

UNIVERSITAT POLITÈCNICA DE CATALUNYA – BARCELONATECH
OPE – ORGANIZACIÓN DE LA PRODUCCIÓN Y DE EMPRESA (ASPECTOS TÉCNICOS, JURÍDICOS
Y ECONÓMICOS EN PRODUCCIÓN)

Modelos y herramientas de decisión. Programación dinámica I

MODELOS Y HERRAMIENTAS DE DECISIÓN 240EO023 – Máster Universitario en Ingeniería de Organización
(240MUEO) - ETSEIB

Joaquín Bautista-Valhondo

OPE-PROTHIUS – OPE-MSc.2018/22 240EO023 (20180408) - <http://futur.upc.edu/OPE-PROTHIUS> - www.prothius.com -
Departamento de Organización de Empresas – ETSEIB · UPC



PROTHIUS
Càtedra Organització Industrial

MHD' 18 – Pro-Din (I): 0
J. Bautista

Contenido

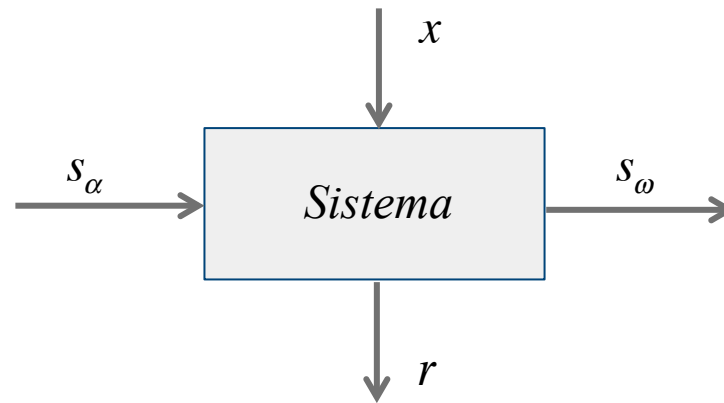
- Introducción
- Ejemplo 1: *El despertar de la Fuerza · El encuentro*. Presentación, formalización y resolución
- Programación Dinámica. Características de los problemas
- Programación Dinámica. Tipología
- Programación Dinámica determinista. Nomenclatura
- El problema de distribución de esfuerzos
- Ejemplo 2: *El problema de los 4 Equipos de investigación*. Presentación, formalización y resolución
- El problema Dynamic Lot Sizing (DLS)
- Ejemplo 3: *El problema del lote dinámico*. Presentación y resolución lotes estáticos
- Ejemplo 3. Programa matemático y resolución Programación Dinámica
- Ejemplo 3. Resolución WAGNER-WHITIN y Grafo polietápico



Introducción

Decisiones multietápicas

Una etapa:



$n = \text{etapa } (n = 1, 2, \dots, N)$

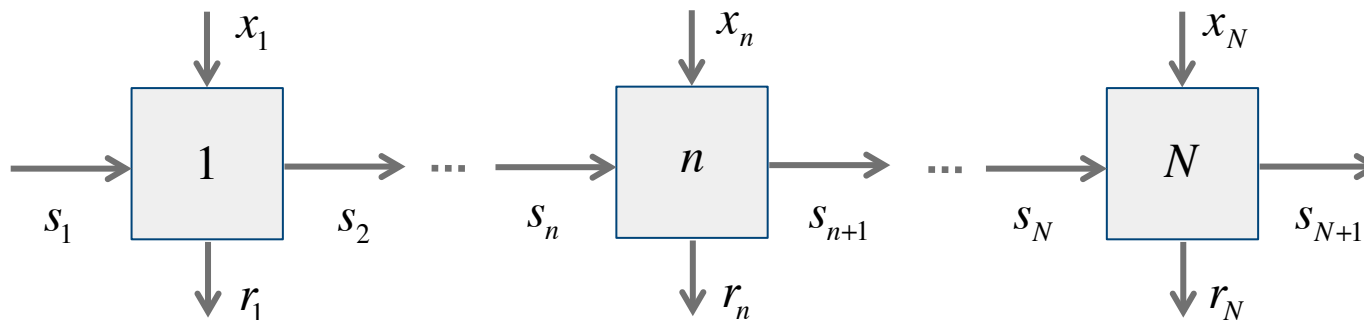
$x_n = \text{variable de decisi3n en la etapa } n$

$s_n = \text{estado al inicio de la etapa } n$

$s_{n+1} = \text{estado al final de la etapa } n$

$r_n(s_n, s_{n+1}, x_n) = \text{rendimiento en la etapa } n$

N etapas:



Ejemplo 1. Presentación (1)

Ejemplo 1 · El despertar de la Fuerza · El encuentro ⁽¹⁾ · Enunciado:

La joven REY (recolectora de chatarra en *Jakku*) está ahora al mando del HALCÓN MILENARIO. Junto a CHEWBACCA y R2-D2, REY parte hacia *Ahch-to* (J) para contactar con LUKE SKYWALKER. El viaje, programado desde *D'Qar* (A) en 4 etapas (figura 1), debe tener el menor riesgo posible.

(1) Basado en *el problema de la Diligencia* de HARVEY M. WAGNER (Stanford University)

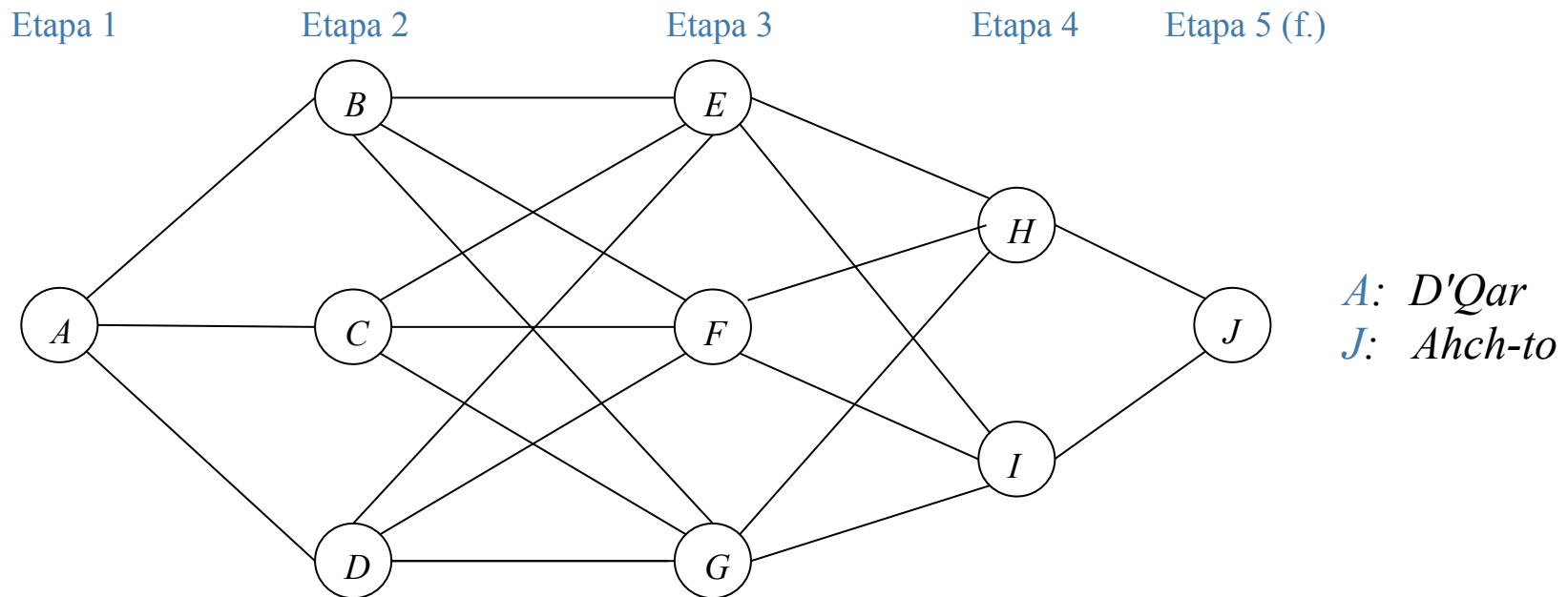


Figura 1: Esquema polietápico del viaje del *Halcón Milenario* desde D'Qar hasta Ahch-To.

