4.12 Cálculo de una fórmula de colocación lineal para aplicar el procedimiento de momentos estadísticos.
Sergio Radrigán V. 4.183.

4.13 Estimación de las características de un embalse subterráneo.
Fernando Mimica S. y Gastón Galleguillos B. 4.205.

4.14 Generación estocástica de series cíclicas.
Patricio Ferrer A. y Ricardo Seguel G. 4.227.

4.15 Análisis de la consistencia de estadísticas pluviométricas.
Gerardo Ahumada T. 4.243.

4.16 Discriminación de situaciones de crecidas y recesión con fines de previsión.
Bonifacio Fernández y Eduardo Varas. 4.151.

4. HIDROLOGIA Y RECURSOS HIDRAULICOS

SOCIEDAD CHILENA DE INGENIERIA HIDRAULICA

IV COLOQUIO NACIONAL

CONSIDERACIONES SOBRE LA APLICACION DEL METODO DE PROGRAMACION DINAMICA EN EL ESTUDIO DE OPERACION DE SISTEMAS DE EMBALSES

Ernesto Brown F. (1) Rodrigo Soler B. (2)

RESUMEN

La técnica de optimización llamada programación dinámica, presenta características que la hacen teóricamente atractiva para ser aplicada en el estudio y optimización de la operación de un embalse o de un sistema de embalses. En la bibliografía existen numerosas referencias en las que esta técnica ha sido aplicada con el objetivo señalado. Existen sin embargo algunas limitaciones para que la aplicación generalizada de la técnica, en la resolución de problemas de la práctica, resulte atractiva. Estas limitaciones se refieren al planteamiento de una función objetivo adecuada, a la discretización de los valores de las variables involucradas y fundamentalmente, al gran volúmen de valores y cálculos numéricos, que en algunos casos hace inmanejable un problema por limitaciones en la capacidad de memoria de los computadores y/o por el excesivo tiempo de computación, lo que se traduce en un excesivo costo. Con respecto al último problema señalado, se han propuesto algunas modificaciones a la técnica de programación dinámica, que permiten ampliar la gama de problemas que es factible resolver por este método; estas modificaciones presentan interés en los problemas de operación de sistemas de embalses. Sin embargo, las técnicas modificatorias a la programación dinámica clásica, presentan a su vez limitaciones para su aplicación generalizada.

El objetivo de este trabajo, es examinar algunos aspectos de la aplicabilidad de las técnicas señaladas. El trabajo corresponde a una etapa de avance parcial, en una de las líneas de investigación que actualmente está realizándose en el Centro de Recursos Hidráulicos de la Universidad de Chile.

---

(1) Ingeniero Civil, Investigador Centro de Recursos Hidráulicos, Departamento de Obras Civiles, Universidad de Chile.

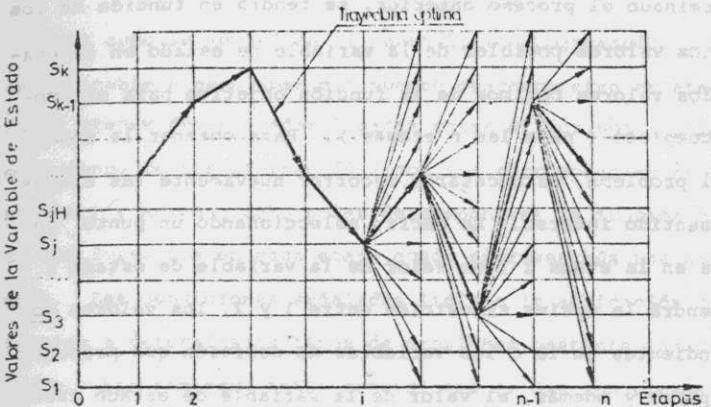
(2) Ayudante Investigador, Centro de Recursos Hidráulicos, Departamento de Obras Civiles, Universidad de Chile

