



Objetivos

1. Aprender diferentes codificaciones para resolver problemas de optimización por colonia de hormigas.
2. Aprender a inicializar correctamente los parámetros de entrada.
3. Resolución de problemas reales.

Introducción

En esta práctica se van a aplicar los conceptos aprendidos en las clases de EB (Tema 6) relativos a la estrategia de optimización por colonia de hormigas (muy conocido por su acepción inglesa *ants colony optimization*). Se muestra el esquema de búsqueda:

Inicialización de parámetros:

Asignación de la cantidad inicial a los rastros de feromona $\tau[i][j] \leftarrow \tau_0$

nIter: número de iteraciones

α, β

m: número de hormigas

Para it=1 hasta Número_de_Iteraciones hacer

/* Colocar a cada hormiga en diferentes nodos al comenzar */

Paso 1. Para k=1 hasta m hacer

$L[k][1] \leftarrow$ nodo inicial

/* Construcción de soluciones por las hormigas */

Paso 2. Para i=2 hasta Número_de_Nodos hacer

Para k=1 hasta Número_de_Hormigas hacer

$L[k][i] \leftarrow$ Regla_transición ($L[k], \tau, \eta$)

/* Mejor solución encontrada */

Paso 3. Para k=1 hasta Número_de_Hormigas hacer

$Coste[k] \leftarrow C(L[k])$

$Mejor_Actual \leftarrow$ Mejor ($L[k]$)

/* Actualización feromonas */

Paso 4. Para i=1 hasta Número_de_Nodos hacer

Para j=1 hasta Número_de_Nodos hacer

Actualización_feromona($\tau[i][j], L, C(L), \rho$)

Paso 5. Si $C(Mejor_Actual)$ es mejor que $C(Mejor_Global)$

$Mejor_Global \leftarrow$ Mejor_Actual



Antes de desarrollar la práctica es muy recomendable que el estudiante dedique unos minutos a recordar los conceptos básicos que se vieron en teoría. En concreto, es fundamental recordar:

1. Hormiga artificial.
2. Feromona.
3. Evaporación.
4. Regla probabilística de transición.
5. Condición de parada.

Problema propuesto

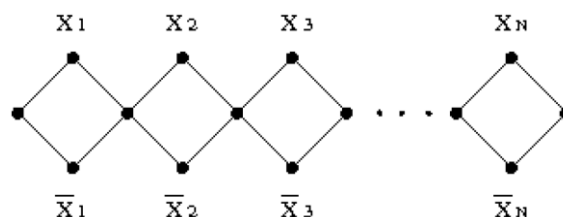
Se pide implementar en Java un algoritmo de ACO para el problema de la mochila, siguiendo las directrices generales aquí descritas. No obstante, el estudiante puede realizar alguna modificación o plantear una solución alternativa, siempre que la justifique debidamente.

Suponga que se dispone de N objetos y que cada uno de ellos tiene asociado un peso (w) y un valor (v). El problema consiste en encontrar el subconjunto de objetos de valor máximo que cumpla la restricción de que la suma de sus pesos sea menor o igual que una cota C prefijada (la capacidad de la mochila).

Se propone que siga el siguiente pseudocódigo:

Inicializar los rastros de feromonas
Colocar hormigas en posición inicial
Repetir
 Para K desde 1 **Hasta** el numero de hormigas **Hacer**
 Construir la solución para la hormiga K
 Seleccionar la mejor solución
 Actualizar los rastros de feromonas
Hasta Alcanzar el número de iteraciones

Dado que el objetivo del problema es encontrar un subconjunto y las hormigas lo que hacen es buscar caminos, se puede considerar el siguiente grafo:



Su interpretación es la que sigue. Para cada hormiga, si su camino pasa por el nodo X_i , interpretaremos que el objeto i -ésimo pertenece al subconjunto, y si por el contrario el camino pasa por \bar{X}_i entonces interpretaremos que el objeto i -ésimo no pertenece al subconjunto.

También se puede realizar el diseño prescindiendo del grafo y asociando los niveles de feromona directamente a los objetos, en lugar de a las aristas. El estudiante puede escoger libremente entre ambas opciones, sin que por ello se vea mermada su nota.



En cuanto a la construcción de una solución por una hormiga, en cada instante la probabilidad $p(i)$ de que un objeto X_i sea elegido depende del nivel de feromona $\tau(i)$, del peso del objeto $w(i)$ y de su valor $v(i)$:

$$p(i) = \frac{f(i)}{\sum_{j=1, \dots, N} f(j)}^{-1}$$
$$f(i) = \tau(i)^\alpha w(i)^{-\beta} (1 - v(i)^{-1})$$

Tanto α como β pueden ajustarse experimentalmente.

En cuanto a la actualización del nivel de feromona τ , hay que considerar por un lado que en cada vuelta del bucle se añade una cantidad a las aristas del mejor camino (o a los objetos de la mejor solución, según la estrategia escogida), y por otro lado que los niveles anteriores deben disminuirse en función de una tasa de evaporación. La cantidad a añadir y la tasa de evaporación se pueden ajustar experimentalmente.

El algoritmo debe diseñarse para que lea la instancia que tiene que resolver a través de un fichero de texto, donde aparecerán tres columnas (una línea por cada objeto): la primera columna con el índice que identifica al objeto, y las otras dos para el peso y el valor del objeto, respectivamente. Se presentarán al menos cinco ficheros con instancias. Si, por cualquier motivo no se pudiera implementar la lectura por fichero de texto, se podrán escribir estas instancias dentro del propio código, aunque esta solución sí acarreará una penalización de 1.5 puntos (la nota máxima a la que se podrá optar es 8.5).

Entregables

En la plataforma se habilitará una actividad para entregar esta práctica con fecha límite lunes 8 de mayo, a las 23:59.

Cada alumno, de manera individual, deberá subir un único fichero comprimido que contenga:

1. El proyecto completo comprimido. Los códigos no comentados debidamente estarán penalizados con 1.5 puntos (la nota máxima a la que se podrá optar es 8.5).
2. Los cinco ficheros con las instancias usadas (si decidió trabajar con ficheros de texto).
3. Documento PDF en el que se expliquen las decisiones tomadas y cómo ha ajustado los valores necesarios para que el algoritmo converja en un tiempo adecuado. Igualmente, incluya una estimación del número de iteraciones necesarias para obtener el resultado óptimo.