Fundamentos de análisis y diseño de algoritmos

Programación dinámica

El problema de la mochila 0/1

Programación dinámica

Se tienen N objetos y una mochila de capacidad (de peso) M, cada objeto tiene un peso w_i, $1 \leq i \leq N$. Cada objeto puede estar, o no, en la mochila. Ademas, se tiene un beneficio b_i por cada objeto

El problema consiste en maximizar el beneficio. La solución se representa indicando para cada objeto si se debe colocar o no en la mochila

Programación dinámica

Se tienen N objetos y una mochila de capacidad (de peso) M, cada objeto tiene un peso w_i, $1 \leq i \leq N$. Cada objeto puede estar, o no, en la mochila. Además, se tiene un beneficio b_i por cada objeto

De manera formal, el problema consiste en encontrar $\langle x_1, x_2, \ldots, x_n \rangle$ tal que:

$$\sum_{1 \leq i \leq N} b_i x_i \text{ sea máximo, sujeto a}$$

optimización

$$\sum_{1 \leq i \leq N} w_i x_i \leq M$$

$x_i \in \{0,1\}$, donde 0 significa que el objeto i no se coloca en la mochila y 1 que si

Programación dinámica

Se tienen N objetos y una mochila de capacidad (de peso) M, cada objeto tiene un peso w_i, $1 \le i \le N$. Cada objeto puede estar, o no, en la mochila. Además, se tiene un beneficio b_i por cada objeto

De manera formal, el problema consiste en encontrar $\langle x_1, x_2, \ldots, x_n \rangle$ tal que:

$$\sum_{1 \le i \le N} b_i x_i \text{ sea máximo, sujeto a}$$

Problema mochila(1, N, M)

$$\sum_{1 \le i \le N} w_i x_i \le M$$

$x_i \in \{0,1\}$, donde 0 significa que el objeto i no se coloca en la mochila y 1 que si

Programación dinámica

N=3, M=9, b=<10,6,8>, w=<3,4,5>

<1,0,1> es una solución que indica colocar en la mochila los objetos 1 y 3, esto implica un beneficio de 18

<1,1,0> es una solución que indica colocar en la mochila los objetos 1 y 2, esto implica un beneficio de 16

<0,1,1> es una solución que indica colocar en la mochila los objetos 2 y 3, esto implica un beneficio de 14

<1,0,0>, <0,1,0>, <0,0,1>

Debido a los pesos no se pueden colocar todos los objetos en la mochila

Programación dinámica

N=3, M=9, b=<10,6,8>, w=<7,4,5>

Muestre soluciones indicando el beneficio

<1, 0, 0> b = 10

<0, 1, 0> b=6

<0, 1, 1> b=14

<0, 0, 1> b=8

Programación dinámica

N=3, M=9, b=<10,6,8>, w=<7,4,5>

<1,0,0>: beneficio 10

<0,1,0>: beneficio 6

<0,0,1>: beneficio 8

<0,1,1>: beneficio 14

Solución óptima: <0,1,1>

Programación dinámica

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Muestre todas las soluciones indicando el beneficio.
Presente la solución óptima

$\rightarrow \langle 1, 1, 0, 1 \rangle \quad b=9$

Programación dinámica

N=4, M=20, b=$\langle 3,2,1,4 \rangle$, w=$\langle 7,5,6,8 \rangle$

$\langle 1,1,1,0 \rangle$ W=18, B=6

$\langle 0,1,1,1 \rangle$ W=19, B=7

$\langle 1,0,1,1 \rangle$ W=21 X

$\langle 1,1,0,1 \rangle$ W=20, B=9

Programación dinámica

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Considere la solución óptima <1,1,0,1>

Encuentre subproblemas y soluciones óptimas a esos subproblemas (utilice la notación mochila(L,N,M) para indicar el problema de la mochila utilizando de L a N elementos y una capacidad M)

Programación dinámica

Problema: encontrar $\langle x_k, x_{k+1}, \ldots, x_l \rangle$ tal que:

$$\sum_{k \le i \le l} b_i x_i$$ sea máximo, sujeto a

$$\sum_{k \le i \le l} w_i x_i \le P$$

Problema mochila(k, l, P)

