

EI1022/MT1022 - PROBLEMAS

SESIÓN 8

4-Dic-2018

Programación dinámica. Implementación de ecuaciones recursivas

Problemas

- Mochila discreta
- Asignación óptima de recursos
- Desglose de moneda



Mochila discreta

Mochila discreta

- Sea N el número de objetos disponibles, W la capacidad de la mochila, v el vector de beneficios y w el de pesos.
- Podemos describir el conjunto de soluciones factibles del siguiente modo:

$$X = \left\{ (x_0, x_1, \dots, x_{N-1}) \in \{0, 1\}^N \mid \sum_{0 \leq i < N} x_i w_i \leq W \right\}$$

- Si queremos maximizar el valor de los objetos en la mochila sin exceder el peso, obtenemos la siguiente ecuación recursiva:

$$B(n, c) = \begin{cases} 0, & \text{si } n = 0; \\ \max\{B(n-1, c), B(n-1, c - w_{n-1}) + v_{n-1}\}, & \text{si } n > 0 \text{ y } w_{n-1} \leq c; \\ B(n-1, c), & \text{si } n > 0 \text{ y } w_{n-1} > c. \end{cases}$$

Mochila discreta

- Implementa, en el programa `sesion_8_mochila.py` que se suministra:
 - ▣ Las tres versiones recursivas:
 - Directa
 - Con memoización
 - Con memoización y recuperación de camino
 - ▣ Las dos versiones iterativas:
 - Con recuperación de camino
 - Con reducción del coste espacial

Problema de test - mochila discreta

□ Problema:

- $W = 6$
- $v = [90, 75, 60, 20, 10]$
- $w = [4, 3, 3, 2, 2]$

□ Solución:

- Beneficio: 135
- Objetos: $[0, 1, 1, 0, 0]$



Asignación óptima de recursos

Asignación óptima de recursos

- Sea U el número de unidades de un recurso y deseamos asignar cierta cantidad a cada una de N actividades distintas. Tenemos:
 - $v[i, u]$ es un diccionario que se nos proporciona y que devuelve el beneficio de asignar u unidades del recurso a la actividad i .
 - m es un vector que nos indica el número máximo de unidades del recurso que podemos asignar a cada actividad. La asignación a la actividad i debe ser menor o igual a $m[i]$.
- Podemos describir el conjunto de soluciones factibles del siguiente modo:

$$X = \left\{ (x_0, x_1, \dots, x_{N-1}) \in \mathbb{N}^N \mid 0 \leq x_i \leq m_i, 0 \leq i < N; \sum_{0 \leq i < N} x_i \leq U \right\}$$

- Obtenemos la siguiente ecuación recursiva si queremos maximizar el beneficio obtenido:

$$B(n, u) = \begin{cases} 0, & n = 0 \\ \max_{0 \leq k \leq \min(m_{n-1}, u)} (B(n-1, u-k) + v[n-1, k]), & n > 0 \end{cases}$$

Asignación óptima de recursos

- Implementa, en el programa `sesion_8_recursos.py` que se suministra:
 - ▣ Las tres versiones recursivas:
 - Directa
 - Con memoización
 - Con memoización y recuperación de camino
 - ▣ Las dos versiones iterativas:
 - Con recuperación de camino
 - Con reducción del coste espacial

Problema de test – asig. recursos

□ Problema:

- $U = 12$

- $m = [2, 4, 2, 4, 2]$

- El diccionario v se genera automáticamente en el código suministrado.

□ Solución:

- Beneficio: 441

- Recursos: $[1, 4, 1, 1, 0]$



Desglose de moneda

Desglose de moneda con limitación del número de monedas

- Sea N el número de monedas del sistema monetario, Q la cantidad a desglosar, v el vector con los valores de las monedas y w el vector con los pesos de las monedas.
- Además, ahora tenemos un vector m que nos indica cuántas monedas tenemos de cada tipo.

- Podemos describir el conjunto de soluciones factibles del siguiente modo:

$$X = \left\{ (x_0, x_1, \dots, x_{N-1}) \in \mathbb{N}^N \mid 0 \leq x_i \leq m_i, 0 \leq i < N; \sum_{0 \leq i < N} x_i v_i = Q \right\}$$

- Obtenemos la siguiente ecuación recursiva si queremos minimizar el peso del desglose:

$$L(q, n) = \begin{cases} 0, & \text{si } q = 0 \text{ y } n = 0; \\ +\infty, & \text{si } q > 0 \text{ y } n = 0; \\ \min_{0 \leq i \leq \min\{m_i, \lfloor q/v_n \rfloor\}} \left(L(q - i v_{n-1}, n-1) + i w_{n-1} \right), & \text{en otro caso.} \end{cases}$$

Desglose de moneda con limitación del número de monedas

- Implementa un programa `sesion_8_desglose.py` que incluya:
 - ▣ Las tres versiones recursivas:
 - Directa
 - Con memoización
 - Con memoización y recuperación de camino
 - ▣ Las dos versiones iterativas:
 - Con recuperación de camino
 - Con reducción del coste espacial

Problema de test – desglose con lim.

□ Problema:

- $Q = 24$
- $v = [1, 2, 5, 10]$
- $w = [1, 1, 4, 6]$
- $m = [3, 1, 4, 1]$

□ Solución:

- Peso: 17
- Desglose: $[2, 1, 2, 1]$