UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades

PROGRAMACION DINAMICA

Manuel de Jesús Guzmán Valdés

Guatemala

1982

PROGRAMACION DINAMICA

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ciencias y Humanidades

PROGRAMACION DINAMICA

Manuel de Jesús Guzmán Valdés

Trabajo de Investigación presentado para
optar al grado académico de
Licenciado en Matemática

Guatemala

1982

Vo. Bo. :

(f) ___________________________
Doctor Bernardo René Morales F.
Asesor

Tribunal:

(f) ___________________________
Doctor Bernardo René Morales F.

(f) ___________________________
Ingeniero Jacinto Quan Chú.

(f) ___________________________
Licenciado Angel Arévalo.

Fecha de aprobación: Enero de 1982.

A mis hijas

A mi esposa

A mi familia en general

Cuyo cariño y compren -
sión ha sido el mejor -
estímulo que he encon-
trado en las distintas
etapas de mi vida.

PREFACIO

En nuestro país un alto porcentaje de la educación superior es cubierto por profesionales, expertos en su campo, pero que pocas veces se han puesto a meditar en la forma adecuada de transmitir un conocimiento. El presente trabajo hecho con un enfoque didáctico, ha sido realizado tomando en cuenta tres aspectos:

a. El campo de acción de los matemáticos en nuestro país es muy extenso, pero en la actualidad un porcentaje muy alto se dedica a la enseñanza.

b. Algunos temas con fuerte fundamentación matemática a veces son cubiertos en forma superficial, lo cual limita el campo de aplicación de los mismos.

c. Al dedicarnos a la docencia debemos tomar en cuenta la responsabilidad que conlleva el hecho de formar profesionales, que tomarán decisiones que necesariamente, repercutirán en el desarrollo de nuestro país.

Sea este trabajo, una mínima contribución, ante el anhelo de estar cada día en una patria mejor.

También quiero dejar constancia de mi gratitud al Dr. Bernardo René Morales F., por su sabia orientación y tiempo dedicado a la asesoría para la realización de este trabajo.

CONTENIDO

I. INTRODUCCION

El impacto de la Investigación de Operaciones se ha sentido en muchas áreas del saber humano.

Diversas organizaciones consultoras en administración están actualmente comprometidas en actividades de Investigación de Operaciones, las cuales han ido mas allá de las aplicaciones empresariales y militares, para incluir hospitales, instituciones financieras, bibliotecas, planeación urbana, sistemas de transporte, investigación criminológica, etc. Tenemos pues, que la Investigación de Operaciones se aplica no solo al sector privado, sino también en el sector de servicios públicos. Todo esto ha obligado a la mayoría de instituciones académicas a ofrecer cursos de esta materia, en todos los niveles.

Dentro de los cursos de Investigación de Operaciones se han trabajado diversas técnicas que permiten resolver gran cantidad de problemas, sin embargo muchos de ellos dejan de ser prácticos cuando el número de variables y/o restricciones crece demasiado, lo cual señala la necesidad de ampliar los conocimientos en este campo.

La técnica que viene a resolver, en parte, el problema presentado por la dimensionalidad de los problemas, es la Programación Dinámica. Esto fue lo que motivó la realización de este trabajo. Hacer un desarrollo de Programación Dinámi ca con un enfoque didáctico ya sea para ser considerado como una unidad dentro de un curso de Investigación de Operaciones o bien para que aquellas personas que se sientan atraídas por el tema puedan estudiarlo por su cuenta, es lo que se pretende.

Al considerar el objetivo principal del trabajo, éste se ha desarrollado así:

Primero, antecedentes, que permiten la ubicación plena de la Programación dinámica dentro del amplio campo de la Investigación de Operaciones.

Segundo, el capítulo de Programación Dinámica, propiamente dicho, en él se hace el desarrollo del tema, presenta los fundamentos matemáticos a lo largo del capítulo, acompañándolos de ejemplos, de tal manera que el lector se sienta motivado a continuar con el estudio del tema, a la vez que se despierta su curiosidad por otras aplicaciones del mismo.

Tercero, la Programación Dinámica es una técnica que puede utilizarse para resolver problemas deterministas o bien probabilistas. Aunque el desarrollo se hace con mayor énfasis en los problemas de carácter determinista, al final del capítulo "Programación Dinámica", se trata el tema de Programación Dinámica Probabilista, presentando un ejemplo. La intención es introducir al lector en el estudio de la Programación Dinámica Probabilista , si alguien desea ampliar sus conocimientos, puede ver las referencias.

Cuarto, se presentan dos modelos, que caen dentro del tema de Inventarios, pero que se trabajan utilizando Programación Dinámica.

Al final se enuncian algunas conclusiones que encierran de alguna manera la meta o la intención que este trabajo tiene.

II. ANTECEDENTES

La Investigación de Operaciones es un enfoque científico en la toma de decisiones; aunque dentro de este contexto la investigación operacional puede remontarse a Frederick W. Taylor, Los Gilbreths y Henry Gantt, solo fue hasta la Segunda Guerra Mundial en que ésta se utilizó y empezó a desarrollarse hasta alcanzar el concepto actual:

"La Investigación de Operaciones, es la aplicación por grupos interdisciplinarios, del método científico a problemas relacionados con el control de las organizaciones o sistemas (hombre-máquina) a fin de que se produzcan soluciones que mejor sirvan a los objetivos de toda la organización".

Desde 1947 cuando el matemático norteamericano, George B. Dantzig, resumiendo el trabajo de muchos de sus precursores inventó el método Simplex, con lo cual dió inicio a la Programación Lineal, las nuevas técnicas se han venido desarrollando gracias a los esfuerzos y cooperación de personas interesadas, tanto en las instituciones académicas como en la industria, y apoyada por el avance de las computadoras digitales; así tenemos en la década de los cincuenta las áreas de: Programación Dinámica (Bellman), Programación no lineal (Kuhn y Tucker), Programación Entera (Gomory), Redes de Optimización (Ford y Fulkerson), Simulación (Markowitz), Inventarios (Arrow, Karlin, Scarf, Whitin), Análisis de --

decisiones (Raiffa) y Procesos Markovianos de Decisión (Howard). La generalización de la Investigación de Operaciones han tratado de darla Churckman, Ackoff y Arnoff.

La investigación de Operaciones describe un sistema mediante un modelo que luego se manipula para determinar la mejor forma de operación del sistema.

Entre las principales fases de un estudio de investigación de Operaciones están:

1. Formulación del problema.
2. Construcción de un modelo para representar el modelo bajo estudio.
3. Deducción de una solución a partir del modelo.
4. Prueba del modelo y de la solución inducida por éste.
5. Establecimiento de controles sobre la solución.
6. Ejecución. Poner la solución a trabajar.

Es esencial que el problema en consideración esté bien definido, para luego construir el modelo, usualmente éste es un modelo matemático.

Hay problemas de optimización que a la hora de plantearlos, resultan muy extensos, ya sea en el número de variables o bien en el número de restricciones, lo cual complica tremendamente su solución. Este tipo de problemas solo pueden ser resueltos cuando se les descompone en una serie de etapas.

Por el "Principio de Optimalidad" de Bellman, la solución secuencial de los problemas de decisión asociados con cada etapa es equivalente a la solución del problema de decisión del Sistema original. Este procedimiento secuencial se conoce como PROGRAMACION DINAMICA.

La Programación Dinámica, es entonces, una técnica matemática, cuyo desarrollo se debe en gran parte a Richard Bellman (1957), principalmente para mejorar la eficiencia de cómputo en ciertos problemas de optimización, cuya idea básica es descomponer el problema en subproblemas (más pequeños) los cuales son computacionalmente más manejables.

Es la técnica más apropiada para resolver problemas que requieren decisiones interrelacionadas, es decir, decisiones que se deben tomar en forma secuencial y las cuales influyen en las decisiones futuras de esa secuencia. Es aplicable a una gran variedad de problemas incluyendo distribución, inventarios y reemplazo.

Conviene mencionar que de todas las técnicas de Investigación de Operaciones, la Programación Dinámica es la que emplea conceptos más sencillos y sin embargo es la más difícil de manejar. Una de las dificultades que presenta es la carencia de una formulación definida y de algoritmos de solución, de donde que cada problema requiere decisiones básicas.

III. PROGRAMACION DINAMICA (P.D.)

A. Conceptos Básicos:

En un problema de Programación Dinámica debemos tomar - una serie de decisiones en una secuencia determinada, que - persigue una política óptima, sin que importe cuáles hayan sido el (o los) estado (s) y decisión (es) anteriores, las decisiones restantes constituirán una política óptima con - respecto al estado que se obtenga con la primera decisión. - Por ejemplo, si se han tomado decisiones equivocadas durante la primera y segunda semanas, esto no impedirá que en el futuro se tomen decisiones correctas. La Programación Dinámica nos permite llegar a decisiones óptimas para los períodos o etapas que todavía están en el futuro, a pesar de las decisiones incorrectas que se hayan tomado en el pasado.

Para introducir el concepto de Programación Dinámica consideremos el siguiente problema (simplificado) de presupuesto de capital.

Una compañía tiene N fábricas y en cada una está considerando una posible ampliación, el capital total asignado es QC. El número de planes posibles para la fábrica i es M_i, - con $i \in \{1, 2, \ldots, N\}$ la cual incluye la posibilidad de ninguna ampliación. El costo esperado adicional de la alternativa m_i (fábrica i) es C_{i,m_i} y el ingreso resultante es R_{i,m_i}. El objetivo del problema de decisión es seleccionar un plan factible m_i para cada fábrica i tal que se maximice el ingreso total resultante.

El método directo para resolver este problema es listar las diferentes combinaciones de alternativas (existen $\prod_i^N M_i$ de tales combinaciones), la óptima es la que proporciona el valor mayor de $\sum_i \sum_{m_i} R_{i,m_i}$ que satisface las restricciones de capital $\sum_i \sum_{m_i} C_{i,m_i} \leq C$.

El cálculo asociado al procedimiento anterior presenta las siguientes dificultades:

1. Cada combinación comprende una política de decisión para el problema completo, lo que implica la enumeración de "todas" las combinaciones posibles, lo cual puede no ser factible para N y M_i grandes.
2. La combinación óptima puede estar disponible desde el principio de los cálculos pero la optimidad no puede verificarse hasta que se consideran todas las combinaciones.
3. Las combinaciones infactibles (las que infringen las restricciones de capital) no se eliminan por adelantado.

La Programación Dinámica está diseñada para evitar estas dificultades de la manera siguiente:

1. El problema se descompone en sub problemas (llamados etapas) y cada subproblema se optimiza sobre sus alternativas, de manera que nunca es necesario enumerar

todas las combinaciones anticipadamente.

ii. Debido a que la optimización se aplica a cada sub problema, todas las combinaciones no óptimas se descartan sistemáticamente.

iii. Los sub problemas están ligados de manera especial de tal forma que nunca es posible optimizar sobre combinaciones infactibles.

Lo anterior nos permite señalar los tres elementos básicos de programación:

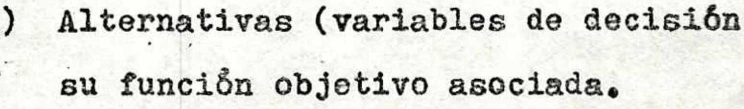
a) Etapa.

b) Alternativas (variables de decisión) en cada etapa y su función objetivo asociada.

c) Estado del sistema en cada etapa.

Esto nos muestra la estructura de la Programación Dinámica que proporciona la capacidad de tomar decisiones relacionadas con el problema en diversas etapas o puntos del tiempo. En cada paso del problema se toma una decisión para cambiar el estado y de ese modo aumentar al máximo la utilidad.

Un modelo típico de Programación Dinámica con N etapas puede ilustrarse gráficamente como:



Donde el proceso de optimización normalmente comienza con la etapa 1.

El estado del sistema es quizá el concepto más importante en un modelo de Programación Dinámica, representa la "liga" entre etapas subsecuentes, de tal manera que aunque cada etapa se meximiza por separado, la decisión resultante es automáticamente factible para el problema completo.

