor de prácticas pueda compilarlo (usando un sistema automático como make o similar) y cómo ejecutarlo, dando algún ejemplo de ejecución**.
- g) Experimentos y análisis de resultados:
 - Descripción de los casos del problema empleados y de los valores de los parámetros considerados en las ejecuciones de cada algoritmo (**incluyendo las semillas utilizadas**).
 - Resultados obtenidos según el formato especificado.
 - Análisis de resultados. El análisis deberá estar orientado a **justificar** (según el comportamiento de cada algoritmo) **los resultados** obtenidos en lugar de realizar una mera “lectura” de las tablas. Se valorará la inclusión de otros elementos de comparación tales como gráficas de convergencia, boxplots, análisis comparativo de las soluciones obtenidas, representación gráfica de las soluciones, etc.
- h) Referencias bibliográficas u otro tipo de material distinto del proporcionado en la asignatura que se haya consultado para realizar la práctica (en caso de haberlo hecho).

Aunque lo esencial es el contenido, también debe cuidarse la presentación y la redacción. **La documentación nunca deberá incluir listado total o parcial del código fuente.**

En lo referente al **desarrollo de la práctica**, se entregará una carpeta llamada **software** que contenga una versión ejecutable de los programas desarrollados, así como el código fuente implementado o los ficheros de configuración del framework empleado. El código fuente o los ficheros de configuración se organizarán en la estructura de directorios que sea necesaria y deberán colgar del directorio *src* en el raíz. Junto con el código fuente, hay que incluir los ficheros necesarios para construir los ejecutables según el entorno de desarrollo empleado (tales como *.prj, makefile, *.ide, etc.). En este directorio se adjuntará también un pequeño fichero de texto de nombre **LEEME**

que contendrá breves reseñas sobre cada fichero incluido en el directorio. Es importante que los programas realizados puedan leer los valores de los parámetros de los algoritmos desde fichero, es decir, que no tengan que ser recompilados para cambiar éstos ante una nueva ejecución. Por ejemplo, la semilla que inicializa la secuencia pseudoaleatoria debería poder especificarse como un parámetro más.

En el caso de que en el lenguaje de programación elegido se haya ofrecido un API, deberá de **obligatoriamente implementarlo siguiendo el API ofrecido**. Si no fuese el caso, se adaptará el API recomendado.

El fichero pdf de la documentación y la carpeta software serán comprimidos en un fichero .zip etiquetado con los apellidos y nombre del estudiante (Ej. Pérez Pérez Manuel.zip). Este fichero será entregado por internet a través del espacio de la asignatura en PRADO.

9 MÉTODO DE EVALUACIÓN

Al principio de la práctica se ha indicado la puntuación máxima que se puede obtener por cada algoritmo y su análisis. **La inclusión de trabajo voluntario** (desarrollo de variantes adicionales, experimentación con diferentes parámetros, prueba con otros operadores o versiones adicionales del algoritmo, análisis extendido, etc.) **podrá incrementar la nota final** por encima de la puntuación máxima definida inicialmente, o compensar parcialmente errores en la práctica.

En caso de que el comportamiento del algoritmo en la versión implementada/ desarrollada no coincida con la descripción en pseudocódigo o no incorpore las componentes requeridas, se podría reducir hasta en un 50 % la calificación del algoritmo correspondiente.