Ejemplo 1. Presentación (2)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Riesgo aditivo entre trayectos (Valoración de 0 a 100):

	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	20	40	30

Riesgo Etapa 1

	<i>E</i>	<i>F</i>	<i>G</i>
<i>B</i>	70	40	60
<i>C</i>	30	20	40
<i>D</i>	40	10	50

Riesgo Etapa 2

	<i>H</i>	<i>I</i>
<i>E</i>	10	40
<i>F</i>	60	30
<i>G</i>	30	30

Riesgo Etapa 3

	<i>J</i>
<i>H</i>	30
<i>I</i>	40

Riesgo Etapa 4

A – B – F – I – J

A – D – F – I – J

...

Rutas: $3 \times 3 \times 2 = 18$



Ejemplo 1. Formalización

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Variables y costes:

Variables de decisión: Destino inmediato de la etapa n : n -ésimo viaje del *Halcón Milenario*.

$$x_n (n = 1, 2, 3, 4) \quad \text{Ruta : } A \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4, \text{ donde } x_4 = J$$

Coste de la mejor política (s, x_n) : Coste (riesgo) al inicio de la etapa n (n -ésimo viaje), en función del estado de partida y de la decisión a tomar, para afrontar las etapas restantes.

$$f_n(s, x_n) \quad (n = 1, 2, 3, 4) \quad \text{Estado: } s, \text{ Decisión: } x_n$$

Coste mínimo estado-etapa : Coste mínimo (riesgo) al inicio de la etapa n (n -ésimo viaje), en función del estado de partida, tomando la mejor decisión para afrontar las etapas restantes.

$$f_n^*(s) = \min_{x_n} f_n(s, x_n) = f_n(s, x_n^*) \quad (n = 1, 2, 3, 4, 5)$$

$$f_n(s, x_n) = c_{s, x_n} + f_{n+1}^*(x_n) \quad (n = 1, 2, 3, 4)$$



Ejemplo 1. Resolución (1)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Resolución Etapa 4:

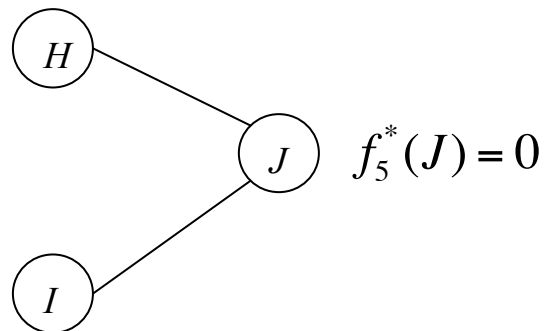
$$n = 4 : s_4 \in \{H, I\}, x_4 \in \{J\}$$

$$f_4(s_4, x_4) = c_{s_4, x_4} + f_5^*(x_4)$$

$$f_4^*(s_4) = \min_{x_4} f_4(s_4, x_4) = c_{s_4, J} + f_5^*(J)$$

$$x_4^* = J, f_5^*(J) = 0$$

	<i>J</i>
<i>H</i>	30
<i>I</i>	40



$$f_4^*(s_4) = \min_{x_4} f_4(s_4, x_4) = c_{s_4, J} + f_5^*(J)$$

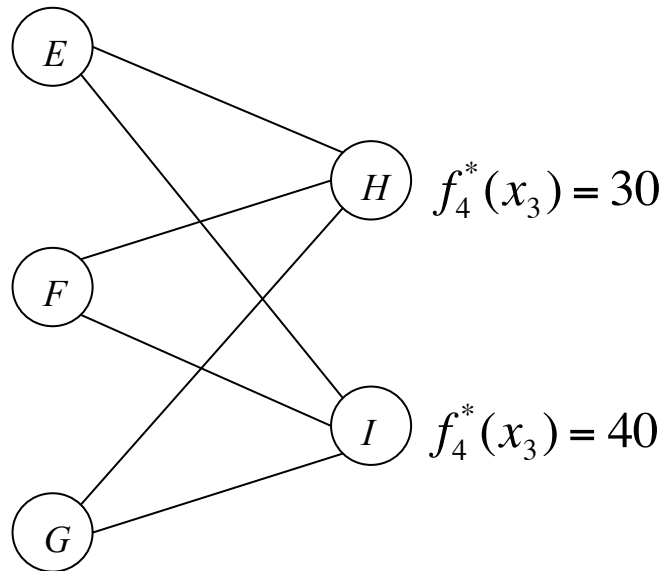
	$f_4(s, x_4)$		
s_4 / x_4	<i>J</i>	$f_4^*(s)$	x_4^*
<i>H</i>	30	30	<i>J</i>
<i>I</i>	40	40	<i>J</i>

Ejemplo 1. Resolución (2)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Resolución Etapa 3:

$$n = 3: s_3 \in \{E, F, G\}, x_3 \in \{H, I\}$$

$$f_3(s_3, x_3) = c_{s_3, x_3} + f_4^*(x_3)$$



$$f_3^*(s_3) = \min_{x_3} f_3(s_3, x_3) = f_3(s_3, x_3^*)$$

	H	I
E	10	40
F	60	30
G	30	30

s_3 / x_3	$f_3(s_3, x_3)$		$f_3^*(s)$	x_3^*
	H	I		
E	40	80	40	H
F	90	70	70	I
G	60	70	60	H

