UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

PROGRAMACIÓN DINÁMICA

- Problema de la Diligencia
- Problema de la Mochila
- Programación de producción e inventario

UAM
Universidad Autónoma de Madrid

Asignatura ..... **Grupo** .....

Apellidos ..... **Nombre** .....

Ejercicio del día .....

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

PROGRAMACIÓN DINÁMICA

Presenta un enfoque general para la solución de problemas en donde se necesitan tomar decisiones en etapas sucesivas. De forma que las decisiones tomadas en una etapa condicionan la evolución futura del sistema, afectando a las situaciones en las que el sistema se encontrará en el futuro (denominadas estados), y a las decisiones que se plantearán en el futuro.

A diferencia de la programación lineal, el modelado de problemas de programación dinámica no sigue una forma estándar, una solución está formada por una serie de decisiones.
Para un problema de programación dinámica es necesario especificar cada uno de los problemas que lo caracterizan.

La resolución general de un problema de programación dinámica se fracciona en orden inverso en el análisis recursivo de casa una de las etapas. Es decir, se comienza por la última etapa pasando en cada iteración a la etapa antecesora. De este modo, el análisis de la primera etapa finaliza con la obtención del óptimo del problema.

FUNCIÓN RECURSIVA DILIGENCIA: Dados unos nodos y unos arcos que conectan estos nodos, el problema de la diligencia intenta encontrar la ruta más corta que conecta un nodo de arranque con el nodo final (destino).

Siendo $s \equiv$ estado de inicio y $j \equiv$ estado final

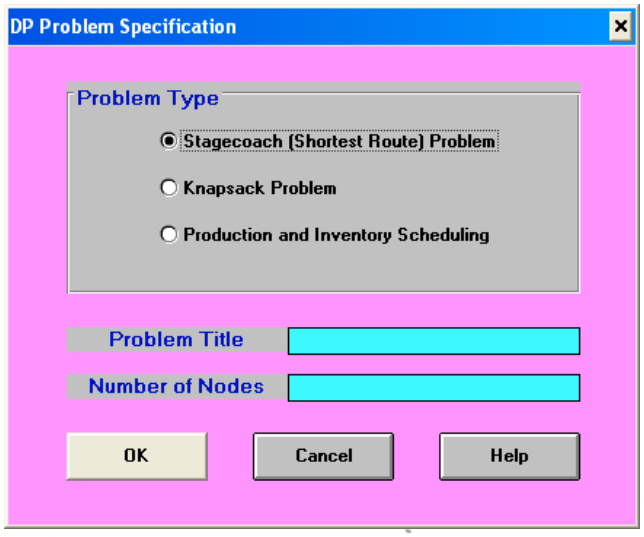
$n \equiv$ fase (normalmente, representa el número de arcos hasta el destino)

$C(s, j) \equiv$ costo o distancia de ir de s hasta j

$f(n, s) \equiv$ política de costo mínimo al encontrarse en el estado s de la etapa n

Función recursiva dinámica: $f(n, s) = \text{mínimo}\left[C(s, j) + f(n-1, j)\right] \quad \forall \text{arco}(s, j)$

WinQSB / Dynamic Programming /Problem Specification



Winqsb incorpora tres modelos diferentes:

- Stagecoach Problem (Problema Diligencia)
- Knapsack Problem (Problema Mochila)
- Production and Inventory Scheduling (programación de producción e inventario)

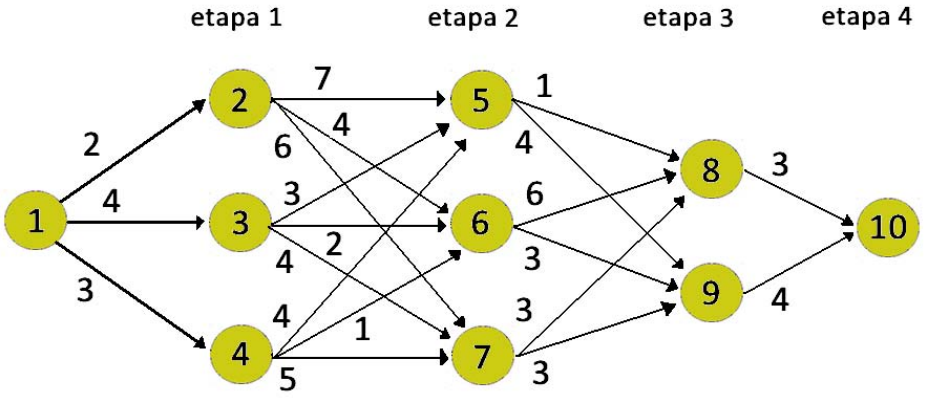
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

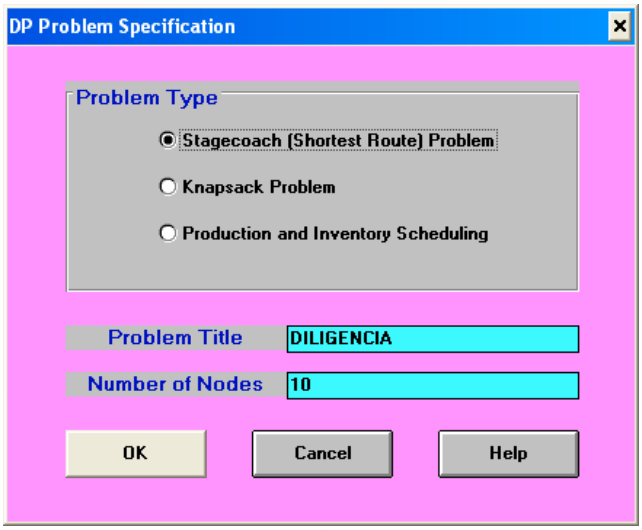
Apellidos ..... Nombre .....

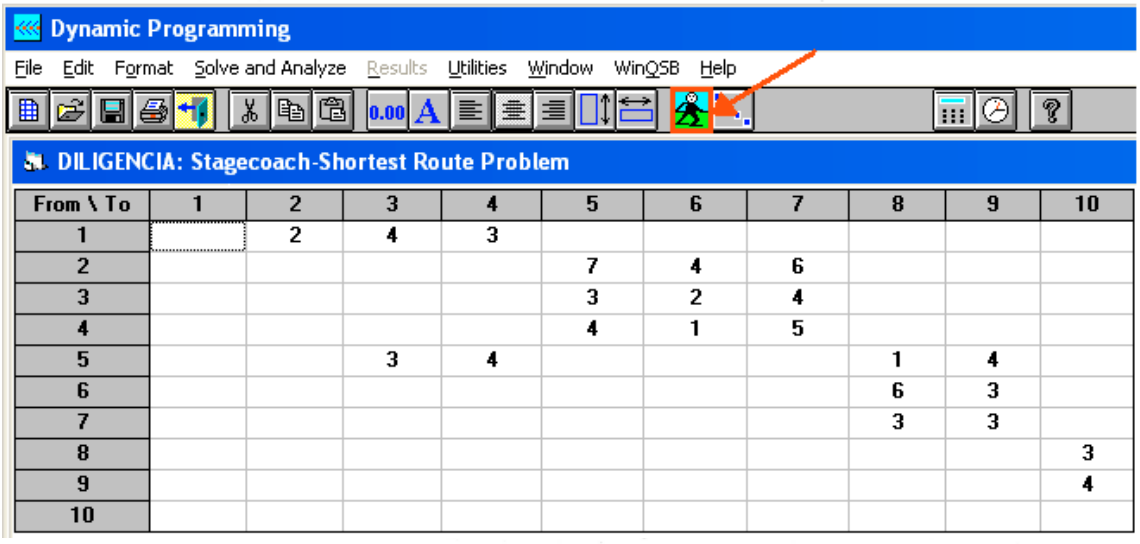
Ejercicio del día .....