Programación dinámica

N=4, M=20, b=$\langle 3,2,1,4\rangle$, w=$\langle 7,5,6,8\rangle$

Si $\langle 1,1,0,1\rangle$ es una solución óptima de mochila(1,4,20) …

$\langle 1,1,0\rangle$ es una sol mochila(1,3,12)

$\langle 1,1\rangle$ es sol mochila(1,2,12)

$\langle 1\rangle$ es sol mochila(1,1,7)

Programación dinámica

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Si <1,1,0,1> es una solución óptima de mochila(1,4,20)

entonces <1,1,0> es una solución óptima de mochila(1,3,20-8)

mochila(1,3,12) es el problema de colocar los elementos 1, 2 y 3 en la mochila de capacidad 12

Programación dinámica

N=4, M=20, b=$\langle 3,2,1,4 \rangle$, w=$\langle 7,5,6,8 \rangle$

Si $\langle 1,1,0,1 \rangle$ es una solución óptima de mochila(1,4,20)

entonces $\langle 1,1,0 \rangle$ es una solución óptima de mochila(1,3,20-8)

Si $\langle 1,1,0 \rangle$ es una solución óptima de mochila(1,3,12)

entonces $\langle 1,1 \rangle$ es una solución óptima de mochila(1,2,12)

Un 0 en la solución óptima significa que el elemento no se coloca en la mochila. Note que no se disminuye la capacidad de la mochila

Programación dinámica

N=4, M=20, b=$\langle 3,2,1,4 \rangle$, w=$\langle 7,5,6,8 \rangle$

Si $\langle 1,1,0,1 \rangle$ es una solución óptima de mochila(1,4,20)

entonces $\langle 1,1,0 \rangle$ es una solución óptima de mochila(1,3,20-8)

Si $\langle 1,1,0 \rangle$ es una solución óptima de mochila(1,3,12)

entonces $\langle 1,1 \rangle$ es una solución óptima de mochila(1,2,12)

Si $\langle 1,1 \rangle$ es una solución óptima de mochila(1,2,12)

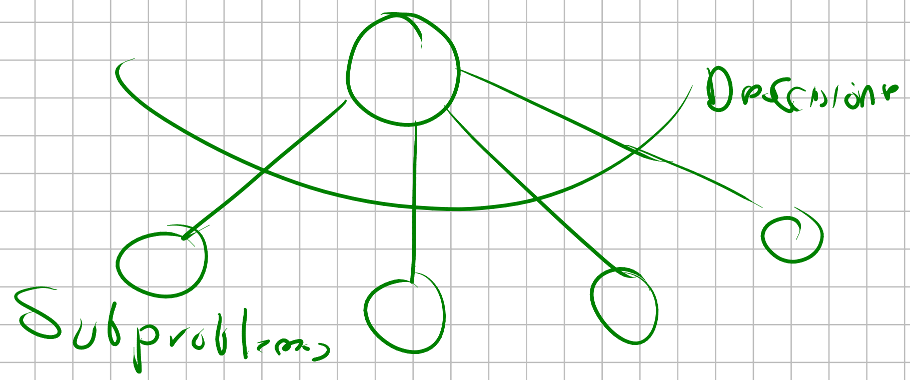
entonces $\langle 1 \rangle$ es una solución óptima de mochila(1,1,12-5)

Programación dinámica

En términos generales se tiene que, sea $\langle y_1, y_2,\ldots,y_N \rangle$ una secuencia óptima para $\langle x_1,x_2,\ldots x_N \rangle$, dada una mochila de capacidad M, entonces:

- Si $y_N=0$ entonces $\langle y_1,\ldots,y_{N-1} \rangle$ es una secuencia óptima para mochila(1,N-1, M)

- Si $y_N=1$ entonces $\langle y_1,\ldots,y_{N-1} \rangle$ es una secuencia óptima para mochila(1,N-1, M-w_N)