La definición del estado es el concepto mas sutil en formulaciones de Programación Dinámica, no existe forma fácil de definirlo, pero usualmente pueden encontrarse pistas haciendo las dos preguntas siguientes:

1. ¿ Qué relaciones enlazan las etapas ?
2. ¿ Qué información es necesaria para tomar decisiones factibles en la etapa actual sin verificar la factibilidad de decisiones hechas en etapas anteriores ?

Los siguientes ejemplos nos ayudarán a comprender la definición de estado.

Primero, consideremos el problema de presupuesto de capital. Cada fábrica representa una etapa para la cual debe tomarse una decisión, la alternativa dada por la variable de decisión m_i, es la etapa i que designa un plan específico de expansión. En este caso la función de rendimiento es R_{i,m_i}.

¿Qué define el estado en la etapa i? notemos que las etapas están ligadas por el hecho de que todas las fábricas (etapas) están compitiendo por una participación de capital limitado C. Lo cual sugiere que el estado deberá estar definido en términos de la asignación de capital.

La definición del estado en la etapa i está dada como la "cantidad de capital asignado a las etapas 1,2,...,i (es decir, a las primeras i etapas)" donde la diferencia entre el capital asignado a las primeras i etapas y el capital asignado a las i-1 etapas, dará la cantidad asignada únicamente a la etapa i.

Este es un ejemplo típico de asignación en el cual un recurso (o recursos) se distribuye, óptimamente, entre un número de actividades (etapas).

La definición de estado para todos los problemas de asignación es el mismo: la cantidad de recurso asignado a un número sucesivo de etapas comenzando con la etapa 1.

Segundo, consideremos una situación de reemplazo de equipo, donde al final de cada año se toma la decisión de conservar una máquina otro año o reemplazarla inmediatamente. Si una máquina se retiene, su beneficio declina, por otro lado el reemplazarla implica el costo de reemplazo. El problema es decidir cuándo deberá reemplazarse una máquina a fin de maximizar el beneficio neto total.

En este problema la etapa i representa el año i. Las alternativas en cada etapa son: conservar la máquina o reemplazarla. Nos preguntamos ¿ Cuál es la relación entre dos etapas sucesivas? ¿ Qué información es necesaria de las etapas anteriores para tomar una decisión ? La respuesta es: la -

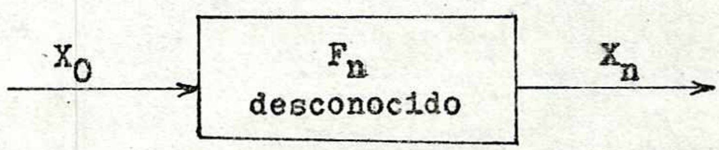
edad de la máquina.

Por lo tanto el estado del sistema en una etapa se define como la edad de la máquina al inicio del período asociado.

Desarrollaremos ahora los mecanismos de que se vale la Programación Dinámica para resolver este tipo de problemas por medio de funciones recursivas discretas o contínuas, con o sin restricciones.

B. <u>El Principio de Descomposición</u>

Supongamos que la compañía ABC se le presenta como el bloque:



Sistema ABC

dentro del cual se encuentra toda la estructura de la compañía, todas sus componentes (controlables y no controlables) y todas las interrelaciones existentes entre las mismas. Supongamos que en un momento específico de tiempo, t_0, por ejemplo, el 1o. de Enero de 1981 a las 8:00 hrs., a la compañía se le puede describir por el vector, X_0 :

$$X_0 = \begin{pmatrix} X^0{}_1 \\ X^0{}_2 \\ \vdots \\ X^0{}_i \\ \vdots \\ X^0{}_m \end{pmatrix}$$

Supóngase que un año después, t_n, es decir, el 1o. de Enero de 1982, a las 8:00 hrs. se vuelven a medir todas las m componentes de la compañía, dando los valores:

$$X_n = \begin{pmatrix} X^n{}_1 \\ X^n{}_2 \\ \vdots \\ X^n{}_i \\ \vdots \\ X^n{}_m \end{pmatrix}$$

Es claro que los valores de las componentes en X_n están relacionadas de alguna manera con las de X_0. Por ejemplo si la componente i representa empleados tenemos:

No. de empleados en el tiempo t_n	=	No. de empleados en el tiempo t_0	+	No. de empleados contratados en el período t_n-t_0	−	Empleados de baja en el período t_n-t_0

Esto nos indica que $X^n{}_i$ es función conocida de $X^0{}_i$, es -

decir, de alguna manera X^0_i se transformó en X^n_i en -- $(t_n - t_0)$ períodos después. Denotemos esta función por - F^n_i, formalizando tenemos:

$$X^n_i = F^n_i (X^0_i)$$

Lo cual ocurre con cada una de las componentes. Formando el vector de funciones tenemos:

$$F_n = \begin{pmatrix} F^n_1 \\ F^n_2 \\ \vdots \\ F^n_i \\ \vdots \\ F^n_m \end{pmatrix} \qquad y \qquad X_n = F_n (X_0)$$

O sea

$$\begin{pmatrix} X^n_1 \\ X^n_2 \\ \vdots \\ X^n_i \\ \vdots \\ X^n_m \end{pmatrix} = \begin{pmatrix} F^n_1 (X^0_1) \\ F^n_2 (X^0_2) \\ \vdots \\ F^n_i (X^0_i) \\ \vdots \\ F^n_m (X^0_m) \end{pmatrix}$$

Necesitamos representar explícitamente el vector de -- transformaciones F_n.

Puede ocurrir que un año sea demasiado tiempo y esto complique mucho el análisis, necesitamos descomponerlo en partes que sean más fáciles de analizar.

Una vez desmembrada la función, se analizan sus partes y se les vuelve a juntar para formar el conjunto original.

Supongamos que desconocemos qué es lo que F_n hace a X_0 para transformarla en X_n, pero se conoce la transformación f_n que actuando sobre X_{n-1} la transforma en X_n, suponiendo X_{n-1}, el valor de todas las componentes el 1o. de Diciembre de 1981, a las 8:00 hrs.

Es decir:

$$X_n = f_n (X_{n-1})$$

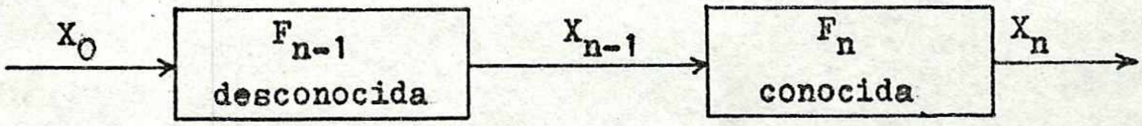
O bien:

$$\begin{pmatrix} X^n{}_1 \\ X^n{}_2 \\ \vdots \\ X^n{}_i \\ \vdots \\ X^n{}_m \end{pmatrix} = \begin{pmatrix} f^n{}_1 (X^{n-1}{}_1) \\ f^n{}_2 (X^{n-1}{}_2) \\ \vdots \\ f^n{}_i (X^{n-1}{}_i) \\ \vdots \\ f^n{}_m (X^{n-1}{}_m) \end{pmatrix}$$

Pero para resolver el sistema original debemos encontrar una transformación que conecte X_0 con X_{n-1}, ya que conocemos f_n tal que $X_n = f_{\_}(X_{n-1})$; o sea, debemos encontrar F_{n-1} que satisfaga:

$$X_{n-1} = F_{n-1}(X_0)$$

Tenemos:



que equivale a

$$X_{n-1} = F_{n-1} (X_0)$$
$$X_n = f_n (X_{n-1})$$
$$X_n = f_n \left(F_{n-1} (X_0)\right)$$

Si nuevamente conocemos otra transformación f_{n-1} tal que actuando sobre X_{n-2} nos da X_{n-1}, tenemos:

$$X_n = f_n (X_{n-1})$$
$$X_{n-1} = f_{n-1} (X_{n-2})$$
$$X_{n-2} = F_{n-2} (X_0)$$

o bien:

$$X_n = f_n (f_{n-1} (F_{n-2} (X_0)))$$

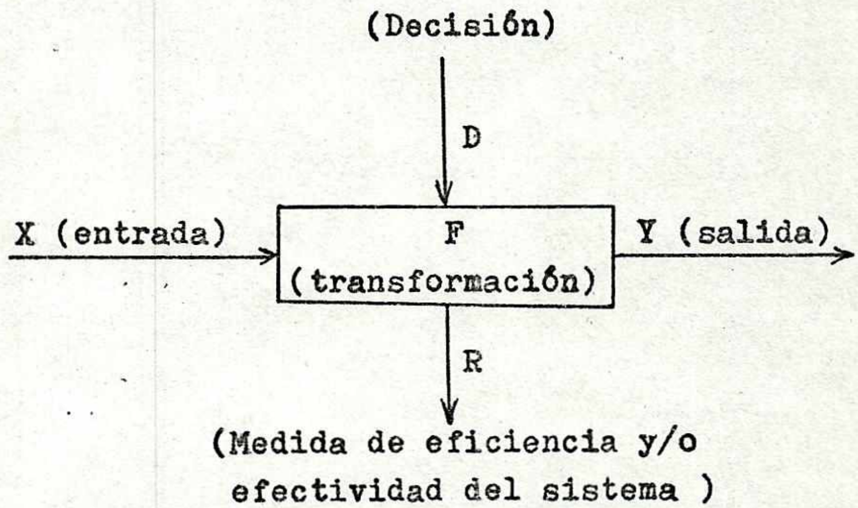
Notamos que es necesario descomponer el problema original en n etapas, es decir tantas etapas como sea necesario para descubrir la estructura exacta de las transformaciones -- $f_1, f_2, \ldots, f_n$ de tal manera que:

$$X_n = f_n(f_{n-1}(f_{n-2}(\ldots f_2(f_1(X_0))\ldots)))$$

C. Problema de Decisión de una Etapa

Analizaremos el problema de decisión que puede surgir en cualquiera de las etapas en las que se ha descompuesto el problema original.

Supongamos que una etapa cualquiera está representada -- por:



Nótense los siguientes elementos:

a) X el vector de entrada, proporciona toda la información relevante respecto a las componentes importantes del sistema, antes de que se tome una decisión.

b) Y vector de salida, proporciona la información relevante después de haber tomado una decisión.

c) Conjunto de decisiones agrupadas en el vector D, son los elementos que se pueden emplear para alcanzar los objetivos del sistema.

d) Una transformación F, tal que $Y = F(X,D)$.

e) Un vector de medida de eficiencia y/o efectividad del sistema, dado por R, el cual puede ser positivo (utilidad, salud, etc.) o negativo (costo, tiempo, etc..). Este vector es una función $\bar{F}$ de la entrada, la salida y la decisión que se toma.

$$R = \bar{F}(X,Y,D)$$
$$= \bar{F}(X,F(X,D),D)$$

Componiendo las funciones $\bar{F}$ y F para obtener r, tenemos que:

$$R = r(X,D)$$

En todo problema debe existir por lo menos un valor de decisión que lo optimice. Sea este valor óptimo g(X) Donde para valores positivos tenemos:

$$g(X) = \underset{D}{\text{Máx}}\ r(X,D)$$

Y para valores negativos

$$g(X) = \underset{D}{\text{mín}}\ r(X,D)$$

Suponiendo que D^* genera el valor óptimo tenemos:

$$g(X) = r(X,D^*) = \underset{D}{\text{Máx}}\ r(X,D) \geq r(X,D)$$

ó

$$g(X) = r(X,D^*) = \underset{D}{\text{mín}}\ r(X,D) \leq r(X,D)$$

NOTA: Si X y D tienen una sola componente, solo hay una decisión por tomar. Si X tiene una componente y D tiene varias, entonces por cada componente de D habrá un valor de - f(X).

Ejemplo:

Una persona vive a 20 km de su trabajo, todas las mañanas de lunes a viernes, tiene que presentarse a las 9:00 hrs Si llega 10 minutos tarde le descuentan el día de trabajo. - Sale a las 17:00 hrs. y regresa a su casa. Esta persona tiene varias maneras de dirigirse al trabajo:

a) Maneja su automóvil de su casa al trabajo, lo estaciona por 8 hrs. y regresa en su automóvil a su casa.

b) Maneja su automóvil a la estación del tren, donde lo estaciona en la calle. Toma el tren al trabajo. Al salir del trabajo toma el tren hasta donde está su automóvil - y de allí maneja a su casa.

c) Toma un taxi de su casa al trabajo y viceversa.

