UNIVERSIDAD
DE GRANADA

E.T.S. DE INGENIERÍAS INFORMÁTICA y DE TELECOMUNICACIÓN

Departamento de Ciencias de la Computación e Inteligencia Artificial

Algorítmica

Guión de Prácticas

Práctica 5: Algoritmos de Programación Dinámica

Curso 2020-2021

Grado en Informática

Objetivo

El objetivo de esta práctica es que el estudiante comprenda y sepa utilizar la técnica de Programación Dinámica para resolver problemas. Para ello cada equipo de estudiantes deberá diseñar e implementar un algoritmo de Programación Dinámica para resolver uno de los problemas (asignado al azar) que se describen a continuación en las secciones 1–5.

1. Asignación de recursos

Se tienen r unidades de un recurso (monetarias, personal,...) que deben asignarse a n proyectos. Si se asignan j, $0 \leq j \leq r$, unidades al proyecto i, se obtiene un beneficio $N(i, j) \geq 0$.

Diseñad e implementad un algoritmo de Programación Dinámica que asigne recursos a los n proyectos maximizando el beneficio total obtenido.

Aplicadlo para un problema con $n = 4$ proyectos y $r = 6$ recursos, y una matriz de beneficios $N(i, j)$

i/j	0	1	2	3	4	5	6
1	0	1	1	2	3	4	5
2	0	2	3	3	3	4	6
3	5	0	1	1	2	2	2
4	0	2	3	4	5	6	7

2. Envases de pintura

Una fábrica de pinturas necesita envasar, para satisfacer un pedido, la cantidad de V litros de cierta pintura. Para ello dispone de una determinada cantidad de envases de n tipos distintos, e_1, e_2,..., e_n. La capacidad de cada envase de tipo e_i es de l_i litros, y tiene un coste (el envase) de c_i euros. Además, de cada tipo de envase disponemos de u_i unidades.

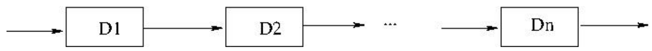
Se desea determinar la forma de satisfacer el pedido utilizando la combinación de envases que genere el mínimo coste. Nótese que es posible que en la solución propuesta algunos envases no queden completamente llenos. Diseñad e implementad un algoritmo de Programación Dinámica para calcular la solución óptima.

Aplicadlo al siguiente caso: $n = 3$, $V = 10$, y

	capacidad l_i	número u_i	coste c_i
e_1	3	3	5
e_2	5	3	9
e_3	7	2	12

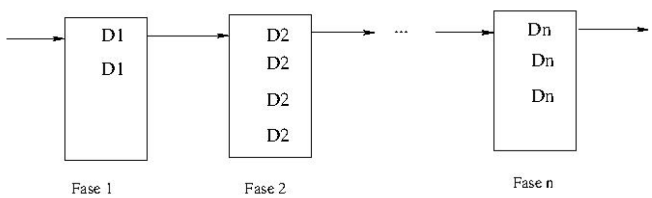
3. Diseño de Sistemas Fiables

Se desea diseñar un sistema compuesto de n dispositivos $D_1, D_2, \ldots, D_n$ conectados en serie:



Sea r_i la fiabilidad del dispositivo D_i, es decir la probabilidad de que ese dispositivo funcione correctamente en un instante de tiempo dado. La fiabilidad del sistema completo es el producto de las fiabilidades de los dispositivos conectados en serie, $\prod_{i=1}^{n} r_i$. Por ejemplo si $n = 10$ y $r_i = 0{,}99$, $i = 1, \ldots, 10$, la fiabilidad del sistema es $0{,}99^{10} = 0{,}904$.

Una forma de aumentar la fiabilidad es colocar varios dispositivos iguales pero en paralelo en cada fase, como se muestra en la figura siguiente:



Si en la fase i se ponen m_i copias del dispositivo D_i, la probabilidad de que toda la fase falle es $(1 - r_i)^{m_i}$ (probabilidad de que fallen todos los dispositivos), y por tanto su fiabilidad (probabilidad de que no falle, es decir que al menos algún dispositivo funcione) es $1-(1-r_i)^{m_i}$. Por ejemplo si $r_i = 0{,}99$ y $m_i = 2$, la fiabilidad de la fase i pasará de 0.99 a 0.9999.

Denotemos mediante $\phi(m_i)$ la fiabilidad de la fase i si conectamos m_i dispositivos en paralelo. La fiabilidad del sistema completo será $\prod_{i=1}^{n} \phi(m_i)$. Supongamos que el coste de cada unidad del dispositivo D_i es de c_i unidades monetarias, y que disponemos de un máximo de C unidades monetarias para construir el sistema completo.

Diseñad e implementad un algoritmo de Programación Dinámica que determine cuántas unidades m_i de cada componente D_i se deben comprar para maximizar la fiabilidad del sistema completo, respetando la limitación presupuestaria, es decir que $\sum_{i=1}^{n} c_i * m_i \leq C$.

Aplicadlo al caso de un sistema con $n = 3$ fases, con dispositivos de fiabilidad y costo:

i	1	2	3
r_i	0.9	0.95	0.9
c_i	1	2	3

Y se dispone de un total de $C = 10$ unidades monetarias.

4. Mudanza con dos camiones

Disponemos de n objetos de pesos $p[i]$ y beneficios $b[i]$. Disponemos de 2 vehículos de transporte iguales capaces de soportar (cada uno) un peso máximo W. Queremos seleccionar qué objetos debemos transportar en los dos vehículos que maximicen el beneficio obtenido, respetando las limitaciones de carga.

Diseñad e implementad un algoritmo de Programación Dinámica que resuelva este problema.

Aplicadlo a un problema con $n = 5$ objetos, peso máximo $W = 5$ y con pesos y beneficios:

p_i	b_i
3	2
4	6
1	3
2	4
4	5

5. Trabajos sucesivos

Supongamos una serie de n trabajos denominados $t_1, t_2, \ldots, t_n$ y una tabla o matriz B de tamaño $n \times n$, en la que cada posición $B(i, j)$ almacena el beneficio de ejecutar el trabajo t_i y a continuación el trabajo t_j. Se quiere encontrar la sucesión de m trabajos (con $m > 1$) que dé un beneficio óptimo. No hay límite en el número de veces que se puede ejecutar un trabajo concreto.

Diseñad e implementad un algoritmo de Programación Dinámica que resuelva este problema.

Aplicado para un problema con $n = 3$ trabajos y $m = 5$, y una matriz B:

$B(i,j)$	t_1	t_2	t_3
t_1	2	2	5
t_2	4	1	3
t_3	3	2	2

Nota importante

Se debe entregar una memoria detallada. En todos los casos se deben especificar claramente todos los pasos para la aplicación de programación dinámica al problema en cuestión: planteamiento de la solución como una secuencia de decisiones y verificación del principio de optimalidad, definición recursiva de la solución óptima, cálculo del valor de la solución óptima usando un enfoque ascendente, y determinación de la solución óptima empleando la información almacenada en la(s) tabla(s). Hay que especificar también el algoritmo que resuelve el problema (y su eficiencia) así como algún ejemplo detallado de su aplicación y, en su caso, los detalles del estudio empírico realizado. Se deben incluir en ficheros aparte los códigos de los programas desarrollados e instrucciones de compilación.