Grado en Ingeniería Informática

Metaheurísticas

Curso 2020/2021

Universidad de Jaén

Práctica 3

1. Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de las Metaheurísticas basadas en adaptación social, y en concreto de los Sistemas de Colonias de Hormigas.

La fecha límite de entrega será el **martes 12 de enero de 2020 antes de las 23:59 horas**. La entrega se realizará a través de la plataforma virtual.

2. Trabajo a realizar

Este guión será una continuación de los anteriores, por tanto, el trabajo a realizar y los casos de estudio permanecen sin cambios.

3. Sistemas de Colonias de Hormigas

Se aplicará un sistema de colonias de hormigas tal y como hemos visto en teoría, pero para el problema de la máxima diversidad. A continuación, entramos en más detalle sobre este sistema.

Como el problema que estamos trabajando es el mismo **tanto la representación de la solución, como la función objetivo se mantiene idéntica a los guiones anteriores.**

Estamos ante un algoritmo constructivo y por tanto la solución se va generando paso a paso (igual que un algoritmo *greedy*) pero hay que tener en cuenta los distintos componentes del algoritmo:

- *Ciclo de vida del sistema:* El sistema de colonias de hormigas viene controlado por un ciclo de vida o iteraciones totales, en el que se construyen tantas soluciones de población de hormigas como iteraciones tengamos en este ciclo.

- *Población de hormigas:* Este sistema de colonias de hormigas estará formado por una población con un número determinado de hormigas que siempre permanecerá fijo.
- *Matriz de heurística:* Esta matriz contiene los valores de la máxima diversidad **invertidos (*1/valor*)**, al igual que hacemos en la transparencia 39 de la Unidad 4.1.
- *Memorias:* Los sistemas de colonias de hormigas cuentan con una matriz de información o matriz de feromonas que es la forma de colaborar entre las hormigas de las distintas poblaciones para conseguir obtener la mejor solución posible. Es importante anotar que cuando se incluye un elemento dentro de una solución constructiva (en este problema) hay que actualizar la relación entre el elemento introducido y todos los ya presentes en la propia solución.
- *Demonio:* Este componente del sistema de colonias de hormigas se encarga de realizar todos los procesos de actualización y evaporación de la feromona al final de cada iteración.
- *Regla de la transición:* Este mecanismo es clave en los sistemas de colonias de hormigas, sin embargo, en un problema como el de máxima diversidad que vamos construyendo paso a paso, el número de posibles caminos es demasiado grande y para restringir el número de soluciones posibles vamos a emplear una lista restringida de candidatos (LRC), de la siguiente forma:

```
Calcular distancia de todos los elementos no
seleccionados
Obtener la LRC
```

$$LRC = \{s_i \in S \mid Dist(s_i, Sel) \geq d_{min} + \delta(d_{max} - d_{min})\}$$

```
Aplicar la regla de la transición entre todos los
elementos de la LRC
```

o Parámetros del algoritmo

Se detallan a continuación todos los parámetros de los algoritmos, considerando todas las fórmulas de la Unidad 4.1:

- Número de iteraciones: **10.000**.
- Tamaño de la población: **10** hormigas (*m*).
- Feromona inicial: $1/(m*C_{greedy})$
- Valores de la transición: α*=1 y 2,* β*=1 y 2*

- Valor de la regla de la transición: $q_0=0.95$
- Actualización de la feromona: ρ=0.1 (recordamos que hemos invertido y por tanto estamos minimizando)

$$\Delta\tau_{rs}^{mejor} = \frac{1}{C(S)}$$

- Actualización local de la feromona: ϕ=0.1
- Valor del parámetro para LRC: δ=0.6

NOTA: C_{greedy} es el coste de la solución greedy del guión 1 para el problema.

En resumen, tenemos los siguientes algoritmos:

- ***SCH*** $\alpha=1, \beta=1$
- ***SCH*** $\alpha=1, \beta=2$
- ***SCH*** $\alpha=2, \beta=1$

o Tablas de resultados y análisis

Seguiremos trabajando el mismo formato que en guiones anteriores.

Para ello, se cogerán los resultados obtenidos en la Práctica 2 y se incluirán los mismos obtenidos por los algoritmos diseñados en este guión. En primer lugar, se analiza cuál es el mejor SCH y luego se enfrenta con el mejor del guión 2, y el mejor del guión 1.

4. Documentación y normativa

En este caso también seguiremos con la misma normativa que ya hemos trabajado en la Práctica 1 y 2:

- Solo se admitirá el formato **ZIP**. No se corregirán guiones en cualquier otro formato al indicado.

- La memoria que acompañará en la entrega será exclusivamente en formato **PDF** y se incluirá una portada con:
  - Identificación de los dos alumnos (Nombre, apellidos y DNI).
  - Identificación del guión.

  - Identificación del grupo de los alumnos.
  - Algoritmos implementados en la práctica.

- Los nombres de los ficheros siguen el mismo formato anterior junto con el código que se detalla más adelante donde:

- Una breve descripción del problema.

- Descripción en **pseudocódigo** de las **estructuras de los SCH** y de todas aquellas **operaciones relevantes**. El pseudocódigo **deberá forzosamente reflejar la implementación y el desarrollo realizados incluyendo los diseños y representación del sistema, y** no ser una descripción genérica extraída de las transparencias de clase o de cualquier otra fuente. La descripción no deberá ocupar más de **2 páginas**.

- Breve explicación del **procedimiento considerado para desarrollar la práctica**: implementación a partir del código proporcionado en prácticas o a partir de cualquier otro, o uso de un framework de metaheurísticas concreto. Inclusión de un pequeño **manual de usuario describiendo el proceso para que el profesor de prácticas pueda replicarlo**.

- Experimentos y análisis de resultados:
  - Descripción 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.
  - Análisis de tiempos y desviaciones, e influencia del tamaño de la instancia en los resultados obtenidos, etc.
  - Por último, es clave obtener un fichero de registro que vaya almacenando la información de la ejecución del algoritmo indicando las soluciones, en qué generación fueron creadas y costes que va obteniendo en las distintas iteraciones los mejores de cada generación.
- 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 los ficheros de datos de los casos del problema y 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 FUENTES 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.). La versión ejecutable de los programas y los ficheros de datos se incluirán en un subdirectorio del raíz de nombre BIN. En este mismo directorio se adjuntará 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.

La entrega se llevará a cabo a través de la actividad correspondiente en ILIAS. El plazo de entrega termina el 12 de enero de 2021 a las 23:59.

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.