

# UNIVERSIDAD DE GRANADA

## ETSIIT INFORMÁTICA Y TELECOMUNICACIÓN



UNIVERSIDAD  
DE GRANADA

Departamento de Ciencias de la  
Computación e Inteligencia Artificial

### Metaheurísticas

### Guión de Prácticas

#### Práctica 2.a

Técnicas de Búsqueda 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 Poblaciones* en la resolución del Problema de Maximizar Influencia En Redes Sociales (SNIMP) descrito en las transparencias del Seminario 2, y planteados en el Seminario 3. Para ello, se requerirá que el estudiante adapte los siguientes algoritmos a dicho 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 modelos de AGGs 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) totalmente aleatorio desarrollado en la Práctica 1.

El estudiante deberá comparar los resultados obtenidos en la serie de casos del problema con los proporcionados por los algoritmos *Greedy* y los modelos de BL implementados en la práctica 1.

La práctica se evalúa sobre un total de **3 puntos**, distribuidos de la siguiente forma:

- AGGs (**1 punto**).
- AGEs (**1 punto**).
- AMs (**1 punto**).

La fecha límite de entrega será el **el domingo 4 de mayo 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 ESQUEMA DE REPRESENTACIÓN

Aunque el algoritmo en su conjunto devolverá una representación entera, se puede utilizar también la representación binaria  $Sol = (x_1, \dots, x_n)$  en el que la posición  $x_i$  representa si se ha elegido el nodo  $i$  o no (un 1 si se ha escogido, 0 si no). Es decisión del estudiante decidir la representación.

## 4 ALGORITMOS GENÉTICOS

### 4.1 Descripción

Los AGs de esta práctica presentarán los siguientes componentes:

*Inicialización de la población inicial:* La población inicial se generará o bien escogiendo  $m$  valores distintas (representación entera), o bien escogiendo  $m$  posiciones, y asignándole un 1 a esas posiciones, y mantener un cero al resto de posiciones (representación binaria).

*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 con  $k=3$ , consistente en elegir aleatoriamente tres individuos (permitiendo repetición) 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).

*Operador de mutación:* Se considerará el operador de intercambio explicado en las transparencias del Seminario 3. Para aplicarlo, o bien se cambia uno de los valores enteros por otro, o bien se intercambia el valor del gen  $a$  mutar  $x_i$  por el de otro gen  $x_j$  escogido aleatoriamente con el valor contrario.

*Operador de cruce:* Se emplearán los dos operadores de cruce explicados en las transparencias del Seminario 3. Uno de ellos será el cruce en dos puntos sin orden, que requiere reparación porque puede generar repeticiones, y el otro el cruce basado en el orden. **Esto resultará en el desarrollo de cuatro AGs distintos, dos generacionales (AGG-sinorden y AGG-conorden) y dos estacionarios (AGE-sinorden y AGE-conorden).**

*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) compiten entre sí, y el mejor de ellos sustituye a la peor solución de la población actual, en caso de ser mejor que ella. Se podría hacer competir ambos descendientes, pero escoger solo el mejor es más eficiente, y tiende a dar mejores resultados.

## 4.2 Valores de los parámetros

El tamaño de la población será de 30 cromosomas. La probabilidad de cruce será 0,7 en el AGG y 1 en el AGE (siempre se cruzan los dos padres), en caso de decimales se redondeará hacia arriba. La probabilidad de mutación (por individuo) será de 0,1 en ambos casos. El criterio de parada en las dos versiones del AG consistirá en realizar 1000 evaluaciones de la función objetivo. Será necesario hacer 5 ejecuciones por problema, y aplicar la media para cada variable de salida (fitness y tiempo).

## 5 ALGORITMOS MEMÉTICOS

### 5.1 Descripción

El AM consistirá en hibridar el algoritmo genético generacional (AGG) con el operador de cruce que dé mejor resultado. A dicho operador se combinará con la BL totalmente aleatoria desarrollada en la Práctica 1. Para poder usar el código original sin cambiar a la hora de aplicar la BL será necesario pasar de la codificación binaria a la codificación de conjunto de enteros. 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 (donde  $N$  es el tamaño de ésta).

### 5.2 Valores de los parámetros

El tamaño de la población del AGG será de 30 cromosomas. Las probabilidades de cruce y mutación serán 0,7 y 0,1 (por individuo) en ambos casos (en caso de decimales se redondeará hacia arriba). Se detendrá la ejecución de la BL aplicada sobre un cromosoma bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan evaluado 20 vecinos distintos en la ejecución. El criterio de parada del AM consistirá en realizar 1000 evaluaciones de la función objetivo, teniendo en cuenta, por supuesto, las evaluaciones realizadas durante la BL. Se ejecutará tantas veces como los algoritmos genéticos.

## 6 TABLAS DE RESULTADOS

Se diseñará una tabla para cada algoritmo (AGG-1, AGG-2, AGE-1, AGE-2, AM-(10,1.0), AM-(10,0.1) y AM-(10,0.1mej)) donde se recojan los resultados de la ejecución de dicho algoritmo al conjunto de casos del problema. Tendrá la misma estructura que la Tabla 1 del guión de la Práctica 1.

Igualmente, se construirá una tabla de resultados global que recoja los resultados medios de calidad y tiempo de todos los algoritmos considerados para cada problema, según se ve en la tabla 1.

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
LSall	x	x	x	x
LSsmall	x	x	x	x
AGG-sinorden	x	x	x	x
AGG-conorden	x	x	x	x
AGE-sinorden	x	x	x	x
AGE-conorden	x	x	x	x
AM(10, 1.0)	x	x	x	x
AM(10, 0.1)	x	x	x	x
AM(10, 0.1mej)	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
LSall	x	x	x
LSsmall	x	x	x
AGG-sinorden	x	x	x
AGG-conorden	x	x	x
AGE-sinorden	x	x	x
AGE-conorden	x	x	x
AM(10, 1.0)	x	x	x
AM(10, 0.1)	x	x	x
AM(10, 0.1mej)	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 los de los distintos métodos de BL corresponden a los de la Práctica 1.

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.

## 7 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:

- 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**): 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 y los operadores comunes, como los operadores de selección, mutación y los distintos operadores de cruce.
- 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**.

Para esta primera práctica, se incluirán al menos las descripciones en pseudocódigo de:

- Para los AGs, el esquema de evolución y de reemplazamiento considerados.
  - Para los AMs, el esquema de búsqueda seguido por cada algoritmo en lo que
- 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.

## 8 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.**