Diseño de Algoritmos - Algoritmos II

Nazareno Aguirre
Departamento de Computación
Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Universidad Nacional de Río Cuarto

Clase 8: Algoritmos Greedy - Matroides

Algoritmos Greedy

Los algoritmos greedy proponen resolver problemas algorítmicamente (específicamente problemas de optimización) mediante la construcción incremental de una solución:

- se comienza con una solución parcial inicial

- se ejecuta una serie de pasos, en cada uno de los cuales se extiende la solución parcial actual tomando una decisión localmente optimal

- cada decisión de extensión es "irrevocable", no se vuelve atrás

Una sucesión de pasos localmente optimales no necesariamente deriva en un óptimo global. La expectativa es, sin embargo, que en general la sucesión de óptimos locales se aproxima a un óptimo global.

Corrección/Exactitud en Algoritmos Greedy

Un algoritmo greedy es correcto (o exacto) cuando, al terminar, siempre se obtiene un óptimo global.

¿Bajo qué circunstancias un algoritmo greedy es correcto?

Se requiere esencialmente de dos propiedades:

- subestructura optimal: se puede construir una solución optimal a una instancia del problema en base a soluciones optimales de instancias más pequeñas (subproblemas)

- elección greedy: una solución globalmente optimal se puede construir mediante una serie de elecciones localmente optimales

Demostrar que un algoritmo greedy es correcto suele ser más difícil que proponer/diseñar el algoritmo en sí. Veremos un concepto que suele ayudar en la tarea de diseñar algoritmos greedy correctos, o demostrar su corrección a posteriori.

Matroides

Un matroide es un par (S, l), donde

S es un conjunto finito

l es una familia no vacía de subconjuntos de S, denominados subconjuntos independientes de S, que cumplen las siguientes propiedades:

$\boxed{\forall A, B \cdot B \in l \wedge A \subseteq B \Rightarrow A \in l}$ (hereditariedad)

$\boxed{\forall A, B \in l \cdot |A| < |B| \Rightarrow \exists\, x \in B - A \cdot A \cup \{x\} \in l}$ (intercambio)

Matroides con Costos

Un matroide M = (S, I) tiene costos si a cada elemento x de S se puede asignar un costo no negativo w(s). Cada elemento de I tendrá igualmente un costo asociado, definido por:

$$\boxed{w(A) = \sum_{x \in A} w(x)}$$

Algoritmo Greedy para Matroides

Sea M = (S, I) un matroide con costo w. Consideremos el siguiente algoritmo:

```
GREEDY(M, w)
  A <- Ø
  ordenar S[M] en orden decreciente por peso w
  para cada x en S[M] (tomados en orden decreciente)
     if A u {x} pertenece a I[M] entonces A <- A u {x}
  retornar A
```

Teorema: Si M = (S, I) es un matroide con costo w, entonces GREEDY(M, w) retorna un subconjunto independiente maximal, de peso máximo (con respecto a w).

Ejemplo: Árbol Abarcador de Costo Mínimo (revisitado)

Problema: dado un grafo conexo no dirigido G = (V, E) con costos en los arcos, construir un árbol abarcador de G, de costo mínimo.

- Árbol abarcador: subgrafo de G, acíclico y conexo
- Mínimo costo: menor costo entre todos los árboles abarcadores

Intentaremos resolver este problema usando un algoritmo greedy, y garantizando que la solución es óptima. Recurriremos a mostrar que podemos plantear el problema como un problema de optimización en un matroide con costos.

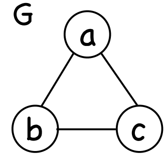
Matroides para Subgrafos Acíclicos

A partir del grafo no dirigido G = (V, E), podemos construir un matroide MG = (S_G, I_G), de la siguiente manera:

- S_G = E
- Si A es un subconjunto E, A pertenece a I_G si y sólo si A es acíclico.

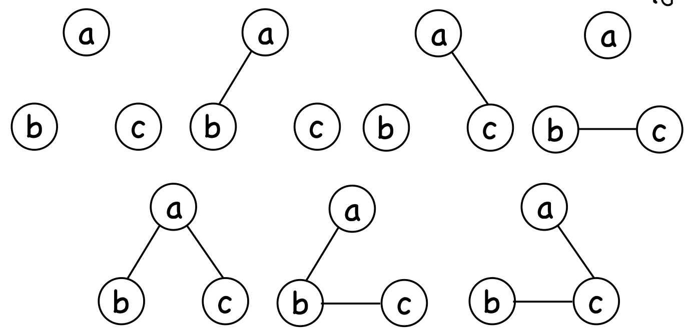
Ej:

G



S_G

I_G



Matroides para Subgrafos Acíclicos (cont.)

Veamos que la estructura $M_G = (S_G, I_G)$ es efectivamente un matroide.

Finitud: $S_G = E$ es obviamente finito.

Hereditariedad: Sea A un subconjunto de E, acíclico, y B subconjunto de A. Claramente, A es un subconjunto acíclico de E, con lo cual A pertenece a I_G.

Intercambio: Sean A y B subconjuntos acíclicos de E, tales que $|A| < |B|$. Dado que A tiene menos arcos que B, A está compuesto por más árboles que B. Luego, B debe contener un árbol T y un par de vértices (adyacentes) x e y en él, tales que x e y se encuentran en árboles diferentes en A. Dado que x e y están en árboles diferentes en A, A u {(x,y)} es acíclico (conecta dos componentes acíclicas no conexas), y por lo tanto A u {(x,y)} pertenece a I_G.

Propiedad: los subconjuntos independientes maximales de M_G son árboles abarcadores.

Costo en el Matroide

Sea w: E -> Num la función de costo del grafo G. Como E es finito, podemos calcular el mayor valor MAX, en la imagen de w. Definimos una nueva función de costo w': S_G -> Num de la siguiente manera:

$$w'(e) = MAX - w(e)$$

Propiedad: Maximizar w' equivale a minimizar w.

El algoritmo greedy para matroides con costos calcula correctamente el árbol abarcador de costo mínimo para el grafo G.

```
GREEDY(M, w)
    A <- Ø
    ordenar S[M] en orden decreciente por peso w
    para cada x en S[M] (tomados en orden decreciente)
        if A u {x} pertenece a I[M] entonces A <- A u {x}
    retornar A
```