Estudio de complejidad y aproximabilidad

Aitor Godoy Fresneda

Universidad Complutense de Madrid
Facultad de Ciencias Matemáticas

Julio 2020

Índice

Motivación.

- ¿P = NP?
- Soluciónes óptimas para problemas NP-Completos costosas.
- Algoritmos de aproximación polinómicos.

Preliminares.

La clase P.

Decimos que un problema pertenece a la clase P cuando existe una máquina de Turing determinista que lo resulеve en tiempo polinómico.

Ejemplo.

- El problema de primalidad.
- Problema de ordenación de un vector de enteros.
- Encontrar el camino más corto entre 2 vértices de un grafo.

Preliminares.

La clase NP.

Decimos que un problema pertenece a la clase NP cuando existe una máquina de Turing no determinista que lo resuelve en tiempo polinómico.

Ejemplo.

- El problema de la mochila.
- El problema MIN TSP.
- El problema CLIQUE.

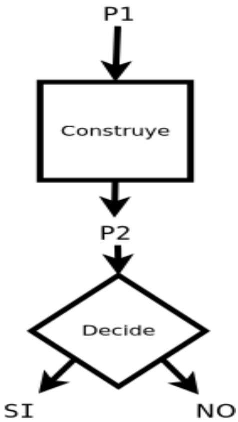
Observaciones.

- $P \subseteq NP$.
- La solución se puede comprobar en tiempo polinómico.

Preliminares.

Reducciones polinómicas.

- Transformación de P_1 a P_2 en tiempo polinómico.
- Transforma entradas de P_1 a P_2 en tiempo polinómico.
- Nueva entrada polinómica respecto de la entrada de P_1.



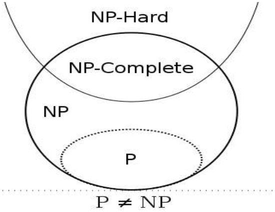
Preliminares.

Problemas NP-Duros y NP-Completos.

- Un problema P es NP-Duro si para todo problema P' $\in$ NP existe una reducción polinómica de P' a P
- Un problema P es NP-Completo si P es NP-Duro y P $\in$ NP

Ejemplo.

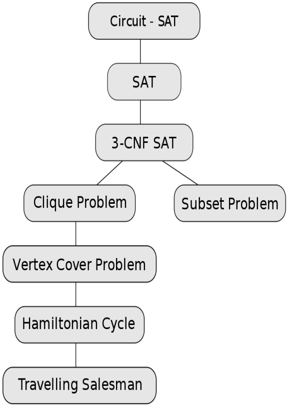
- El problema de la mochila.
- El Problema de PCI.
- El problema SAT.



Preliminares.

Ejemplo.

- El Teorema de COOK.
- Reducción de SAT a 3-SAT.
- Reducción de 3-SAT a PCI.
- Reducción de PCI a CLIQUE.



Clases de aproximación.

- Contraparte de optimización de los problemas de decisión.
- Buscamos soluciones "buenas" en tiempo polinómico.
- Las diferentes clases tienen una jerarquía.

La clase NPO.

- Contraparte de optimización de los problemas NP.
- 4-tupla (I, Sol, m, obj).
- Distinción entre maximización y minimización.

Clases de aproximación.

Dos paradigmas de aproximación principales:

Paradigma de aproximación estándar.

- $\rho_A(I) = \frac{m_A(I,S)}{opt(I)}$.
- $[1, \infty)$ para minimización.
- $(0, 1]$ para maximización.

Paradigma de aproximación diferencial.

- $\delta_A(I) = \frac{|\omega(I) - m_A(I,S)|}{|\omega(I) - opt(I)|}$.
- $[0, 1]$ para minimización.
- $[0, 1]$ para maximización.

Clases de aproximación.

Clases poco optimistas.

Exp-APX y Exp-DAPX.

El coeficiente de aproximación crece (o decrece) de forma exponencial

Poly-APX y Poly-DAPX.

El coeficiente de aproximación crece (o decrece) de forma polinómica.

Log-APX y Log-DAPX.

El coeficiente de aproximación crece (o decrece) de forma logarítmica.

Clases de aproximación.

Clases más optimistas.

APX y DAPX.

El coeficiente de aproximación es constante.

PTAS y DPTAS.

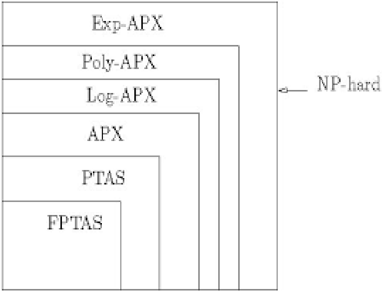
Podemos aproximar nuestra solución a la óptima tanto como queramos de modo que respecto del tamaño del problema obtengamos un algoritmo en tiempo polinómico pero respecto de cuánto queremos aproximar no necesariamente.

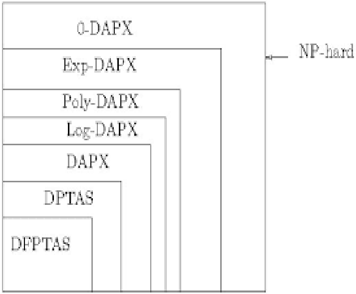
FPTAS y DFPTAS.

FPTAS es como PTAS pero si que es polinómico respecto de cuánto lo quieres aproximar.

Clases de aproximación.