1.- FUNDAMENTOS DE LA PROGRAMACION DINAMICA

La técnica de programación dinámica es aplicable a problemas de optimización que puedan descomponerse en una serie de etapas secuenciales. En cada etapa se realiza una transformación de una o más variables de estado, de acuerdo a un conjunto de ecuaciones y restricciones que a su vez llevan incorporadas una o más variables de decisión; además, existe para cada etapa una función cuantificadora de la efectividad de la transformación, que es función de los valores que tomen la o las variables de decisión y la o las variables de estado, en dicha etapa. Es requisito fundamental del método que la función objetivo para el problema completo, pueda formarse por una composición adecuada de las funciones cuantificadoras de efectividad de cada una de las etapas (por ejemplo por suma de los valores de esta función). El principio básico del método es el llamado principio de optimalidad, que puede enunciarse de la siguiente manera (8) : " una política óptima tiene la propiedad que cualquiera sean la decisión y estado iniciales, las restantes decisiones deben constituir una política óptima con respecto al estado resultante de la primera decisión ".

Lo anterior señala que, tal como se ilustra en la figura 1, para un problema que tenga sólo una variable de estado por etapa, el método de programación dinámica permite determinar una trayectoria óptima a través de un conjunto de etapas y esto a su vez determina el valor óptimo de la variable de estado en cada etapa y también, el valor óptimo de la o las variables de decisión por etapa.

Para aplicar el principio de optimalidad enunciado anteriormente, el problema se resuelve secuencialmente considerando las transiciones entre etapas a partir de uno de los extremos. Por ejemplo, partiendo de la última etapa, se considerarán todas las posibles transiciones entre la etapa n-1 y la etapa n, para cada uno de los valores posibles de la variable de estado (S_j en



F I G. 1

la figura 1) en la etapa n-1; como función de cada valor posible de S_j en esta etapa, se guardará el valor de la o las variables de decisión que generen la transición óptima entre n-1 y n, es decir, aquella que optimice el valor de la función cuantificadora de efectividad definida para la citada transición. Enseguida, se considerarán las transiciones entre la etapa n-2 y n-1, para cada uno de los valores posibles de la variable de estado S_j en la etapa n-2; en este caso, la óptima transición para cada S_j se determinará de acuerdo al óptimo de la función objetivo conformada por la composición de dos términos : aquel debido a la transición entre n-2 y n-1 y aquel que representa al óptimo trayecto desde n-1 en adelante. Terminado el proceso indicado, para la etapa n-2 se tendrá, como función de cada valor posible de la variable de estado, el óptimo valor de la función objetivo desde esta etapa en adelante. El procedimiento esquematizado se continuará en forma análoga hasta llegar a la etapa 1, para la que se considerarán todas las posibles transiciones entre esta etapa y la etapa 2, como función de los valores posibles de la variable de estado en la etapa 1; la función objetivo en este caso considerará la componente debida a la transición entre las etapas 1 y 2 y la componente que corresponde al óptimo de la etapa 2 en adelante, como función de la variable de estado en la etapa 2.

Terminado el proceso anterior, se tendrá en función de los distintos valores posibles de la variable de estado en la etapa 1, los valores óptimos de la función objetivo para el problema completo (para las n etapas). Para obtener la solución al problema, es necesario recorrer nuevamente las etapas en el sentido inverso. Es decir, seleccionando un punto de partida en la etapa 1 (un valor de la variable de estado) se obtendrá la óptima transición entre 1 y 2, los valores correspondientes de la o las variables de decisión que producen este óptimo y además, el valor de la variable de estado resultante para la etapa 2; a partir de este último valor, se determinará la óptima transición entre las etapas 2 y 3, los valores de la o las variables de decisión correspondientes y el valor de la variable de estado resultante para la etapa 3. Este proceso debe continuarse hasta la etapa n, para obtener la trayectoria óptima para el problema completo, tal como se indica esquemáticamente en la figura 1. La trayectoria óptima así determinada, fija una secuencia de valores de la o las variables de decisión, y una secuencia de valores de la variable de estado.

Es importante hacer notar, que la resolución del problema en la forma señalada en los párrafos precedentes, implica lo siguiente :

- Calcular en cada etapa todas las posibles transiciones a la etapa siguiente.
- Almacenar en memoria en cada etapa, en función de todos los valores posibles de la variable de estado en dicha etapa, los valores óptimos de la función objetivo que representa a la trayectoria óptima desde esa etapa en adelante y además, almacenar como función de la variable de estado, el valor de la o las variables de decisión asociadas con la transición entre esa etapa y la siguiente, que generan la trayectoria óptima.

