Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Industrial

IN34A: Clase Auxiliar

Programación Dinámica

Marcel Goic F.[1]

[1] Esta es una versión bastante preliminar por lo que puede contar con numerosas faltas de ortografia y errores no forzados. Si encuentran alguno favor de denunciarlo a mgoic@ing.uchile.cl

1. Introducción

Muchos problemas de programación matemática determinan soluciones que repercuten en la formulación de los problemas a resolver en el pro'ximo período o etapa. Una alternativa es construir un único modelo completo que tenga un gran conjunto de variables indexadas por etapas e iternalizar las relaciones entre etapas como una restricción del problema. Sin embargo esto pude agrandar mucho el tamaño del problema. Surge así Programación Dinámica (PD) como una alternativa de descomposición en que resolvemos subproblemas mas pequeños y luego los ligamos [2]. Así, programación dinámia consiste en solucionar el presente suponiendo que en cada etapa futura siempre se tomaran las decisiones correctas.

2. Características de un Problema de Programación Dinámica

Para que un problema pueda ser resuelto con la técnica de programación dinámica, debe cumplir con ciertas características:

- Naturaleza secuencial de las decisiones: El problema puede ser dividido en etapas.
- Cada etapa tiene un numero de estados asociados a ella.
- La decisión óptima de cada etapa depende solo del estado actual y no de las decisiones anteriores.
- La decisión tomada en una etapa determina cual será el estado de la etapa siguiente.

En síntesis, la política óptima esde un estado s de la etapa k a la etapa final esta constituida por una decisión que transforma s en un estado s' de la etapa $k+1$ y por la política óptima desde el estado s' hasta la etapa final.

3. Resolución de un Problema de Programación Dinámica

Para resolver un problema de programación dinámica debemos al menos:

- Identificación de etapas, estados y variable de decisión:

[2] Recordemos que mucho de los algoritmos de resolución de problemas lineales (Simplex en particular) son de orden exponencial por lo que resolver m problemas de tamaño n es mas rápido que resolver un problema de tamaño m · n

- Cada etapa debe tener asociado una o mas decisones (problema de optimización), cuya dependencia de las decisiones anteriores esta dada exclusivamente por las variables de estado.
- Cada estado debe contener toda la información relevante para la toma de decisión asociada al período.
- Las variables de decisión son aquellas sobre las cuales debemos definir su valor de modo de optimizar el beneficio acumulado y modificar el estado de la próxima etapa.

- DESCRIPCIÓN DE ECUACIONES DE RECURRENCIA: Nos deben indicar como se acumula la función de beneficios a optimizar (función objetivo) y como varían las funciones de estado de una etapa a otra.
- RESOLUCIÓN Debemos optimizar cada subproblema por etapas en función de los resultados de la resolución del subproblema siguiente. Notar que las para que las recurrencias estén bien definidas requerimos de condiciones de borde.

4. Problemas

4.1. Problema 1

La familia Sampsons va a salir de vacaciones desde su ciudad natal Sprangfield. La familia desea visitar n ciudades y dispone de un total de M días para hacerlo, con $M \geq n$. La familia desea saber cuantos días permanecer en cada ciudad de modo de maximizar la satisfacción total de sus vacaciones sabiendo que para cada ciudad i existe una función de satisfacción g_i que es función del número de días de permanencia.

1. Plantee un modelo de programación dinámica para resolver la planificación de las vacaciones de los Sampsons.
2. Suponga que $n = 3$ y $M = 5$ y que las funciones de beneficio $g_k(x_k)$ vienen dadas por:

	$g_1(x_1)$	$g_2(x_2)$	$g_3(x_3)$
$x_k = 0$	0	0	0
$x_k = 1$	1	1	1
$x_k = 2$	2	4	3
$x_k = 3$	3	6	3
$x_k = 4$	4	8	2
$x_k = 5$	5	8	1

3. ¿Cual es la política óptima de permanencia de los Sampsons en cada ciudad si la función de beneficios viene dada por $g_i(x_i) = \frac{1}{x_i}$ para todo i donde x_i es el numero

de días que permanecerán los Sampsons en la ciudad i y existen 3 ciudades para ser distribuidas en 7 días?

Hint: Suponga que no se pierde un tiempo considerable en el traslado de una ciudad a otra.