Las clases de aproximación presenta una jerarquía:



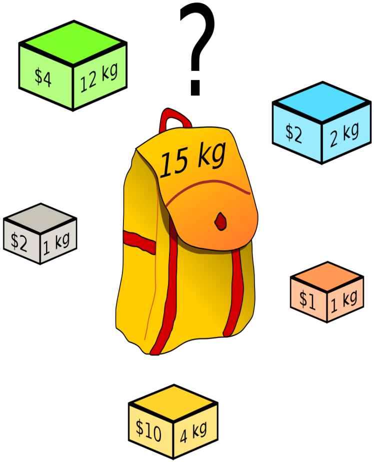


Aproximabilidad en problemas NP-Duros.

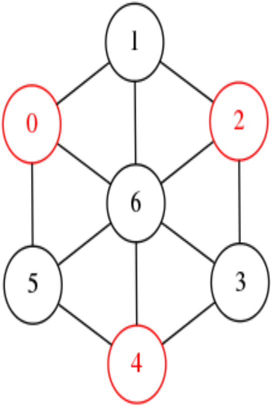
El problema de la mochila.

Está en FPTAS. El algoritmo de aproximación es el siguiente:

1. Fijamos un $\varepsilon \in (0,1)$ y consideramos la entrada $I' = ((a'_i, c_i)_{i=1,\dots,n}, b)$ con $a'_i = \lfloor \frac{a_i}{t} \rfloor$ con $t = \frac{a_{max}\varepsilon}{n}$.
2. Salida S := PROGRAMACIÓN DINÁMICA(I').



Aproximabilidad en problemas NP-Duros.



El problema del Conjunto Independiente maximal (PCI).

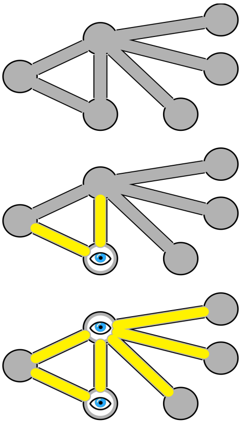
Está en Poly-Apx. Algoritmo IS:

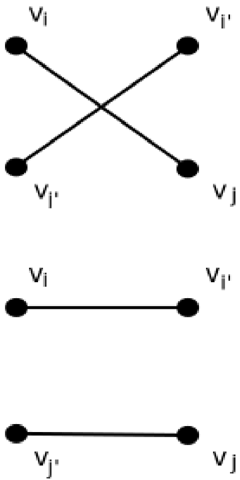
1. Resolver la relajación lineal del PCI para hallar V_0, V_1 y $V_{1/2}$.
2. Colorear $G[V_{1/2}]$ con a lo sumo $\Delta(G[V_{1/2}])$ colores. y definir $\hat{S}$ como el subconjunto de vértices de $G[V_{1/2}]$ cuyo color es el que más vértices cubre del grafo.
3. Devolver $S = V_1 \bigcup \hat{S}$

MIN VERTEX COVER (MVC).

Está en Apx. Algoritmo MATCHING:

1. Computar un matching maximal en G.
2. Devolver el conjunto C de los vértices de las aristas de M.





MIN TSP.

Está en DAPX. Algoritmo 2_OPT:

1. Construir un ciclo hamiltoniano T.
2. consideramos dos aristas a_{ij}, $a_{i'j'}$ y si procede cambiarlas las cambiamos por $a_{ii'}$, $a_{jj'}$.
3. repetir el paso 2 hasta que no haya que cambiar ninguna arista.
4. Devolver T.

Aproximabilidad mediante reducciones.

- Reducciones polinómicas para problemas de optimización.
- Preservan aproximabilidad.
- Transformación del coeficiente de aproximabilidad.

Aproximabilidad mediante reducciones.

Reducciones que preservan aproximabilidad.

Dados dos problemas NPO Π y Π' , una reducción que preserva aproximabilidad R de Π a Π' ($\Pi \leq_R \Pi'$) es una terna (f, g, c) de funciones cumputables en tiempo polinómico tal que:

- f transforma una entrada $I \in \mathbf{I}$ en una entrada $f(I) \in \mathbf{I'}$.
- g transforma una solución de Π' en una solución de Π.
- c transforma el coeficiente de aproximación de aproximación de Π' en uno de Π.

Aproximabilidad mediante reducciones.

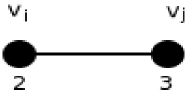
Esta clase de reducciones son muy útiles principalmente por dos motivos:

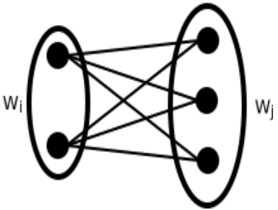
- Por motivos estructurales:
  - Gerarquía en los problemas NPO.
  - No todos los problemas son igual de difíciles de aproximar.
- Por motivos operacionales:
  - No siempre se puede hallar aproximabilidad directamente.
  - Herramienta muy útil para hallar nuevos resultados.

Aproximabilidad mediante reducciones.

Ejemplos.

1. Reducción de MAX PCIV a MAX PCI.
2. Reducción de MIN COLORING a MIN Partition Into Independent Sets.
3. Reducción de MAX PCI a MAX CLIQUE.





Conclusiones.

- La aproximabilidad es muy útil ya que no sabemos si $P = NP$.
- Necesaria ya que no se ha conseguido encontrar algoritmos “rápidos” para algunos problemas a la hora de hallar la solución óptima.
- Nos ayuda a comprender el mundo que nos rodea.
- No siempre vamos a poder hallar la mejor solución a nuestros problemas y vamos a tener que encontrar otra manera de hallar una solución lo suficientemente buena