Decisions
Subproblems

Programación dinámica

Si $\langle y_1, y_2,\ldots,y_N\rangle$ una secuencia óptima para mochila(1,N,M) entonces $\langle y_1, y_2,\ldots,y_i\rangle$ y $\langle y_{i+1}, y_{i+2},\ldots,y_N\rangle$ son soluciones optimas a los problemas:

$$mochila(1, j, \sum_{1\le i\le j} w_i x_i) \qquad \text{y} \qquad mochila(j+1, N, M - \sum_{1\le i\le j} w_i x_i)$$

Programación dinámica

Sea $g_j(M)$ el valor de la solución óptima de mochila(1,j,M), por la subestructura óptima, se puede decir que:

$$g_j(M)=\max(g_{j-1}(M), g_{j-1}(M-w_j)+b_j)$$

$$g_0(M)=0$$

esto es, seleccionar el máximo entre los beneficios de 1) no colocar el elemento j en la mochila y 2) colocar j en la mochila, se debe entonces colocar el beneficio b_j y disminuir el peso máximo de la mochila, que ahora será $M-w_j$

Programación dinámica

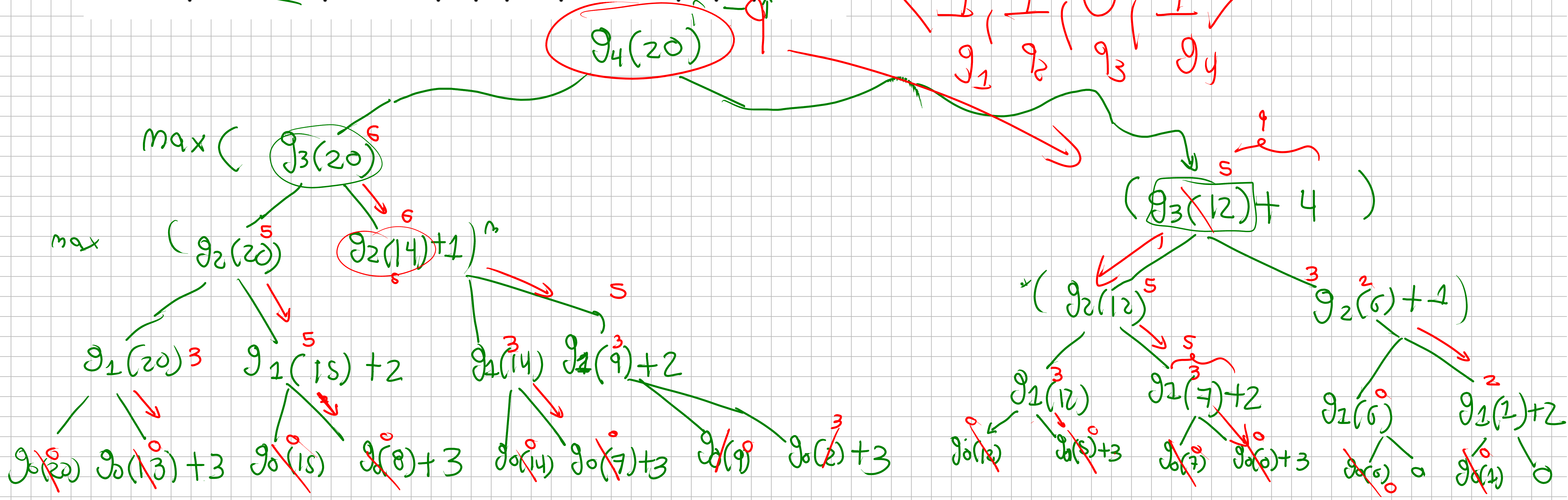
El valor de $g_N(M)$ se expresa en términos de
$g_{N-1}(M)$ y $g_{N-1}(M-w_N)$

El valor de $g_{N-1}(M)$ se expresa en términos de
$g_{N-2}(M)$, $g_{N-2}(M-w_{N-2})$ y $g_{N-2}(M-w_{N-2}-w_{N-1})$

hasta llegar a $g_0(M)$ que vale 0

$N=4,\ M=20,\ b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

$\langle 1, 1, 0, 1\rangle$
$\ \ g_1\ \ g_2\ \ g_3\ \ g_4$



Programación dinámica

$N=4$, $M=20$, $b=\langle 3,2,1,4\rangle$, $w=\langle 7,5,6,8\rangle$

mochila(1,4,20) tiene valor $g_4(20)$, donde:

$g_4(20)=\max(g_3(20), g_3(12)+4)$

$$\boxed{g_j(M)=\max(g_{j-1}(M), g_{j-1}(M-w_j)+b_j)}$$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(20)=\max(g_1(20),\ g_1(15)+2)$

$g_2(14)=\max(g_1(14),\ g_1(9)+2)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(20)=\max(g_1(20),\ g_1(15)+2)$

$g_2(14)=\max(g_1(14),\ g_1(9)+2)$

$g_1(20)=\max(g_0(20),\ g_0(13)+3)$

$g_1(15)=\max(g_0(15),\ g_0(8)+3)$

$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$

$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(20)=\max(g_1(20),\ g_1(15)+2)$

$g_2(14)=\max(g_1(14),\ g_1(9)+2)$

$g_1(20)=\max(g_0(20),\ g_0(13)+3)$

$g_1(20)=\max(0,3)$

$g_1(15)=\max(g_0(15),\ g_0(8)+3)$

$g_1(15)=\max(0,3)$

$$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$$

$$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$$g_2(20)=\max(g_1(20),\ g_1(15)+2)$$
$$=5$$

$$g_2(14)=\max(g_1(14),\ g_1(9)+2)$$

$$g_1(20)=\max(g_0(20),\ g_0(13)+3)$$

$$g_1(20)=3$$

$$g_1(15)=\max(g_0(15),\ g_0(8)+3)$$

$$g_1(15)=\max(0,3)$$

$$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$$

$$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(20)=\max(g_1(20),\ g_1(15)+2)$
$=5$

$g_2(14)=\max(g_1(14),\ g_1(9)+2)$

$g_1(20)=\max(g_0(20),\ g_0(13)+3)$

$g_1(20)=3$

$g_1(15)=\max(g_0(15),\ g_0(8)+3)$

$g_1(15)=\max(0,3)$

$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$

$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$$g_2(20)=\max(g_1(20),\ g_1(15)+2)$$
$$=5$$

$$g_2(14)=\max(g_1(14),\ g_1(9)+2)$$

$$g_1(14)=\max(g_0(14),\ g_0(7)+3)$$
$$g_1(14)=3$$

$$g_1(9)=\max(g_0(9),\ g_0(2)+3)$$
$$g_1(9)=3$$

$$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$$

$$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$$g_2(20)=\max(g_1(20),\ g_1(15)+2)$$
$$=5$$

$$g_2(14)=\max(g_1(14),\ g_1(9)+2)$$
$$=\max(3,5)=5$$

$$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$$

$$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$

$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1$$
$$=\max(5,6)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$$g_2(12)=\max(g_1(12),\ g_1(7)+2)$$

$$g_2(6)=\max(g_1(6),\ g_1(1)+2)$$

$$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$$

$$\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$$

Programación dinámica

$$g_4(20)=\max(g_3(20),\ g_3(12)+4)$$

$$g_3(20)=\max(g_2(20),\ g_2(14)+1$$
$$=\max(5,6)$$

$$g_3(12)=\max(g_2(12),\ g_2(6)+1)$$

$$g_2(12)=\max(g_1(12),\ g_1(7)+2)$$

$$g_2(6)=\max(g_1(6),\ g_1(1)+2)$$

$$g_1(12)=\max(g_0(12),\ g_0(5)+3)$$

$$g_1(7)=\max(g_0(7),\ g_0(0)+3)$$

$$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$$

$$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$
$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(12)=\max(g_1(12),\ g_1(7)+2)$

$g_2(6)=\max(g_1(6),\ g_1(1)+2)$

$g_1(12)=\max(g_0(12),\ g_0(5)+3)$
$=\max(0,3)$

$g_1(7)=\max(g_0(7),\ g_0(0)+3)$
$=\max(0,3)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$
$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(12)=\max(g_1(12),\ g_1(7)+2)$
$=\max(3,5)$