Tal como se señaló en el punto 1, la programación dinámica es aplicable a problemas que puedan descomponerse en etapas; estas etapas deben quedar ligadas entre sí por una (o más) ecuaciones de transformación (con sus correspondientes restricciones) y además, debe poder descomponerse la función objetivo de tal forma que en cada etapa quede representada una parte de esta. Las condiciones señaladas limitan la aplicación de esta técnica a determinados tipos de problemas bastante característicos. Sin embargo, estas condiciones se dan con cierta frecuencia en problemas de análisis de sistemas de recursos hidráulicos, particularmente, en el caso de problemas de operación de embalses. Siempre y cuando la función objetivo pueda descomponerse en la forma señalada, la programación dinámica tiene la gran ventaja sobre otras técnicas de optimización, que esta función objetivo puede tener prácticamente cualquier forma (lineal, no lineal, discreta, etc.).

En aquellos problemas en que es posible plantear la resolución por programación dinámica, es conveniente tener presente algunas consideraciones adicionales.

En primer lugar, en los problemas prácticos de interés en la ingeniería, es necesario siempre discretizar a un conjunto finito de valores, tanto las variables de estado como las variablesade decisión por etapa. Es cierto que esta discretización puede hacerse con una malla todo lo fina que se desee, pero esto contribuye a aumentar los requerimientos de memoria y los requerimientos de tiempo de computación, que se discuten más adelante. Por otra parte esta discretización implica que en la mayoría de los problemas será necesario recurrir a interpolaciones en cada cálculo de cada etapa.

El segundo aspecto que es indispensable tener en cuenta, se refiere a las necesidades de memoria y tiempos de computación involucrados en un problema de programación dinámica. Estos pueden crecer tan rápidamente en función de la discretización

de las variables, número de variables de estado, número de variables de decisión y etapas del problema, que pueden llegar a impedir la resolución de este, aún cuando teóricamente sea factible plantear su resolución por programación dinámica, debido a las limitantes de disponibilidad de capacidad de memoria del computador, o a que el tiempo de computación requerido, y consecuentemente el costo de computación, resulte excesivo frente a los beneficios que se estimen para los resultados del análisis.

Con el objeto de ilustrar la última situación planteada, se considera el siguiente problema. Supóngase un problema de programación dinámica con una variable de estado y una variable de decisión por etapa; supóngase que la variable de estado en cada etapa se ha discretizado a 100 valores; finalmente, supóngase que el problema se ha dividido en 360 etapas. Este problema es similar al que se plantearía, tal como se visualizará más adelante, en el caso de la operación de un embalse de un sólo propósito, cuando se analiza mensualmente, con una estadística de 30 años de gastos medios mensuales; dado que se han tomado 100 valores discretos para la variable de estado, podría corresponder a un caso de un embalse de 200×10^6 m^3 de capacidad, discretizada a intervalos de 2×10^6 m^3.

En el problema planteado había que almacenar en memoria, en cada etapa, por lo menos, un valor de la variable de decisión para cada valor posible de la variable de estado. Si esto se considera para cada una de las etapas, el total de valores a almacenar en memoria, asciende a 36.000. A esta capacidad de memoria es preciso agregar algunos requerimientos menores para los cálculos parciales necesarios en cada etapa. Además, en algunos tipos de problemas de programación dinámica también resulta indispensable mantener en memoria los óptimos de la función objetivo, para cada etapa y para cada valor discreto de la variable de estado; si esto fuera necesario para nuestro problema, la capacidad de memoria se duplicaría a 72.000

valores, lo que implica 288 K bytes en términos de capacidad de memoria del computador. Finalmente, a esto hay que agregar las necesidades de memoria que se requieren para el programa, compilador y otras funciones especiales.

