ALGORITMOS VORACES Y TEORIA DE MATROID

YADDY MELISSA SÁNCHEZ OFO
WILFREDO YATE TORRES

ALGORITMOS VORACES

Los algoritmos voraces, ávidos o de avance rápido (en inglés greedy) son utilizados para solucionar problemas de optimización y toman decisiones basándose en la información que tienen de modo inmediato, sin tener en cuenta los efectos que estas decisiones pueden tener a futuro. Por tanto resultan fáciles de diseñar, fáciles de implementar y cuando funcionan son eficientes. Algunos ejemplos de problemas que se pueden solucionar utilizando un algoritmo voraz son: en un grafo ponderado dar la ruta más corta para ir de un nodo a otro, suponiendo que el sistema monetario de un país está formado por un numero finito de denominaciones, para cualquier cantidad dada, suministrarla en el menor número posible de monedas; en tales contextos, un algoritmo voraz funciona seleccionando la arista o la moneda que parezca más prometedora en un determinado instante, nunca reconsidera su decisión sea cual fuere la situación que pueda surgir más adelante. No hay necesidad de evaluar alternativas, ni de emplear sofisticados procedimientos de seguimiento que permitan deshacer las decisiones anteriores. Los Algoritmos voraces presentan la siguiente estructura:

Función *Voraz* (C: Conjunto): Conjunto
inicio
  $S \leftarrow \emptyset$ {en S se almacenan los elementos de la solución}
  mientras ($C \neq \emptyset$) **y no** *solución(S)* **hacer**
    $x \leftarrow seleccionar(C)$
    $C \leftarrow C - \{x\}$
    Si $factible(S \cup \{x\})$ **entonces**
      $S \leftarrow S \cup \{x\})$
    fin_si
  fin_mientras
  Si *Solución (S)* **entonces devolver** S
  Sino devolver "No hay soluciones"
fin_*Voraz*

Funciones genéricas;

- *Solución*. Comprueba si un conjunto de candidatos es una solución (independientemente de que sea óptima o no).
- *Seleccionar*. Devuelve el elemento más "prometedor"[1] del conjunto de candidatos pendientes (no seleccionados ni rechazados).
- *Factible*. Indica si a partir del conjunto S y añadiendo x, es posible construir una solución (posiblemente añadiendo otros elementos).

Para resolver el problema, se busca un conjunto de candidatos que constituya una solución, y que optimice (minimice o maximice, según el caso) el valor de la solución encontrada. Los algoritmos voraces avanzan paso a paso, la solución es buscada entre los subconjuntos del conjunto inicial o "Conjunto de Candidatos".

TEORIA DE MATROID

Formalmente la teoría de matroids es usada para determinar si un algoritmo voraz proporciona soluciones óptimas a cierto problema; Aunque esta teoría no abarca la totalidad de los casos en que se utilizan los algoritmos voraces, es de gran utilidad en ciertos problemas comunes en el estudio de este tipo de algoritmos.

MATROIDS

Una **matroid** es un par ordenado $M = (S, \mathcal{I})$ que satisface las siguientes condiciones.

1. S es un conjunto finito no vacío.
2. $\mathcal{I}$ es una familia no vacía de subconjuntos de S, tal que si $B \in \mathcal{I}$ y $A \subseteq B$, entonces $A \in \mathcal{I}$. Decimos que $\mathcal{I}$ es **hereditaria** si satisface la anterior propiedad. $\mathcal{I}$ es llamada subconjuntos **independientes** de S, Nótese que el conjunto vacío $\emptyset$ es un miembro de $\mathcal{I}$.
3. Si $A \in \mathcal{I}$, $B \in \mathcal{I}$, A y B finitos con $|A| < |B|$, donde $|A|$ y $|B|$ es el cardinal de A y B respectivamente, entonces hay algún elemento $x \in B - A$ tal que $A \cup \{x\} \in \mathcal{I}$. Decimos que M satisface la propiedad de cambio.

---

[1] Un elemento es prometedor si al adicionarlo al conjunto se puede construir una solución