Complejidad Computacional

Clasificación de Problemas

Teoría de la Complejidad

- Estudia la manera de clasificar algoritmos como buenos o malos.
- Estudia la manera de clasificar problemas de acuerdo a la dificultad inherente de resolverlos.

Introducción

Problema:

Pregunta general que debe contestarse.

Elementos de un problema:

Lista de parámetros (variables libres).
Descripción de las características de la solución.

Instancia:

Especificación de valores de los parámetros.

Problema

Ejemplo:

Problema del agente viajero.
Cual es la pregunta?

Parámetros:

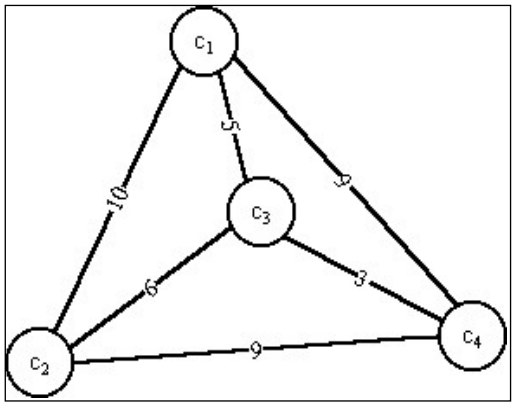
Lista de ciudades y distancia entre ellas.

Solución:

Secuencia de ciudades que minimiza la distancia de un recorrido que inicia en c_1, visita en secuencia todas las ciudades y regresa a c_1 desde la última ciudad c_m.

Instancia de un Problema

Problema del agente viajero.



Solución:

<c_1, c_2, c_4, c_3>
Con una longitud de 27

Encontrar todas las posibles rutas y su longitud

Problemas Intratables

Consistentemente intratables:
•Son aquellos que son tan difíciles que ni un algoritmo de tiempo no polinomial puede resolverlo.

Aparentemente intratables:
•El problema es tan difícil que se requiere una cantidad de tiempo exponencial para encontrar una solución.

•La solución es tan extensa que no puede ser expresada como una función polinomial de la entrada. Ejemplo agente viajero con hasta B CIUDADES POR RECORRER.

Tipos de Problemas

- Problemas decidibles: existen algoritmos capaces de resolverlos
- Problemas indecidibles: no existen algoritmos capaces de resolverlos.
- Ejemplo: Dado un programa de cómputo y su entrada, este siempre para?.
- Problemas aparentemente intratables:
  - No-determinísticos. Intratables→ problemas NP

Problemas Polinomiales y No-Polinomiales (P y NP)

- Un problema es Polinomial (P) si existe un algoritmo determinístico polinomial que lo resuelva.
- Un problema es No-Polinomial (NP) si no existe un algoritmo determinístico polinomial que lo resuelva.

Tipos de Problemas

Intratables: No se ha descubierto un algoritmo que los resuelva.

Clase P: Problemas de decisión que tienen un algoritmo de tiempo polinomial en una máquina determinista que los resuelva.

Clase NP: Problemas de decisión que tienen un algoritmo de tiempo polinomial en una máquina no determinista que los acota.

Clase NP-Completos: Problemas NP más difíciles de resolver.

Teoría de NP-Completes

1.- La clase de problemas P es subconjunto propio de la clase NP.

2.- Un problema es NP-Completo si cualquier otro problema en NP puede ser transformado en él.

3.- Si se encuentra un algoritmo de tiempo polinomial para una máquina determinista que resuelva un problema NP-Completo, entonces se resuelven todos los problemas NP y, entonces:

¿P = NP?

Teorema de Cook

1.- **Reducibilidad en tiempo polinomial:** Transformar un problema en otro usando un algoritmo determinista de tiempo polinomial.

2.- **Clase NP.** Problemas de decisión que son resueltos en tiempo polinomial por una máquina no determinista.

3.- **Satisfactibilidad es NP-Completo**: Todos los problemas NP se pueden reducir al problema de Satisfactibilidad.

4.- **Problemas NP-Completos:** Si SAT es reducible a un problema NP, entonces este NP está en NP-completo.

Satisfactibilidad

Es el problema de decidir si existe una asignación de valores de verdad a los átomos de una fórmula proposicional que la hacen verdadera (satisfactible).

Ejemplo:

(P or ¬Q) and (Q or R) and (¬R or ¬P)

Si P = Q = verdadero y R = falso, la fórmula es satisfactible.

Importancia de SAT

•**Satisfactibilidad es NP:** No existe un algoritmo en tiempo polinomial que lo resuelva en una máquina determinista.

•**Satisfactibilidad es NP-Completo:** Cook demostró que cualquier problema en NP puede ser reducido a SAT.

•SAT fue el primer problema reconocido como NP-Completo.

•Si se encuentra un algoritmo en tiempo polinomial para una máquina determinista que resuelva SAT: Todos los problemas NP se hacen P.

Enfoques para SAT

Métodos Completos: Determinar si la fórmula tiene o no una asignación satisfactible.

Programación lineal entera,

Métodos Incompletos: Determinan si la fórmula es satisfactible, usando búsquedas heurísticas.

GSAT
Algoritmos Genéticos
Recocido Simulado
Búsqueda Tabú, Grasp

Problemas NP-Duros

•Los problemas considerados como los más difíciles en NP-Completo se definen como NP-Duros

•Las versión de optimización de un problema NP-Completo es NP-duro.