La primera alternativa le cuesta \$12, el tiempo de recorrido de ida (o vuelta) es de 30 minutos.

La segunda alternativa le cuesta \$ 2.75, el tiempo total de ida (o vuelta) a su casa es de 60 minutos.

La tercera alternativa le cuesta \$ 15, y le toma 45 minutos de ida (o vuelta) a su casa.

El vector de entrada tiene dos componentes: X_1 cantidad de dinero con el que la persona sale del hogar y X_2 la hora en que sale de su casa. La eficiencia y/o efectividad la mi de en minutos viajados por dolar gastado. El individuo gana \$ 100.00 diarios.

<u>Caso 1</u> : Se supone $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \$\ 100 \\ 8 \text{ Hrs.} \end{pmatrix}$

La decisión D_1 le cuesta \$ 12.00 y lo coloca en el trabajo a las 8:30, por lo tanto

$$r\ (X,D_1) = \frac{30}{12} = 2.5 \quad \text{Min/\$}$$

La decisión D_2, da:

$$r(X,D_2) = \frac{60}{2.75} = 21.82 \quad \text{Min/\$}$$

La decisión D_3, da:

$$r(X,D_3) = \frac{45}{15} = 3 \quad \text{Min/\$}$$

El problema de decisión de acuerdo con su manera de definir su eficiencia y/o efectividad, es encontrar la decisión que optimice su satisfacción,

Entonces:

$$g(X) = r(X,D^*) = \underset{D_1 D_2 D_3}{\text{M á x}} \quad r(X,D) \geqslant r(X,D_i)$$

$g(X) = \text{Máx}\ (2.5, 21.82, 3) = 21.82;\ D^* = D_2$, decisión óptima, puesto que en este caso es la que permite viajar más por menos dinero.

Caso 2 : Se supone:

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \$\ 100 \\ 8{:}30 \text{ hrs.} \end{pmatrix}$$

a) Para D_1 : $r(X,D_1) = \dfrac{30}{12} = 2.5$ min/$

b) Para D_2 : $r(X,D_2) = \dfrac{60}{2.75 + 100} = 0.58$ Min/$

c) Para D_3 : $r(X,D_3) = \dfrac{45}{15 + 100} = 0.39$ Min/$

En este caso

$$g(X) = r\,(X,D^*) = \underset{D_1 D_2 D_3}{\text{Máx}}\ r(X,D)$$

$$= \underset{D_1 D_2 D_3}{\text{Máx}}\ (2.5, 0.58, 0.39) = 2.5$$

Acá la decisión óptima en el caso que $X_1=100$ y $X_2=8.30$ es $D^* = D_1$

Caso 3 : Se supone:

$$X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} = \begin{pmatrix} \$100 \\ 8{:}20 \text{ hrs.} \end{pmatrix}$$

Tenemos:

$$r(X,D_1) = \frac{30}{12} = 2.5 \text{ min/\$}$$

$$r(X,D_2) = \frac{60}{102.75} = 0.58 \text{ min/\$}$$

$$r(X,D_3) = \frac{45}{15} = 3 \text{ min/\$}$$

Por lo tanto

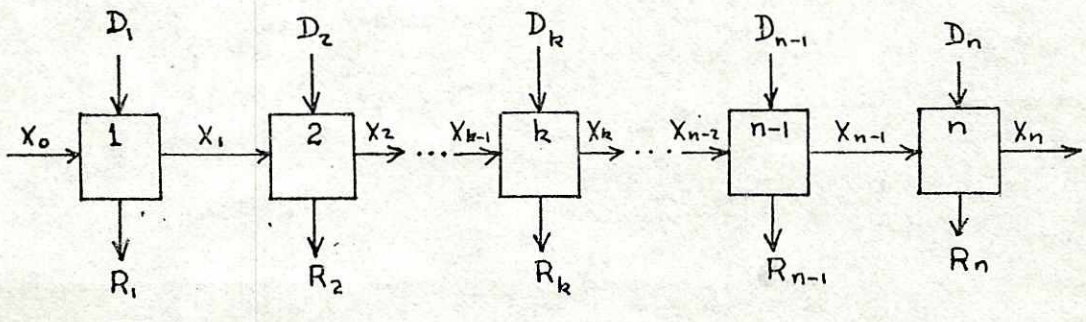
$$g(X) = \underset{D_1 D_2 D_3}{\text{M á x}} \quad (X,D) = \text{Máx } (2.5,\ 0.58,\ 3) = 3$$

y la decisión óptima es $D^{*} = D_3$

Es importante notar que un cambio en el vector de entrada X, puede cambiar la decisión óptima D^{*}, y desde luego cambia g(X).

D. Problema de decisión de n etapas

En un problema con n etapas, para cada etapa k, con - k= 1,2,...,n tendremos un vector de entrada X_{k-1}, uno de salida X_k, una transformación f_k que relaciona la salida con la entrada, un vector de decisión D_k y un vector de medida - de eficiencia de la etapa, R_k.



De tal manera que

$$X_n = f_n(X_{n-1}, D_n) \qquad y \qquad R_n = r_n(X_{n-1}, D_n)$$

$$X_{n-1} = f_{n-1}(X_{n-2}, D_{n-1}) \qquad y \qquad R_{n-1} = r_{n-1}(X_{n-2}, D_{n-1})$$

$$\vdots \qquad \vdots \qquad \vdots$$

$$X_2 = f_2(X_1, D_2) \qquad y \qquad R_2 = r_1(X_1, D_2)$$

$$X_1 = f_1(X_0, D_1) \qquad y \qquad R_1 = r_1(X_0, D_1)$$

Por lo que:

$$X_n = f_n(f_{n-1}(\ldots(f_2(f_1(X_0,D_1)\ D_2)\ldots)D_{n-1})\ D_n)$$

y

$$R_n = r_n(r_{n-1}(\ldots(r_2(r_1(X_0,D_1)\ D_2)\ldots)D_{n-1})\ D_n)$$

Componiendo las funciones $f_n, f_{n-1}, \ldots, f_2, f_1$ y $r_n, r_{n-1}, \ldots, r_2, r_1$ en una sola función a las que llamaremos f_n y r_n tenemos:

$$X_n = f_n\ (X_0, D_1, D_2, \ldots, D_n)$$

$$R_n = r_n\ (X_0, D_1, D_2, \ldots, D_n)$$

Las cuales dependen de la entrada al sistema X_0 y de las decisiones $D_1, D_2, \ldots, D_n$.

El problema de decisión es el de optimizar una composición (ya sea en forma de producto, suma u otra forma) de todos los vectores $R_1, \ldots, R_n$.

Representaremos dicha composición por $*$. Necesitamos encontrar las decisiones que optimicen la composición : $R_1 * R_2^* \ldots {}^* R_n$, es decir, buscamos:

$$\begin{aligned} g_n(X_0) &= \underset{D_1,\ldots,D_n}{\text{Opt.}} (R_1^* R_2^* \ldots {}^* R_n) \\ &= \text{Opt.} \left[r_1(X_0,D_1)^* \, r_2(X_1,D_2)^* \ldots {}^* r_n(X_{n-1},D_n) \right] \\ &= \left[r_1(X_0,D_1^*)^* \, r_2(X_1,D_2^*)^* \ldots \, r_n(X_{n-1},D_n^*) \right] \end{aligned}$$

Esta última expresión indica que la composición de las decisiones óptimas de cada etapa, genera el óptimo del sistema.

Resumiendo, nuestro problema es encontrar las decisiones $D_1,\ldots,D_n$ en cada etapa, a fin de optimizar la eficiencia y/o eficacia total del sistema, $g_n(X_0)$.

Sujeto a $X_k = f_k(X_{k-1}, D_k)$; $\forall$ $k = 1,2,\ldots, n$.

Ahora necesitamos descomponer el problema de optimización, en n problemas de optimización de una sola etapa.

Supongamos que la composición toma la forma de adición; además que optimización significa maximización.

El problema a resolver es:

$$g_n(X_0) = \underset{D_1,\ldots,D_n}{\text{Máx}} \left[r_1(X_0,D_1) + r_2(X_1,D_2) + \ldots + r_n(X_{n-1},D_n) \right]$$

Sujeto a.

$$X_k = f_k(X_{k-1},D_k) \quad ; \quad k = 1,2,\ldots,n$$

El cual se puede escribir como:

$$g_n(X_0)=\underset{D_1}{\text{Máx}}\left[r_1(X_0,D_1)\right] + \underset{D_2,\ldots,D_n}{\text{Máx}}\left[r_2(X_1,D_2)+\ldots+r_n(X_{n-1},D_n)\right]$$

Sujeto a: $X_k=f_k(X_{k-1},D_k)$; $k=1,2,\ldots,n$

luego $g_{n-1}(X_1) = \underset{D_2,\ldots D_n}{\text{Máx}}\left[r_2(X_1,D_2)+\ldots+r_n(X_{n-1},D_n)\right]$

entonces podemos escribir

$$g_n(X_0) = \underset{D_1}{\text{Máx}}\left[r_1(X_0D_1) + g_{n-1}(X_1)\right]$$

Pero como: $X_1= f_1(X_0,D_1)$

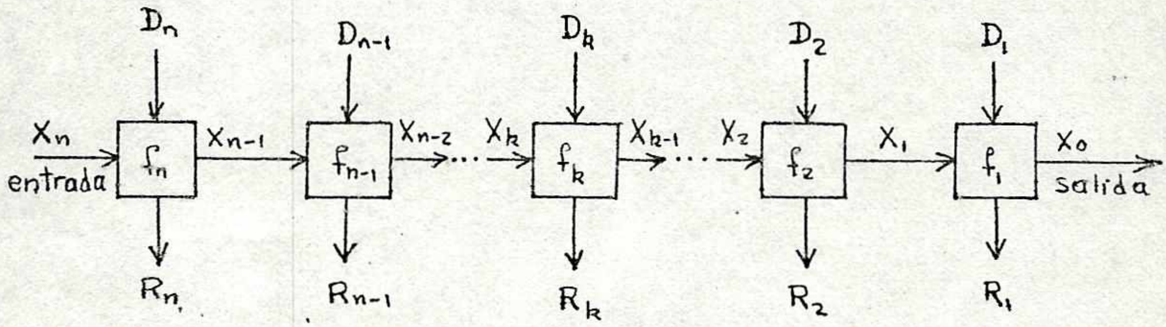
Se tiene finalmente:

$$g_n(X_0) = \underset{D_1}{\text{Máx}}\left[r_1(X_0,D_1)+g_{n-1}(f_1(X_0,D_1))\right] \quad ; \quad n=n,\ n-1,\ldots,2,1$$

Con $g_0(X_n)=0$, que es una condición inicial de la función recursiva.

Esta última expresión es la función recursiva que resuelve el problema original, resolviendo n problemas de optimización, de una etapa cada uno.

Consideremos ahora el diagrama



del cual obtenemos la función recursiva

$$g_n(X_n) = \underset{D_n,\ldots,D_1}{\text{Opt}} \left[r_n(X_n,D_n) + r_{n-1}(X_{n-1},D_{n-1}) + \ . \ . \ . + r_1(X_1,D_1) \right]$$

Sujeto a:

$$X_{k-1} = f_k(X_k,D_k) \ ; \quad k=n,\ n-1,\ldots,2,1.$$

Haciendo:

$$g_{n-1}(X_{n-1}) = \underset{D_{n-1},\ldots,D_1}{\text{Opt}} \left[r_{n-1}(X_{n-1},D_{n-1}) + \ldots + r_1(X_1,D_1) \right]$$

Se tiene que

$$g_n(X_n) = \underset{D_n}{\text{Opt}} \left[r_n(X_n,D_n) + g_{n-1}(X_{n-1}) \right] \ ; \quad \text{con } n=n,\ n-1,\ldots,1$$

Como:

$$X_{n-1} = f_n(X_n,D_n)$$

Obtenemos:

$$g_n(X_n) = \underset{D_n}{\text{Opt}} \left[r_n(X_n,D_n) + g_{n-1}(f_n(X_n,D_n)) \right] \ ; \quad n=n,n-1,\ldots,1$$

Con $g_0(X_0) = 0$ como condición inicial.