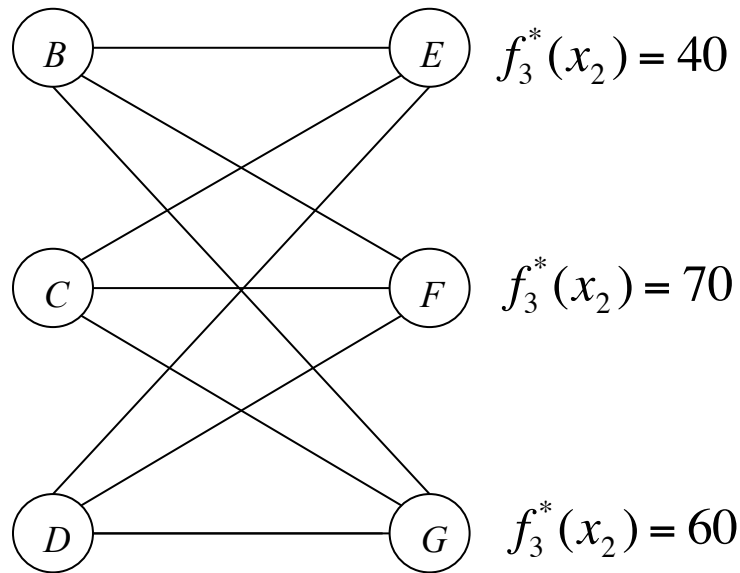


Ejemplo 1. Resolución (3)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Resolución Etapa 2:

$$n = 2 : s_2 \in \{B, C, D\}, x_2 \in \{E, F, G\}$$

$$f_2(s_2, x_2) = c_{s_2, x_2} + f_3^*(x_2)$$



$$f_2^*(s_2) = \min_{x_2} f_2(s_2, x_2) = f_2(s_2, x_2^*)$$

	<i>E</i>	<i>F</i>	<i>G</i>
<i>B</i>	70	40	60
<i>C</i>	30	20	40
<i>D</i>	40	10	50

	$f_2(s_2, x_2)$				
s_2 / x_2	<i>E</i>	<i>F</i>	<i>G</i>	$f_2^*(s)$	x_2^*
<i>B</i>	110	110	120	110	<i>E, F</i>
<i>C</i>	70	90	100	70	<i>E</i>
<i>D</i>	80	80	110	80	<i>E, F</i>

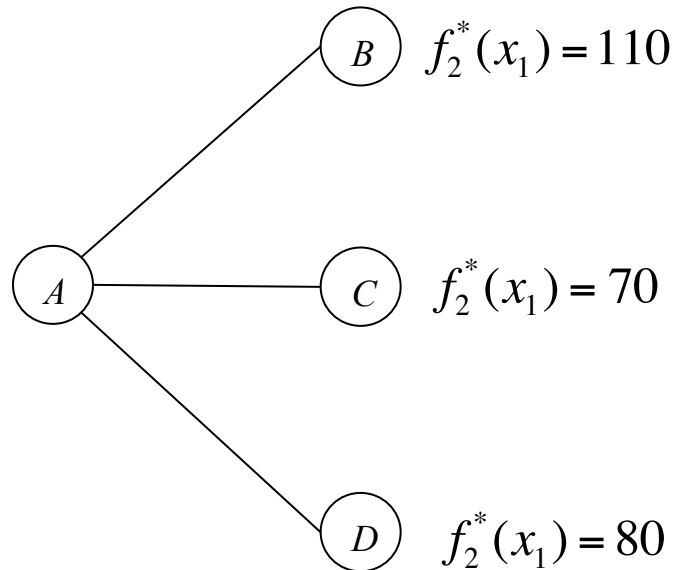


Ejemplo 1. Resolución (4)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Resolución Etapa 1:

$$n = 1: s_1 \in \{A\}, x_1 \in \{B, C, D\}$$

$$f_1(s_1, x_1) = c_{s_1, x_1} + f_2^*(x_1)$$



$$f_1^*(s_1) = \min_{x_1} f_1(s_1, x_1) = f_1(s_1, x_1^*)$$

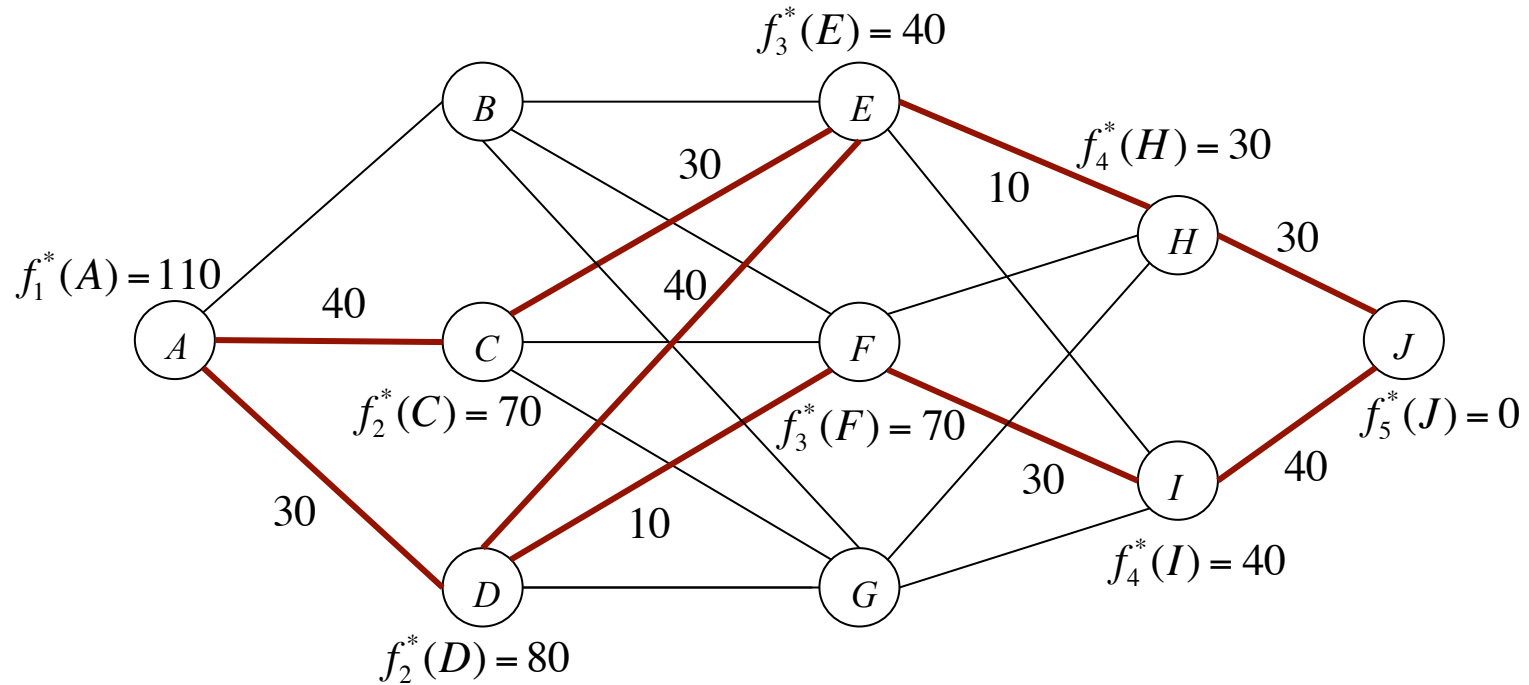
	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	20	40	30

	$f_1(s, x_1)$				
s_1 / x_1	<i>B</i>	<i>C</i>	<i>D</i>	$f_1^*(s)$	x_1^*
<i>A</i>	130	110	110	110	<i>C, D</i>



Ejemplo 1. Resolución (5)

Ejemplo 1 · El despertar de la Fuerza · El encuentro · Resolución Etapas 1 a 4 · Trayectorias:



A: D'Qar
J: Ahch-to

- Ruta 1: A → C → E → H → J
- Ruta 2: A → D → E → H → J
- Ruta 3: A → D → F → I → J



Programación Dinámica. Características de los problemas

Vocabulario y atributos:

- *División en etapas*: Agregación de *decisiones simples* para tomar una *decisión compleja*.
- *Etapas*: Se caracteriza por un conjunto de estados.
- *Decisión*: Transición de un estado a otro con *valor económico*. El efecto de la política de decisión es en cada etapa transformar el estado actual a otro del inicio de la etapa siguiente.
- *Valor Función económica*: Suma de valores económicos de las transiciones.
- *Política*: Secuencia de decisiones.
- *Política óptima*: Dado un estado actual, una política óptima para las etapas restantes es independiente de la política adoptada en etapas anteriores.
- *Principio de optimalidad*: La decisión inmediata óptima depende sólo del estado actual y no de cómo se llegó ahí.
- *Ecuación de recurrencia*: Permite establecer la *política óptima* a partir de cada estado posible.
- *Ecuación funcional*: Nombre de la *ecuación de recurrencia* cuando el número de etapas no está acotado.



