Método devorador : Bibliografía y Ejemplos

Horowitz : Mochila (4.2)

Tareas secuenciales con vencimiento (4.4)

Almacenamiento óptimo en una cinta (4.6)

Mezcla de listas ordenadas, Códigos de Huffman } Arbol extendido de peso mínimo (4.7)

Cormen : Maximización de uso de recursos compartidos (16.1)

Códigos de Huffman (16.3)

Uso de matroides (16.4)

Tareas secuenciales con vencimiento penalizado (16.5)

Brassard Bratley : El problema del cambio (6.1)

El problema de la mochila (6.5)

Minimización de tiempos de espera (6.6.1)

Tareas secuenciales con vencimiento (6.6.2)

Descripción del método general

- Soluciones descomponibles en etapas.

  Cada iteración determina una etapa de la solución (óptima) escogiendo entre una serie de candidatos posibles.

  Número de etapas predeterminado o establecido por agotamiento de los (datos) candidatos.

  El número de candidatos en cada etapa es "moderado" y el criterio de selección sencillo : Jamás hay vuelta atrás.

- El coste suele ser muy bajo, y siempre polinomial.

Técnica de reducción de diferencias para garantizar la corrección (optimalidad)

- Solución devoradora generada en ese orden

$$\left.\begin{array}{llllllll} \mathcal{D} & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 \\ \mathcal{C} & s_1 & s_2 & s_3 & \cancel{s_4} & r_1 & r_2 & r_3 \quad r_4 \quad r_5 \end{array}\right\}$$

Siempre se puede mejorar o dejar igual $\mathcal{C}$ incorporando los elementos que le faltan de $\mathcal{D}$.

- Solución cualquiera ordenada según la devoradora

MATROIDES Y ALGORITMOS DEVORADORES

Def: $M = (S, I)$, $|S| < \infty$, $I \subseteq \mathcal{P}(S)$ the independent subsets of S that are hereditary: $B \in I$, $A \subseteq B \Rightarrow A \in I$ and satisfyes the incrementability property: $A, B \in I$ $|A| < |B| \Rightarrow \exists\, b \in B - A$ $A \cup \{b\} \in I$

Def: The graphic matroid generated by $G = (V, E)$ is $M_G = (S_G, I_G)$ where $S_G = E$; $A \subseteq E$ is independent iff A is acyclic (eq. (V, A) forest)

Lemma: If (V, A) is a forest it contains $|V| - |A|$ trees.

Proof: Whenever we add an arc we remove a tree.

Prop: M_G is a matroid

Proof: $|A| < |B| \Rightarrow$ B contains some tree with at least an arc b connecting two trees of A $\Rightarrow$ $A \cup \{b\}$ is a forest.

Def: $A \in I$, $x \notin A$ is an extension of A iff $A \cup \{x\} \in I$.

Prop: All maximal independent subsets of S have the same size.

Example: All the maximal forests in I_G when G is connected are spanning trees and have $|V| - 1$ arcs.

Def: A matroid can be weighted by means of a function $w: S \to \mathbb{R}^+$. We extend w to $\mathcal{P}(S)$ taking $w(A) = \sum_{a \in A} w(a)$.

Lemma: $a \in A \in I \Rightarrow \{a\} \in I$.

Def: The optimization problem on a weighted matroid M looks for the set $A \in I$ that maximizes w.

Def: The greedy strategy to solve the optimization problem on M starts choosing the element a such that $\{a\} \in I$ which maximizes w. Then we iteratively add to the constructed indepent set A that $b \notin A$ which also maximizes w.

Lemma: The first element selected by the greedy strategy above is contained in some optimal solution to the optimization problem on M.

Proof: Starting from $A = \{a\}$ and B optimal, since M is a matroid we can add some elements of B to A till we get $a \in A$, $|A| = |B|$, so that we have $A = \{a\} \cup (B - \{b\})$ for some $b \in B$. We have $\{b\} \in I \Rightarrow w(b) \leq w(a)$.

Thm: The greedy strategy produces an optimal solution to the optimization problem.

Proof: After introducing a in A the optimization problem reduces to that corresponding to the matroid $M' = (S', I')$ with $S' = \{ b \in S \mid \{a, b\} \in I \}$,

$$I' = \{ B \subseteq S' \mid B \cup \{a\} \in I \}.$$