Nótese que hemos construido dos funciones recursivas, en la primera la solución secuencial recursiva se hace del vector de entrada al vector de salida, mientras que en el segundo caso el proceso secuencial recursivo se hace del vector de salida al vector de entrada.

Aunque muchos problemas de optimización pueden resolverse en ambos sentidos, hay unos que por su estructura solamen te se pueden resolver en uno de ellos.

Ejemplo (utilizando función recursiva de vector de salida al vector de entrada)

Ejemplo: Una compañía lechera va a comprar 6 cisternas para transportar leche, cada una con capacidad de 10,000 litros. Hay cuatro zonas productoras de lecha que abastecen la ciudad, localizadas en los puntos A, B, C, y D.

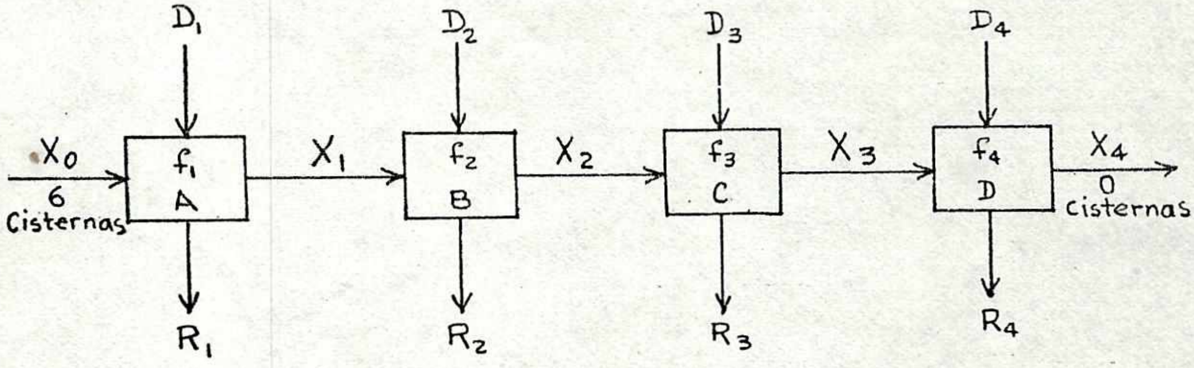
La compañía al no tener que contratar cisternas particulares, y según las cisternas asignadas a cada zona obtendrá los ahorros mensuales siguientes:

No. de cisternas	ZONAS A	B	C	D
0	0	0	0	0
1	4	2	6	2
2	6	4	8	3
3	7	6	8	4
4	8	8	9	5
5	9	9	9	6
6	10	10	10	6

en miles de quetzales

El problema es ¿ Cuántas cisternas deben asignarse a cada zona a fin de maximizar el ahorro mensual estimado para la compañía ?

Tenemos un problema de decisión de cuatro etapas.



El vector de entrada al sistema ,X_0, debe ser igual al número de cisternas (6), y el vector de salida del sistema ,X_4, debe ser igual a cero, puesto que todas las cisternas deben ser asignadas.

La decisión D_k, en la etapa k, es asignar cisternas a cada zona.

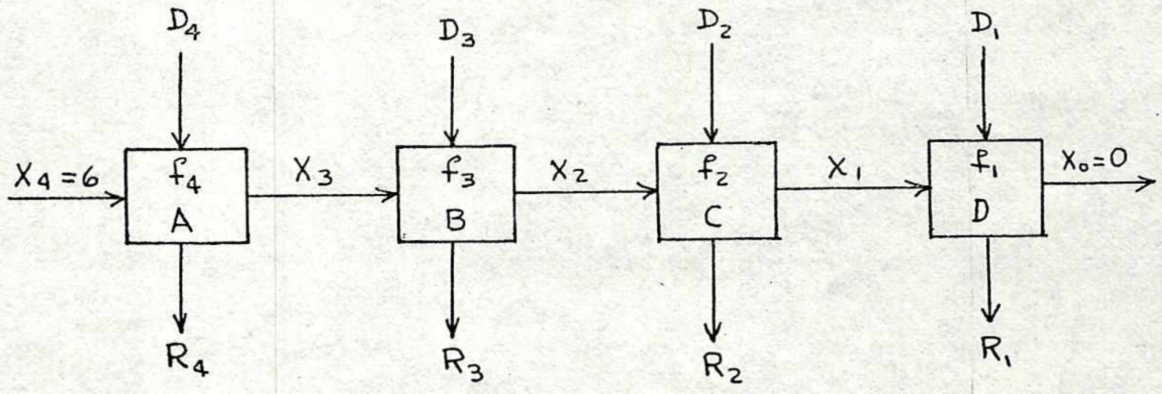
La eficiencia de la etapa k, R_k, es el ahorro mensual que se logra por asignar D_k cisternas cuando aún quedan disponibles X_{k-1}.

La función que relaciona X_k con X_{k-1} es:

$$X_k = f_k(X_{k-1}) = X_{k-1} - D_k \quad ; \quad k = 1,2,3,4,$$

O sea, el número de cisternas disponibles para la etapa k+1 es igual al número de cisternas que se disponían para la asignación de la etapa k menos las cisternas que se asignaron a la etapa k.

Resolveremos el problema en el sentido de salida a entrada, haciendo uso de la función recursiva.



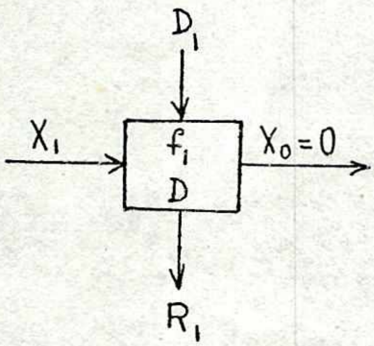
$$g_4(X_4) = \max_{D_4, D_3, D_2, D_1} \left[r_4(X_4, D_4) + r_3(X_3, D_3) + r_2(X_2, D_2) + r_1(X_1, D_1) \right]$$

Sujeto a $\quad X_{k-1} = X_k - D_k$

y $\qquad X_4=6 \quad$ condición inicial del sistema

$\qquad X_0=0 \quad$ condición final del sistema.

<u>Optimización de una etapa</u>



$$g_1(X_1) = \max_{D_1} \left[r_1(x_1, D_1) + g_0(X_0) \right]$$

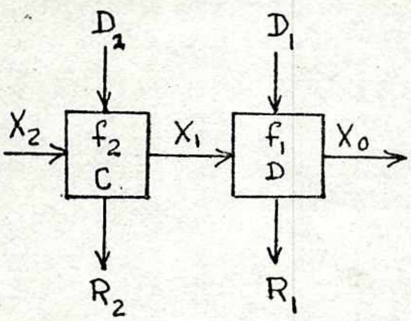
como $g_0(X_0) = 0 \quad ; \quad X_1 = D_1$

Tenemos:

$X_1 \backslash D_1$	0	1	2	3	4	5	6	$g_1(X_1)$	D_1^*
	$r_1(X_1,D_1)+0$								
0	0+0=0							0	0
1		2+0=2						2	1
2			3+0=3					3	2
3				4+0=4				4	3
4					5+0=5			5	4
5						6+0=6		6	5
6							6+0=6	6	5 ó 6

$g_1(X_1)$ nos indica el ahorro máximo asociado a la decisión -- D_1 y a la entrada X_1. La última columna indica la decisión óptima D_1^*.

<u>Optimización de dos etapas</u>



$$g_2(X_2) = \max_{D_2, D_1} \left[r_2(X_2,D_2) + r_1(X_1,D_1) \right]$$

sujeto a $X_1 = X_2 - D_2$

o sea $$g_2(X_2) = \max_{D_2} \left[r_2(X_2,D_2) + \underset{D_1}{\text{Máx}}\ r_1(X_1,D_1) \right]$$

es decir:

$$g_2(X_2) = \underset{D_2}{\text{Máx}} \left[r_2(X_2,D_2) + g_1(X_1) \right]$$

$$= \underset{D_2}{\text{Máx}} \left[r_2(X_2,D_2) + g_1(X_2-D_2) \right]$$

Los valores asociados con la función recursiva para dos etapas $g_2(X_2)$ son :

X_2 \ D_2	$r_2(X_2,D_2) + g_1(X_2-D_2)$ 0	1	2	3	4	5	6	$g_2(X_2)$	D_2^*
0	0+0=0							0	0
1	0+2=2	6+0=6						6	1
2	0+3=3	6+2=8	8+0=8					8	1 ó 2
3	0+4=4	6+3=9	8+2=10	8+0=8				10	2
4	0+5=5	6+4=10	8+3=11	8+2=10	9+0=9			11	2
5	0+6=6	6+5=11	8+4=12	8+3=11	9+2=11	9+0=9		12	2
6	0+6=6	6+6=12	8+5=13	8+4=12	9+3=12	9+2=11	10+0=10	13	2

El renglón D_2 indica el número de cisternas que se asignan a C, X_2 el número disponible de cisternas.

<u>Optimización de tres etapas</u>

$$g_3(X_3) = \underset{D_3}{\text{Máx}} \left[r_3(X_3,D_3) + g_2(X_2) \right]$$

Sujeto a

$$X_2 = X_3 - D_3$$

$$\therefore \quad g_3(X_3) = \underset{D_3}{\text{Máx}} \left[r_3(X_3,D_3) + g_2(X_3-D_3) \right]$$

X_3 \ D_3	$r_3(X_3,D_3)+g_2(X_3-D_3)$ 0	1	2	3	4	5	6	$g_3(X_3)$	D_3
0	0+0=0							0	0
1	0+6=6	2+0=2						6	0
2	0+8=8	2+6=8	4+0=4					8	0 ó 1
3	0+10=10	2+8=10	4+6=10	6+0=6				10	0 ó 1 ó 2
4	0+11=11	2+10=12	4+8=12	6+6=12	8+0=8			12	1 ó 2 ó 3
5	0+12=12	2+11=13	4+10=14	6+8=14	8+6=14	9+0=9		14	2 ó 3 ó 4
6	0+13=13	2+12=14	4+11=15	6+10=16	8+8=16	9+6=15	10+0=10	16	3 ó 4

Optimización de las cuatro etapas

$$g_4(X_4) = \underset{D_4}{\text{Máx}} \left[r_4(X_4,D_4) + g_3(X_3)\right]$$

Sujeto a

$$X_4=6 \quad \text{y} \quad X_3= X_4-D_4 = 6-D_4$$

o sea

$$g_4(X_4) = \underset{D_4}{\text{Máx}} \left[r_4(6,D_4) + g_3(6-D_4) \right.$$

Notemos que ahora el vector de entrada es un vector fijo - $X_4 = 6$

X_4 \ D_4	$r_4(6,D_4)+g_3(6-D_4)$ 0	1	2	3	4	5	6	$g_4(X_4)$	D_4^*
6	0+16=16	4+14=18	6+12=18	7+10=17	8+8=16	9+6=15	10+0=10	18	1 ó 2

Vemos que la solución óptima D_4^* es 1 ó 2 cisternas, que proporciona un ahorro total mensual de Q 18,000.

Supongamos que se asigna una cisterna a la zona A, entonces quedan 5 disponibles para las otras tres zonas (etapas). Luego tenemos que $g_3(5)=14$ que corresponde a la decisión D_3^* igual 2 ó 3 ó 4 cisternas. Supongamos que se asignan dos cisternas a B por lo que habrá una disponibilidad de 3. $g_2(3)=10$, que corresponde a la decisión D_2^* de dos cisternas a C.

Ahora tenemos una disponibilidad de una cisterna y $g_1(X_1) = g_1(1) = 2$ indicando que $D_1^* = 1$; o sea una cisterna para la zona D. Nótese que en este caso la solución no es única.

