

PROBLEMAS COMPUTACIONALES

Problemas “sencillos” y problemas “difíciles”.

Sencillos: Ordenamiento de valores, multiplicación de matrices.

Difíciles: problemas combinatorios.

¿ Cómo evaluar la complejidad computacional ?

Comportamiento en caso promedio y peor caso.

Independiente de la arquitectura y lenguaje de programación (*a priori*).

Se analiza el **orden de magnitud** del número de instrucciones de un algoritmo, en función del **tamaño de la entrada**.

Un algoritmo de complejidad $O(g(n))$ ejecutado con datos de tamaño n , tendrá un tiempo de ejecución menor a $|g(n)|$.

PROBLEMAS NP

Es necesario utilizar algoritmos de resolución **más eficientes** que la búsqueda exhaustiva, a medida que se incrementa el tamaño del problema.

La complejidad del algoritmo de resolución del problema se incrementa de manera superpolinomial con el tamaño de la entrada. Ejemplo: TSP $O(n^2 * 2^n)$

Una clase particular de problemas NP son los **problemas de optimización**.

Objetivo: hallar la(s) solución(es) óptima(s) de un problema (de acuerdo a una función objetivo determinada).

MÉTODOS de RESOLUCIÓN

- Técnicas heurísticas

Basadas en procedimientos conceptualmente simples encuentran soluciones de buena calidad (no necesariamente hallan la solución óptima) a problemas difíciles, de un modo sencillo y eficiente.

Los procedimientos de búsqueda a menudo están basados en el sentido común o emulación de fenómenos bien conocidos.

“Heurística” deriva del griego *heuriskein* (*ὁρίσκειν*), que significa “encontrar” o “descubrir”.

Heurísticos

- ◆ **Métodos heurísticos:** la rapidez del proceso es tan importante cómo la calidad de la solución obtenida
- ◆ Definición formal:
“Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.”

Métodos exactos Vs. Métodos heurísticos

◆ Exactos

- Proporcionan una solución óptima
- El **tiempo** invertido para encontrar la solución óptima de un problema difícil puede llegar a ser desproporcionado

◆ *Heurísticos*

- Proporcionan una buena solución del problema no necesariamente óptima
- El **tiempo** invertido puede ser muy reducido

Heurísticos

Razones para utilizar métodos heurísticos

- ◆ El problema es de una naturaleza tal que no se conoce ningún método exacto para su resolución
- ◆ Existe un exacto para resolver el problema pero su uso es computacionalmente muy costoso
- ◆ El método heurístico es más flexible que el exacto, permitiendo, la incorporación de condiciones de difícil modelado
- ◆ El método heurístico se utiliza como parte de un procedimiento global que garantiza el óptimo de un problema. Existen dos posibilidades
 - El método heurístico proporciona una buena solución inicial de partida (B&B)
 - El método heurístico participa en un paso intermedio del procedimiento (Simplex)

- ◆ **Desventajas** de los Algoritmos heurísticos
 - Dependen en gran medida del **problema concreto** para el que se han diseñado
 - Las técnicas e ideas aplicadas a la resolución de un problema son **específicas** de éste
 - Es difícil **trasladar** el aprendizaje a otros problemas (han de particularizarse en cada caso)

Metaheurísticas

- ◆ En los últimos años han aparecido una serie de métodos bajo el nombre de *Metaheurísticos*
 - propósito de obtener mejores resultados que los alcanzados por los heurísticos tradicionales
- ◆ El término metaheurístico fue introducido por Fred Glover en 1986

“Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que los heurísticos clásicos no son efectivos. Los Metaheurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos”

METAHEURÍSTICAS

- “Heurísticas de nivel más alto de abstracción” (Glover, 1986).
- Características:
 - Estrategias que guían el proceso de búsqueda, incluyendo en general heurísticas subordinadas.
 - De uso genérico (no específicas para una clase de problemas).
 - Admiten descripción a nivel abstracto.
 - Deben instanciarse para cada clase de problemas.

METAHEURÍSTICAS

- **Objetivos:**
 - Encontrar rápidamente soluciones **factibles** (puede ser en sí mismo un problema NP-difícil).
 - Encontrar soluciones factibles de **buena calidad** (valores de la función objetivo cercanos al óptimo).
 - Recorrer el espacio de soluciones sin quedar “atrapados” en una zona.
 - **Nociones fundamentales:**
 - **Exploración** del espacio de búsqueda.
 - **Explotación** de las soluciones obtenidas.

METAHEURÍSTICAS

- **Múltiples clasificaciones, de acuerdo a los criterios utilizados:**
 - Inspiradas en la naturaleza o no.
 - Aleatorias o Determinísticas.
 - Poblacionales o única solución.
 - Constructivas o Trayectorias.
 - Con o Sin Memoria.

Ejemplos de metaheurísticas

Trayectoriales

- ◆ *Búsqueda tabú*
- ◆ *Búsqueda de vecindad variable*
- ◆ *Búsqueda local guiada*
- ◆ *Métodos multi-arranque*
- ◆ *Concentración heurística*
- ◆ *GRASP*
- ◆ *Recocido simulado*
- ◆ *Aceptación del umbral*
- ◆ *Métodos ruidosos*
- ◆ *FANS*
- ◆ *POPMUSIC*
- ◆ *Búsqueda local iterativa*

Poblacionales

- ◆ *Búsqueda dispersa*
- ◆ *Re-encadenamiento de trayectorias*
- ◆ *Algoritmos evolutivos*
- ◆ *Algoritmos genéticos*
- ◆ *Algoritmos meméticos*
- ◆ *Algoritmos de estimación de la distribución*
- ◆ *Equipos asíncronos*
- ◆ *Sistemas de hormigas*
- ◆ *Algoritmos culturales*
- ◆ *Inteligencia de enjambre*