PROBLEMA DE LA DILIGENCIA: En el grafo se representan las posibles rutas de ir de la Ciudad 1 hasta la Ciudad 10. Cada nodo representa a una Ciudad y los arcos el costo o distancia de ir e un nodo a otro. Se supone que los desplazamientos tienen la misma duración. Calcular la solución óptima.



Solución:





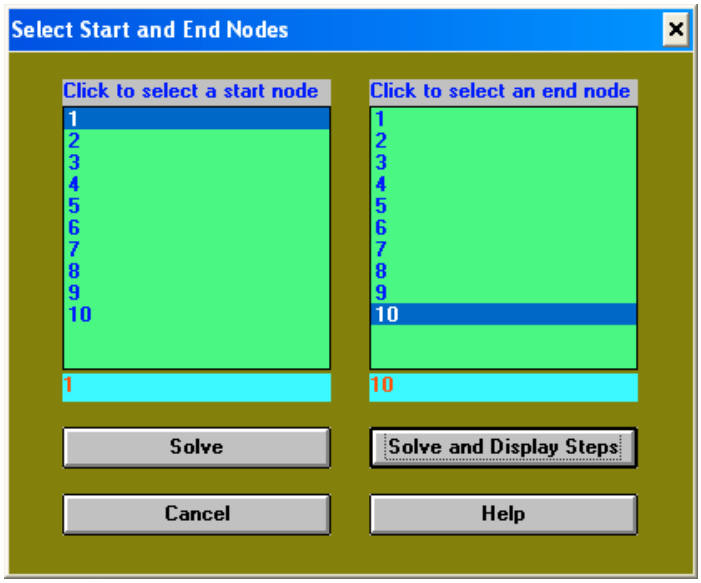
From \ To	1	2	3	4	5	6	7	8	9	10
1		2	4	3						
2					7	4	6			
3					3	2	4			
4					4	1	5			
5			3	4				1	4	
6								6	3	
7								3	3	
8										3
9										4
10										

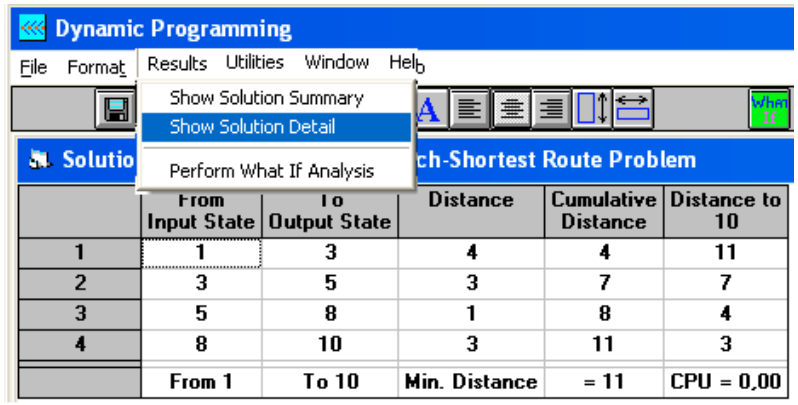
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

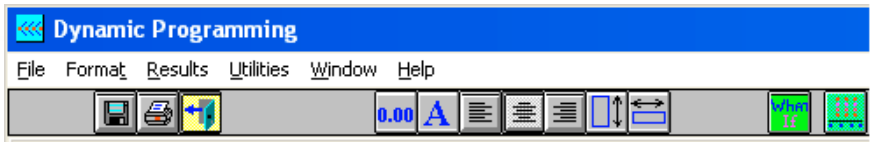
Apellidos ..... Nombre .....

Ejercicio del día .....





	From Input State	To Output State	Distance	Cumulative Distance	Distance to 10
1	1	3	4	4	11
2	3	5	3	7	7
3	5	8	1	8	4
4	8	10	3	11	3
	From 1	To 10	Min. Distance	= 11	CPU = 0,00



Solution Steps for DILIGENCIA: Stagecoach-Shortest Route Problem

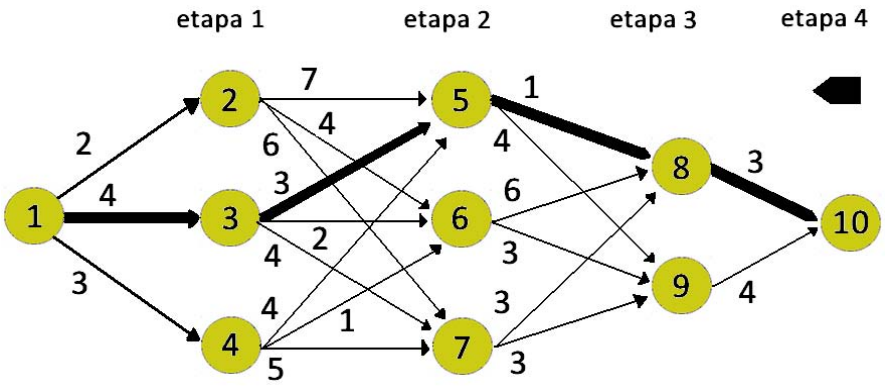
	Stage	From Input State	To Output State	Distance	Distance to 10	Status
1	1	1	3	4	11	Optimal
2	2	2	5	7	11	
3	2	3	5	3	7	Optimal
4	2	4	5	4	8	
5	3	5	8	1	4	Optimal
6	3	6	9	3	7	
7	3	7	8	3	6	
8	4	8	10	3	3	Optimal
9	4	9	10	4	4	
	From 1	To 10	Minimum	Distance =	11	CPU = 0,00

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....