Solución

1. Consideremos que la familia ya definió cual será el orden en que visitará las ciudades (si efectivamente decide visitarlas). En dicho caso, una etapa estará en relación unívoca con una ciudad (Cada ciudad es una etapa y se pasará a la siguiente etapa cuando se pase a la siguiente ciudad). Además, el estado vendrá dado por el número de días que le restan a la familia para completar el total de días disponibles [3]. Así, podemos definir: x_i = Número de días en la ciudad i (variable de decisión de la etapa i). y_i = Número de días sobrantes despues de visitar la ciudad $i-1$ ó justo antes de visitar la ciudad i (variable de estado).

   Analicemos las ecuaciones recursivas:

   - Condición de borde (ultima etapa):
     En este caso habremos visitado las ciudades $1, 2, \ldots, n-1$ y tendremos y_n días disponibles para usar (dependiendo de cuantos días hayamos decidido quedarnos en las ciudades anteriores, y_n puede tener varios valores posibles. Luego, el problema a resolver viene dado por:

     $$\begin{aligned} f_n(y_n) &= \text{máx } g_n(x_n) \\ & \quad\ s.a \quad 0 \leq x_n \leq y_n \\ & \quad\ x_n \text{ entero} \\ &= \text{valor de la política óptima de estadía en la ciudad n si la familia} \\ & \quad\ \text{ya ha visitado n-1 ciudades y aun dispone de } y_n \text{ días disponibles} \end{aligned}$$

   - Recursión genérica k:
     En este caso habremos visitados las ciudades $1, 2, \ldots, k-1$ y nos quedan por visitar las ciudades $k, k+1, \ldots, n$ siendo y_k el numero de días que aun nos quedan disponibles. Luego nuestro problema a resolver será encontrar el numero de dias a permanecer en la ciudad k de modo de maximizar el beneficio de actual mas el beneficio de visitar las próximas ciudades suponiendo que de ahora en adelante tomaremos las decisiones óptimas dado los dias que nos quedarán luego de tomar nuestra decisión hoy:

     $$\begin{aligned} f_k(y_k) &= \text{máx}\{g_k(x_k) + f_{k+1}(y_k - x_k)\} \\ & \quad\ s.a \quad 0 \leq x_k \leq y_k \\ & \quad\ x_n \text{ entero} \\ &= \text{satisfacción total óptima por visitar a las ciudades } k, k+1, \ldots, n. \end{aligned}$$

[3] Tambien puede considerarse el numero de días que ya han gastado

Finalmente el óptimo de la satisfacción de las vacaciones de la familia Sampson viene dado por

$$f = f_1(M)$$

2. Debemos resolver los problemas asociados a cada etapa partiendo desde la última [4], para cada uno de los posibles estados en que puede llegar el problema a la etapa en cuestión.

   Para resolver cada uno de estos problemas haremos una enumeración explícita de los casos posibles para cada etapa y seleccionaremos la mejor [5].

   - Ciudad 3:

y_3	$g_3(0)$	$g_3(1)$	$g_3(2)$	$g_3(3)$	$g_3(4)$	$g_3(5)$	$f_3(y_3)$	x_3^*
0	0	i	i	i	i	i	0	0
1	0	1	i	i	i	i	1	1
2	0	1	3	i	i	i	3	2
3	0	1	3	3	i	i	3	2,3
4	0	1	3	3	2	i	3	2,3
5	0	1	3	3	2	1	3	2,3

   i: infactible, es decir, la familia no puede quedarse esa cantidad de días porque ya no le quedan tantos.

   - Ciudad 2:

	$\overbrace{\qquad\qquad}^{g_2(x_2)+f_3(y_2-x_2)}$							
y_2	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$x_2 = 5$	$f_2(y_2)$	x_2^*
0	0+0	i	i	i	i	i	0	0
1	0+1	1+0	i	i	i	i	1	0,1
2	0+3	1+1	4+0	i	i	i	4	2
3	0+3	1+3	4+1	6+0	i	i	6	4
4	0+3	1+3	4+3	6+1	8+0	i	8	4
5	0+3	1+3	4+3	6+3	8+1	8+0	9	4,3

   i: infactible, es decir, la familia no puede quedarse esa cantidad de días porque ya no le quedan tantos.

