Antonio García Arana
Curso 872

PROGRAMACION DINAMICA

Programación dinámica es una técnica matemática de utilidad para tomar una secuencia de decisiones interrelacionadas. A diferencia con P.L. no existe una formulación o una técnica estandarizada. En este caso las ecuaciones particulares usadas deben ser desarrolladas para ajustarse a cada situación individual. Un cierto grado de ingenio de la estructura general de problemas de programación dinámica es requerida para saber cuando se puede resolver por programación dinámica y como debe ser hecho.

Programación dinámica empieza con una pequeña porción del problema y encuentra la solución óptima para este pequeño problema. Después empieza gradualmente a engrandar el problema encontrando la correspondiente solución óptima utilizando la anterior solución, así hasta que el problema es resuelto enteramente.

Características típicas para resolver un problema por programación dinámica:

1.- Decisiones son tomadas en secuencia.

2.- Los problemas pueden ser divididos en etapas.

3.- Cada etapa tiene asociado un número de estados.

4.- Un estado corriente (cualquiera) contiene toda la información

necesaria - como fue alcanzado ese estado es irrelevante para futuras decisiones.

5.- Una decisión transforma el estado corriente con uno asociado - con la siguiente etapa.

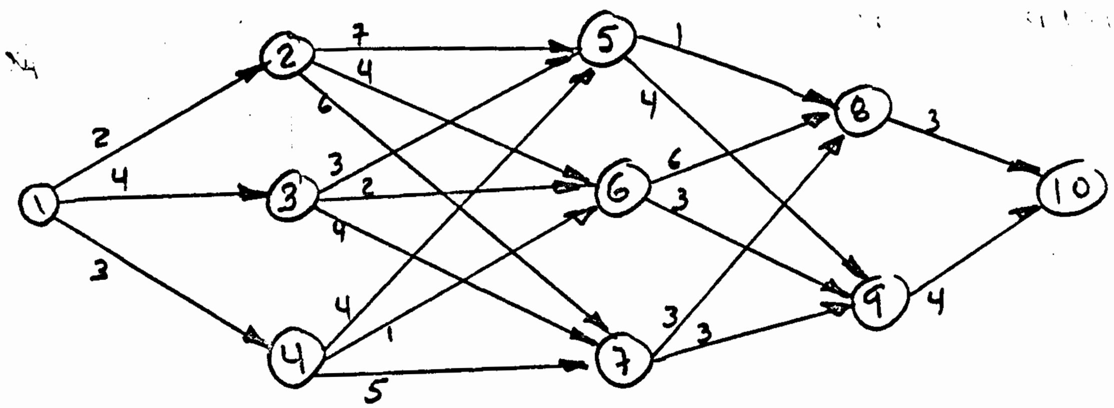
6.- Recompensa o castigos están asociados con transformaciones - de estado.

7.- Objetivo.- Maximizar recompensas o minimizar castigos.

Principio de Optimalidad de Bellman:

Una decisión óptima tiene la propiedad de que cualquiera que sea el estado inicial y decisión inicial las decisiones restantes deben - constituir una política óptima de acuerdo al estado que resultó de la - primera decisión.

Ejemplo: Minimizar costo de transporte para ir de (1) a (10)



Una forma de atacar el problema sería analizando las 18 rutas de (1) a (10), método que no resulta muy alentador. Programación dinámica prove una solución con mucho menos esfuerzo que el analizar todas las rutas.

Llamando: X_n a la variable decisión $(n = 1, 2, 3, 4)$, la cual indica el destino inmediato cuando hay n etapas más por recorrer. Así la ruta escogida sería $(1) \longrightarrow X_4 \longrightarrow X_3 \longrightarrow X_2 \longrightarrow X_1$ $(X_1 = (10))$.

n = No. de etapas restantes (de izquierda a derecha).

Sea $f_n(S, X_n)$ el costo total de la mejor política de todo el problema para los últimos n etapas, dado que el individuo está en el estado S y elije X_n como el destino inmediato.