Las necesidades de capacidad de memoria analizadas en el párrafo precedente, son perfectamente compatibles con las disponibilidades de los computadores corrientes en la actualidad. Sin embargo, la situación cambia sustancialmente, si se complica levemente el problema base. En efecto, considérese ahora un problema con dos variables de estado y dos variables de decisión por etapa; nuevamente, cada variable de estado discretizada a 100 valores y considerando 360 etapas. Este problema podría originarse en el análisis de la operación conjunta de un sistema de dos embalses, en que las variables de estado corresponden al almacenamiento en cada embalse al inicio de cada intervalo de tiempo, y las variables de decisión son las entregas desde cada embalse en cada intervalo de tiempo. Al igual que en el problema base, se estaría examinando la operación del sistema a lo largo de un período de 30 años de estadísticas de gastos medios mensuales. En este caso, habría que almacenar en memoria en cada etapa, por lo menos un valor para cada una de las variables de decisión, para cada par de valores de la variable de estado; esto significa $100 \cdot 100 = 10^4$ valores. Si esto lo ampliamos a las 360 etapas, obtenemos $3{,}6 \times 10^6$ valores, que es un número que ya excede la capacidad de memoria de acceso rápido, de los computadores corrientes actuales. En consecuencia, ya en un caso que es levemente más complejo que el problema base, habría que entrar a recurrir al uso de unidades periféricas para almacenar información, con las consecuentes ineficiencias en términos de los tiempos de computación involucrados.

Con respecto a los tiempos de proceso, llamando : t_e al tiempo de cálculo por estado y por etapa, para encontrar la decisión óptima, N_t al número de etapas y N_e al número de va

lores discretos de la variable de estado, el tiempo total de proceso para el problema base, sería de :

$$T = t_e \cdot N_e \cdot N_t$$

De acuerdo a la experiencia obtenida en este trabajo, en un computador IBM 370/145, t_e puede estimarse en 0,02 seg. Por lo tanto, para el problema base :

$$T = 0{,}02 \cdot 100 \cdot 360 \approx 12 \text{ min.}$$

Lo anterior significa del orden de $ 4.000 (*) para una partición de 320 K de memoria rápida.

Considerando ahora el problema de dos variables de estado y dos variables de decisión por etapa, el tiempo total sería de :

$$T = t_e \cdot (N_e)^2 \cdot N_t$$

Suponiendo que t_e se mantiene igual, lo que podría lograrse con un buen método de búsqueda de las decisiones óptimas para cada estado, y además, siempre que el computador tuviese capacidad de memoria rápida suficiente, lo que no es efectivo, según se analizó anteriormente, el tiempo total de proceso sería:

$$T = 0{,}02 \cdot (100)^2 \cdot 360 \approx 20 \text{ hrs.}$$

Traduciendo a costos el tiempo de 20 horas, significa del orden de $ 850.000.- considerando una ocupación total de la capacidad de memoria rápida del computador (1024 K para el IBM 370/145). En la práctica, para este problema sería absolutamente indispensable transferir información a unidades periféricas (cintas magnéticas o discos) e ir recuperando esta información en la medida que se necesita; en estas condiciones, el tiempo de proceso sería mayor, aún cuando quizás, se podría trabajar con una partición menor en cuanto a memoria rápida se refiere.

(*) Valores en $ de Abril de 1979.

Los ejemplos planteados ilustran las dificultades de aplicar en la práctica, el método tradicional de programación dinámica, cuando se trata de problemas que requieren ser descritos por más de una variable de estado y más de una variable de decisión por etapa.

3.- MODIFICACIONES A LOS PLANTEAMIENTOS DEL METODO TRADICIONAL DE PROGRAMACION DINAMICA

Existen en la bibliografía algunos procedimientos modificatorios a la programación dinámica tradicional (1, 3, 5, 8, 9), cuyo principal objetivo es el solucionar los problemas, presentados en el punto anterior, de grandes requerimientos de capacidades de memoria y grandes tiempos de proceso, que se plantean en la resolución de modelos de programación dinámica.

En este trabajo se examinan dos procedimientos alternativos para el efecto señalado. El primero de ellos consiste en un método de aproximaciones sucesivas ; en este método se descompone el problema original de programación dinámica con n variables de estado y n variables de decisión por etapa, en n subproblemas de una variable de estado y una variable de decisión por etapa, cada uno de ellos. Estos subproblemas se ligan entre sí con un número adecuado de ecuaciones que permitan mantener las características del problema (1,9). Estos subproblemas se van resolviendo en forma secuencial; se inicia el proceso optimizando el primer subproblema por programación dinámica tradicional, manteniendo fijas las ecuaciones que ligan a este con los demás subproblemas y manteniendo fijas las decisiones en los otros subproblemas ; enseguida se toma el segundo subproblema y así se continúa hasta el subproblema n; luego se vuelve a iniciar el proceso, con el primer subproblema, etc. En cada aproximación se van dejando fijas las variables de decisión determinadas por los subproblemas anteriores. El esquema de resolución señalado implica ir "mejorando" sucesivamente la solución, y tiene incorporado un criterio de término, consistente en parar las iteraciones cuando no se observa una modificación sustancial de la solución, entre u