   - Ciudad 1:

   En este caso, como sabemos que inicialmente (antes de visitar la primera ciudad), la familia dispone de 5 días para sus vacaciones, solo analizamos el caso $y_1 = 5$

$$\overbrace{\qquad\qquad\qquad\qquad}^{g_1(x_1)+f_2(y_1-x_1)}$$

[4] Por que este es un problema que podremos resolver directamente sin necesitar los resultados de las próximas etapas

[5] En un problema general no se resolverá de forma tan ineficiente, sino que se recurrirá a otras técnicas: Simplex, Branch & Bound, métodos de descenso, etc.

y_1	$x_1=0$	$x_1=1$	$x_1=2$	$x_1=3$	$x_1=4$	$x_1=5$	$f_1(y_1)$	x_1^*
5	0+9	1+8	2+6	3+4	4+1	5+0	9	0,1

Finalmente existen 3 soluciones óptimas, todas con beneficio óptimo =9, cuyas estadías en cada ciudad viene dadas por:

Solución	Ciudad 1	Ciudad 2	Ciudad 3
Itinerario 1	0	3	2
Itinerario 2	0	4	1
Itinerario 3	1	4	0

3. Propuesto.

4.2. Problema 2

Considere el clásico problema de la mochila [6].

$$\text{máx } c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

$$\begin{array}{llll} s.a & a_1x_1 + a_2x_2 + \ldots + a_nx_n & \leq & b \\ & x_j \in \{0,1\} & & \forall\, j = 1,2,\ldots,n \end{array}$$

Modele dicho problema con la técnica de programación dinámica, describiendo las etapas del problema, las variables de estado y las ecuaciones recursivas que definen la formulación.

Solución

- Decisiones: $x_1, x_2, \ldots, x_n$ donde cada x_i viene dado por:

$$x_i = \begin{cases} 1 & \text{Si se echa el producto i en la mochila.} \\ 0 & \text{Si no se echa el producto i en la mochila.} \end{cases}$$

- Etapas: $1, 2, \ldots, n$. En la etapa i se decide el valor de x_i.
- Estado de la etapa i: Espacio disponible en la mochila. Si esta decidido el valor de $x_1, x_2, \ldots, x_{i-1}$ el lado derecho de la restricción de capacidad viene dado por $s_i = b - a_1x_1 - a_2x_2 - \ldots - a_{i-1}x_{i-1}$. Como en la etapa i no conocemos el valor de $x_1, x_2, \ldots, x_{i-1}$ vamos a tomar como estados $s_i \in \{0, 1, \ldots, b\}$ [7].

Con esto, debemos tratar de construir las ecuaciones de recurrencia. Para ello definimos:

[6] Para los no iniciados, el problema de la mochila (knapsack problem) consiste encontrar un subconjunto de productos que echar en una mochila de modo de maximizar el beneficio y respetar la capacidad de la mochila.

[7] Supondremos que los a_i son enteros.

$$\begin{array}{rrcl} & f_i(s_i) = \max\ c_i x_i + c_{i+1}x_{i+1} + \ldots + c_n x_n & & \\ s.a & a_i x_i + a_{i+1}x_{i+1} + \ldots + a_n x_n & \leq & s_i \\ & x_j \in \{0,1\} & & \forall\, j = i, i+1, \ldots, n \end{array}$$

Y tambien:

$$\begin{array}{rrcl} & f_n(s_n) = \max\ c_n x_n & & \\ s.a & a_n x_n & \leq & s_n \\ & x_n \in \{0,1\} & & \end{array}$$

Notamos que en cada etapa tendremos distintos problemas dependiendo del espacio disponible con que lleguemos a la etapa. Estos problemas son muy fáciles de resolver pues en cada problema pueden haber 2 casos posibles:

- Si la capacidad disponible al llegar a la etapa i es menor que el espacio que ocupa el producto i que estamos decidiendo si echar o no, no tendremos mas remedio que no echar el producto ($x_i = 0$)y el espacio disponible para la siguiente etapa será la misma con la que llegamos a esta etapa ($s_{i+1} = s_i$).
- Si la capacidad disponible al llegar a la etapa es mayor que el espacio que ocupa el producto que estamos decidiendo si echar o no, deberemos comparar las siguientes 2 alternativas:
  1. Echar el producto perdiendo capacidad para la próxima etapa.
  2. No echar el producto manteniendo la misma capacidad actual para la próxima etapa.

Finalmente, la ecuación de recurrencia viene dada por:

$$f_i(s_i) = \begin{cases} f_{i+1}(s_i) & \text{Si } s_i \geq a_i. \\ \max\{f_{i+1}(s_i), c_i + f_{i+1}(s_i - a_i)\} & \text{Si } s_i \leq a_i. \end{cases}$$

4.3. Problema 3

Un prestigioso taller mecánico, especialista en mantención y reparación de motores, tiene una máquina especializada para estos fines y desea saber cuando cambiar dicha máquina. Para ello cuenta con los siguientes datos:

- Una maquina nueva cuesta C [u.m].
- El taller puede mantener una máquina por 1, 2 o 3 años.
- Una máquina con i años de uso puede ser vendida en el mercado en v_i [u.m].