Dado S y n, sea X_n^* el valor de X_n que minimiza $f_n(S, X_n)$, y sea $f_n^*(S)$ el valor mínimo correspondiente de $f_n(S, X_n)$. Así $f_n^*(S) = f_n(S, X_n^*)$. El objetivo es encontrar $f_4^*(1)$ y la política correspondiente a programación dinámica hace esto encontrando sucesivamente $f_1^*(S)$, $f_2^*(S)$, $f_3^*(S)$ y finalmente $f_4^*(1)$.

Así cuando el vendedor tiene solamente una etapa por recorrer su ruta está determinado enteramente por su destino final. La solución inmediata para problema de una etapa es como sigue:

S	$f_1^*(S)$	X_1^*
8	3	10
9	4	10

Cuando el vendedor tiene dos etapas por recorrer, la solución requiere algunos cálculos. Así, si está en el estado (5) S = 5 puede ir a 8 ó 9 a un costo de 1 ó 4. Si se va a (8) el costo total será 1 + 3 = (4). Si escoge 9 : 4 + 4 = (8) $\therefore$ el escogerá el estado 8 $X_2^* = 8$ ya que da un costo total $f_2^*(5) = 4$. En igual forma para S = 6 y S = 7 se tiene:

S \ X_2	$f_2(S, X_2) = C_{SX_2} + f_1^*(X_2)$: 8	9	$f_2^*(S)$	X_2^*
5	4	8	4	8
6	9	7	7	9
7	6	7	6	8

Para el problema de tres etapas:

$$f_3\ (S, X_3) = C_S\ X_3 + f_2^*\ (X_3)$$

X_3 / S	$f_3(S, X_3) = C_{SX_3} + f_2^*(X_3)$			$f_3^*(S)$	X_3^*
	5	6	7		
2	7+4=11	11	12	11	5 ó 6
3	7	9	10	7	5
4	8	8	11	8	5 ó 6

Para n = 4

X_4 / S	$f_4(S, X_4) = C_{SX_4} + f_3^*(X_4)$			$f_4^*(S)$	X_4^*
	2	3	4		
1	13	11	11	11	3 ó 4

Una relación recursiva es disponible, la cual identifica la política óptima para cada estado con n etapas restantes, dando la política óptima para cada estado con (n-1) etapas por recorrer.

$r_{X_n}(S)$ = Recompensa o castigo asociado con decisión X_n en el estado S.

$T_{X_n}(S)$ = El estado de la etapa n-1 (desde el final) en el cual decisión X_n transforma S.

Así, $f_n(S)$ = máximo o mínimo absoluto, recompensa o castigo, empezando en el estado S con n etapas por recorrer.

$$= \underset{X_n}{\text{máximo o mínimo}} \left\{ r_{X_n}(S) + f_{n-1}(\underbrace{T_{X_n}(S)'}_{\substack{\text{recompensa o castigo} \\ \text{de previas etapas.}}}) \right\}$$

En nuestro ejemplo la relación recursiva fue:

$$f_n(S) = \underset{X_n}{\text{mínimo}} \left\{ C_{SX_n} + \underbrace{f^*_{n-1}(X_n)}_{\substack{\text{castigo si escogemos } X_n \\ \text{en etapa } n}} \right\}$$

ANALISIS DE DIFERENTES PROBLEMAS

CASOS DETERMINISTAS

1.- Minimización del Costo ~~del Costo~~ en una Red.- (3)

$$f_n(s) = \min_{x_n} \left\{ C_{s x_n} + f_{n-1}(x_n) \right\}$$

Ver Ej. de pags. 2, 3, 4, 5.

2.- Asignación de Recursos.- (2), (3)

a) Problema de Producción

$$\min z = \sum_{i=1}^{n} C_i X_i$$

Sujeta a:

$$\sum_{i=1}^{n} X_i = S$$

$$f_n(s) = \min_{0 \le X_n \le s} \left\{ C_n X_n + f_{n-1}(s - X_n) \right\}$$

X_i = Cantidad producida en el mes i.

C_i = Costo de producción en el mes i.