na iteración y la anterior. En un planteamiento como el recién descrito, el aumento de tiempo de computación es lineal en función del número de variables de estado por etapa, en cada iteración, en lugar de ser exponencial como en la programación dinámica tradicional.

Si consideramos el problema ejemplo planteado en el punto 2, con dos variables de estado y dos variables de decisión por etapa, y 360 etapas, el tiempo de computación por iteración podría estimarse como :

$$T_i = t_e \cdot 2N_e \cdot N_t$$

Suponiendo $t_e = 0{,}02$ seg, igual que en los casos anteriores, resulta :

$$T_i = 0{,}02 \cdot 2 \cdot 100 \cdot 360 = 24 \text{ min.}$$

Lo anterior indica que, si llegar al resultado óptimo requiere menos de 50 iteraciones, el método resultaría favorable frente a la programación dinámica tradicional, aún sin considerar los menores requerimientos en cuanto a capacidad de memoria (en este caso se requiere una capacidad de memoria similar a la que se necesita en un problema de una variable de estado y una variable de decisión por etapa).

El segundo procedimiento que se examina en este trabajo es un método por estados incrementales. Este método consiste, al igual que el anterior, en aproximarse a la solución óptima al problema por iteraciones sucesivas. En este caso se trabaja con todas las variables de decisión y de estado por etapa, pero considerando en cada iteración un número restringido de valores para las variables de estado y consecuentemente un número restringido de valores para las variables de decisión. El método requiere iniciar el proceso con una solución inicial factible; luego se busca una solución "mejor" que la inicial, sólo dentro del entorno de ella; la segunda iteración se inicia con la solución mejorada, buscando en el entorno de esta nueva solución; el procedimiento continúa hasta que en alguna

iteración no se observe mejoría con respecto a la solución inicial de esa iteración. En la figura 2 se ilustra este procedimiento para un problema de sólo una variable de estado y una variable de decisión por etapa; la ilustración corresponde al espacio de busqueda que se considera en una iteración cualquiera, en el entorno de la solución inicial de esa iteración.

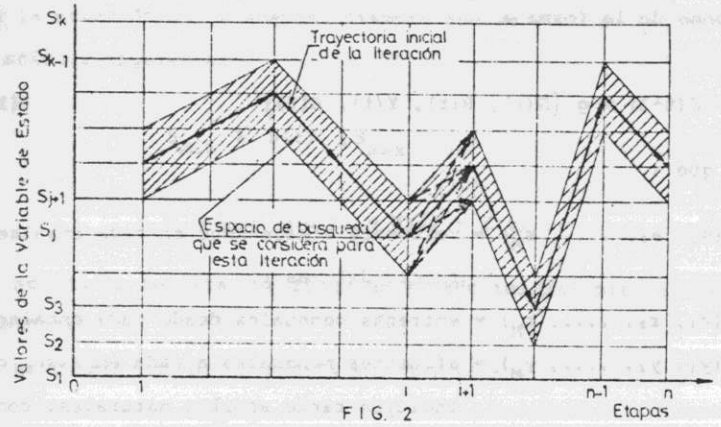


FIG. 2

De la figura 2 se observa que entre las etapas i e i+1 se consideran, en este método, sólo ° transiciones posibles, en cada iteración.

Haciendo referencia ahora, al mismo problema ejemplo señalado en el punto 2, con dos variables de estado y dos variables de decisión por etapa, y en base a los tiempos de computación que se han experimentado en este trabajo, se ha estimado para este problema, en la modalidad de resolución por el método de estados incrementales, un tiempo de computación de 0,518 seg/etapa. De acuerdo con este valor, el tiempo total por iteración alcanzaría a 3,11 min. En estas condiciones, este método resultaría favorable frente a la programación dinámica tradicional, siempre que el resultado óptimo de alcance en menos de 385 iteraciones, sin siquiera considerar los menores requerimientos de capacidad de memoria de acceso rápido, que en este caso resultan ser mínimos.