Distancia mínima
$= 4 + 3 + 1 + 3 = 11$

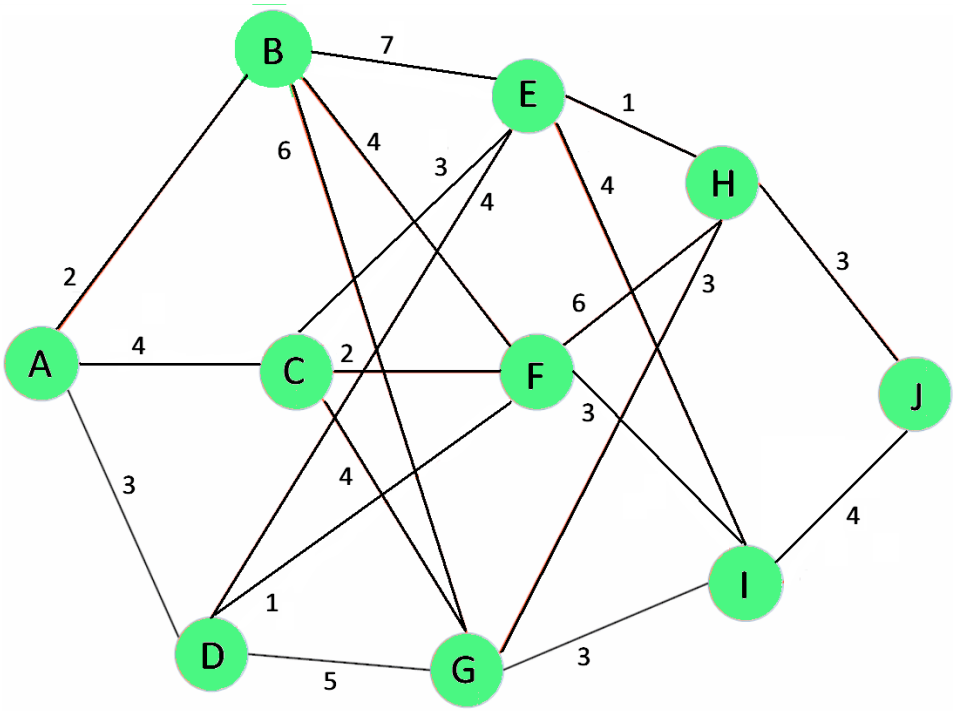
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

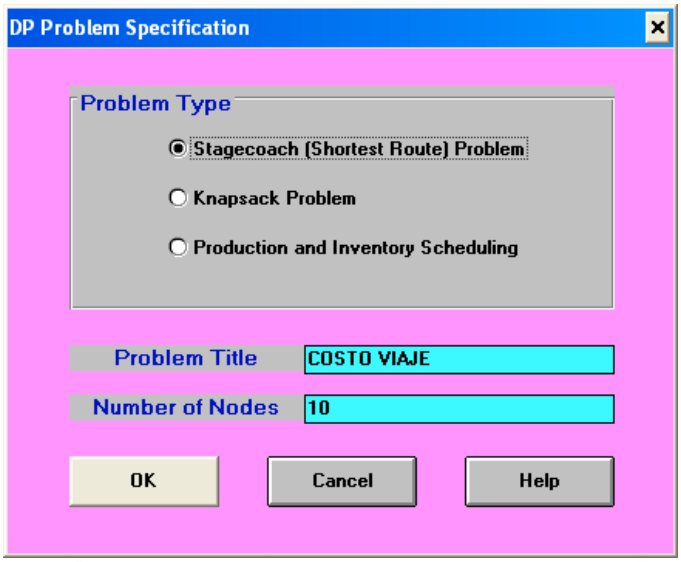
Apellidos ..... Nombre .....

Ejercicio del día .....

PROBLEMA DE LA DILIGENCIA: Se ofrecen pólizas de seguro de vida a los pasajeros. El costo de la póliza de cualquier jornada está basado en una evaluación de la seguridad del recorrido, la ruta más segura debe ser aquella cuya póliza de seguro tenga el menor costo total.
En cada arco, del estado i al estado j, se denota el costo de la póliza de viaje. ¿Cuál es la ruta que minimiza el costo total?



WinQSB / Dynamic Programming /Problem Specification

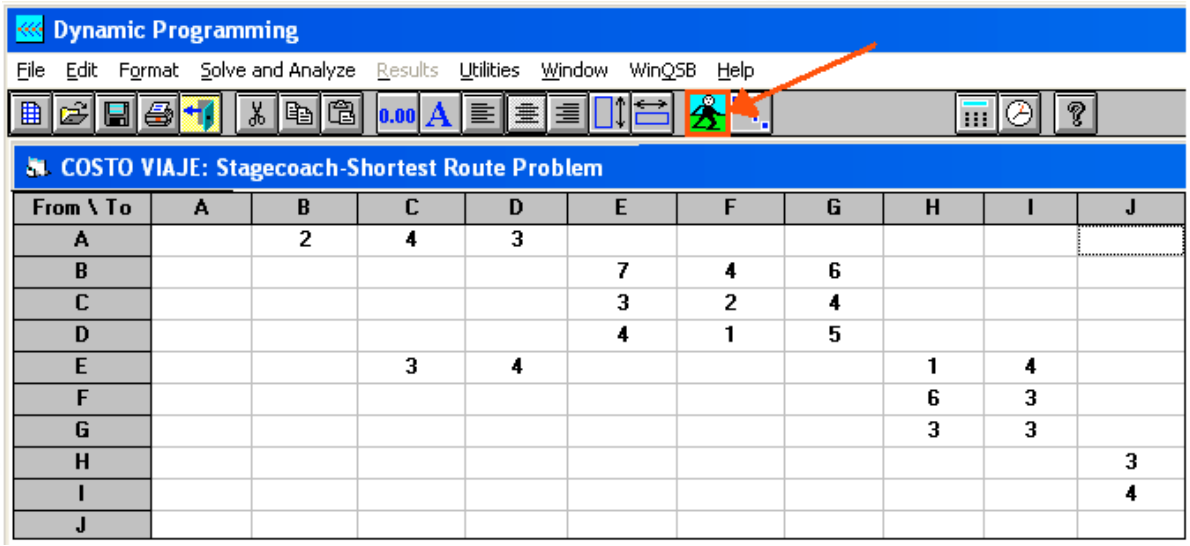


UAM
Universidad Autónoma de Madrid

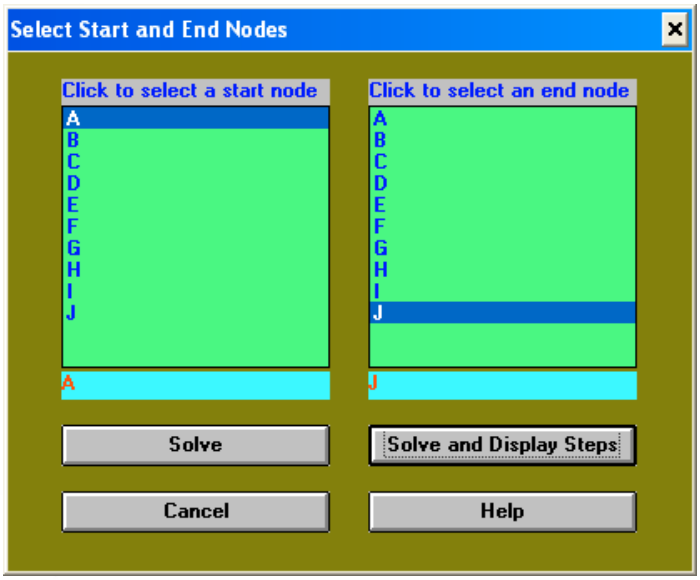
Asignatura ..... Grupo .....

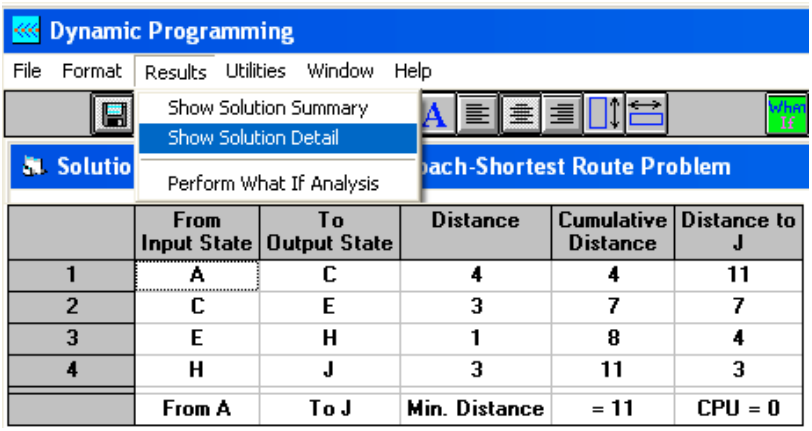
Apellidos ..... Nombre .....