S = Cantidad restante por ser producida

Ej. de un problema de producción

Min: $2X_1^2 + 3X_2^2 + 2X_3^2$

Sujeto: $X_1 + X_2 + X_3 = 9$

Producir 9 unidades a costo mínimo

$$f_n(s) = \min_{0 \le X_n \le s} \left\{ w_n X_n^2 + f_{n-1}(s - X_n) \right\}$$

S = Cantidad remanente por ser producida

X_i^* = Variable decisión = cantidad por ser producida en ~~el mes~~ la etapa i

Para n = 1 :

s	0	1	2	3	4	5	6	7	8	9
x_1^*	0	1	2	3	4	5	6	7	8	9
$f_1^*(s)$	0	2	8	18	32	50	72	98	128	162

Parando a n = 2

$$f_2(s,x_2) = \min_{x_2=0,\dots,9} \{3x_2^2 + f_1^*(s-x_2)\}$$

$s \backslash x_2$	$x_2 =$ 0	1	2	3	...	$f_2^*(s)$	x_2^*
0	$0+f_1(0)=0$					0	0
1	$0+f_1(1)=2$	$3+f_1(0)=3$				2	0
2	$0+f_1(2)=8$	$3+f_1(1)=5$	$12+f_1(0)=12$			5	1
3	$0+f_1(3)=18$	$3+f_1(2)=11$	$12+f_1(1)=14$	$27+f_1(0)=27$		11	1
4						20	2
5						30	2
6						44	2
7						59	3
8						77	3
9						98	4

Para n = 3

x_3:	0	1	2	3	4	...	x_3^*	f_3^*
$s=9$	$0+98=98$	$6+77=83$	$24+59=83$	98	126	...	1,2	83

$\therefore\ x_3^* = 1$ ó 2

Si $x_3^* = 1$; $x_2^* = 3$; $x_1^* = 5$ DOS SOLUC. OPTIMAS

Si $x_3 = 2$; $x_2^* = 3$; $x_1^* = 4$

OPTIMIZACION USANDO CALCULO :

En la ecuación recursiva $f_n(s) = \max_{x_n} \{ r_{x_n}(s) + f_{n-1}(T_{x_n}(s)) \}$

La maximización se puede realizar vía diferenciación si x_n es una variable de decisión contínua, y formas funcionales pueden ser encontradas para $r_{x_n}(s)$ y f_{n-1} ...

Así, para el Ej. de Producción:

Para n = 1 $\quad f_1(s) = w_1 s^2 \qquad x_1^* = s$

$$\therefore \; f_2(s) = \min_{0 \le x_2} \{ w_2 x_2^2 + w_1 (s - x_2)^2 \}$$

$$\frac{\partial}{\partial x_2}(f_2) = 2 w_2 x_2 - 2 w_1 (s - x_2)$$

$$= 0 \text{ cuando } x_2^* = \frac{w_1 s}{w_1 + w_2}$$

$$\frac{\partial^2}{\partial x_2^2}\{f_2\} > 0, \text{ dando un mínimo}$$

$f_2(s) = w_2 \frac{w_1^2 s^2}{(w_1+w_2)^2} + w_1\left(s - \frac{w_1 s}{w_1+w_2}\right)^2 =$ $\quad f_2(s) = \dfrac{w_1 w_2 s^2}{w_1 + w_2}$, reemplazando x_2^* en $\{f_2\}$

n = 3 da, similarmente $\quad x_3^* = w_1 w_2 s / (w_1 w_3 + w_2 w_3 + w_1 w_2)$

$$f_3 = w_1 w_2 w_3 s^2 / (w_1 w_3 + w_2 w_3 + w_1 w_2)$$

(Se puede demostrar, por inducción, que la forma general de solución es)

$$x_n^* = s \frac{1}{w_n} \Big/ \sum_{i=1}^{n} \frac{1}{w_i}$$

$$f_n = s^2 \Big/ \sum_{i=1}^{n} \frac{1}{w_i}$$

Así, para este caso de 3 etapas con $\quad S = 9, \; w_1 = 2, \; w_2 = 3, \; w_3 = 6$

$$f_3(9) = \underline{81}$$

$$x_3^*(9) = \underline{1.5}, \text{ dando } s - x_3 = 7.5$$

Así: $\quad x_2^*(7.5) = \underline{3}$, dando $s - x_2 = 4.5$

$$x_1^*(4.5) = \underline{4.5}$$

b) Optimización de cargamentos.-

Maximizar el valor de la carga = $\sum v_i X_i$

Sujeto a $\sum w_i X_i \leq S$

v_i = Valor del producto de tipo i

X_i = No. de productos de tipo i para ser cargado

m_i = No. de artículos de tipo i (disponibles)