E. <u>Composiciones más generales de las eficiencias y/o efectividades de un sistema.</u>

La función recursiva resuelve en cada etapa un problema de optimización asociado a un subsistema tomando en cuenta la posible variación en la entrada a ese subsistema. Cuando se termina la optimización se añade una etapa al subsistema creando un subsistema mayor; se continúa el proceso de optimización hasta que el subsistema abarque el sistema original.

Mitten derivó la condición suficiente que una composición de funciones debe satisfacer, a fin de que las funciones recursivas trabajen.

Consideremos un problema de optimización de dos etapas.

Sea:

$$g_2(X_2) = \underset{D_2,\, D_1}{\text{M á x}} \left[r_2(X_2,D_2)^* \; r_1(X_1,D_1)\right]$$

Sujeto a:

$$X_1 = f_2(X_2,D_2)$$

Eliminando X_1 tenemos:

$$g_2(X_2) = \underset{D_2,D_1}{\text{M á x}} \left[r_2(X_2,D_2)^* \; r_1(X_2,D_2,D_1)\right]$$

Sea

$$\bar{g}_2(X_2) = \underset{D_2}{\text{máx}} \left[r_2(X_2,D_2)^* \; \underset{D_1}{\text{Máx}} \; r_1(X_2,D_2,D_1)\right]$$

de la definición de máximo tenemos que $\bar{g}_2(X_2) \leq g_2(X_2)$. Deseamos saber, bajo qué condiciones se logra que $\bar{g}_2(X_2) = g_2(X_2)$.

Una de las condiciones suficientes que debe satisfacerse para que exista la igualdad, es que $\left[r_2(X_2,D_2)^* \, r_1(X_2,D_2,D_1)\right]$ sea una función monótona no decreciente de r_1 para cada valor factible de r_2.

Por la definición de monotonicidad se tiene, que si

$r_1(X_2,D_2,D_1') \geq r_1(X_2,D_2,D_1'')$, con X_2 y D_2 fijas,

entonces

$$\left[r_2(X_2,D_2) * \; r_1(X_2,D_2,D_1')\right] \geq \left[r_2(X_2,D_2) * \; r_1(X_2,D_2,D_1'')\right]$$

Pero por cada valor de X_2 y D_2 se tiene

$$\left[r_2(X_2,D_2) * r_1(X_2,D_2D_1^*)\right] \geq \left[r_2(X_2,D_2) * r_1(X_2,D_2,D_1)\right]$$

Para todos los valores de D_1.

Pero esta desigualdad también debe cumplirse para el máximo, del lado derecho de la misma, es decir,

$$\left[r_2(X_2,D_2) * r_1(X_2,D_2,D_1^*)\right] \geq \underset{D_1}{\text{Máx}} \left[r_2(X_2,D_2) * r_1(X_2,D_2,D_1)\right]$$

en consecuencia

$$\bar{g}_2(X_2) = \underset{D_2}{\text{máx}} \left[r_2(X_2,D_2) * r_1(X_2,D_2,D_1^*)\right]$$

$$\geq \underset{D_2}{\text{Máx}} \; \underset{D_1}{\text{Máx}} \left[r_2(X_2,D_2) * r_1(X_2,D_2,D_1)\right] = g_2(X_2)$$

Como teníamos que $\bar{g}_2(X_2) \leq g_2(X_2)$ y acabamos de obtener $\bar{g}_2(X_2) \geq g_2(X_2)$

Se satisface la igualdad $\bar{g}_2(X_2) = g_2(X_2)$.

Las condiciones suficientes de Mitten son:

Primera: Si $\left[R_1^* R_2^* \dots ^* R_n\right]$ es una función monótona no decreciente de $r_1, r_2, \dots, r_n$, la posición del operador maximización (minimización) con respecto a $D_1, D_2, \dots, D_n$ puede cambiarse sin ninguna alteración del resultado óptimo.

Segunda: Se refiere a la separabilidad de la función:

$$\left[R_1^* R_2^* \dots ^* R_n\right] = \left[r_1(X_1,D_1)^* r_2(X_2,D_2)^* \dots ^* r_n(X_n,D_n)\right]$$

Entendiendo por separabilidad lo siguiente:

$$\left[R_1^* R_2^* \dots ^* R_n\right] = R_1^* \left[R_2^* R_3^* \dots ^* R_n\right]$$

Bajo estas condiciones las siguientes composiciones de - eficiencias y/o efectividades de un sistema son válidas para la función recursiva obtenida.

a) $* \equiv +$, es decir, $\left[R_1^* R_2^* \ldots {}^* R_n\right] = \left[R_1 + R_2 + \ldots + R_n\right]$

b) $* \equiv \cdot$, es decir, $\left[R_1^* R_2^* \ldots {}^* R_n\right] = \left[R_1 \cdot R_2 \cdot \ldots \cdot R_n\right]$

c) $* \equiv$ maximización o minimización, es decir,

$$\left[R_1^* R_2^* \ldots {}^* R_n\right] = \text{Máx}\left[R_1, R_2, \ldots R_n\right] \text{ ó Mín } \left[R_1, R_2, \ldots R_n\right]$$

d) $* \equiv \circ$, composición.

Analizaremos un ejemplo en el cual el óptimo total es el producto de los óptimos parciales.

Ejemplo (problema de confiabilidad)

Se quiere diseñar un mecanismo electrónico que consiste en tres componentes principales. Las tres componentes se - disponen en serie de tal manera que la falla de una causará la falla del mecanismo completo. La confiabilidad (probabilidad de que no fracase) el mecanismo puede ser mejorado - instalando en paralelo unidades de reserva, en cada componente. Cada uno de estos puede incluir 1, 2 ó 3 unidades en paralelo. El capital total, en miles de quetzales, disponibles para el diseño es 10.

El siguiente cuadro, nos proporciona los datos para la - confiabilidad R_{i,m_i} y costos c_{i,m_i} para el i-ésimo componente ($i=1,2,3,$) dadas m_i unidades en paralelo ($m_i = 1,2,3,$).

m_i \ i	1		2		3	
	R	c	R	c	R	c
1	0.5	2	0.7	3	0.6	1
2	0.7	4	0.8	5	0.8	2
3	0.9	5	0.9	6	0.9	3

El problema es determinar m_i, que maximice la confiabilidad del sistema sin exceder el capital asignado.

La confiabilidad R , de un sistema de N componentes en serie y m_i unidades en paralelo por componente $i(i=1,2,\ldots,N)$ está dada por $R=\prod_{i=1}^{N} R_{i,m_i}$ el problema por tanto será:

$$\text{Maximizar } R = \prod R_{i,m_i}$$

$$\text{Sujeto a } \sum_{i=1}^{N} c_{i,m_i} \leq C$$

donde C es el capital total disponible. (Nótese que la alternativa $m_i = 0$ no tiene sentido en este caso).

La función recursiva está, en este caso, basada en una descomposición multiplicativa.

Sea $g_i(X_i)$ la confiabilidad de las componentes (etapas), donde X_i es el capital asignado a estas i componentes; con $0 \leq X_i \leq C$. Los estados del sistema, por consiguiente, están dados por X_i.

La función recursiva se desarrolla como:

$$g_1(X_1) = \underset{m_1}{\text{Máx}} \left\{R_{1,m_1}\right\}$$

$$0 \leq c_{1,m_1} \leq X_1$$

y

$$g_i(X_i) = \underset{m_i}{\text{Máx}} \left\{R_{i,m_i} \cdot g_{i-1}(X_i - c_{i,m_i})\right\} \quad ; \text{ con } i = 2,3,\ldots,N$$

$$0 \leq c_{i,m_i} \leq X_i$$

Los cálculos tabulares son:

Como cada componente tiene al menos una unidad, X_1, X_2 y X_3 deben satisfacer

$c_{11} \leq X_1 \leq C - c_{21} - c_{31}$; $c_{11} + c_{12} \leq X_2 \leq C - c_{31}$ y $c_{11} + c_{21} + c_{31} \leq X_3 \leq C$.

de donde: $2 \leq X_1 \leq 6$

$5 \leq X_2 \leq 9$

$6 \leq X_3 \leq 10$

Optimización etapa 1

X_1 \ m_1	$g_1(m_1 \mid X_1) = R_{1,m_1}$			óptimo	
	1	2	3		
	R=0.5, c=2	R=0.7, c=4	R=0.9; c=5	$g_1(X_1)$	m_1^*
2	0.5			0.5	1
3	0.5			0.5	1
4	0.5	0.7		0.7	2
5	0.5	0.7	0.9	0.9	3
6	0.5	0.7	0.9	0.9	3

Optimización etapa 2

X_2 \ m_2	$g_2(m_2\|X_2) = R_{2,m_2} \cdot g_1(X_2 - C_{2,m_2})$			óptimo	
	1	2	3	$g_2(X_2)$	m_2^*
	R=0.7 ; C=3	R=0.8 ; C=5	R=0.9 ; C=6		
5	0.7×0.5 = 0.35			0.35	1
6	0.7×0.5 = 0.35			0.35	1
7	0.7×0.7 = 0.49	0.8×0.5 = 0.40		0.49	1
8	0.7×0.9 = 0.63	0.8×0.5 = 0.40	0.9×0.5 = 0.45	0.63	1
9	0.7×0.9 = 0.63	0.8×0.7 = 0.56	0.9×0.5 = 0.45	0.63	1

Optimización etapa 3

X_3 \ m_3	$g_3(m_3\|X_3) = R_{3,m_3} \cdot g_2(X_3 - C_{3,m_3})$			óptimo	
	1	2	3	$g_3(X_3)$	m_3^*
	R=0.6 ; C=1	R=0.8 ; C=2	R=0.9 ; C=3		
6	0.6×0.35 = 0.210			0.210	1
7	0.6×0.35 = 0.210	0.8×0.35 = 0.280		0.280	2
8	0.6×0.49 = 0.294	0.8×0.35 = 0.280	0.9×0.35 = 0.315	0.315	3
9	0.6×0.63 = 0.378	0.8×0.49 = 0.392	0.9×0.35 = 0.315	0.392	2
10	0.6×0.63 = 0.378	0.8×0.63 = 0.504	0.9×0.49 = 0.441	0.504	2

De lo anterior tenemos que la solución óptima es: $m_1^* = 3$

$m_2^* = 1$

$m_3^* = 2$

Con una confiabilidad correspondiente $R^* = 0.504$.

F. Diferentes estructuras de Programación Dinámica

Hemos visto que los programas dinámicos secuenciales, de acuerdo a la forma en que combinan las eficiencias y/o efectividades, estos pueden ser en forma de suma, multiplicación o bien calcular el máximo o mínimo del mismo. Otra clasificación se hace en función de la optimización total. Cuando éstas son positivas (utilidades, rendimiento, etc) la función se maximiza. Si éstas son negativas se minimiza.

En cuanto a la eficiencia y/o efectividad en sí, ésta puede ser una función contínua o discreta. En cuanto al sistema que se está optimizando puede tener restricciones o no tenerlas. En cuanto a la certeza de la función, puede ser determinística o estocástica (probabilística). En cuanto a la manera de resolver el problema, puede ser en sentido de entrada a salida o de salida a entrada.

Se dice que una función de eficiencia y/o efectividad del sistema es discreta, en el contexto de Programación Dinámica, cuando a ésta se le representa por medio de tablas. Cuando se le representa matemáticamente por medio de una función, entonces se dice que el programa dinámico es contínuo.

Nótese que un problema dado, puede incluir cualquiera de las distintas combinaciones de estas diferentes clasificaciones, así por ejemplo:

- Problema discreto, determinista, finito, con suma de eficiencias parciales y método de solución de entrada a salida.

- Problema contínuo, determinista, finito con solución de salida a entrada, etc.

G. El Problema de dimensionalidad en la Programación Dinámica.

Hemos trabajado donde tanto el vector de salida como el de entrada tienen una sola componente. En general estos vectores tienen n componentes. La dificultad de resolver un problema dinámico crece exponencialmente con el número de componentes de los vectores de entrada y salida del sistema.

Supongamos, por ejemplo, que en un estudio de mercado se tienen P quetzales y H horas hombre disponibles para conducir una serie de encuestas. La ciudad se supone dividida en D zonas.