Ejercicio del día .....



From \ To	A	B	C	D	E	F	G	H	I	J
A		2	4	3						
B					7	4	6			
C					3	2	4			
D					4	1	5			
E			3	4				1	4	
F								6	3	
G								3	3	
H										3
I										4
J										





	From Input State	To Output State	Distance	Cumulative Distance	Distance to J
1	A	C	4	4	11
2	C	E	3	7	7
3	E	H	1	8	4
4	H	J	3	11	3
	From A	To J	Min. Distance	= 11	CPU = 0

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

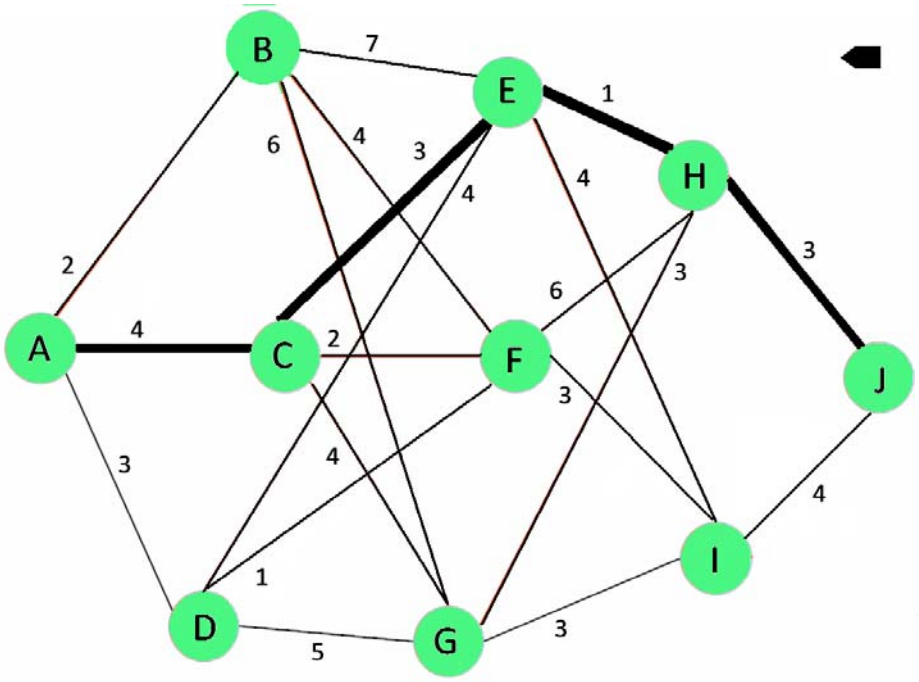
Ejercicio del día .....

Dynamic Programming

File Format Results Utilities Window Help

Solution Steps for COSTO VIAJE: Stagecoach-Shortest Route Problem

	Stage	From Input State	To Output State	Distance	Distance to J	Status
1	1	A	C	4	11	Optimal
2	2	B	E	7	11	
3	2	C	E	3	7	Optimal
4	2	D	E	4	8	
5	3	E	H	1	4	Optimal
6	3	F	I	3	7	
7	3	G	H	3	6	
8	4	H	J	3	3	Optimal
9	4	I	J	4	4	
	From A	To J	Minimum	Distance =	11	CPU = 0



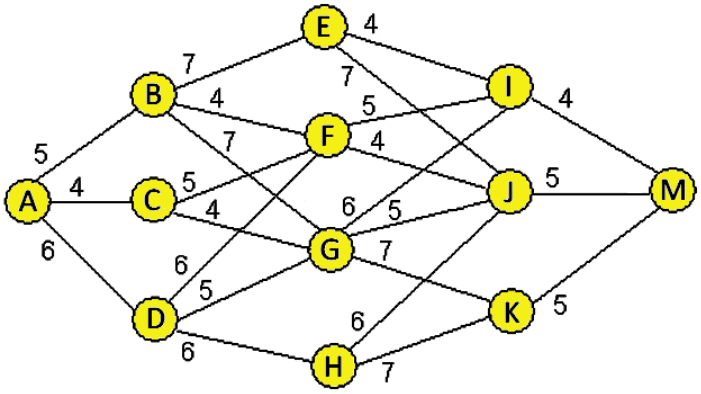
UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

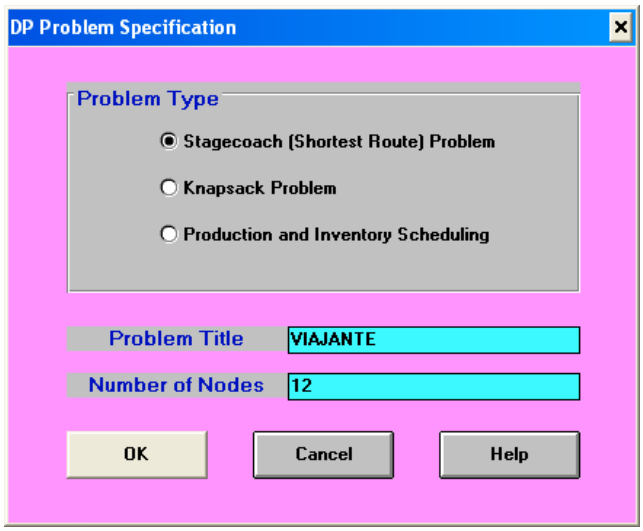
Apellidos ..... Nombre .....

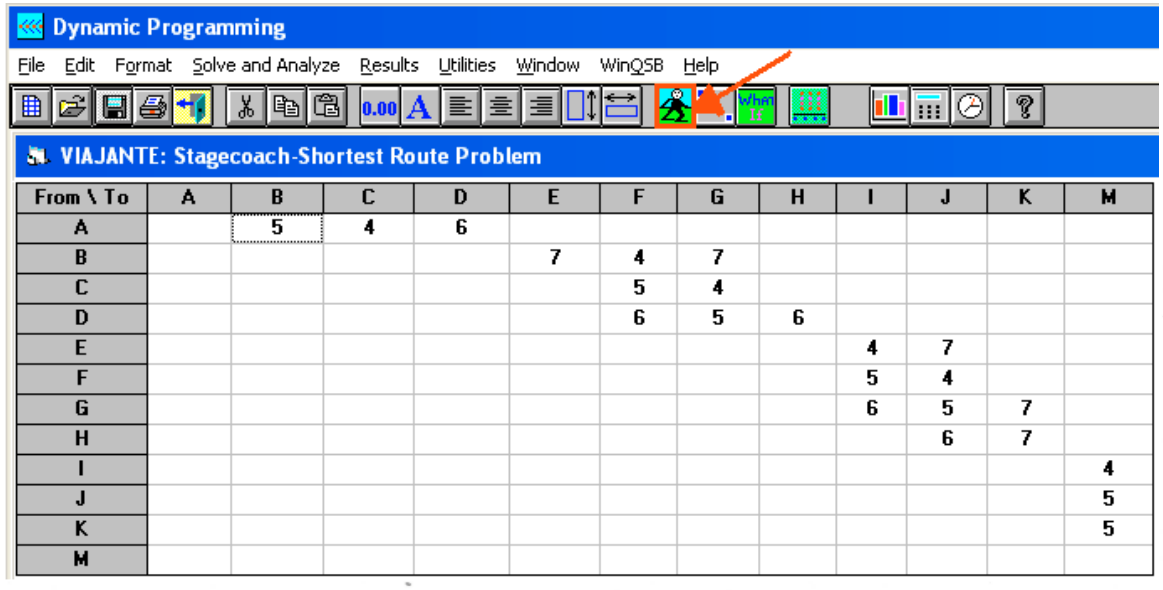
Ejercicio del día .....

