



UNIVERSIDAD  
DE GRANADA

# P4: Programación Dinámica

Eugenio Martínez Cámara  
[emcamara@decsai.ugr.es](mailto:emcamara@decsai.ugr.es)



# Recordatorio Tema 4 – Objetivo de la Programación Dinámica

Algoritmos que **tratan** de optimizar una función objetivo.

# Recordatorio Tema 4 – Características

1. La solución se construye paso a paso. Cada selección puede originar varias soluciones (*Greedy* solo una).
2. Se divide un problema de tamaño  $n$  en problemas de tamaño  $n-1$ , que se solapan (En DyV son independientes).
3. Mantiene en memoria los subproblemas solucionados para ahorrar cálculos.
4. Siempre devuelve la solución óptima.
5. Eficiente en tiempo, pero no en memoria.

# Recordatorio Tema 4 – Ventajas vs Inconvenientes

## Ventajas

1. Eficientes.
2. Devuelven la solución óptima.
3. Fáciles de implementar.

## Inconvenientes

1. Ineficientes en espacio.
2. Difíciles de diseñar.
3. Tiene varios requisitos para poder aplicar la técnica.

# Recordatorio Tema 4 – Elementos y plantilla

- No existe una plantilla como en el resto de técnicas.
- Se debe primero determinar *qué se quiere optimizar*.
- Es un método eficiente de diseñar algoritmos recursivos que almacenan los resultados parciales sin necesidad de recalcular estos.
- Definir el proceso de optimización como una función recurrente.
- Definir casos base ( $T(0)$ ); definir caso  $T(i)$ .
- Definir función de optimización.
- Demostrar que se cumple el Principio de Optimalidad de Bellman: si una secuencia de pasos para resolver un problema es óptima, cualquier subsecuencia de estos mismos pasos también es óptima.

# Problema del cambio

- Objetivo: Devolver cambio con el **mínimo número de monedas**.
- Restricciones:
  - Cambio de  $C$  céntimos.
  - Tipos de monedas:  $n$ ,  $\{1 \dots n\}$ .
  - Valor monedas:  $d$ ,  $\{1 \dots d\}$ .
  - Número finito de monedas por tipo:  $\{m_1, \dots m_n\}$ .



# Problema del cambio – Solución Greedy

- Función de selección: Se selecciona aquella moneda de mayor valor siempre y cuando no supere el valor del cambio y haya monedas de ese tipo disponibles.
- ¿Es optimo? Puede no ser óptimo.
- Cantidad a devolver: 63 ctmos.
- Monedas:
  - 50 ctmos: 3 monedas.
  - 20 ctmos: 5 monedas.
  - 1 ctmos: 12 monedas.
  - No quedan más tipos de monedas.

```
S = Ø; s = 0
Mientras s != n hacer:
    x = el mayor elemento de C tal que s + x ≤ n
    Si no existe ese elemento, entonces
        Devolver "no encuentro la solución"
    S = S U {una moneda de valor x}
    C = C \ {x}
    s = s + x
Fin-Mientras
Devolver S
```

# Problema del cambio – Solución Greedy

- Función de selección: Se selecciona aquella moneda de mayor valor siempre y cuando no supere el valor del cambio y haya monedas de ese tipo disponibles.
- ¿Es optimo? Puede no ser óptimo.
- Cantidad a devolver: 63 ctmos.
- Monedas:
  - 50 ctmos: 3 monedas.
  - 20 ctmos: 5 monedas.
  - 1 ctmos: 12 monedas.
  - No quedan más tipos de monedas.
  - Algoritmo: NO HAY SOLUCIÓN.
  - Óptima: 3 monedas de 20 céntimos y 3 monedas de 1 céntimo.

```
S = Ø; s = 0
Mientras s != n hacer:
    x = el mayor elemento de C tal que s + x ≤ n
    Si no existe ese elemento, entonces
        Devolver "no encuentro la solución"
    S = S U {una moneda de valor x}
    C = C \ {x}
    s = s + x
Fin-Mientras
Devolver S
```



# Problema del cambio – Solución por programación dinámica

- ¿Se puede resolver el problema por etapas?
  - Sí, porque se pueden ir devolviendo sucesivamente una moneda de un tipo dado.
  - Si suponemos que las monedas están ordenadas de mayor valor  $d_i$  a menor valor, se asegura una solución óptima desde las etapas más tempranas.
  - La elección en cada etapa depende de:
    - Del tipo de monedas disponibles.
    - De la cantidad aún por devolver en cada etapa.
    - Y dependiendo del problema, de la cantidad de monedas disponibles.
  - En cada etapa se debe evaluar si se devuelve un tipo de moneda u otro.
- Dos fases:
  - Diseño de la memoria.
  - Recuperación de la solución.

# Problema del cambio – Solución por PD – Diseño de Memoria

- La memoria se representa con una matriz (T):
- Filas: tipo de moneda.
- Columnas: cantidad de dinero a devolver, desde 1 céntimo a N céntimos. Si hay que devolver 100 céntimos (1 €), tendríamos 100 columnas.
- $T(i, j)$ : Cantidad mínima de monedas a devolver para una cantidad de cambio  $j$ .