$g_2(6)=\max(g_1(6),\ g_1(1)+2)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$
$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(12)=\max(g_1(12),\ g_1(7)+2)$
$=\max(3,5)$

$g_2(6)=\max(g_1(6),\ g_1(1)+2)$

$g_1(6)=0$
(no cabe)

$g_1(1)=0$
(no cabe)

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$
$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$

$g_2(12)=\max(g_1(12),\ g_1(7)+2)$
$=\max(3,5)$

$g_2(6)=\max(g_1(6),\ g_1(1)+2)$
$=\max(0,2)$

$\boxed{g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)}$ $\boxed{b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle}$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$g_3(20)=\max(g_2(20),\ g_2(14)+1$
$=\max(5,6)$

$g_3(12)=\max(g_2(12),\ g_2(6)+1)$
$=\max(5,\ 3)$

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=\langle 3,2,1,4\rangle,\ w=\langle 7,5,6,8\rangle$

Programación dinámica

$g_4(20)=\max(g_3(20),\ g_3(12)+4)$

$=\max(6,9)$

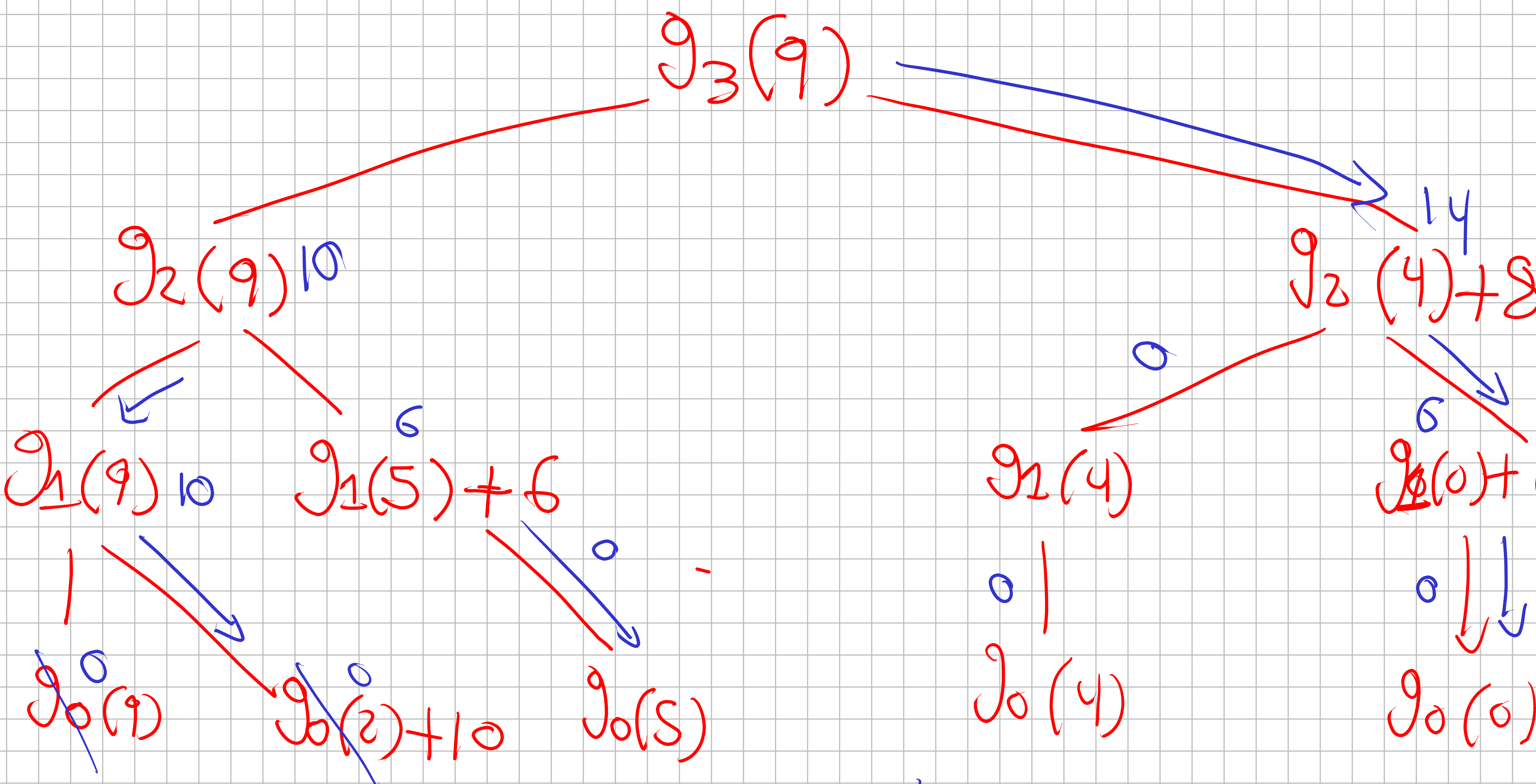
$=9$

9 es el valor óptimo

$g_j(M)=\max(g_{j-1}(M),\ g_{j-1}(M-w_j)+b_j)$

$b=<3,2,1,4>,\ w=<7,5,6,8>$

$N=3$, $M=9$, $b=\langle 10,6,8\rangle$, $w=\langle 7,4,5\rangle$



$\langle 0, 1, 1\rangle$

Programación dinámica

Construir una matriz BMAX de dimensiones MXN de forma que BMAX[i,j] contenga el beneficio máximo que se puede obtener si la capacidad de I y se cuenta con los primeros J elementos

$$
BMAX(I,1)= \begin{cases} B(1) \text{ si } I \geq W(1) \\ 0 \text{ si } I < W(1) \end{cases}
$$

no llevar

$$
BMAX(I,J)= MAX(BMAX(I,J-1), \\ BMAX(I-W(J), j-1) + B(J))
$$

llevar

$N=4$, $M=20$, $b=\langle 3,2,1,4\rangle$, $w=\langle 7,5,6,8\rangle$

Se crea una matriz de 20x4

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

$$BMAX(I,1)=\begin{cases} B(1) \text{ si } I \geq W(1) \\ 0 \text{ si } I < W(1) \end{cases}$$

	$\boxed{1}$	2	3	4
1	0			
2	0			
3	0			
4	0			
5	0			
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