PROBLEMA DE LA DILIGENCIA:
Encontrar el mejor camino para un viajante que tiene que desplazarse de A hasta M, si cada nodo representa una ciudad y las ramas la distancia en Km.



WinQSB / Dynamic Programming /Problem Specification





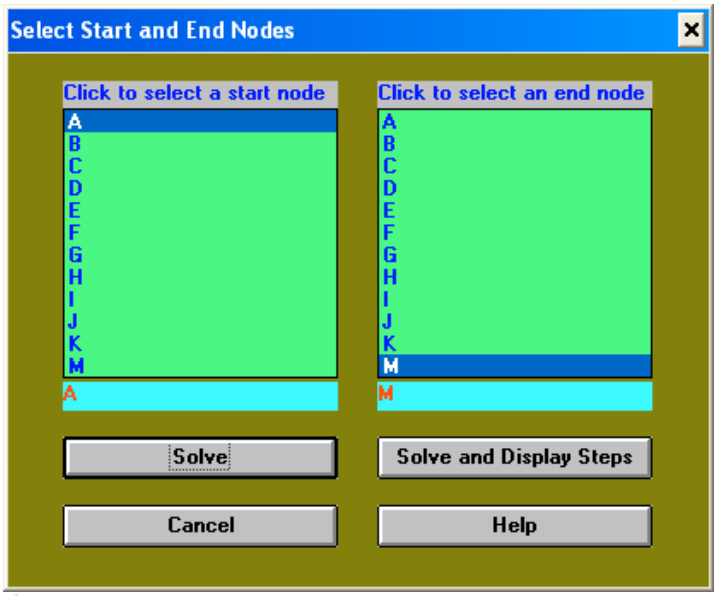
From \ To	A	B	C	D	E	F	G	H	I	J	K	M
A		5	4	6								
B					7	4	7					
C						5	4					
D						6	5	6				
E									4	7		
F									5	4		
G									6	5	7	
H										6	7	
I												4
J												5
K												5
M												

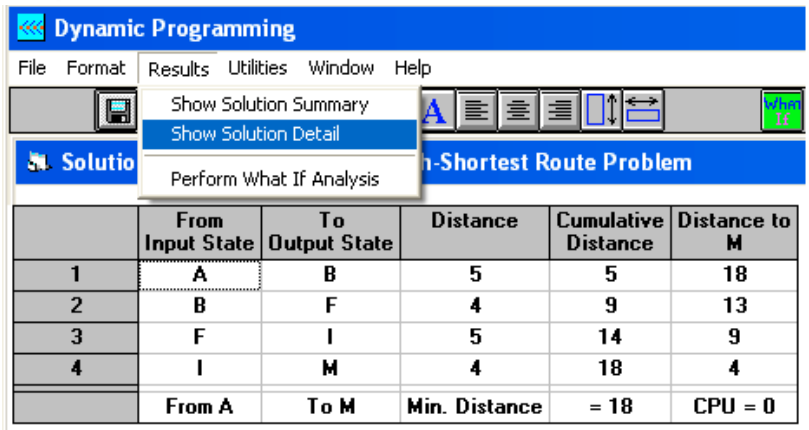
UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

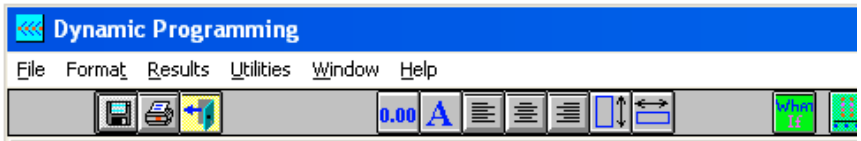
Apellidos ..... Nombre .....

Ejercicio del día .....





	From Input State	To Output State	Distance	Cumulative Distance	Distance to M
1	A	B	5	5	18
2	B	F	4	9	13
3	F	I	5	14	9
4	I	M	4	18	4
	From A	To M	Min. Distance	= 18	CPU = 0



	Stage	From Input State	To Output State	Distance	Distance to M	Status
1	1	A	B	5	18	Optimal
2	2	B	F	4	13	Optimal
3	2	C	F	5	14	
4	2	D	F	6	15	
5	3	E	I	4	8	
6	3	F	I	5	9	Optimal
7	3	G	I	6	10	
8	3	H	J	6	11	
9	4	I	M	4	4	Optimal
10	4	J	M	5	5	
11	4	K	M	5	5	
	From A	To M	Minimum	Distance =	18	CPU = 0

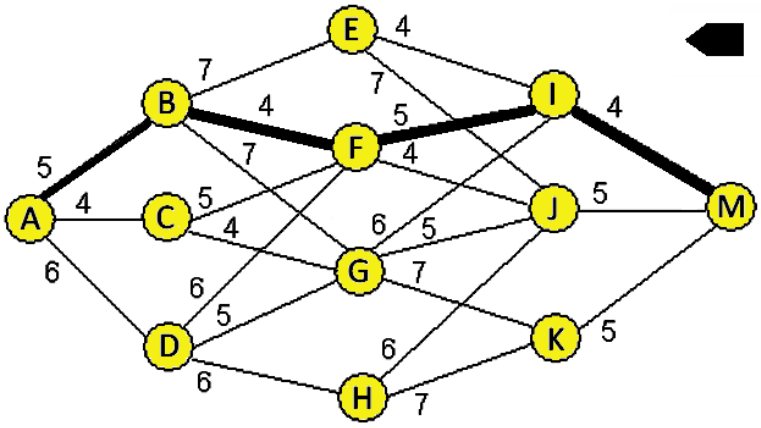
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

Distancia mínima $= 5 + 4 + 5 + 4 = 18$ km



UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

PROBLEMA DE LA MOCHILA O CANASTA DE EQUIPAJE (Knapsack Problem)

Existen n tipos distintos de artículos que pueden cargarse en una mochila. Cada artículo tiene asociado un peso y un valor. El problema consiste en determinar cuántas unidades de cada artículo se deben colocar en la mochila para maximizar el valor total.

Este enfoque resulta útil para la planificación del transporte de artículos en algún medio (carga de un buque, avión, camión). Este modelo también se utiliza en planificación de producción (distribuir la producción a través de varias máquinas).

El problema se desarrolla considerando el peso o el volumen. También se puede resolver mediante programación lineal entera binaria.

FUNCIÓN RECURSIVA DE LA MOCHILA:

Sea x_j el elemento j a transportar.