Cambio: 15 céntimos.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1																
2																
5																
10																
20																
50																
100																
200																

# Problema del cambio – Solución por PD – Diseño de Memoria

- Casos base:
  - Si cambio igual a 0 ( $j=0$ ):  $T(i, 0)=0$ . Independientemente del tipo de moneda.
  - Monedas de valor unitario ( $i=1$ , monedas de 1 ctmo.):  $T(1,j)=j$ .
- Ecuación recurrente:
  - Devolver moneda tipo  $i$ . En este caso, las monedas a devolver sería:  $1 + T(i, j - d_i)$ .
  - No devolver moneda tipo  $i$ . En este caso, el mínimo número de monedas a devolver es el mismo, porque no hemos devuelto ninguna, por tanto,  $T(i-1, j)$ .
- Ecuación recurrente:  
$$T(i, j) = \min\{T(i - 1, j), 1 + T(i, j - d_i)\}$$

Cambio: 15 céntimos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1															
2															
5															
10															
20															
50															
100															
200															

# Problema del cambio – Solución por PD – Diseño de Memoria

- Casos base:

- Si cambio igual a 0 ( $j=0$ ):  $T(i, 0)=0$ . Independientemente del tipo de moneda.

- Monedas de valor unitario ( $i=1$ , monedas de 1 ctmo.):  $T(1,j)=j$ .

- Ecuación recurrente:

- Devolver moneda tipo  $i$ . En este caso, las monedas a devolver sería:  $1 + T(i, j - d_i)$ .

- No devolver moneda tipo  $i$ . En este caso, el mínimo número de monedas a devolver es el mismo, porque no hemos devuelto ninguna, por tanto,  $T(i-1, j)$ .

- Ecuación recurrente:

$$T(i, j) = \min\{T(i - 1, j), 1 + T(i, j - d_i)\}$$

Cambio: 15 céntimos.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio – Solución por PD – Construir la solución

Cambio: 15 céntimos.

$S \leftarrow \emptyset$

$i \leftarrow n, j \leftarrow N$

Mientras  $j \neq 0$ , hacer:

    Si  $i > 1$  y  $T(i, j) = T(i-1, j)$ , hacer:

$i \leftarrow i-1$

    En otro caso, hacer:

$j \leftarrow j - d_i$

        Añadir moneda de tipo  $i$  a  $S$

Devolver  $S$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio – Solución por PD – Construir la solución

Cambio: 15 céntimos.

$S \leftarrow \emptyset$

$i \leftarrow n, j \leftarrow N$

Mientras  $j \neq 0$ , hacer:

    Si  $i > 1$  y  $T(i, j) = T(i-1, j)$ , hacer:

$i \leftarrow i-1$

    En otro caso, hacer:

$j \leftarrow j - d_i$

        Añadir moneda de tipo  $i$  a  $S$

Devolver  $S$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio – Solución por PD – Construir la solución

Cambio: 15 céntimos.  
Monedas: 1x10+1x5

$S \leftarrow \emptyset$   
 $i \leftarrow n, j \leftarrow N$   
Mientras  $j \neq 0$ , hacer:  
    Si  $i > 1$  y  $T(i, j) = T(i-1, j)$ , hacer:  
         $i \leftarrow i-1$   
    En otro caso, hacer:  
         $j \leftarrow j - d_i$   
        Añadir moneda de tipo  $i$  a  $S$   
Devolver  $S$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio con cantidad de monedas – Solución por PD – Diseño de Memoria

- **Mismos casos base y ecuación recurrente.**

- Casos base:

- Si cambio igual a 0 ( $j=0$ ):  $T(i, 0)=0$ . Independientemente del tipo de moneda.
- Monedas de valor unitario ( $i=1$ , monedas de 1 ctmo.):  $T(1,j)=j$ .

- Ecuación recurrente:

- Devolver moneda tipo  $i$ . En este caso, las monedas a devolver sería:  $1 + T(i, j - d_i)$ .
- No devolver moneda tipo  $i$ . En este caso, el mínimo número de monedas a devolver es el mismo, porque no hemos devuelto ninguna, por tanto,  $T(i-1, j)$ .

- Ecuación recurrente:

$$T(i, j) = \min\{T(i - 1, j), 1 + T(i, j - d_i)\}$$

Cambio: 15 céntimos.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2



# Problema del cambio con cantidad de monedas – Solución por PD – Construir la solución

Monedas:

- 1 ctms: 15
- 2 ctms: 20
- 5 ctms: 2
- 10 ctms: 0
- 20 ctms: 1.
- 50 ctms: 0.
- 100 ctms: 3
- 200 ctms: 5.

Cambio: 15 céntimos.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio con cantidad de monedas – Solución por PD – Construir la solución

Monedas:

- 1 ctms: 15
- 2 ctms: 20
- 5 ctms: 2 → 1
- 10 ctms: 0
- 20 ctms: 1.
- 50 ctms: 0.
- 100 ctms: 3
- 200 ctms: 5.

Cambio: 15 céntimos.

Monedas: 1x5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2

# Problema del cambio con cantidad de monedas – Solución por PD – Construir la solución

Monedas:

- 1 ctms: 15
- 2 ctms: 20
- 5 ctms: 2 → 1 → 0
- 10 ctms: 0
- 20 ctms: 1.
- 50 ctms: 0.
- 100 ctms: 3
- 200 ctms: 5.

Cambio: 15 céntimos.

Monedas: 1x5 + 1x5+2x2+1x1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8
5	0	1	1	2	2	1	2	2	3	3	2	3	3	4	4	3
10	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
20	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
50	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
100	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2
200	0	1	1	2	2	1	2	2	3	3	1	2	2	3	3	2



**UNIVERSIDAD  
DE GRANADA**