S = Capacidad disponible

w_i = Peso de un artículo de tipo i

$$f_n(s) = \max \{ v_n X_n + f_{n-1}(s - w_n X_n) \}$$

$$X_n \leq \min \{ \frac{S}{w_n}, m_n \}$$

3.- Programación dinámica estocástica.-

Cuando existe la probabilidad de pasar de un estado i a un estado j ($p_{ij}(X_n)$), entonces se trata de maximizar el valor esperado de la recompensa $f_n(s)$.

Se estudiarán problemas de asignación de recursos utilizando probabilidades.

a) Confiabilidad de aparatos con multicomponentes(2)

Se trata de minimizar la falla de un equipo con N fases. En cada fase "n" se pueden instalar 1 + X_n unidades en paralelo. Cada unidad cuesta C_i .

$$\varphi_n(X_n) = p_r \text{ (no falla en fase n)}$$

Maximizar la probabilidad de que el sistema opere = $\sum_{i=1}^{n} \varphi_i(X_i)$

Sujeto a $\sum C_i X_i = S$

X_n = No. de unidades por instalar en la fase n

S = Cantidad disponible de dinero

$f_u(S)$= Máx. prob. de que las "n" últimas fases trabajan

$$f_u(s) = \max_{X_u \leq \frac{S}{C_u}} \{ \varphi_u(X_u) f_{n-1}(S - C_u X_u) \}$$

$$f_1(s) = \varphi_1 \left(\frac{S}{C_1} \right)$$

b) Asignación de personal (Científicos)
Ver pag. 247, ejercicio 3
Introduction to O.R. - Lieberman - Hillier

c) Juegos
Ver pag. 249, ejercicio 4
Introduction to O.R. - Lieberman - Hillier

4.- Control de Inventarios.-

Se tienen cierta demanda d_t en el período t (t=1,2,...,N) la cual debe ser satisfecha; un costo fijo por ordenar K; costo de compra ó producción proporcional al número ordenado (c) (el cual puede ser ignorado) y un costo de almacenamiento h por unidad por período.

La producción X_t en el período t está disponible al comienzo del período, y el costo de almacenamiento de una unidad es cargado al inicio del período.

j es un índice que indica el tiempo de la última producción (al principio del período).

Se solicitará producto únicamente cuando el almacenamiento es cero:

C_i = Costo para los primeros i períodos usando una política óptima.

$$C_n = \min_{j=1,2,\dots,n} \left[C_{j-1} + K + c\,(d_j + \cdots + d_n) + h\,\{ d_j + 2d_{j+1} + \cdots (n-j+1)d_n \} \right]$$

$C_0 = 0$

Ver pag. 362

Introduction to O.R. - Lieberman - Hillier

5.- PROCESOS DE DECISION DE MARKOV.-[1]

Problemas con un número infinito de etapas.-

Un proceso de Markov se puede caracterizar por la matríz de transición $P = \{p_{ij}\}$, donde p_{ij} es la probabilidad de que si el sistema se encuentra en el estado i en el periodo t, alcanzará el estado j en el período (t+1), independiente de como se llegó al estado i e independiente de t.

Llamando $\pi_i(n)$ a la probabilidad de que el sistema ocupará el estado i después de n transiciones ó pasos, si el estado para $n=0$ es conocido ($\pi(0)$).

$$\pi_j(n+1) = \sum_{i=1}^{N} \pi_i(n)\, p_{ij} \qquad n=0,1,2\ldots$$

$$\pi(n+1) = \pi(n)\, P$$

$$\pi(n) = \pi(0)\, P^n$$

En un proceso de Markov donde las probabilidades límites de estado π_i son independientes de las condiciones iniciales, se dice entonces que se trata de un proceso completamente ergódico.

$$\lim_{n\to\infty} \pi(n) \to \pi \qquad\qquad \pi = \pi P, \quad \sum_{i=1}^{N} \pi_i = 1$$

Si se considera una recompensa r_{ij} al pasar del estado i al estado j se tendrá:

$$v_i(n) = E\left[\text{recompensa total en "n" transiciones desde el estado i}\right]$$

$$v_i(n) = \sum_{j=1}^{N} p_{ij} \left\{ r_{ij} + v_j(n-1) \right\}$$

$$v_i(n) = q_i + \sum_{j=1}^{N} p_{ij}\, v_j(n-1) \rightarrow \text{acumula experiencias pasadas}$$

Donde q_i es la recompensa inmediata esperada.