$x(j) \equiv$ número de unidades cargadas del artículo j

$w(j) \equiv$ espacio o peso que demanda cada unidad del artículo j

$R(j, x(j)) \equiv$ función de retorno del artículo j si se llevan $x(j)$ unidades en la mochila del artículo j

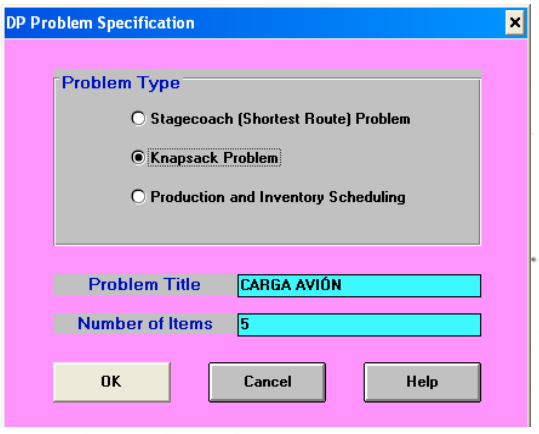
$g(j, w) \equiv$ retorno del total acumulativo dado el espacio w disponible para el artículo j

Función recursiva dinámica: $g(j, w) = \text{máximo}\left[R(j, x(j)) + g(j-1, w - w(j).x(j)\right] \quad \forall x(j)$

PROBLEMA DE LA MOCHILA: La carga de un avión se distribuye con el propósito de maximizar el ingreso total. Se consideran 5 elementos y solo se necesita uno de cada uno. La compañía gana 4.000 euros por elemento más una bonificación por elemento. El avión puede transportar 1400 kg, con un volumen máximo de 100 metros cúbicos.
Atendiendo a las especificaciones de la tabla, ¿cuál es el máximo ingreso que puede obtenerse? Y ¿cuántos elementos deben transportarse?

Elemento	Peso (kg)	Volumen (m^3)	Bonificación
1	600	30	900
2	650	40	800
3	450	35	1100
4	520	25	1000
5	300	30	700

WinQSB / Dynamic Programming / Knapsack Problem

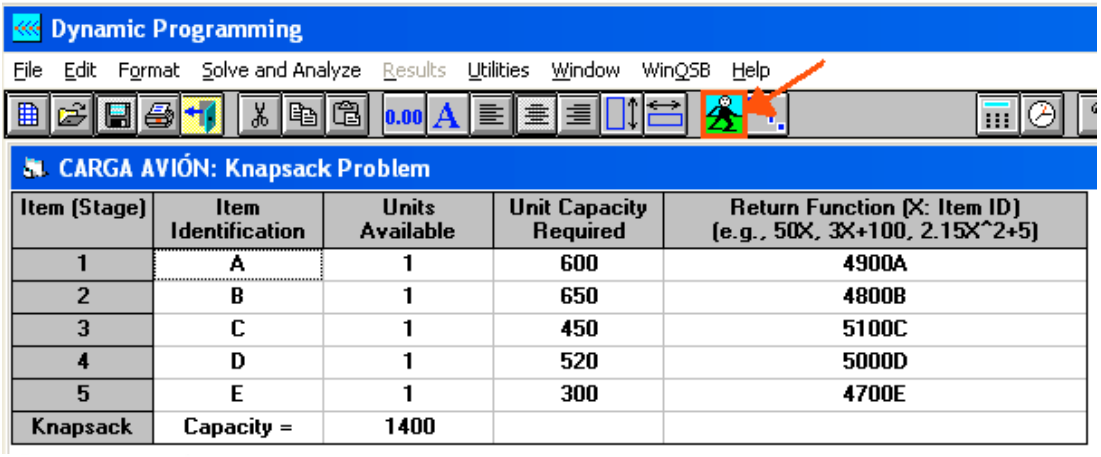


UAM
Universidad Autónoma de Madrid

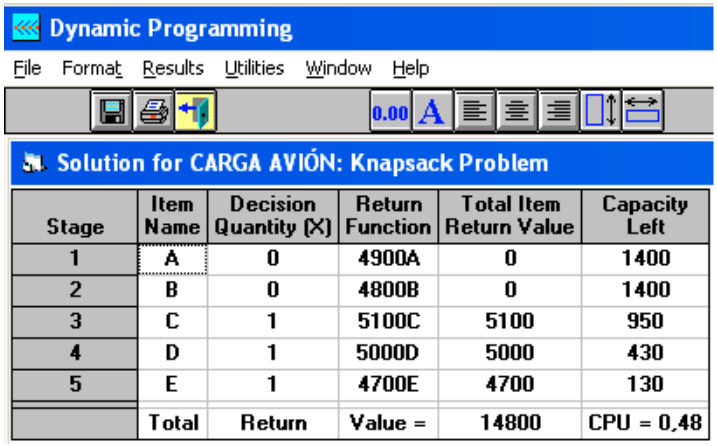
Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....



Item (Stage)	Item Identification	Units Available	Unit Capacity Required	Return Function (X: Item ID) (e.g., 50X, 3X+100, 2.15X^2+5)
1	A	1	600	4900A
2	B	1	650	4800B
3	C	1	450	5100C
4	D	1	520	5000D
5	E	1	300	4700E
Knapsack	Capacity =	1400		

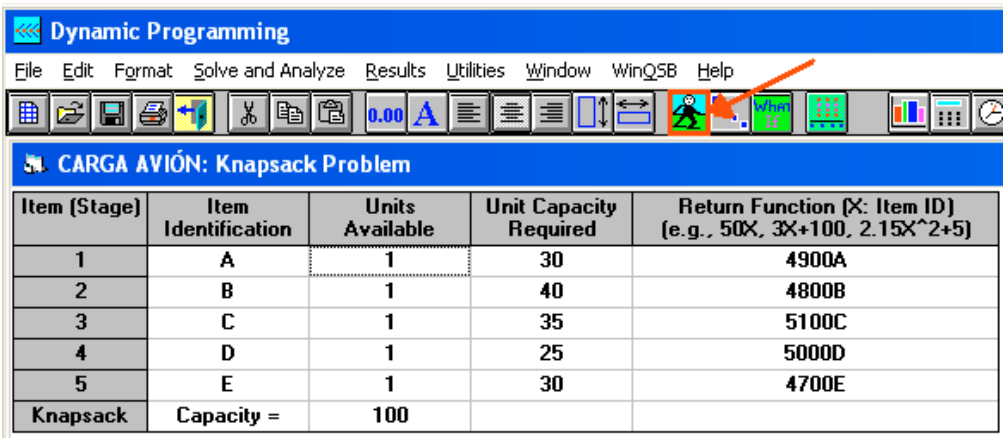


Stage	Item Name	Decision Quantity (X)	Return Function	Total Item Return Value	Capacity Left
1	A	0	4900A	0	1400
2	B	0	4800B	0	1400
3	C	1	5100C	5100	950
4	D	1	5000D	5000	430
5	E	1	4700E	4700	130
	Total	Return	Value =	14800	CPU = 0,48

La solución indica que se deben transportar los ítems 3, 4 y 5 con un retorno total de 14.800 euros.

CPU: Tiempo aproximado de procesamiento.