Sin embargo, cabe señalar que el tiempo de computación en el método por estados incrementales, aumenta exponencialmente con el número de variables (de estado y de decisión) que es necesario manejar.

4.- PLANTEAMIENTO GENERAL POR PROGRAMACION DINAMICA, PARA EL ANALISIS DE OPERACION DE UN SISTEMA DE EMBALSES

Si se considera un conjunto de M embalses interrelacionados entre sí tanto en serie como en paralelo, el flujo de agua entre ellos puede en general describirse por un sistema de ecuaciones de la forma :

$$S(t+1) = g\,[S(t),\ R(t),\ Y(t),\ E(t)] \qquad (1)$$

en que :

$S=(s_1, s_2, \ldots, s_M)$ = volúmenes almacenados en cada embalse al inicio del mes t.

$R=(r_1, r_2, \ldots, r_M)$ = entregas mensuales desde cada embalse.

$Y=(y_1, y_2, \ldots, y_M)$ = afluentes mensuales a cada embalse, que incluyen tanto aportes naturales, como entregas desde embalses ubicados aguas arriba.

$E=(e_1, e_2, \ldots, e_M)$ = pérdidas por evaporación mensual desde los embalses.

t = índice que señala un intervalo de tiempo mensual.

$g[\cdot]$ = funcional vectorial, en general M-dimensional.

El problema de operación, se plantea tanto en el caso del diseño de uno o más embalses del sistema, diseño que queda condicionado por una política de operación óptima y metas de entrega desde cada embalse, así como también, en el caso que se desee determinar políticas de operación y metas de entrega para un sistema de embalses existentes, con cualquier fin de uso del agua. En cualquier de los dos casos señalados, para cada embalse del sistema se tendrá una capacidad máxima $s_{máx_i}$ (que podría variarse paramétricamente en un problema de diseño) y una capacidad mínima aprovechable s_{min_i}. Generalmente el análisis del problema se hará considerando un cierto número de años de esta

dística afluente a los embalses, en términos de gastos medios mensuales; sea N el número de períodos mensuales considerados. En estas condiciones la política de operación óptima, será una secuencia de decisiones de entrega R(1), R(2), ..., R(N) que optimice una función objetivo adecuada, y tales que se cumpla la ecuación (1) y además, siempre que se cumplan las restricciones siguientes :

$$S_{min} \le S(t) \le S_{máx} \qquad (2)$$

$$0 \le R(t) \le S(t) + Y(t) - E(t) \qquad (3)$$

Además se tendría que fijar un estado inicial para el sistema S(0).

La función objetivo podría ser del tipo :

$$\text{máx} \sum_{t=1}^{N} B_t\,[S(t),\ R(t)] \qquad (4)$$

En este caso se estaría suponiendo que el máximo beneficio de la operación del sistema es equivalente a la suma de los beneficios obtenidos en cada intervalo de tiempo, y que estos a su vez, son una función de la entrega R(t) realizada en ese intervalo y del estado de almacenamiento S(t) al inicio del mismo intervalo.

Otro tipo de función objetivo que se ha utilizado en algunos casos (5,9) con el objeto de maximizar la entrega de agua garantizada de un sistema de embalses, es del siguiente tipo :

$$\text{máx}\ \left\{ \min_t \left[\frac{\sum_{i=1}^{M} r_i(t)}{\alpha_j} \right] \right\} \qquad (5)$$

en que α_j es la fracción de la demanda total anual que debe entregarse en el mes j y por lo tanto j es un índice que varía entre 1 y 12.

Las implicancias de utilizar un tipo de función objetivo u otro son variadas, y meritan un análisis y discusión más am -

plios, que quedan fuera del alcance del presente trabajo.