	1	2	3	4
1	0			
2	0			
3	0			
4	0			
5	0			
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

Capacidades de la mochila

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(1,2)=MAX(BMAX(1,1),

BMAX(1-W(2), 1) + B(2))

No cabe en la mochila, el elemento menos pesado es el 2 que pesa 5

	1	2	3	4
1	0			
2	0			
3	0			
4	0			
5	0			
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(5,2)=???

BMAX(0, 1)+2

0

BMAX(i,j) → BMAX(i, j-1)

BMAX(i,j) → BMAX(i-w(j), j-1) + B(j)

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2		
6	0	2		
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

B(5,2)

↓

B(1,1) +2

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(5,2)=MAX(BMAX(5,1),

BMAX(5-W(2), 1) + B(2))

=MAX(0,

BMAX(0, 1) + 2)

=MAX(0,2)=2

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2		
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(5,3)=MAX(BMAX(5,2),

BMAX(5-W(3), 1) + B(3))

como 3 no cabe, el máximo sigue

siendo BMAX(5,2)=2

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(5,4)=MAX(BMAX(5,3),

BMAX(5-W(4), 1) + B(4))

como 4 no cabe, el máximo sigue

siendo BMAX(5,3)=2

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0			
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(6,2)=MAX(BMAX(6,1),

BMAX(6-W(2), 1) + B(2))

=MAX(0,

BMAX(1, 1) + 2)

=2

donde BMAX(1,1) ya se conoce

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2		
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(6,3)=MAX(BMAX(6,2),

BMAX(6-W(3), 2) + B(3))

=MAX(2,

BMAX(0, 1) + 1)

=2

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	
7	3			
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(7,2)=MAX(BMAX(7,1),

BMAX(7-W(2), 1) + B(2))

=MAX(3,

BMAX(2,1) + 2)

=MAX(3, 2) = 3

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3		
8	3			
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(8,2)=MAX(BMAX(8,1),