Considerando sólo el volumen, el nuevo modelo será:



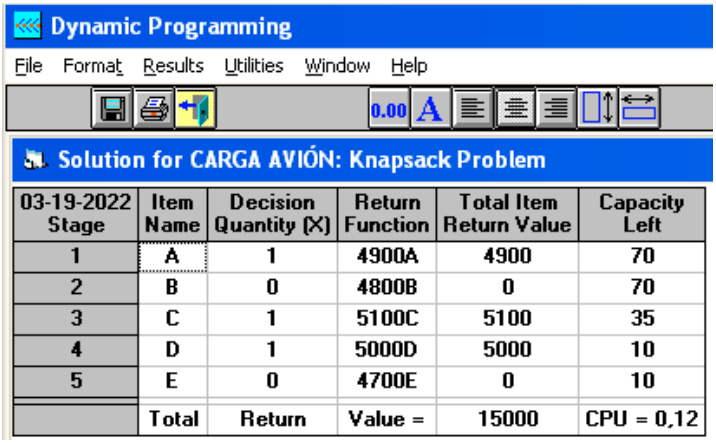
Item (Stage)	Item Identification	Units Available	Unit Capacity Required	Return Function (X: Item ID) (e.g., 50X, 3X+100, 2.15X^2+5)
1	A	1	30	4900A
2	B	1	40	4800B
3	C	1	35	5100C
4	D	1	25	5000D
5	E	1	30	4700E
Knapsack	Capacity =	100		

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....



03-19-2022 Stage	Item Name	Decision Quantity (X)	Return Function	Total Item Return Value	Capacity Left
1	A	1	4900A	4900	70
2	B	0	4800B	0	70
3	C	1	5100C	5100	35
4	D	1	5000D	5000	10
5	E	0	4700E	0	10
	Total	Return	Value =	15000	CPU = 0,12

La solución indica que se deben transportar los ítems 1, 3 y 4 con un retorno total de 15.000 euros.

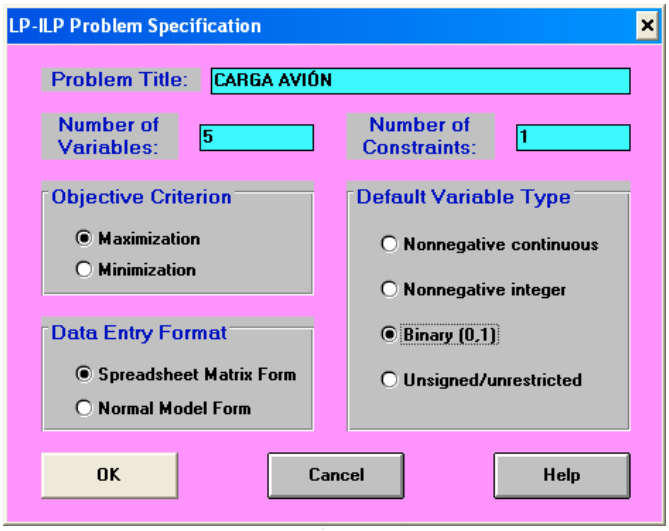
CPU: Tiempo aproximado de procesamiento.

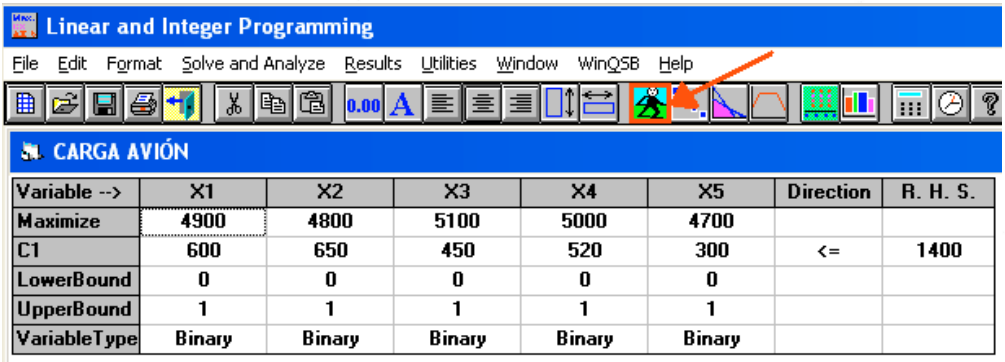
WinQSB / Linear and Integer Programming / Nonnegative Interger

Se puede resolver como un problema de programación lineal entera binaria

Maximizar $z = 4900x_1 + 4800x_2 + 5100x_3 + 5000x_4 + 4700x_3$

Restricciones: $600x_1 + 650x_2 + 450x_3 + 520x_4 + 300x_5 \leq 1400$ kg $\quad x_i \geq 0$ entero





Variable -->	X1	X2	X3	X4	X5	Direction	R. H. S.
Maximize	4900	4800	5100	5000	4700		
C1	600	650	450	520	300	<=	1400
LowerBound	0	0	0	0	0		
UpperBound	1	1	1	1	1		
VariableType	Binary	Binary	Binary	Binary	Binary		

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

Linear and Integer Programming

File Format Results Utilities Window Help

Combined Report for CARGA AVIÓN

	Decision Variable	Solution Value	Unit Cost or Profit c[j]	Total Contribution	Reduced Cost	Basis Status
1	X1	0	4900	0	4900	at bound
2	X2	0	4800	0	4800	at bound
3	X3	1	5100	5100	0	basic
4	X4	1	5000	5000	0	basic
5	X5	1	4700	4700	0	basic
	Objective	Function	(Max.) =	14800		

	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price
1	C1	1270	<=	1400	130	0

Análogamente para el volumen:

Maximizar $z = 4900x_1 + 4800x_2 + 5100x_3 + 5000x_4 + 4700x_3$

Restricciones: $30x_1 + 40x_2 + 35x_3 + 25x_4 + 30x_5 \leq 100\ m^3$ $\quad x_i \geq 0$ entero

Linear and Integer Programming

File Edit Format Solve and Analyze Results Utilities Window WinQSB Help

CARGA AVIÓN

Variable -->	X1	X2	X3	X4	X5	Direction	R. H. S.
Maximize	4900	4800	5100	5000	4700		
C1	30	40	35	25	30	<=	100
LowerBound	0	0	0	0	0		
UpperBound	1	1	1	1	1		
VariableType	Binary	Binary	Binary	Binary	Binary		

Linear and Integer Programming

File Format Results Utilities Window Help

Combined Report for CARGA AVIÓN

	Decision Variable	Solution Value	Unit Cost or Profit c[j]	Total Contribution	Reduced Cost	Basis Status
1	X1	1	4900	4900	0	basic
2	X2	0	4800	0	4800	at bound
3	X3	1	5100	5100	5100	at bound
4	X4	1	5000	5000	0	basic
5	X5	0	4700	0	4700	at bound
	Objective	Function	(Max.) =	15000		

	Constraint	Left Hand Side	Direction	Right Hand Side	Slack or Surplus	Shadow Price
1	C1	90	<=	100	10	0

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

PROGRAMACIÓN DE PRODUCCIÓN E INVENTARIO (Production and Inventory Scheduling)

Los sistemas de producción con inventarios son muy variados, cada uno presenta un nivel de complejidad diferente y es innegable la importancia que tienen para la planificación y el desarrollo de la economía.