Suponiendo que se utilice la función objetivo representada por la ecuación (4), y considerando descrito el problema con el sistema de relaciones (1), (2) y (3), la formulación por programación dinámica es muy directa. En efecto, la variable S es una variable de estado (M-dimensional) y la variable R es una variable de decisión (M-dimensional); por otra parte, el sistema de ecuaciones (1), (2) y (3), describe las transiciones de la variable S desde la etapa t a la etapa t+1 (etapas en función del tiempo, considerando intervalos mensuales). Las ecuaciones de recurrencia de programación dinámica pueden escribirse como :

$$f_1[S(1)] = \max_{R(1)} \{B_1[S(1), R(1)]\} \qquad (6)$$

y $$f_{i+1}[S(i+1)] = \max_{R(i+1)} \{B_{i+1}[S(i+1), R(i+1)] + f_i[S(i)]\} \qquad (7)$$

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

Las ecuaciones (6) y (7), en conjunto con las relaciones (1), (2) y (3) representan un problema de programación dinámica de N etapas, con M variables de estado y M variables de decisión por etapa.

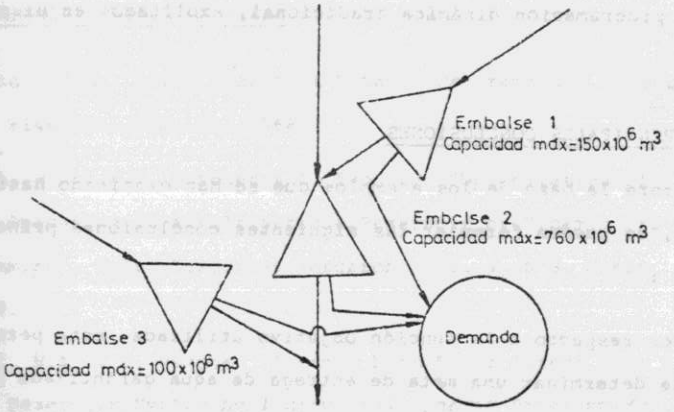
Esta formulación es bastante general y conceptualmente sería aplicable a cualquier problema en que se pretende analizar la operación de un sistema de embalses. Los problemas prácticos para aplicarla han sido señalados en el punto 2 de este trabajo.

5.- EJEMPLOS ANALIZADOS EN ESTE TRABAJO

El procedimiento seguido en este trabajo, se inició con el análisis de la operación de un ambalse, para un sólo propósito. Este análisis se realizó utilizando el método de programación dinámica tradicional, y además los dos procedimientos modificatorios señalados en el punto 3; se utilizó una

función objetivo del tipo (5). Esta etapa tuvo como objetivo lograr una familiarización con los métodos señalados y cuantificar las necesidades de tiempo y capacidades de memoria de computador requeridas.

En una segunda fase se abordó un problema ejemplo tal como el esquematizado en la figura 3. Es decir, un sistema de tres embalses, con las capacidades señaladas en la figura, para abastecer un centro de demanda. La demanda se definió con una distribución mensual típica de riego. El problema se describió mediante tres variables de estado por etapa (los estados de almacenamiento al comienzo de cada mes en cada embalse) y cuatro variables de decisión por etapa (las entregas mensuales desde cada embalse y además una posible entrega de agua desde el embalse 1 al embalse 2). En general se ha operado con distinto número de etapas (meses) y distinto número de valores discretos para cada variable. La función objetivo para este ejemplo toma la forma :



F I G. 3

$$f_k(S_{1k},S_{2k},S_{3k}) = \max_{r_{1k},r_{2k},r_{3k},r_{4k}} \{\min[(\frac{r_{1k}}{\alpha_{1j}} + \frac{r_{2k}}{\alpha_{2j}} + \frac{r_{3k}}{\alpha_{3j}}), f_{k-1}]\} \quad (8)$$

en que :

f_k : valor de la función objetivo en la etapa k, como función de las variables de estado S_{1k}, S_{2k}, S_{3k}

r_{1k}, r_{2k}, r_{3k} : entregas de los embalses 1, 2 y 3, en el período k, para satisfacer la demanda

r_{4k} : entrega desde el embalse 1 al embalse 2, en el período k

$\alpha_{1j}, \alpha_{2j}, \alpha_{3j}$: es la fracción de la demanda total anual que debe entregarse en el mes j desde cada uno de los embalses 1, 2 y 3 (j = 1, 2, ..., 12)

Además se completa la descripción del problema, con las ecuaciones de continuidad y las restricciones del tipo (2) y (3) señaladas en el punto 4.