La exactitud de la encuesta en la zona i, denotada por $E_i(P_i,H_i)$ depende del dinero que se eroga y las horas hombre que se utilizan en dicha zona.

Se quiere determinar la cantidad de dinero a gastar y las horas hombre a utilizar en cada zona, para que la exactitud de la encuesta total se maximice.

Las etapas son cada una de las zonas. La entrada para cada etapa está dada por la cantidad de dinero P_i, y las horas-hombre H_i, aun disponibles para esa etapa y las etapas que siguen a ésta. Es decir, el vector de entrada tiene dos componentes (P_i, H_i). La decisión consiste en una asignación de P_i quetzales, $(0 \leq p_i \leq P_i)$ y de h_i horas-hombre $(0 \leq h_i \leq H_i)$. El vector de salida será entonces, $P_i - p_i = P_{i+1}$ quetzales y $H_i - h_i = H_{i+1}$ horas-hombre.

La función recursiva, utilizando el método de salida a entrada, es:

$$g_i(P_i, H_i) = \max_{\substack{0 \leq p_i \leq P_i \\ 0 \leq h_i \leq H_i}} \left\{ E_i(p_i, h_i) + g_{i+1}(P_i - p_i, H_i - h_i) \right\}; \; i = 1, 2, \ldots, n-1$$

con $$g_n(P_n, H_n) = \max_{\substack{0 \leq h_n = H_n \\ 0 \leq p_n = P_n}} \left\{ E_n(P_n, h_n) \right\}$$

Como condición inicial.

La dificultad ahora, es clara, si por ejemplo, tanto P_i como H_i son valores discretos (enteros) que pueden oscilar del 0 al 9 (10 valores en total) y existen 10 zonas en la ciudad, el número total de alternativas diferentes es $10^3 = 1000$.

En general si existen k componentes en un vector de entrada y estos pueden tomar w valores diferentes y existen n etapas, entonces el número total de alternativas a analizar es nw^k.

Ejemplo: (resolución de un programa lineal utilizando Programación Dinámica)

Sea Máx $z = \sum_{i=1}^{n} c_i X_i$

Sujeto a $\sum_{j=1}^{n} a_{ij}X_j \leq b_i$; $i= 1,2,...,m$

$X_j \geq 0$; $j= 1,2,...,n$

Consideramos cada actividad X_j como una etapa. El nivel de la actividad $X_j \geq 0$, representa la decisión en la etapa j. Esta decisión es una variable contínua. La entrada a la etapa j es un vector de disponibilidad de recursos, que tiene m componentes. Este vector en la etapa j, se representa por:

$$H_j = \begin{pmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{mj} \end{pmatrix}$$

El vector de entrada a todo el sistema H_1, tiene como componentes

$$H_1 = \begin{pmatrix} h_{11} \\ h_{21} \\ \vdots \\ h_{m1} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \text{ de donde que } 0 \leq H_j \leq H_1 \ ; \ \forall j$$

La función de transferencia la dan las mismas restricciones del problema lineal, es decir, $h_{ij} - a_{ij}H_j$.

La eficiencia y/o efectividad total, en este caso, es la función objetivo del programa lineal.

La función recursiva, siguiendo un método de solución de salida a entrada es:

$g_k(h_{1k}, h_{2k}, \ldots, h_{mk}) =$

$$\underset{0 \leq a_{ik}X_k \leq h_{ik}}{\text{Máx}} \left\{ c_k X_k + g_{k+1}(h_{1k} - a_{1k}X_k, \ldots, h_{mk} - a_{mk}X_k) \right\}$$

donde $i = 1, \ldots, m$

$k = 1, 2, \ldots, n-1$

con

$$g_n(h_{1n}, h_{2n}, \ldots, h_{mn}) = \underset{0 \leq a_{in}X_n \leq h_{in}}{\text{Máx}} \left\{ c_n X_n \right\} \quad ; \quad i = 1, 2, \ldots, m$$

Como condición inicial, donde se sobre entiende que

$$0 \leq h_{ij} \leq h_{i1} = b_i \quad ; \quad i = 1, 2, \ldots, m$$

Resolución de un problema (ejemplo)

$$\text{Máx } Z = 2X_1 + 5X_2$$

Sujeto a

$$2X_1 + X_2 \leq 430$$
$$2X_2 \leq 460$$
$$X_1 \geq 0; \quad X_2 \geq 0$$

<u>Optimización de una etapa</u>

Como solo hay dos variables, X_1 y X_2, empezamos con la etapa correspondiente a X_2. Tenemos:

$$H_2 = \begin{pmatrix} h_{12} \\ h_{22} \end{pmatrix} \quad ; \quad g_2(h_{12}, h_{22}) = \underset{\substack{0 \leq X_2 \leq h_{12} \\ 0 \leq 2X_2 \leq h_{22}}}{\text{Máx}} \left\{ 5X_2 \right\}$$

Por ser la función $5X_2$ monótona creciente, es decir, que aumenta a medida que X_2 aumenta, el máximo se obtiene ya sea cuando $X_2 = h_{12}$ ó $X_2 = \frac{h_{22}}{2}$. Tratando de no violar las restricciones, vemos cual de las dos es mehor, por lo que tomamos:

$$X_2^* = \text{Mín} \left(h_{12}, \frac{h_{22}}{2}\right)$$

o sea que

$$g_2(h_{12}, h_{22}) = \text{Mín} \left(5h_{12}, \frac{5h_{22}}{2}\right)$$

$$= 5 \text{ Mín} \left(h_{12}, \frac{h_{22}}{2}\right)$$

<u>Optimización de dos etapas</u>

En este caso $H_1 = \begin{pmatrix} h_{11} \\ h_{12} \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 430 \\ 460 \end{pmatrix}$

Por lo que

$$g_1(430,460) = \underset{0 \leq 2X_1 \leq 430}{\text{Máx}} \left\{2X_1 + g_2(430-2X_1, 460-0)\right\}$$

$$= \underset{0 \leq X_1 \leq \frac{430}{2}}{\text{Máx}} \left\{2X_1 + 5 \text{ Mín} \left(430-2X_1, \frac{460}{2}\right)\right\}$$

$$= \underset{0 \leq X_1 \leq 215}{\text{Máx}} \left\{2X_1 + 5 \text{ mín} \left(430-2X_1, 230\right)\right\}$$

De esta última expresión tenemos:

$$g_1(430,460) = \text{Máx} \begin{cases} 2X_1 + 5(230) & ; \text{ si } 0 \leq X_1 \leq 100 \\ 2X_1 + 5(430-2X_1) & ; \text{ si } 100 \leq X_1 \leq 215 \end{cases}$$

De donde:

$$g_1\ (430,460) = \text{Máx} \begin{cases} 2X_1 + 1{,}150 & ; \quad \text{si} \quad 0 \leq X_1 \leq 100 \\ -8X_1 + 2{,}150 & ; \quad \text{si} \quad 100 \leq X_1 \leq 215 \end{cases}$$

La expresión $2X_1 + 1{,}150$, obtiene su máximo para $X_1 = 100$, y dá valor de 1,350.

Para la segunda expresión, el máximo lo da $X_1=100$, generando $-8(100) + 2{,}150 = 1{,}350$.

Por lo tanto $X_1^* = 100$.

Sabemos que la relación que hay entre H_1 y H_2 es:

$$H_2 = \begin{pmatrix} h_{12} \\ h_{22} \end{pmatrix} = \begin{pmatrix} h_{11} - a_{11}X_1^* \\ h_{21} - a_{21}X_1^* \end{pmatrix} = \begin{pmatrix} 430 - 2(100) \\ 460 - 0(100) \end{pmatrix} = \begin{pmatrix} 230 \\ 460 \end{pmatrix}$$

y como $X_2 = \text{Mín}\ (h_{12}, \dfrac{h_{22}}{2})$, tenemos :

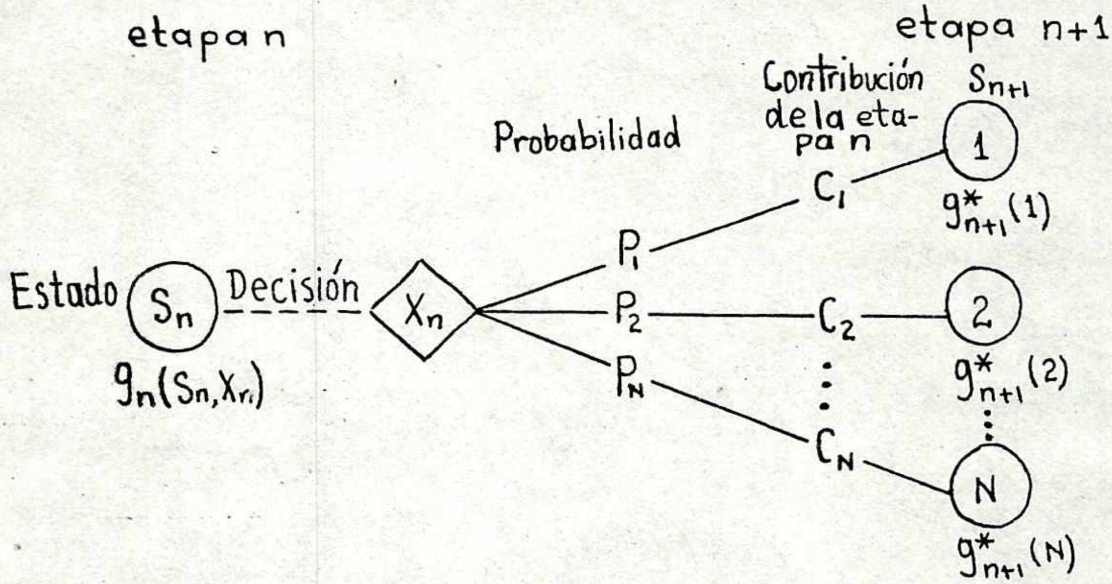
$$X_2^* = \text{Mín}\ (230, \frac{460}{2}) = 230$$

$\therefore$ la solución es $X_1^* = 100$; $X_2^* = 230$; y $Z^* = 1{,}350$.

H. Programación Dinámica Probabilista

La Programación Dinámica Probabilista difiere de la Programación Dinámica Determinista en que el estado en la etapa siguiente no está completamente determinado por el estado y la política de decisión de la etapa anterior. En lugar de - ello hay una distribución de probabilidad para el próximo estado. Sin embargo la distribución de probabilidad está plenamente determinada por el estado y la política de decisión de la etapa anterior.

La estructura básica de la Programación Dinámica Probabilista está dada, gráficamente, por el dibujo siguiente:



donde N es el número de posibles pasos (situaciones) en la etapa n+1; $(p_1, p_2, \ldots, p_N)$ es la distribución de probabilidad para el estado n+1 dada la decisión X_n en el estado S_n ; y c_i es la contribución resultante para la función objetivo de la etapa n si el estado pasa a la situación i.

Si la gráfica anterior es expandida hasta incluir todos los estados y decisiones posibles entonces obtenemos un árbol de decisiones.

Si el árbol de decisiones no es muy grande resulta ser -

una forma útil de resumir las distintas posibilidades que pueden presentarse.

Debido a la estructura probabilista la relación entre $g_n(S_n,X_n)$ y $g^*_{n+1}(S_{n+1})$ es de alguna manera más complicada que para la Programación Dinámica Determinista. La forma precisa de esta relación dependerá de la forma total de la función objetivo.

Para ilustrar supongamos que el objetivo es minimizar la suma esperada de las contribuciones de los estados individuales. En este caso $g_n(S_n,X_n)$ representa la suma mínima esperada desde la etapa n en adelante, dado que el estado y la política de decisión en la etapa n son S_n y X_n respectivamente. Por lo tanto

$$g_n(S_n,X_n) = \sum_{i=1}^{N} p_i \left[c_i + g^*_{n+1}(i)\right]$$

con $$g^*_{n+1}(S_{n+1}) = \underset{X_{n+1}}{\text{Mín}} \; g_{n+1}(S_{n+1}, X_{n+1})$$

donde esta minimización es tomada de los valores factibles de X_{n+1}.

