

WUOLAH



TEAM_GETPPID__
www.wuolah.com/student/TEAM_GETPPID__



Metahuristica Tema 1.pdf

Resúmenes Metaheurística



3º Metaheurísticas



Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba**

 **escuela
de negocios**
CÁMARA DE SEVILLA

MÁSTER EN DIRECCIÓN Y GESTIÓN DE RECURSOS HUMANOS

www.mastersevilla.com

Inscríbete



BECAS

Tema 1: Introducción a las Metaheurísticas

Contenido

1. Introducción	2
2. Optimización de problemas.....	2
2.1 Teoría de la complejidad	2
2.2 Teoría computacional	2
3. Algoritmos de búsqueda	2
3.1 Viajante de comercio.....	2
3.2 Mochila.....	3
3.2.1 Simple.....	3
3.2.2 Múltiple	3
3.3 Tipos de codificación.....	3
4. Algoritmos aproximados	3
4.1 Elementos de un algoritmo aproximado	3
4.2 Heurística Greedy	3
4.3 Metaheurísticas	4
5. No free lunch.....	4

1. Introducción

Hoy en día existen muchos problemas de optimización que son muy difíciles de resolver o que no pueden ser resueltos en un tiempo razonable, por eso hacemos uso de los **algoritmos aproximados**, que aunque sacrifican exactitud, obtienen soluciones aproximadas.

- **Heurísticas:** Diseñadas para un problema, por lo que son dependientes.
- **Metaheurísticas:** Ofrecen como ventaja una capacidad de generalización para varios problemas, además de mayor rapidez y robustez mediante el uso de metáforas naturales.

2. Optimización de problemas

2.1 Teoría de la complejidad

La definición de la teoría de la complejidad se puede definir como el orden del algoritmo, es decir como crece el coste computacional según aumenta el tamaño del problema. Definimos entonces la complejidad del problema como el mejor algoritmo que lo resuelva.

2.2 Teoría computacional

En la teoría computacional se definen los siguientes conjuntos de problemas:

- Conjunto de problemas **P**: Soluciones en tiempos razonables (Orden polinomial).
- Conjunto de problemas **NP**: Se puede determinar en tiempo razonable (Orden polinomial) si una solución que se proporciona es correcta al problema.

- Todos los problemas que están en P están también en NP.

- No está tan claro que un problema que esté en NP esté también en P

3. Algoritmos de búsqueda

La mayoría de este tipo de problemas son denominados NP-hard, es decir, los algoritmos exactos son ineficientes.

Los problemas NP plantean una función objetivo que guía el proceso y busca el máximo o mínimo de una función, donde el espacio de búsqueda serán los valores de las variables que serán evaluados.

3.1 Viajante de comercio

Tenemos una serie de nodos y un grafo con estos nodos conectados. Se debe de buscar el ciclo que pase por todos los nodos minimizando el camino (Sin repetir). A mayor número de nodos mayor será la duración de la resolución. Con 50 nodos el tiempo de resolución deja de ser razonable.



3.2 Mochila

3.2.1 Simple

Tenemos una mochila con una capacidad máxima y hay que llenarla maximizando el beneficio (Optimizando el valor final) sin pasarnos del máximo. Cada objeto tiene un peso y un coste.

La solución es una secuencia binaria (0 no entra el objeto, 1 si entra).

3.2.2 Múltiple

A diferencia del anterior, cada objeto en vez de entrar 1 vez puede entrar 0 o m veces. Por lo que la solución es un vector discreto (0 no entra, m entra m veces).

3.3 Tipos de codificación

Hemos visto en los problemas anteriores que podemos codificar la solución de distintas formas:

- **Binaria** (1 0 0 1 0 1): Mochila simple
- **Entera o discreta** (1 0 4 1 5 2): Mochila múltiple.
- **Ordinal** (3 1 5 2 4 6): Viajante de comercio. No hay repetición.
- **Real** (2.45 1.01): Optimización de funciones.

4. Algoritmos aproximados

Aportan soluciones cercanas a la óptima (Sacrifican precisión) en problemas NP en tiempos razonables.

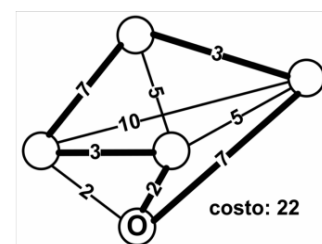
Búsqueda se usa para construir/mejorar soluciones y obtener el óptimo o soluciones casi-óptimas.

4.1 Elementos de un algoritmo aproximado

- **Solución**
- **Entorno**: Soluciones cercanas.
- **Movimiento**: Transformación de una solución a otra. Las soluciones debes de ser modificables (De un estado a otro) fácilmente.
- **Evaluación**: Calidad de la solución y función objetivo.

4.2 Heurística Greedy

Se trata de una solución construida seleccionando de forma iterativa los elementos de menor coste. Las soluciones que aporta no tienen por qué ser las óptimas. En la figura podemos ver un ejemplo del viajante de comercio que empieza en el nodo 2.



4.3 Metaheurísticas

- Familia de algoritmos aproximados
- Suelen ser procedimientos iterativos
- Guían una heurística subordinada de búsqueda
- Combina diversificación (exploración global) con intensificación (explor. local)

Ventajas	Inconvenientes
Algoritmos de propósito general	Aproximados, no exactos
Gran éxito	Estocásticos
Fácil de implementar	Sin base teórica establecida
Fácil de paralelizar	

Mayoritariamente estas están basadas en la naturaleza (Hormigas)

5. No free lunch

Este teorema define que no existe ningún algoritmo que sea universal, es decir que sea capaz de resolver todos los problemas.