Programación Dinámica. Tipología

Tipos de problemas y técnicas:

Según tipo de universo:

- Programación Dinámica Determinista (PDD).
- Programación Dinámica Probabilística (PDP).

Según horizonte (número de etapas):

- Finito determinado: número de etapas N fijado.
- Finito indeterminado: número de etapas N finito desconocido.
- Infinito: número de etapas N infinito.

Técnicas:

- Iteración en el espacio de los estados.
- Iteración en el espacio de las políticas.



Programación Dinámica determinista. Nomenclatura

Parámetros, variables y funciones:

N = número de etapas.

n = etiqueta de la etapa actual ($n = 1, 2, \dots, N$).

s_n = estado actual de la etapa n .

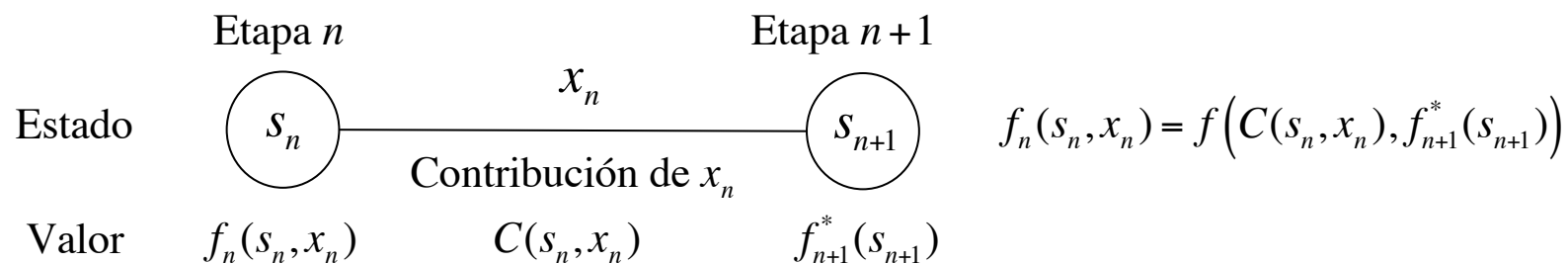
x_n = variable de decisión de la etapa n .

x_n^* = valor óptimo de x_n (dado s_n)

$f_n(s_n, x_n)$ = contribución a la función objetivo de las etapas $n, n+1, \dots, N$, desde s_n con decisión x_n .

$f_n^*(s_n) = f_n(s_n, x_n^*)$ = contribución óptima, desde s_n con la mejor decisión.

$f_n^*(s_n) = \min_{x_n} \{f_n(s_n, x_n)\}$ o $f_n^*(s_n) = \max_{x_n} \{f_n(s_n, x_n)\}$



El problema de distribución de esfuerzos

Descripción:

Grupo de problemas en el que existe sólo una clase de *Recurso* que debe asignarse a un número de *Actividades*. El objetivo es determinar cómo *distribuir el Recurso* (Esfuerzo) de la forma más *eficaz*.

Formulación:

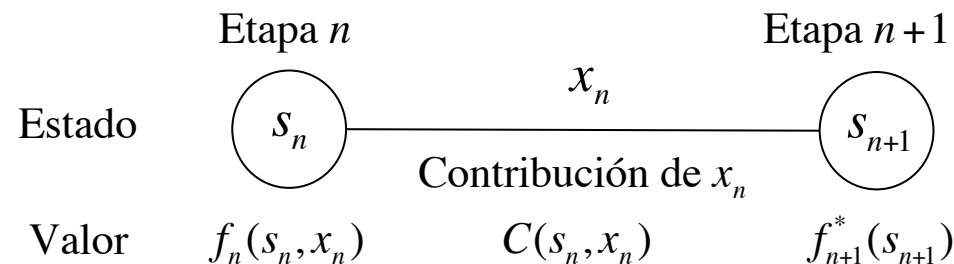
N = número de Actividades.

n = etiqueta de la Actividad ($n = 1, 2, \dots, N$). Orden arbitrario.

s_n = (estado) Cantidad de recursos disponibles para asignarse a las Actividades $n, n + 1, \dots, N$.

x_n = cantidad de recursos asignados a la Actividad n .

x_n^* = valor óptimo de x_n (dado s_n)



$$s_{n+1} = s_n - x_n$$

$$f_n(s_n, x_n) = f(C(s_n, x_n), f_{n+1}^*(s_n - x_n))$$



Ejemplo 2. Presentación

Ejemplo 2 · El problema de los 4 Equipos de investigación · Enunciado:

El departamento encargado de un proyecto del Gobierno debe resolver un problema. Se dispone de 4 equipos de investigación (E1, E2, E3 y E4). A priori, las probabilidades de fracaso de los equipos son: 0.55 (E1), 0.60 (E2), 0.70 (E3) y 0.80 (E4). Si se asignan hasta dos científicos adicionales, las probabilidades de fracaso de los equipos se alteran según Tabla 1.

<i>Nuevos científicos</i>	<i>Probabilidades de fracaso</i>			
	E1	E2	E3	E4
0	0.55	0.60	0.70	0.80
1	0.30	0.45	0.50	0.60
2	0.25	0.30	0.35	0.40

Tabla 1. Probabilidades de fracaso del problema del proyecto del Gobierno

Objetivo: Asignar dos científicos en total con el propósito de minimizar la probabilidad de que los cuatro equipos fracasen.



Ejemplo 2. Formalización

Ejemplo 2 · El problema de los 4 Equipos de investigación · Objetivo, variables y costes:

$p_i(x_i)$ = probabilidad de fracaso del Equipo i si se le asignan x_i científicos adicionales.