Ejemplo:

Una compañía manufacturera ha recibido una orden para suministrar un artículo de determinado tipo. Pero el cliente ha especificado ciertas condiciones de calidad, por lo que el fabricante tendrá que producir más de un ejemplar para obtener uno que sea aceptable.

El fabricante estima que cada artículo de dicho tipo, producido por él, será aceptable con una probabilidad de $\frac{1}{2}$ y defectuoso (sin posibilidad de rehacerlo) con probabilidad de $\frac{1}{2}$.

Entonces el número de artículos aceptables producidos en un lote de tamaño L tendrá una distribución binomial. Esto es, la probabilidad de producir cero artículos aceptables en el lote es $(\frac{1}{2})^L$.

Los costos de producción marginal para este producto están estimados en Q100 por artículo (aunque sean defectuosos) y el exceso de artículos es innecesario. El costo para montar e iniciar la producción es de Q 300. Si la revisión de un lote, revela que aún no se ha logrado la producción deseada, un costo de Q 300 tendrá que ser agregado. El fabricante solo tiene tiempo para fabricar tres lotes. Si ningún ejemplar aceptable se ha logrado el fabricante tendrá una pérdida entre venta y costos de Q 1,600.

El objetivo es determinar la política a seguir considerando el tamaño del lote para la producción requerida que minimice el costo total esperado por el fabricante.

Solución:

Consideraremos las producciones como las etapas. Las variables de decisión X_n, $(n=1,2,3,)$ son, el tamaño de cada lote en la etapa n.

El número de artículos aceptables por obtener (1 ó 0) -

servirá para describir el estado del sistema en cualquier - etapa. Es decir, el estado S= 1 en la etapa 1.

Si por lo menos un artículo aceptable es obtenido subsecuentemente el estado vuelve o cambia a S = 0, después del cual ningún costo adicional debe sucederse.

Aunque una función complicada puede darse para toda la - función objetivo, es mucho más apropiado definir $g_n(S,X_n)$ - directamente. En este caso $g_n(S,X_n)$ es el costo mínimo total esperado para la etapa n; puesto que el estado y la política de decisión en el estado n son S y X_n respectivamente.

Por lo tanto $$g_n^*(S) = \min_{X_n=0,1,\ldots} g_n(S,X_n)$$

donde $g_n^*(0) = 0$

Usando Q 100 como unidad monetaria, la contribución del costo de la etapa n es $(K+X_n)$, haciendo caso omiso del siguiente estado donde

$$K = \begin{cases} 0 & \text{si} \quad X_n = 0 \\ 3 & \text{si} \quad X_n > 0 \end{cases}$$

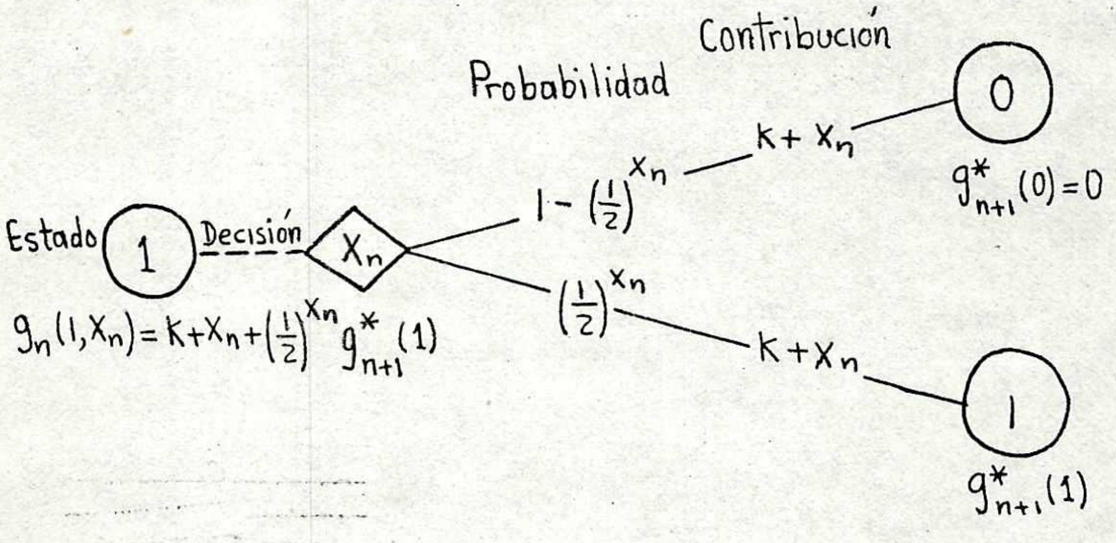
Por lo tanto para S = 1

$$g_n(1,X_n) = K + X_n + \left(\frac{1}{2}\right)^{X_n} g_{n+1}^*(1) + \left[1-\left(\frac{1}{2}\right)^{X_n}\right] g_{n+1}^*(0)$$

$$= K + X_n + \left(\frac{1}{2}\right)^{X_n} g_{n+1}^*(1)$$

donde $g_4^*(1)$ es definido igual a 16, el costo terminal si no se ha llegado a obtener ningún artículo aceptable .

Esta relación básica la podemos representar por la figura:



Por lo tanto la función recursiva para los cálculos es:

$$g_n(1) = \underset{X_n=0,1,\ldots}{\text{Mín}} \quad K + X_n + \frac{1}{2}^{X_n} \; g_{n+1}(1)$$

para $n = 1,2,3.$

<u>Para $n = 3$</u>

X_3 \ S	$g_3(1,X_3)=K+X_3+16\left(\frac{1}{2}\right)^{X_3}$: 0	1	2	3	4	$g^*_3(S)$	X^*_3
0	0					0	0
1	16	12	9	8	8	8	3 ó 4

Para n = 2

$S \backslash X_2$	$g_2(1,X_2)=K+X_2+\left(\frac{1}{2}\right)^{X_1} g_3^*(1)$ 0	1	2	3	4	$g_2^*(S)$	X_2^*
0	0					0	0
1	8	8	7	7	7.5	7	2 ó 3

Para n = 1

$S \backslash X_1$	$g_1(1,X_1)=K+X_1+\left(\frac{1}{2}\right)^{X_2} g_2^*(1)$ 0	1	2	3	4	$g_1^*(S)$	X_1^*
1	7	7.5	6.75	6.875	7.0625	6.75	2

De donde tenemos que la política óptima es producir dos artículos en la primera producción, si ninguno es aceptable entonces fabricar dos o tres artículos en la segunda producción, si ninguno es aceptable se fabricarán tres o cuatro artículos en la tercera producción. El costo total esperado para esta política es Q 675.

IV. ALGUNOS MODELOS

Consideremos ahora algunos modelos de inventario, cuya solución se puede obtener por medio de la Programación Dinámica. Un problema de inventario existe cuando es necesario guardar bienes físicos o mercancías con el propósito de satisfacer la demanda sobre un horizonte de tiempo especificado (finito o infinito). Debemos tomar en cuenta que casi toda empresa debe almacenar bienes para asegurar un trabajo uniforme y eficiente en sus operaciones.

A. Modelo dinámico de un solo artículo y N-períodos.

En este modelo se supone que la demanda, aunque conocida con certeza, puede variar de un período al siguiente. También el nivel de inventario se revisa periódicamente en lugar de contínuamente. No obstante que puede permitirse la demora en la entrega (expresada como un número fijo de períodos) el modelo supone que el almacén se reabastece instantáneamente al inicio del período. Finalmente, no se permite ninguna escasez.

El desarrollo de modelos deterministas dinámicos está limitado al estudio de horizontes de tiempo finito. Esto es así ya que una solución numérica de estos modelos requiere el uso de la técnica de Programación Dinámica, la cual en este caso es factible únicamente para un número finito de períodos (etapas). Esta no es una limitación seria, sin embargo, ya que las demandas futuras distantes usualmente tienen poco efecto sobre las decisiones del horizonte de tiempo finito presente. Además, en la mayoría de las situaciones no es práctico suponer que el artículo se mantendrá en existencia indefinidamente.

Defina para el período $i, i = 1,2,\ldots,N$,

$z_i =$ cantidad ordenada

$\xi_i =$ cantidad demandada

$x_i =$ inventario que entra (al inicio del período i)

$h_i =$ costo de mantenimiento por unidad de llevar adelante el inventario del período i al período i + 1.

$K_i =$ costo fijo

$c_i(z_i) =$ función de costo de compra marginal (producción) dada z_i.

Sea

$$C_i(z_i) = \delta_i K_i + c_i(z_i)$$

donde

$$\delta_i = \begin{cases} 0, & z_i = 0 \\ 1, & z_i > 0 \end{cases}$$

La función $C_i(z_i)$ es de interés únicamente si el costo de compra unitario varía de un período al siguiente o si existen rebajas en los precios.

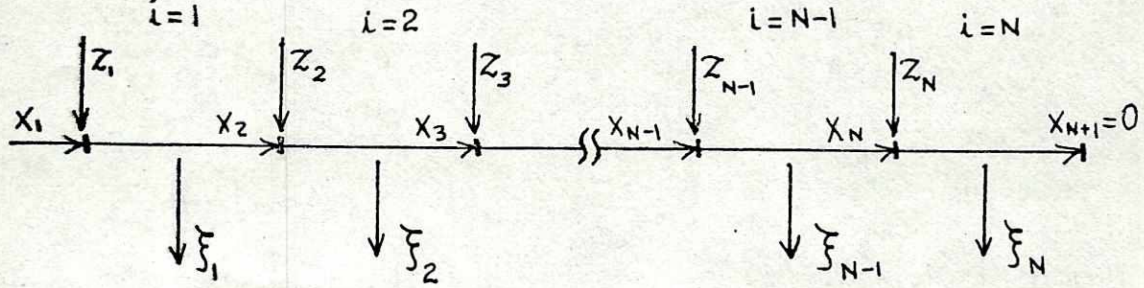
Puesto que no existe excasez permitida, el objetivo es -

determinar los valores óptimos de z_i que minimizen la suma de los costos fijos, de compra y de mantenimiento para todos los N períodos. El costo de mantener el inventario se supone proporcional a:

$$x_{i+1} = x_i + z_i - \xi_i$$

que es la cantidad de inventario que se lleva desde i hasta i+1. Esto significa que el costo de mantener el inventario en el período i es $h_i x_{i+1}$. La hipótesis se introduce únicamente para simplificar ya que el modelo puede extenderse fácilmente hasta cubrir cualquier función de costo de mantenimiento de inventario $H_i(x_{i+1})$ reemplazando $h_i x_{i+1}$ por $H_i(x_{i+1})$. Análogamente, el costo de mantenimiento de inventario puede estar basado en x_i ó $(x_i + x_{i+1})/2$.

El desarrollo del modelo de Programación Dinámica se simplifica representando gráficamente el problema, como:



Cada período representa una etapa. Usando la ecuación - recursiva hacia atrás, se pueden definir los estados del sis tema en la etapa i como la cantidad de inventario que entra x_i.

Sea $g_i(x_i)$ el costo de inventario mínimo para los períodos $i, i+1, \ldots,$ y N. La ecuación recursiva completa está - dada por:

$$g_N(x_N) = \underset{\substack{z_n + x_N = \xi_N \\ z_N \geq 0}}{\text{Mín}} \left\{ C_N(Z_N) \right\}$$

$$g_i(x_i) = \underset{\xi_i \leq x_i + z_i \leq \xi_i + \ldots + \xi_N}{\text{Mín}} \left\{ C_i(Z_i) + h_i(x_i + Z_i - \xi_i) + g_{i+1}(x_i + Z_i - \xi_i) \right\}$$

$$i = 1, 2, \ldots, N-1$$

La ecuación recursiva hacia adelante puede desarrollarse definiendo los estados en la etapa i como la cantidad de inventario al final del período i. En la gráfica anterior, - estos estados son equivalentes a x_{i+1}. En cualquier etapa, los valores de x_{i+1} están limitados por

$$0 \leq x_{i+1} \leq \xi_{i+1} + \ldots + \xi_N$$

Esto significa que, en el caso extremo, la cantidad Z_i en el período $j \leq i$ puede ser ordenada lo suficientemente - grande de manera que el inventario restante x_{i+1} satisfaga la demanda para todos los períodos restantes.

