



Práctica 2 Metaheurísticas Curso 2021/2022



Esta segunda práctica está orientada al desarrollo de algoritmos genéticos con distintas representaciones. La práctica se divide en dos partes: *Codificación binaria*; *Codificación entera*. El alumno deberá realizar todo lo que se solicita en este documento, y generar un informe detallado en el que se justifiquen las respuestas y los análisis llevados a cabo. Tanto el **código nuevo generado como el informe detallado deberán entregarse por el grupo de prácticas antes de la fecha establecida**, teniendo en cuenta que esta segunda práctica tiene una duración de 3 sesiones.

1. *Codificación binaria*. En este primer apartado vamos a trabajar con una representación binaria del problema de la mochila para resolverlo con un algoritmo genético generacional. El código (incompleto) en Python se encuentra en el fichero `Kanpsack.py`. La representación binaria $[1, 0, 1, 1, 1]$ indica que se han escogido los objetos 0, 2, 3, y 4. Analiza el código y comprueba que es correcto y cumple con los requisitos. A continuación, completa el código (sólo función *aplicarOperadoresGeneticos*). Utiliza el operador de cruce y mutación que quieras. Puedes variar el número de objetos, así como los pesos y los precios para comprobar que tu algoritmo funciona.

Contesta a las siguientes cuestiones de manera justificada y ayudándote de gráficas cuando sea necesario:

- ¿Cómo se comporta este algoritmo a medida que cambiamos el número de soluciones, generaciones, tamaño del torneo, probabilidad de cruce y probabilidad de mutación? Amplía el problema con más objetos para ver el comportamiento del algoritmo.
- ¿Obtiene siempre la mejor solución? ¿Por qué? ¿De qué depende?
- Modifica el código para incorporar elitismo. La mejor solución se guarda en la élite y nunca se pierde hasta que venga una nueva mejor. Esta solución es la que se devuelve al final. ¿Has conseguido mejorar? ¿Por qué?
- Hasta ahora, hemos partido de soluciones válidas al comenzar el algoritmo. Cámbialo para que pueda generarse cualquier solución. ¿Afecta al rendimiento? Analiza esta nueva situación

2. *Codificación entera*. En este segundo apartado vamos a trabajar con una representación entera del problema de la mochila para resolverlo con un algoritmo genético generacional. A diferencia de antes, ahora cada objeto puede aparecer más de una vez por lo que la representación binaria $[1, 0, 1, 1, 1]$ ya no es válida. Usa una representación $[0, 2, 0, 1, 3]$ para indicar que se tienen 2 objetos del tipo 1, 1 objeto del tipo 3 y 3 objetos del tipo 4. Utiliza el código del apartado anterior para ajustarlo al nuevo problema y representación. Puedes variar el número de objetos, así como los pesos y los precios para comprobar que tu algoritmo funciona.

Contesta a las siguientes cuestiones de manera justificada y ayudándote de gráficas cuando sea necesario:

- ¿Cómo se comporta este algoritmo a medida que cambiamos el número de soluciones, generaciones, tamaño del torneo, probabilidad de cruce y probabilidad de mutación? Amplía el problema con más objetos para ver el comportamiento del algoritmo.
- ¿Obtiene siempre la mejor solución? ¿Por qué? ¿De qué depende?
- Modifica el código para incorporar elitismo. La mejor solución se guarda en la élite y nunca se pierde hasta que venga una nueva mejor. Esta solución es la que se devuelve al final. ¿Has conseguido mejorar? ¿Por qué?
- Hasta ahora, hemos partido de soluciones válidas al comenzar el algoritmo. Cámbialo para que pueda generarse cualquier solución. ¿Afecta al rendimiento? Analiza esta nueva situación