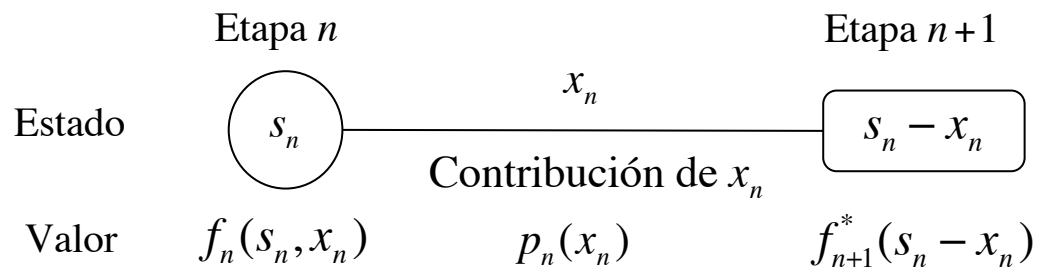
$$\text{Minimizar } P = \prod_{i=1}^4 p_i(x_i) = p_1(x_1)p_2(x_2)p_3(x_3)p_4(x_4)$$

s.a.:

$$\sum_{i=1}^4 x_i = 2$$

$$x_i \in \{0,1,2\} \quad \forall i = 1, \dots, 4$$

$$f_n(s_n, x_n) = p_n(x_n) \cdot \min \prod_{i=n+1}^4 p_i(x_i); \quad \sum_{i=n}^4 x_i = s_n; \quad f_n^*(s_n) = \min_{0 \leq x_n \leq s_n} f_n(s_n, x_n)$$



$$f_n(s_n, x_n) = p_n(x_n) \cdot f_{n+1}^*(s_n - x_n)$$



Ejemplo 2. Resolución (1)

Ejemplo 2 · El problema de los 4 Equipos de investigación · Resolución Equipos 4 y 3:

$$n = 4: s_4 \in \{0,1,2\}, x_4 \in \{0,1,2\}$$

$$f_4(s_4, x_4) = p_4(x_4) \cdot f_5(0)$$

$$f_5(0) = 1$$

$$f_4^*(s_4) = f_4(s_4, x_4) = p_4(x_4)$$

	$f_4(s_4, x_4)$				
s_4 / x_4	0	1	2	$f_4^*(s_4)$	x_4^*
0	0.80			0.80	0
1		0.60		0.60	1
2			0.40	0.40	2

$$n = 3: s_3 \in \{0,1,2\}, x_3 \in \{0,1,2\}$$

$$f_3(s_3, x_3) = p_3(x_3) \cdot f_4^*(s_3 - x_3)$$

$$f_3^*(s_3) = \min_{0 \leq x_3 \leq s_3} f_3(s_3, x_3)$$

	$f_3(s_3, x_3)$				
s_3 / x_3	0	1	2	$f_3^*(s_3)$	x_3^*
0	0.560			0.560	0
1	0.420	0.400		0.400	1
2	0.280	0.300	0.280	0.280	0,2



Ejemplo 2. Resolución (2)

Ejemplo 2 · El problema de los 4 Equipos de investigación · Resolución Equipos 2 y 1:

$$n = 2 : s_2 \in \{0,1,2\}, x_2 \in \{0,1,2\}$$

$$f_2(s_2, x_2) = p_2(x_2) \cdot f_3^*(s_2 - x_2)$$

$$f_2^*(s_2) = \min_{0 \leq x_2 \leq s_2} f_2(s_2, x_2)$$

	$f_2(s_2, x_2)$				
s_2 / x_2	0	1	2	$f_2^*(s_2)$	x_2^*
0	0.336			0.336	0
1	0.240	0.252		0.240	0
2	0.168	0.180	0.168	0.168	0,2

$$n = 1 : s_1 \in \{2\}, x_1 \in \{0,1,2\}$$

$$f_1(s_1, x_1) = p_1(x_1) \cdot f_2^*(s_1 - x_1)$$

$$f_1^*(s_1) = \min_{0 \leq x_1 \leq s_1} f_1(s_1, x_1)$$

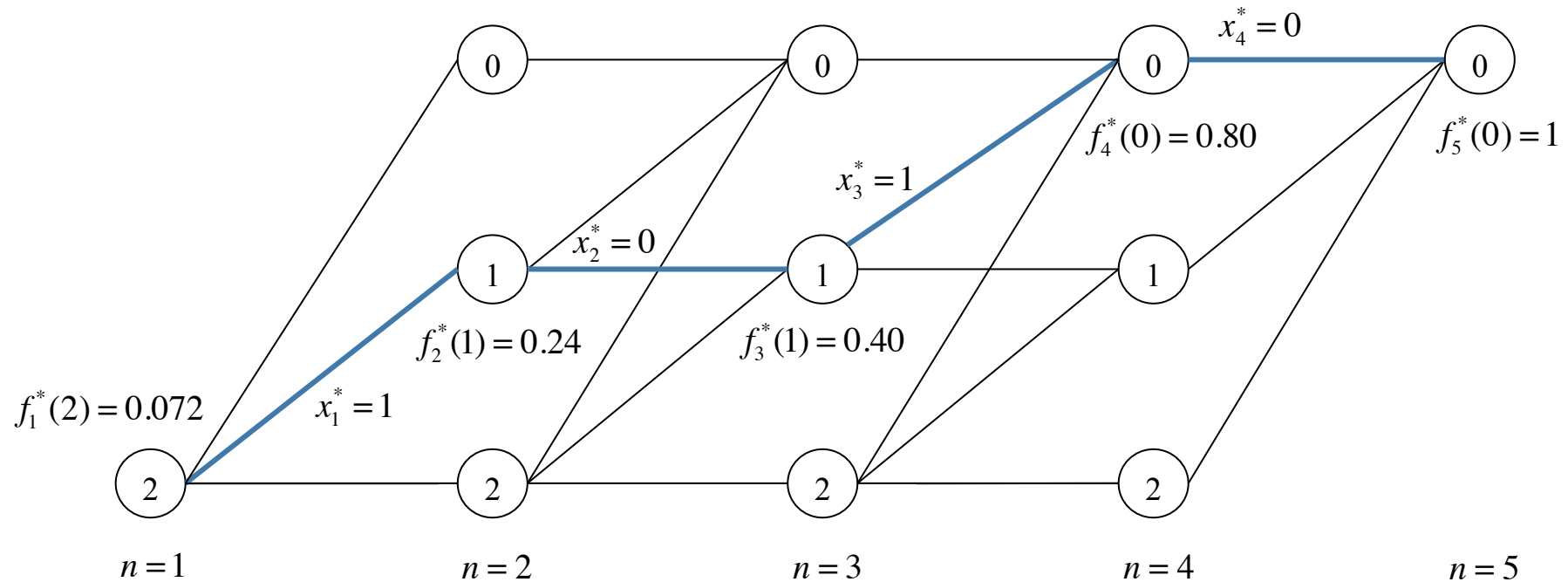
	$f_1(s_1, x_1)$				
s_1 / x_1	0	1	2	$f_1^*(s_1)$	x_1^*
2	0.092	0.072	0.084	0.072	1

$$x_1^* = 1 \rightarrow s_2 = s_1 - x_1^* = 1 \rightarrow x_2^* = 0 \rightarrow s_3 = 1 \rightarrow x_3^* = 1 \rightarrow s_4 = 0 \rightarrow x_4^* = 0 \quad P_0 = 0.185, P = 0.072$$



Ejemplo 2. Resolución (3)

Ejemplo 2 · El problema de los 4 Equipos de investigación · Resolución Equipos 1 a 4:



El problema Dynamic Lot Sizing (DLS)

Descripción:

- Grupo de problemas de stocks que considera un tipo de artículo con una demanda variable en el tiempo.
- Se establece un horizonte finito y a cada periodo se asigna una demanda concreta.
- No se admiten roturas y se consideran dos tipos de costes: fabricación de lote (lanzamiento), y posesión de stock.
- El objetivo es determinar cuando fabricar y el tamaño de los lotes para satisfacer la demanda total de la forma más **eficaz**.

Enfoques:

- Programación dinámica
- Programación lineal entera mixta.
- Algoritmo de Wagner-Whitin.
- Camino extremo en grafo polietápico.