Sea $g_i(x_{i+1})$ el costo del inventario mínimo para los períodos 1,2,..., e i dado x_{i+1}, la cantidad de inventario al final del período i. La ecuación recursiva completa entonces está dada como.

$$g_1(x_2) = \min_{0 \leq z_1 \leq \xi_1 + x_2} \{C_1(Z_1) + h_1 x_2\}$$

$$g_i(x_{i+1}) = \min_{0 \leq z_i \leq \xi_i + x_{i+1}} \{C_i(Z_i) + h_i x_{i+1} + g_{i-1}(X_{i+1} + \xi_i - Z_i)$$

$$i = 2,3,\ldots,N$$

Las formulaciones hacia adelante y hacia atrás del modelo son computacionalmente equivalentes. Esto es así ya que las transformaciones de estado son básicamente las mismas en ambos casos.

El algoritmo hacia adelante, es útil si se desarrolla un caso como el del modelo anterior.

B. <u>Modelo dinámico para programar la producción en N períodos.</u>

Consideremos el problema de programar la producción sobre N períodos sucesivos. Las demandas para los diferentes períodos son variables pero deterministas. Estas demandas pueden satisfacerse ya sea por un inventario que fluctúe mientras se mantiene la producción constante, o fluctuando la producción mientras se mantiene el inventario constante, o por una combinación de ambas. Las fluctuaciones en producción pueden lograrse trabajando tiempo extra, mientras que la fluctuación en inventario puede satisfacerse manteniendo una cantidad positiva en almacén o permitiendo que la demanda que no se satisfaga quede pendiente para llenarse en períodos posteriores. El objetivo aquí es determinar el programa de producción para los N períodos, lo cual minimiza los costos totales relevantes.

Este modelo supone costo fijo cero en cada período. En general, la escasez se permite, excepto que toda la demanda que queda pendiente de un período a los siguientes debe satisfacerse en el N-ésimo período. Esta situación puede representarse como un modelo de transporte. En particular, notando las características especiales del modelo para el caso donde no se permite escasez, el problema puede resolverse en una forma fácil sin tener que aplicar el procedimiento iterativo de la técnica de transporte.

Definamos los siguientes símbolos para el período i; $i=1,...N$.

c_i = costo de producción por unidad durante el tiempo regular.

d_i = costo de producción por unidad durante el tiempo extra, $c_i < d_i$.

h_i = costo de mantener el inventario por unidad desde el período i al período i + 1.

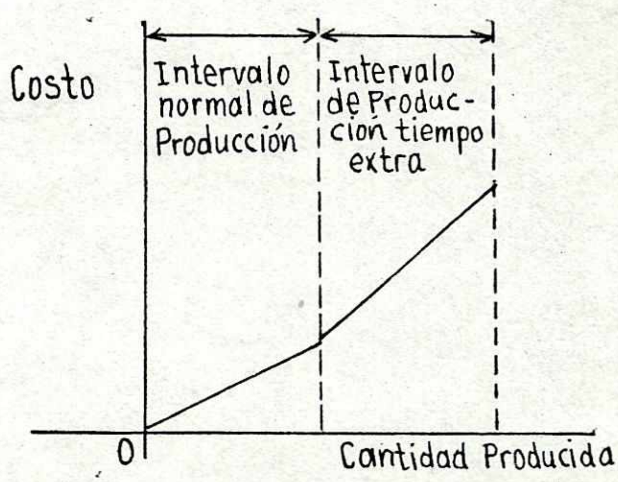
p_i = costo de escasez por unidad demandada en el período i y satisfecha en el período $i + 1$.

a_{Ri} = capacidad de producción (en número de unidades) durante el tiempo regular.

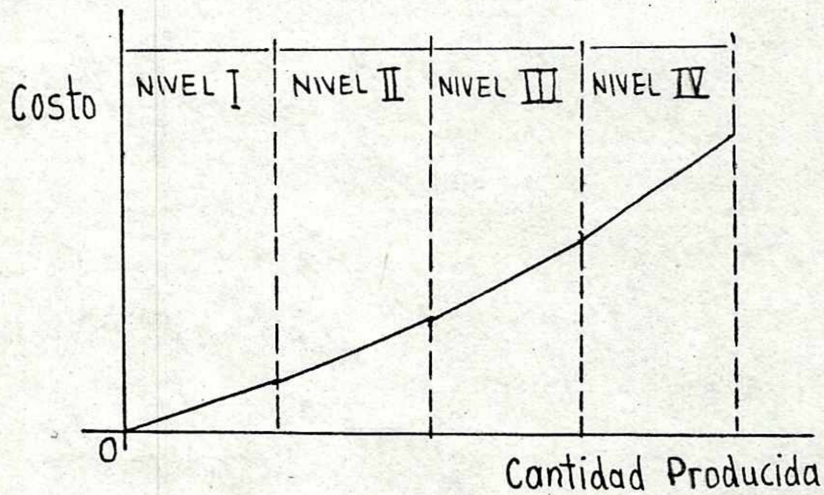
a_{Ti} = capacidad de producción (en número de unidades) durante el tiempo extra.

b_i = demanda (en número de unidades).

Observemos que c_i, el costo de producción por unidad durante el tiempo normal, es menor que d_i, el costo de producción por unidad durante el tiempo extra. Esto se demuestra gráficamente en la figura siguiente.



La situación puede generalizarse al caso en que existen k niveles de producción tales que los costos de producción por unidad aumentan con el nivel de producción. Un ejemplo típico se demuestra en esta figura:



En tales condiciones, la función de costo de producción se dice que tiene costos marginales crecientes. Matemáticamente, se dice que la función es convexa.

La restricción anterior sobre la función de costo de producción debe mantenerse, de otra manera el modelo que sigue no será aplicable. Este punto se justificará posteriormente después que se hayan presentado los detalles del modelo.

i. Modelo sin escasez

Primero, considere el caso donde no se permite la escasez en el sistema. De acuerdo con la terminología del modelo de transporte tenemos : las "fuentes" se represen tan por las producciones normales y de tiempo extra para los diferentes períodos. Los "destinos" están dados por las demandas para los períodos respectivos. El costo uni tario de "transporte" de cualquier origen a cualquier - destino se representa por la producción correspondiente por unidad más los costos de mantener el inventario.

La matriz de costos completa para el modelo de transporte equivalente (suponiendo que no hay escasez) se da en la tabla siguiente:

Período de demanda j

Período de producción i	1	2	3	...	N	Excedente	
R_1	c_1	c_1+h_1	$c_1+h_1+h_2$		$c_1+h_1+\cdots+h_{N-1}$	0	a_{R1}
T_1	d_1	d_1+h_1	$d_1+h_1+h_2$		$d_1+h_1+\cdots+h_{N-1}$	0	a_{T1}
R_2		c_2	c_2+h_2		$c_2+h_2+\cdots+h_{N-1}$	0	a_{R2}
T_2		d_2	d_2+h_2		$d_2+h_2+\cdots+h_{N-1}$	0	a_{T2}
$\vdots$							$\vdots$
R_N					c_N		a_{RN}
T_N					d_N	0	a_{TN}
	b_1	b_2	b_3	...	b_N	S	

La columna de excedente se usa para balancear el modelo de transporte; o sea, $S = \sum_i a_i - \sum_j b_j$. Esto se basa en la hipótesis razonable de que la demanda siempre es menor que - la capacidad de producción del sistema. (El costo por unidad en la columna de excedente es igual a cero.) Ya que no se - permite ninguna escasez el período de producción i no puede emplearse para satisfacer las demandas para los períodos --

1,2,... e i-1. Esto se muestra por los cuadros sombreados en el cuadro anterior, lo cual es equivalente en realidad a - asignar a cada uno un costo unitario muy grande.

Debido a que no se permiten órdenes atrasadas en este modelo, es necesario incluir la restricción de que para cada - período k la cantidad acumulada de demanda hasta ese período inclusive no exceda a la cantidad acumulada correspondiente de producción; esto es,

$$\sum_{i=1}^{k}(a_{Ri} + a_{Ti}) \geq \sum_{j=1}^{k} b_j, \text{ para } k = 1,2,\ldots,N$$

La solución del problema se simplifica mucho por su formulación como un modelo de transporte. Puesto que la demanda en el período i debería estar satisfecha antes que en los períodos i + 1, i+2,..., y N y debido a la condición especial impuesta sobre la función de costo de producción, no será necesario usar el algoritmo de transporte regular al resolver el problema. En lugar de esto, la demanda para el período 1 se satisface primero asignando sucesivamente tanta cantidad como sea posible a los registros más baratos de la primera - columna (período 1).

Los nuevos valores de a_i se actualizan luego para reflejar las capacidades restantes para los períodos diferentes.

Después, se considera el período 2 y su demanda se satisface en la forma más barata posible dentro de las nuevas limitaciones de capacidad. El proceso continúa hasta que se -

satisface la demanda para el período N.

Debido a los costos marginales crecientes en la función de costos de producción, la capacidad de producción regular quedará exhausta antes de que pueda comenzar la producción de tiempo extra. Si esta condición no se satisface, el modelo de transporte no será aplicable, puesto que podría proporcionar resultados sin sentido.

ii. Modelo con escasez

Consideremos ahora una generalización del modelo anterior en el cual se permite la escasez. Se supone que la demanda que no se satisface puede serlo al final del horizonte de N períodos.

Si necesitamos incluir el efecto de dejar órdenes pendientes. Esto se logra introduciendo los costos de "transporte" unitario apropiado en las rutas de los bloques. Por ejemplo, si se define a p_i como el costo de escasez por unidad demandada en el período i y satisfecha en el período i + 1, los costos unitarios de transporte correspondientes a los cuadrados $(R_N, 1)$ y $(T_N, 1)$ son $\left\{c_N+p_1+\dots+p_{N-1}\right\}$ y $\left\{d_N+p_1+\dots+p_{N-1}\right\}$ respectivamente.

Parecería razonable que el procedimiento de solución sin escasez también sería aplicable a la nueva situación donde se permite la misma. Lo cual desafortunadamente no es cierto.

V. CONCLUSIONES

a. La Programación Dinámica es una herramienta importante, para resolver diversidad de problemas donde haya que tomar decisiones, por lo que debiera de formar parte de los pensa de las carreras de Ingeniería, Economía y Administración.

b. La herramienta matemática juega un papel importante en la Programación Dinámica, sin embargo no existen algoritmos de tipo universal para el tratamiento de todos los problemas, por lo que se requiere de una sólida formación en Programación no Dinámica para poder proponer posibles procesos dinámicos de solución.

c. Existen problemas para los cuales es imposible soslayar el uso de métodos de Programación Dinámica tales como los de inventarios sin interrupción de actividad productiva, es conveniente crear algunos algoritmos simples de fácil manejo para su ejecución en este tipo de problemas.

VI. BIBLIOGRAFIA

Ackoff, Russell L.; Saseini, Maurice W. Fundamentos de
1973 Investigación de Operaciones. México. Limusa.

Collatz, L.; Wetterleng, W. Optimization Problems. New
1975 York. Springer-Verlag.

Hillier, Frederick S.; Lieberman, Geral j. Operations
1974 Research. U.S.A. Holden-Day INC.

Kaufmann, A.; Crun, R. Dynamic Programming. New York
1967 Academic Press Inc.

Kaufmann, A,; Faure R. Invitación a la Investigación -
1977 de Operaciones. México. C.E.C.S.A.

Prawda W., Juan. Métodos y Modelos de Investigación de
1979 Operaciones. Vol. 1 Modelos Determinísticos.
México. Limusa.

Prawda W., Juan. Métodos y Modelos de Investigación de
1980 Operaciones. Vol 2 Modelos Estocásticos. -
México. Limusa.

Shamblin, James E.; Stevens G.T. Investigación de Ope-
1979 raciones. Un Enfoque Fundamental. México.
McGraw-Hill.

Taha, Hamdy A. Investigación de Operaciones. Una In-
1981 troducción. México. Representaciones y Ser-
vicios de Ingeniería S.A.

Thierauf, Robert J.; Grosse, Richard A. Toma de Deci-
1981 siones por medio de Investigación de Operacio-
nes. México. Limusa.