BMAX(8-W(2), 1) + B(2))

=MAX(3,

BMAX(3,1) + 2)

=MAX(3, 2) = 3

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3		
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

$N=4$, $M=20$, $b=\langle 3,2,1,4\rangle$, $w=\langle 7,5,6,8\rangle$

Se crea una matriz de 20x4

$$BMAX(I,J)=MAX(BMAX(I,J-1),\; BMAX(I-W(J), j-1) + B(J)\;)$$

$$BMAX(8,4)=MAX(BMAX(8,3),\; BMAX(8-W(4), 1) + B(4)\;)$$

$$=MAX(\;3,\; BMAX(0,1) + 4\;)$$

$$=MAX(\;3, 4) = 4$$

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3			
10	3			
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

$N=4$, $M=20$, $b=\langle 3,2,1,4\rangle$, $w=\langle 7,5,6,8\rangle$

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),
BMAX(I-W(J), j-1) + B(J))

BMAX(8,4)=MAX(BMAX(8,3),
BMAX(8-W(4), 1) + B(4))
=MAX(3,
BMAX(0,1) + 4)
=MAX(3, 4) = 4

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	3	3	4
10	3	3	3	4
11	3			
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(11,2)=MAX(BMAX(11,1),

BMAX(11-W(2), 1) + B(2))

=MAX(3,

BMAX(6,1) + 2)

=MAX(3, 2) = 3

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	3	3	4
10	3	3	3	4
11	3	3		
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),
BMAX(I-W(J), j-1) + B(J))

BMAX(11,3)=MAX(BMAX(11,2),
BMAX(11-W(3), 2) + B(3))
=MAX(3,
BMAX(5,2) + 2)
=MAX(3, 4) = 4

El 4 se obtiene entonces por <0,1,1,0>

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	4	4	4
10	3	4	4	4
11	3	3	4	
12	3			
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(12,2)=MAX(BMAX(12,1),

BMAX(12-W(2), 1) + B(2))

=MAX(3,

BMAX(7,2) + 2)

=MAX(3, 5) = 5

Se continua el proceso, al final se tendrá el valor optimo

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	4	4	4
10	3	4	4	4
11	3	3	4	4
12	3	5		
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

N=4, M=20, b=<3,2,1,4>, w=<7,5,6,8>

Se crea una matriz de 20x4

BMAX(I,J)=MAX(BMAX(I,J-1),

BMAX(I-W(J), j-1) + B(J))

BMAX(12,2)=MAX(BMAX(12,1),

BMAX(12-W(2), 1) + B(2))

=MAX(3,

BMAX(7,2) + 2)

=MAX(3, 5) = 5

Para obtener la respuesta se guardan los valores de j con los que se obtiene el valor máximo

	1	2	3	4
1	0	X	X	X
2	0	X	X	X
3	0	X	X	X
4	0	X	X	X
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	4	4	4
10	3	4	4	4
11	3	3	4	4
12	3	5		
13	3			
14	3			
15	3			
16	3			
17	3			
18	3			
19	3			
20	3			

	1	2	3	4
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	2	2	2
6	0	2	2	2
7	3	3	3	3
8	3	3	3	4
9	3	3	3	4
10	3	3	3	4
11	3	3	3	4
12	3	5	$\boxed{5}$	5
13	3	5	5	6
14	3	5	5	6
15	3	5	5	7
16	3	5	5	7
17	3	5	5	7
18	3	5	6	7
19	3	5	6	7
20	3	5	6	9

9

9

$\langle 1, 1, 0, 1 \rangle$

$b \ \langle 3 \ 2 \ 1 \ 4 \rangle$

$w \ \langle 7 \ 5 \ 6 \ 8 \rangle$

BMAX[i][j]

Capacidad

[illegible]

$$
BMAX[i][j] = \begin{cases} 0 & i = 0 \quad \text{mochila vacía} \\ & i < w[j] \quad \text{mochila sin cap suf} \\ \max\big(BMax[i][j-1], & \text{// no llevar} \\ \quad BMax[i - w[j]][j-1] + b[j] \big) & \end{cases}
$$