Determinar el nivel de inventario para diferentes períodos de forma que se garantice cierto flujo de producción, calcular el tiempo de reaprovisionamiento de materias primas y materiales, estudiar el comportamiento de la demanda, son algunos de los aspectos más importantes analizados por estos sistemas.

La programación dinámica da solución a uno de los casos más sencillos: El problema de producción con inventarios de un solo producto para un determinado período, con demanda conocida para cada subperíodo.

DESCRIPCIÓN GENERAL DEL PROBLEMA TIPO: Se produce un producto A en un determinado período, se conoce la capacidad de producción y la demanda del producto para cada subperíodo y se puede almacenar para ser utilizado posteriormente. Se saben los costos de producción y de almacenamiento. El objetivo es determinar la cantidad a producir en cada subperíodo para minimizar los costos totales.

Elementos que componen un problema de producción con inventario:

Etapas (n): cada uno de los subperíodos de tiempo en que se puede dividir el período de planificación.

Estados (S): cantidad de unidades del producto en inventario al inicio de cada etapa.

Variable de decisión (X_n): cantidad de unidades a producir en cada uno de los subperíodos de tiempo.

$C_{ap}(P)$: capacidad de producción $C_{ap}(A)$: capacidad de almacenamiento

C_p : costo de producción C_a : costo de almacenamiento

Función de recursividad:
$$\begin{cases} f_n^*(S) = \text{mínimo}\left[f_n(S, x_n)\right] \\ f_n^*(S, x_n) = \text{mínimo}\left[Cp(x_n) + Cl(S) + f_{n-1}^*(S+x_n-d_n)\right] \end{cases}$$

PRODUCCIÓN E INVENTARIO: La empresa Ávila para los primeros cinco meses quiere producir tanques de agua para servicio público. El producto puede producirse en un mes y almacenarse para ser vendido posteriormente. En la tabla adjunta se muestra la demanda y la capacidad de producción para cada mes y el costo unitario de producción y almacenamiento. Al comienzo del período no hay tanques en el almacén, ni se desea que haya al final.

La empresa desea conocer cuántos tanques debe producir cada mes para minimizar el costo total al final del período, sabiendo que no permite tener más de 3 tanques en inventario.

Mes	Demanda (tanques)	Capacidad de Producción (tanques)	Costo unitario de Producción (euros/tanque)	Costo unitario de Almacenamiento (euros/tanque)
Enero	2	4	30	2
Febrero	4	5	60	1
Marzo	5	5	30	5
Abril	3	4	50	2
Mayo	1	5	20	5

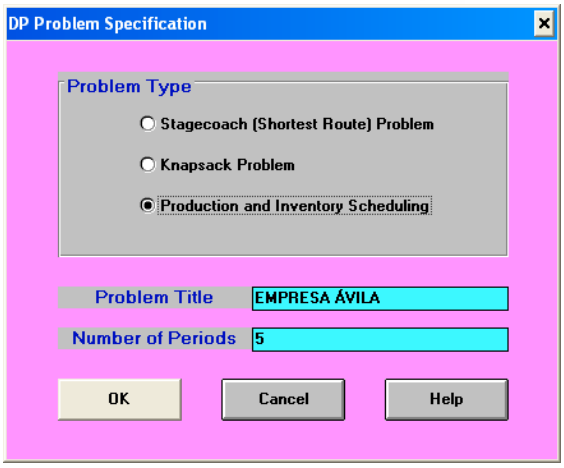
UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

WinQSB / Dynamic Programming / Production and Inventory Scheduling



Dynamic Programming

File Edit Format Solve and Analyze Results Utilities Window WinQSB Help

EMPRESA ÁVILA: Production and Inventory Scheduling Problem

Period (Stage)	Period Identification	Demand	Production Capacity	Storage Capacity	Production Setup Cost	Variable Cost Function (P,H,B: Variables)
1	Enero	2	4	3	0	30P+2H
2	Febrero	4	5	3	0	60P+2H
3	Marzo	5	5	3	0	30P+2H
4	Abril	3	4	3	0	50P+2H
5	Mayo	1	5	3	0	20P+2H
Initial	Inventory =	0				

El problema consiste en determinar un programa de producción para un periodo de tiempo con el fin de minimizar los costos totales relacionados. Hay demandas conocidas para cada periodo, límites de capacidad tanto para la producción como para los inventarios (almacenamiento).

Cuando hay más producción que demanda, se acumula inventario (almacenamiento), y cuando la producción es menor que la demanda, se generarán retrasos en el cumplimiento de pedidos (backorder, pedido pendiente).

Para cada periodo, una producción no-cero incurre en un costo de preparación. En programación dinámica, el costo variable se expresa como una función de la producción (P), el inventario (H), y backorder (B).

$P(n) \equiv$ Número de unidades producidas en el periodo n

$D(n) \equiv$ Demanda en el periodo n

$H(n) \equiv$ Inventario disponible al final del periodo n

$B(n) \equiv$ Pedido pendiente al final del periodo n

$I(n) \equiv$ Posición del pedido pendiente al final del periodo n. Es decir, $I(n) = H(n)$ o $I(n) = B(n)$

$$I(n) = I(n-1) + P(n) - D(n)$$

$S(n) \equiv$ Costo de preparación en el periodo n

$V[P(n), I(n)] \equiv$ Costo variable $\rightarrow$ Función de P(n), H(n), y/o B(n)

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

$$C[n,P(n),I(n)] = \begin{cases} S(n) + V[P(n),I(n)] & P(n) > 0 \\ V[P(n),I(n)] & P(n) = 0 \end{cases}$$

Función recursiva: $f(n,i) = \text{máximo}\left[\{C(n,P(n),i+P(n)-D(n)\} + f\{(n-1),\ i+P(n)-D(n)\}\right] \quad \forall P(n)$

Period Identication: Identificación de los períodos de tiempo en que se divide el estudio (etapas).

Demand: Demanda para cada período.

Production Capacity: Capacidad de producción de cada período.

Storage Capacity: Capacidad de almacenamiento de cada período.

Production Setup Cost: Costo de lanzamiento o preparación para el período (es independiente de la cantidad de unidades a producir en el período).

Variable Cost Function: Función del costo variable.

Dynamic Programming

File Format Results Utilities Window Help

Solution for EMPRESA ÁVILA: Production and Inventory Scheduling Problem

Stage	Period Description	Net Demand	Starting Inventory	Production Quantity	Ending Inventory	Setup Cost	Variable Cost Function (P,H,B)	Variable Cost	Total Cost
1	Enero	2	0	4	2	0	30P+2H	124,00 €	124,00 €
2	Febrero	4	2	2	0	0	60P+2H	120,00 €	120,00 €
3	Marzo	5	0	5	0	0	30P+2H	150,00 €	150,00 €
4	Abril	3	0	3	0	0	50P+2H	150,00 €	150,00 €
5	Mayo	1	0	1	0	0	20P+2H	20,00 €	20,00 €
Total		15	2	15	2	0		564,00 €	564,00 €

UAM
Universidad Autónoma de Madrid

Asignatura ..... Grupo .....

Apellidos ..... Nombre .....

Ejercicio del día .....

Instrumentos Estadísticos Avanzados
Facultad Ciencias Económicas y Empresariales
Departamento de Economía Aplicada
Profesor: Santiago de la Fuente Fernández