- El costo anual de mantención de una máquina con i años de uso es m_i [u.m].

El taller busca una política óptima de reemplazo que minimice los costos totales durante 5 anós restringidos a que siempre debe haber una máquina sabiendo que se compró una máquina el año 1 y que se venderá al final del año 5.

Solución

En el problema identificamos:

- Etapas: Corresponden a los años del horizonte $(t_0, t_1, t_2, ..., t_5)$.
- Decisiones: En cada etapa debemos decidir si conservar o cambiar la máquina.
- Estados: Edad de la máquina al final de la etapa $(l_{t_0}, l_{t_1}, l_{t_2}, ..., l_{t_5})$.

Notamos que:

1. En t_0 estamos obligados a comprar y en t_5 obligados a vender.
2. El espacio de edades posibles varía dependiendo de la etapa:

$$l_0 \in \{0\}, l_1 \in 1, l_2 \in \{1,2\}, l_3, l_4, l_5 \in \{1,2,3\}$$

Definimos:

$f_{t_i}(l_i) =$ Costo de la política óptima desde t_i hasta el final, dado que la edad de la máquina del analizar en t_i es l_i.

Con esto, lo que queremos calcular es $f_{t_0}(0)$.

- **Año 5**
  La máquina debe venderse al final del último año, al valor correspondiente a la edad de la máquina.
  $$f_{t_5}(l_5) = -v_{l_5}$$
- **Año 4**
  El problema a resolver dependerá de la edad que tenga la máquina. En efecto, si tuviera 3 anós no tendriamos mas que cambiar.
  $$f_{t_4}(3) = -v_3 + C + m_1 + f_{t_5}(1)$$
  $$f_{t_4}(2) = \begin{cases} -v_2 + C + m_1 + f_{t_5}(1) & \text{(Cambiar)} \\ m_3 + f_{t_5}(3) & \text{(Conservar)} \end{cases}$$
  $$f_{t_4}(1) = \begin{cases} -v_1 + C + m_1 + f_{t_5}(1) & \text{(Cambiar)} \\ m_2 + f_{t_5}(2) & \text{(Conservar)} \end{cases}$$

- **Año 3**
  El problema a resolver nuevamente dependerá de la edad que tenga la máquina.

$$f_{t_3}(3) = -v_3 + C + m_1 + f_{t_4}(1)$$

$$f_{t_3}(2) = \begin{cases} -v_2 + C + m_1 + f_{t_4}(1) & \text{(Cambiar)} \\ m_3 + f_{t_4}(3) & \text{(Conservar)} \end{cases}$$

$$f_{t_3}(1) = \begin{cases} -v_1 + C + m_1 + f_{t_4}(1) & \text{(Cambiar)} \\ m_2 + f_{t_4}(2) & \text{(Conservar)} \end{cases}$$

- **Año 2**
  El problema a resolver nuevamente dependerá de la edad que tenga la máquina. Sin embargo, en el año 2 (dado que se compra una nueva en t_1), solo podría tener una edad de 1 o 2.

$$f_{t_2}(2) = \begin{cases} -v_2 + C + m_1 + f_{t_3}(1) & \text{(Cambiar)} \\ m_3 + f_{t_3}(3) & \text{(Conservar)} \end{cases}$$

$$f_{t_2}(1) = \begin{cases} -v_1 + C + m_1 + f_{t_3}(1) & \text{(Cambiar)} \\ m_2 + f_{t_3}(2) & \text{(Conservar)} \end{cases}$$

- **Año 1**
  El problema a resolver nuevamente dependerá de la edad que tenga la máquina. Pero, en el año 1 (dado que se compra una nueva en t_1), solo podría tener una edad de 1.

$$f_{t_1}(1) = \begin{cases} -v_1 + C + m_1 + f_{t_2}(1) & \text{(Cambiar)} \\ m_2 + f_{t_2}(2) & \text{(Conservar)} \end{cases}$$

- **Año 0**
  Al inicio del primer período, necesariamente debemos comprar una máquina nueva.

$$f_{t_0}(0) = C + m_1 + f_{t_1}(1)$$

$\odot$