Ejemplo 3. Presentación

Ejemplo 3 · El problema del lote dinámico · Enunciado:

Se dispone del plan de demanda de un artículo, con un horizonte de 10 periodos. En la Tabla 2 se recogen los valores (en cientos de unidades) de la demanda en cada periodo. El coste unitario de posesión de stock se valora en 1 unidad monetaria por periodo, mientras que el coste de fabricar un lote, independientemente de su tamaño, se valora en 30 um por cada lanzamiento a línea. Determinar el plan de producción más eficaz para satisfacer sin roturas la demanda total.

Periodo n	1	2	3	4	5	6	7	8	9	10
d_n	5	6	9	7	3	10	6	4	8	2

Tabla 2. Plan de demanda (no homogénea) de un artículo l. Horizonte 10 periodos.

$$c_A = 30 \frac{um}{orden}; c_h = 1 \frac{um}{up \times ut}$$



Ejemplo 3. Resolución lotes estáticos

Ejemplo 3 · El problema del lote dinámico · Resolución JIT y Harris-Wilson:

$$1. \text{ Lot for Lot - Just in Time: } \left\{ \begin{array}{l} Q_n = d_n (n=1, \dots, N) \\ I_n = 0 (n=1, \dots, N) \end{array} \right\} \Rightarrow F_N = c_A N = 30 \times 10 = 300 \text{ um}$$

$$2. \text{ Regla de Harris-Wilson: } \left\{ D = \sum_{n=1}^N d_n, Q^* \approx \sqrt{\frac{2c_A D}{NC_h}}, T^* \approx \sqrt{\frac{2c_A}{NC_h D}} \right\} \Rightarrow Q^* \approx 20, T^* \approx 3/10$$

Periodo n	1	2	3	4	5	6	7	8	9	10
d_n	5	6	9	7	3	10	6	4	8	2
Q_n	20			20			20			
I_n	15	9	0	13	10	0	14	10	2	0
\dot{f}_n	45	9	0	43	10	0	44	10	2	0

$$F_N = 163 \text{ um}$$



Ejemplo 3. Programa matemático

Ejemplo 3 · El problema del lote dinámico · Programación lineal entera (binaria).

c_A, c_h = Costes de lanzamiento y posesión

d_n = Demanda en el periodo n ($n = 1, \dots, N$): $D = \sum_{n=1}^N d_n$

x_n = Variable binaria de lanzamiento en el periodo n ($n = 1, \dots, N$) ($x_n = 1 \Rightarrow$ lanzamiento)

Q_n = Lote de fabricación en el periodo n ($n = 1, \dots, N$)

I_n = Posición de stock en el periodo n ($n = 1, \dots, N$)

$$\text{Min } F = \sum_{n=1}^N c_A x_n + \sum_{n=1}^N c_h I_n$$

s.a.:

$$I_n = \sum_{t=1}^n Q_t - \sum_{t=1}^n d_t \quad n = 1, \dots, N$$

$$Q_n \leq D \cdot x_n \quad n = 1, \dots, N$$

$$Q_n, I_n \geq 0 \quad n = 1, \dots, N$$

$$x_n \in \{0, 1\} \quad n = 1, \dots, N$$

$$I_N = 0$$



Ejemplo 3. Resolución programación dinámica (1)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Formulación:

n = etapa correspondiente al periodo del horizonte del plan de producción ($n = 1, \dots, N$)

t = periodo en el que se realiza el último lanzamiento ($t = 1, \dots, n$) asociado al estado (n, t)

x_n = variable de decisión en el periodo n ($n = 1, \dots, N$): $x_n \in \{0, 1\}$, $x_n = 1 \Rightarrow$ (lanzar) $t = n$

x_n^* = valor de la mejor decisión en la etapa n ($n = 1, \dots, N$)

$f_n(t, x_n)$ = coste mínimo para cubrir la demanda desde el periodo n ($n = 1, \dots, N$) hasta el N cuando el último lanzamiento se realiza en el periodo t ($t = 1, \dots, n$)

$f_n^*(t)$ = coste mínimo para cubrir la demanda desde el periodo n ($n = 1, \dots, N$) hasta el N adoptando la mejor decisión

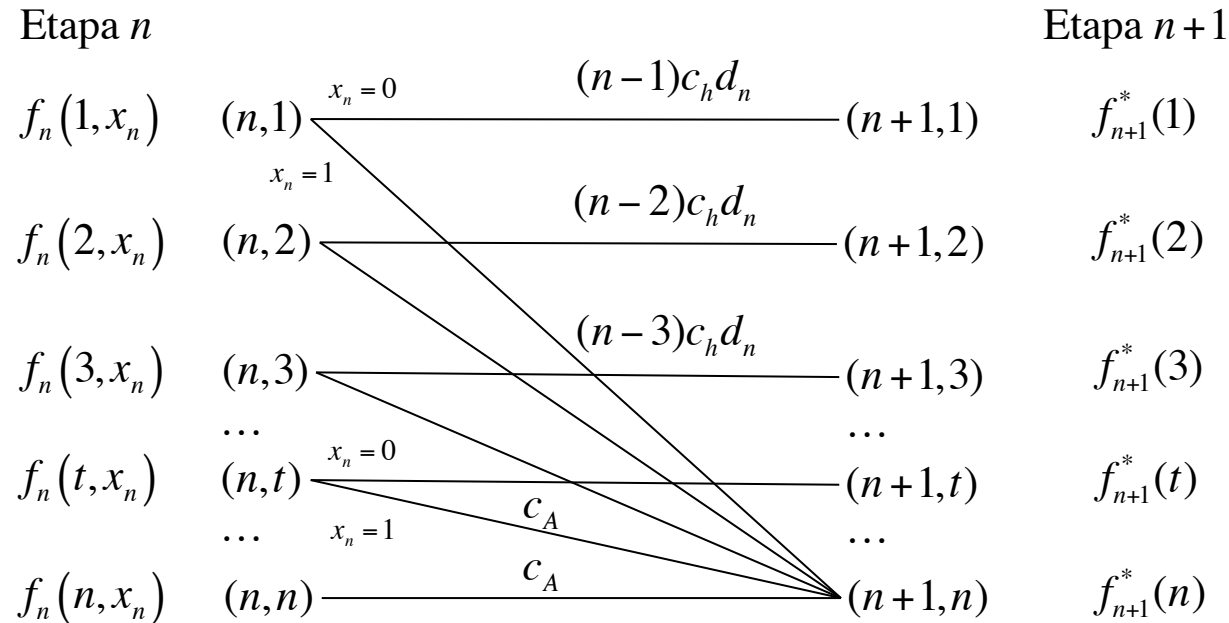
$$(1) \quad f_n(t, x_n) = \begin{cases} f_{n+1}^*(t) + (n-t) \cdot c_h \cdot d_n & (x_n = 0) \\ f_{n+1}^*(n) + c_A & (x_n = 1) \end{cases}$$

$$(2) \quad f_n^*(t) = f_n(t, x_n^*) = \min_{x_n \in \{0, 1\}} \{f_n(t, x_n)\} \Rightarrow f_n^*(t) = \min \left\{ \left[f_{n+1}^*(t) + (n-t) \cdot c_h \cdot d_n \right], \left[f_{n+1}^*(n) + c_A \right] \right\}$$