Definiendo como ganancia: $g = \sum_{i=1}^{N} \pi_i \, q_i$ a la recompensa esperada por transición en el estado fijo.

Se puede demostrar que para "n" grande:

$$v_i(n) = ng + v_i$$

Donde v_i representa el valor del estado i como estado inicial

---

Proceso de Decisión.-

Si existen varias alternativas de decisión en el estado i, sea p_{ij}^k la probabilidad para pasar del estado i al estado j con la alternativa k, r_{ij}^k la recompensa asociada con esa transición (j=1,2,..., N).

Sea $X_i(n)$ la variable decisión en el estado i con n etapas faltantes por recorrer. En este caso, $X_i(n)$ representa el número de la alternativa (k) escogida. Una política de decisión $X(n)$ especifíca una decisión para cada estado.

Si se define a $v_i(n)$ como la recompensa óptima esperada, se tendrá:

$$v_i(n) = \max_k \left\{ q_i^k + \sum_{j=1}^{N} p_{ij}^k \, v_j(n-1) \right\}$$

Se puede demostrar que cuando $n \to \infty$, la política óptima $X(n) \to X$, independiente de n

Existen dos métodos para análizar este problema:

1.- Método iterativo para determinar la recompensa esperada

Este método trabaja como un problema normal de programación dinámica, etapa por etapa, maximizando en cada estado todas las posibles alternativas.

$$v_i(n+1) = \max_k \left\{ q_i^k + \sum_{j=1}^{N} p_{ij}^k \; v_j(n) \right\}$$

Se ha demostrado que para este proceso iterativo se converge a la mejor alternativa, para cada estado, cuando n va siendo mayor.

La diferencia con respecto al segundo método (método iterativo de políticas óptimas) es que en este último se determina la política óptima sin necesidad de estar calculando recompensas esperadas para n=1,2,..., obteniendo con pocas iteraciones el óptimo.

2. Método iterativo de políticas óptimas.-

Este método consiste de dos etapas, las cuales llevan al óptimo.

a) Determinación del valor.

Para <u>una política</u> dada se tiene:

$$v_i(n) = n\,g + v_i$$

$$v_i(n) = q_i + \sum_{j=1}^{N} p_{ij} \left\{ (n-1)\,g + v_j \right\}$$

Esto es: $g + v_i = q_i + \sum_{j=1}^{N} p_{ij} \; v_j \qquad i=1,2,\ldots,N$

Resulta un sistema de N ecuaciones con N v_i y 1 g incógnitas (N+1 incógnitas) por lo que se puede especificar un valor cualquiera de v_i (ej $v_N=0$), quedando N ecuaciones con N incógnitas (v_i ; $i = 1 \ldots,N$ y g).

b) Mejoramiento de política

Para encontrar una política mejor a la analizada en la primera etapa (a), se tenían los valores de v_j encontrados en ésta y se determinan las alternativas k (para cada estado i separadamente) que maximizan.

$$q_i^k + \sum_{j=1}^{N} p_{ij}^k \, v_j \qquad i=1,2,\ldots,N$$

Esto proporciona una nueva política con una <u>ganancia</u> mayor. Con la nueva política calculada se entra a la primera etapa (a), empezando así a iterar hasta encontrar dos políticas iguales, tomando aquella con mayor ganancia (g).

Para el caso de problemas con dos restricciones, se trabaja con las combinaciones resultantes de los términos independientes de las restricciones (ver pag. 259, ej. 7 - Lieberman Hillier).

BIBLIOGRAFIA.-

(1) Dynamic Programming and Markov Process - Howard M.I.T.

(2) Applied Dynamic Programming - Dreyfus and Bellman Princeton.

(3) Introduction to O.R. - Lieberman S. Hillier.