Este ejemplo se abordó por los dos métodos modificatorios de la programación dinámica tradicional, explicados en el punto 3.

6.- PRINCIPALES CONCLUSIONES

Sobre la base de los ejemplos que se han examinado hasta ahora, se pueden formular las siguientes conclusiones principales :

- Con respecto a la función objetivo utilizada, esta permite determinar una meta de entrega de agua garantizada para el sistema, pero no es muy flexible para determinar reglas de operación que consideren entregas que se desvíen de esa meta en algunos períodos.
- El método de aproximaciones sucesivas, de acuerdo a la experimentación con el ejemplo diseñado, requiere de muy pocas iteraciones para alcanzar una política de operación

óptima. Además no manifiesta una dependencia significativa sobre la política inicial de partida.
- El método de estados incrementales, ha requerido un número de iteraciones relativamente grande para alcanzar la política de operación óptima; este método es altamente dependiente sobre la política inicial.
  Para el caso del ejemplo (7 variables de estado y decisión) el tiempo de computación observado fue de 14 seg/etapa; esto en la práctica implica que para este número de variables, el método no sería factible debido al costo de computación.
- El método de estados incrementales trabaja exclusivamente con los valores de las variables de estado definidos con la malla inicial. En este sentido esta malla debe definirse con el grado de aproximación con que se deseen los resultados finales. En el método de aproximaciones sucesivas la malla inicial puede ser más "gruesa", puesto que el método requiere realizar interpolaciones entre los puntos de la malla.

REFERENCIAS BIBLIOGRAFICAS

1.- Larson, R.E. "State Increment Dynamic Programming". American Elsevier Pub. Co. 1968.

2.- Hall, W.A., W.S. Butcher & A. Esogbue. "Optimization of the Operation of a Multipurpose Reservoir by Dynamic Programming". Water Resources Research, Vol. 4, N°3, Junio 1968.

3.- Hall, W.A. & R. Harboe. "Optimun Firm Power Output from a Two Reservoir System by Incremental Dynamic Programming". Water Resources Center, U. de California, Contribution N°130, 1969.

4.- Harboe, R., F. Mobasheri & W.W.-G. Yeh. "Optimal Policy for Reservoir Operation". Journal of the Hydraulic Division A.S.C.E. Nov. 1970

5.- Harboe, R. & F. Mobasheri. "Optimum Firm Water and Power Supply from Linked Surface Reservoirs". Informe Inédito, U. de California, Feb. 1971.

6.- Mechin, Y. & D. Normand. "Exemples d'application de la Gestion Optimale des Réservoirs a buts Multiples". La Houille Blanche, N°2/3, 1977.

7.- Moreno, O. y A. Matte. "Determinación de la Explotación Optima de un Embalse de Uso Múltiple". VII Congreso Latinoamericano de Hidráulica, A.I.I.H., Santiago, Chile, Oct. 1976.

8.- Nemhauser, G.L. "Introduction to Dynamic Programming". John Wiley & Sons, Inc. 1966.

9.- Trott, W.J. & W.W-G. Yeh. "Multi-level Optimization of a Reservoir System". A.S.C.E. Annual & National Environmental Engineering Meeting, St. Louis, Missouri, U.S.A. Oct. 1971.

SOCIEDAD CHILENA DE INGENIERIA HIDRAULICA
IV COLOQUIO NACIONAL

METODO SINTETICO PARA MAXIMIZACION DE CRECIDAS (1)

RICARDO SEGUEL G. (*)
LUDWIG STOWHAS B. (**)

RESUMEN

En este trabajo se propone una metodología que permite, dado el Hidrograma Unitario de una cuenca y la precipitación máxima en 24 hrs., determinar la distribución temporal o hietograma de la tormenta que maximiza la crecida de diseño, haciendo uso de la llamada fórmula de Grunsky o cualquier otro método sintético para la estimación de curvas intensidad-duración.

---

(1) La metodología presentada en este trabajo, fue desarrollada en el "Estudio Integral de Riego de los Valles Aconcagua, Putaendo, Ligua, Petorca, ejecutado por CICA Ingenieros Consultores para la Comisión Nacional de Riego.

(*) Ingeniero Civil (U.S.M.)
(**) Ingeniero Civil (U.de Ch.)