Ejemplo 3. Resolución programación dinámica (2)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Esquema:



$$f_n(t, x_n) = \begin{cases} f_{n+1}^*(t) + (n-t) \cdot c_h \cdot d_n & (x_n = 0) \\ f_{n+1}^*(n) + c_A & (x_n = 1) \end{cases} \quad \begin{matrix} f_{11}^*(1) = f_{11}^*(2) = \dots = f_{11}^*(9) = 0 \\ f_{11}^*(10) = 30 \end{matrix}$$

$$f_n^*(t) = \min \left\{ \left[f_{n+1}^*(t) + (n-t) \cdot c_h \cdot d_n \right], \left[f_{n+1}^*(n) + c_A \right] \right\}$$



Ejemplo 3. Resolución programación dinámica (3)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Cálculos 1:

n	10	9	8	7	6	5	4	3	2	1
$t \setminus d_n$	2	8	4	6	10	3	7	9	6	5
1	18	32	42	56	76	88	109	119	125	155
2	16	32	42	56	76	85	99	108	138	
3	14	32	42	56	76	82	89	119		
4	12	32	42	56	76	79	109			
5	10	32	42	54	64	94				
6	8	32	40	46	76					
7	6	22	26	56						
8	4	12	42							
9	2	32								
10	30									
x_n		1			1			1		1

$$f_{11}^*(1) = f_{11}^*(2) = \dots = f_{11}^*(9) = 0$$

$$f_{11}^*(10) = 30$$



Ejemplo 3. Resolución programación dinámica (4)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Solución 1:

$f_1^*(1) = 155 \text{ um}$ · Coste óptimo para satisfacer la demanda de $n = 1, \dots, N$

Solución 1: $\{x_1 = 1, x_3 = 1, x_6 = 1, x_9 = 1\} \Rightarrow \{Q_1 = 11, Q_3 = 19, Q_6 = 20, Q_9 = 10\}$

Periodo n	1	2	3	4	5	6	7	8	9	10
d_n	5	6	9	7	3	10	6	4	8	2
x_n	1		1			1			1	
Q_n	11		19			20			10	
I_n	6	0	10	3	0	10	4	0	2	0
\dot{f}_n	36	0	40	3	0	40	4	0	32	0

$$F_N = 36 + 40 + 3 + 40 + 4 + 32 = 155 \text{ um}$$



Ejemplo 3. Resolución programación dinámica (5)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Cálculos 2:

n	10	9	8	7	6	5	4	3	2	1
$t \setminus d_n$	2	8	4	6	10	3	7	9	6	5
1	18	32	42	56	76	88	109	119	125	155
2	16	32	42	56	76	85	99	108	138	
3	14	32	42	56	76	82	89	119		
4	12	32	42	56	76	79	109			
5	10	32	42	54	64	94				
6	8	32	40	46	76					
7	6	22	26	56						
8	4	12	42							
9	2	32								
10	30									
x_n					1			1		1

$f_{11}^*(1) = f_{11}^*(2) = \dots = f_{11}^*(9) = 0$
 $f_{11}^*(10) = 30$

Ejemplo 3. Resolución programación dinámica (6)

Ejemplo 3 · El problema del lote dinámico · Programación dinámica · Solución 2:

$f_1^*(1) = 155 \text{ um}$ · Coste óptimo para satisfacer la demanda de $n = 1, \dots, N$

Solución 1: $\{x_1 = 1, x_3 = 1, x_6 = 1\} \Rightarrow \{Q_1 = 11, Q_3 = 19, Q_6 = 30\}$

Periodo n	1	2	3	4	5	6	7	8	9	10
d_n	5	6	9	7	3	10	6	4	8	2
x_n	1		1			1				
Q_n	11		19			30				
I_n	6	0	10	3	0	20	14	10	2	0
\dot{f}_n	36	0	40	3	0	50	14	10	2	0

$$F_N = 36 + 40 + 3 + 50 + 14 + 10 + 2 = 155 \text{ um}$$



Ejemplo 3. Resolución WAGNER-WHITIN (1)

Ejemplo 3 · El problema del lote dinámico · Algoritmo de WAGNER-WHITIN:

f_n = coste mínimo para cubrir la demanda hasta el periodo n ($n = 1, \dots, N$)

$f_{n,t}$ = coste mínimo para cubrir la demanda hasta el periodo n ($n = 1, \dots, N$) cuando el último lanzamiento se realiza en el periodo t ($t = 1, \dots, n$)

$$(1) \quad f_n = \min_{1 \leq t \leq n} \{f_{n,t}\}$$

$$(2) \quad f_{n,t} = f_{t-1} + c_A + c_h(d_{t+1} + 2 \cdot d_{t+2} + \dots + (n-t) \cdot d_n) = f_{t-1} + c_A + c_h \sum_{k=t+1}^n (k-t) \cdot d_k$$

$$f_0 = 0$$

$$f_1 = f_{1,1} = c_A$$

$$f_2 = \min \{f_{2,1}, f_{2,2}\} = \min \{[f_0 + c_A + c_h d_2], [f_1 + c_A]\}$$

$$f_3 = \min \{f_{3,1}, f_{3,2}, f_{3,3}\} = \min \{[f_0 + c_A + c_h d_2 + 2c_h d_3], [f_1 + c_A + c_h d_3], [f_2 + c_A]\}$$

...

$$f_n = \min \{f_{n,1}, f_{n,2}, f_{n,3}, \dots, f_{n,n}\} = \min \left\{ \left[f_0 + c_A + c_h \sum_{k=2}^n (k-1) \cdot d_k \right], \left[f_1 + c_A + c_h \sum_{k=3}^n (k-2) \cdot d_k \right], \dots, [f_{n-1} + c_A] \right\}$$



Ejemplo 3. Resolución WAGNER-WHITIN (2)

Ejemplo 3 · El problema del lote dinámico · Algoritmo de WAGNER-WHITIN:

Periodo n	1	2	3	4	5	6	7	8	9	10
d_n	5	6	9	7	3	10	6	4	8	2

$$c_A = 30 \frac{um}{\text{lanzamiento}}; c_h = 1 \frac{um}{up \times ut}$$

$$f_n = \min_{1 \leq t \leq n} \{f_{n,t}\}, f_{n,t} = f_{t-1} + c_A + c_h(d_{t+1} + 2 \cdot d_{t+2} + \dots + (n-t) \cdot d_n) = f_{t-1} + c_A + c_h \cdot \sum_{\tau=1}^{n-t} \tau \cdot d_{t+\tau}$$

$$f_0 = 0$$

$$f_1 = f_{1,1} = 30 \rightarrow t^* = 1$$

$$f_2 = \min\{(30+6), (30+30)\} = \min\{36, 60\} = 36 \rightarrow t^* = 1$$

$$f_3 = \min\{(36+18), (60+9), (36+30)\} = \min\{54, 69, 66\} = 54 \rightarrow t^* = 1$$

$$f_4 = \min\{(54+21), (69+14), (66+7), (54+30)\} = \min\{75, 83, 73, 84\} = 73 \rightarrow t^* = 3$$

$$f_5 = \min\{(75+12), (83+9), (73+6), (84+3), (73+30)\} = \min\{87, 92, 79, 87, 103\} = 79 \rightarrow t^* = 3$$

...

$$f_n = \min\{f_{n,1}, f_{n,2}, f_{n,3}, \dots, f_{n,n}\} = \min_{1 \leq t \leq n} \{f_{n,t}\}$$



Ejemplo 3. Resolución WAGNER-WHITIN (3)

Ejemplo 3 · El problema del lote dinámico · Algoritmo de WAGNER-WHITIN:

Periodo t	1	2	3	4	5	6	7	8	9	10
d_t	5	6	9	7	3	10	6	4	8	2
$n = t$	30	60	66	84	103	109	137	145	153	177
$n = t + 1$	36	69	73	87	113	115	141	153	155*	
$n = t + 2$	54	83	79	107	125	123	157	157		
$n = t + 3$	75	92	109	125	137	147	163			
$n = t + 4$	87		133	141		155*				
$n = t + 5$			153							

Periodo n	1	2	3	4	5	6	7	8	9	10	
Q_n	11	-	19	-	-	20	-	-	10	-	$f_{10} = 155$
Q_n	11	-	19	-	-	30	-	-	-	-	$f_{10} = 155$



Ejemplo 3. Resolución grafo polietápico (1)

Ejemplo 3 · El problema del lote dinámico · Camino extremo en grafo polietápico:

f_n = coste mínimo para cubrir la demanda hasta la etapa n ($n = 1, \dots, N$)

$f_{n,t}$ = coste mínimo para cubrir la demanda hasta la etapa n ($n = 1, \dots, N$) cuando el último lanzamiento se realiza en la etapa t ($t = 1, \dots, n$)

$$(1) \quad f_n = \min_{1 \leq t \leq n} \{ f_{n,t} \}$$

$$(2) \quad f_{n,t} = \begin{cases} f_{n-1,t} + (n-t) \cdot c_h \cdot d_n & (t = 1, \dots, n-1) \\ \min_{1 \leq t' \leq n-1} [f_{n-1,t'} + c_A] = f_{n-1} + c_A & (t = n) \end{cases}$$

$$f_0 = 0$$

$$f_{1,1} = c_A$$

$$f_{2,1} = f_{1,1} + c_h d_2, \quad f_{2,2} = f_1 + c_A$$

$$f_{3,1} = f_{2,1} + 2c_h d_3, \quad f_{3,2} = f_{2,2} + c_h d_3, \quad f_{3,3} = f_2 + c_A$$

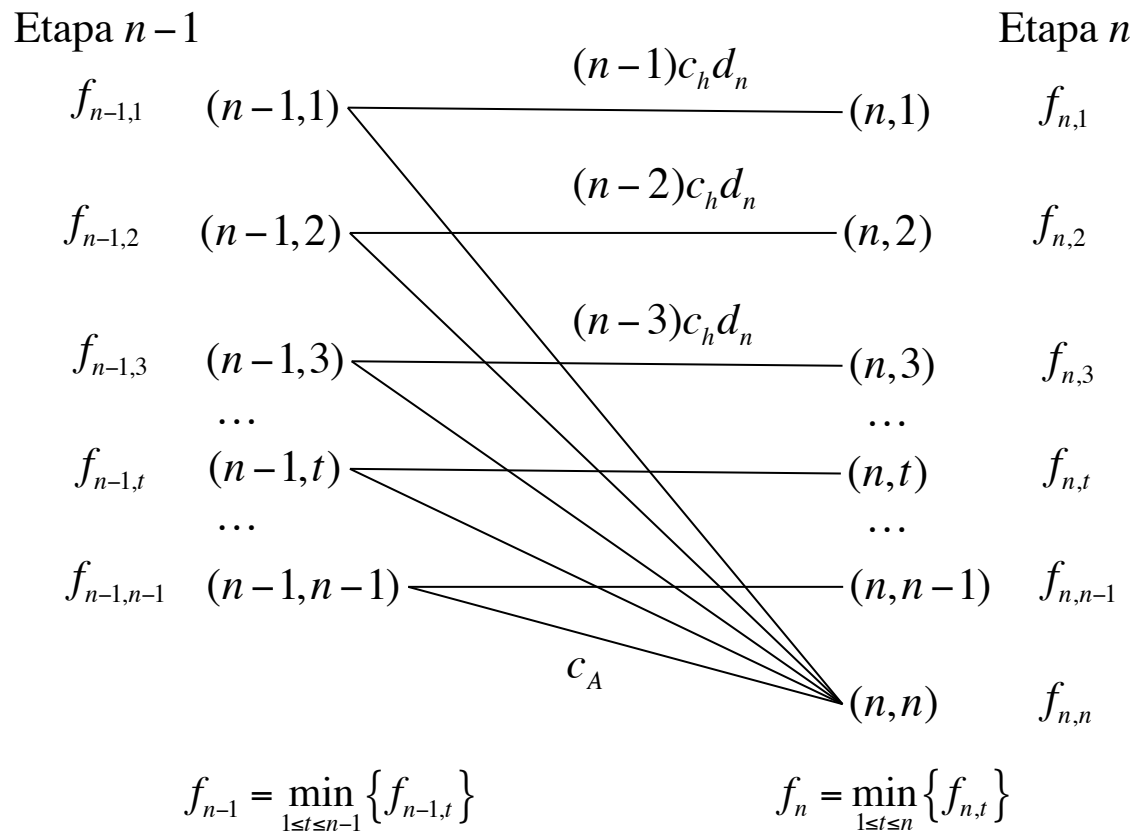
...

$$f_{n,1} = f_{n-1,1} + (n-1) \cdot c_h \cdot d_n, \quad f_{n,2} = f_{n-1,2} + (n-2) \cdot c_h \cdot d_n, \dots, \quad f_{n,n} = f_{n-1} + c_A$$



Ejemplo 3. Resolución grafo polietápico (2)

Ejemplo 3 · El problema del lote dinámico · Camino extremo en grafo polietápico:



Ejemplo 3. Resolución grafo polietápico (3)

Ejemplo 3 · El problema del lote dinámico · Camino extremo en grafo polietápico:

n	1	2	3	4	5	6	7	8	9	10
$t \setminus d_n$	5	6	9	7	3	10	6	4	8	2
1	30	36	54	75	87	137	173	201	265	283
2		60	69	83	92	132	162	186	242	258
3			66	73	79	109	133	153	201	215
4				84	87	107	125	141	181	193
5					103	113	125	137	169	179
6						109	115	123	147	155
7							137	141	157	163
8								145	153	157
9									153	155
10										177
f_n	30	36	54	73	79	107	115	123	147	155



El Hilo de Ariadna

- Sólo hay una manera -recitó, en efecto, Guillermo- de encontrar la salida de un laberinto. Al llegar a cada nudo nuevo, o sea hasta el momento no visitado, se harán tres signos en el camino de llegada. Si se observan signos en alguno de los caminos del nudo, ello indicará que el mismo ya ha sido visitado, y entonces sólo se marcará un signo en el camino de llegada.

Cuando todos los pasos de un nudo ya estén marcados, habrá que retroceder. Pero si todavía quedan uno o dos pasos sin marcar, se escogerá uno al azar, y se lo marcará con dos signos. Cuando se escoja un paso marcado con un solo signo, se marcarán dos más, para que ya tenga tres. Si al llegar a un nudo sólo se encuentran pasos marcados con tres signos, o sea, si no quedan pasos que aún falte marcar, ello indicará que ya se han recorrido todas las partes del laberinto.

UMBERTO ECO (1932 – 2016)

El Nombre de la Rosa (1980)

