A P E N D I C E A

Obtención de la subrutina del algoritmo

La subrutina del algoritmo está basada en programación dinámica. Esta es una técnica matemática útil para la toma de decisiones interrelacionadas, como lo es este caso. Proporciona un procedimiento sistemático para determinar la combinación de decisiones que maximiza la efectividad total.

Son ocho las características básicas que distinguen a los problemas de programación dinámica:

1.- <u>El problema se puede dividir en etapas que requieren una política de decisión en cada una de ellas.</u>

En este algoritmo, el problema se divide en operaciones o etapas. La política de decisión de cada etapa es el destino que tendrá el trabajo en proceso (WIP) en la siguiente operación. De manera parecida, otros problemas de programación dinámica requieren de la toma de una serie de decisiones interrelacionadas, en donde cada decisión corresponde a una etapa del problema.

2.- <u>Cada etapa tiene un cierto número de estados asociados a ella.</u>

Los estados asociados en cada etapa del problema en este caso, son las diferentes máquinas que conforman una operación. Cada máquina determinará un estado diferente, ya que se tendrán condiciones distintas para resolver el problema. En general, los estados son las distintas condiciones posibles en las que se puede encontrar el sistema en cada etapa del problema. El número de estados puede ser en ciertos casos finito o infinito.

3.- El efecto de la política de decisión en cada etapa es transformar el estado actual en un estado asociado con la siguiente etapa.
La decisión del trabajo en proceso en cuanto al siguiente destino lo conduce de su estado actual al siguiente estado. Este procedimiento sugiere que los problemas de programación dinámica se pueden interpretar en términos de redes, donde cada nodo corresponde a un estado. La red consiste en columnas de nodos, en donde cada columna corresponde a una etapa, en forma tal que el flujo de un nodo sólo puede ir a un nodo de la siguiente columna a la derecha. El valor asignado a cada liga que conecta a dos nodos, se interpreta como la contribución a la función objetivo que se obtiene al ir de un estado al siguiente que corresponde a estos nodos. En este caso, el objetivo es encontrar la ruta más corta a través de la red.

4.- El procedimiento de solución está diseñado para encontrar una política óptima para el problema completo. Esto es decir, obtener una solución para las decisiones de la política óptima en cada etapa para cada uno de los estados posibles.

5.- <u>Dado el estado actual, una política óptima para las etapas restantes es independiente de la política adoptada en etapas anteriores.</u>

Dado el estado en el que se encuentra el trabajo en proceso, la ruta más corta que disminuya el tiempo de flujo de este lugar en adelante es independiente de cómo este trabajo en proceso llegó ahí. En general, en los problemas de programación dinámica, el conocimiento del estado actual del sistema expresa todo la información sobre su comportamiento anterior, y esta información es necesaria para determinar la política óptima de ahí en adelante. Esta propiedad es la propiedad Markoviana, y a veces se hace referencia a esta como el principio de optimalidad para programación dinámica.

6.- <u>El procedimiento de solución se inicia al encontrar la política óptima para la última etapa.</u>
La política óptima para la última etapa prescribe la política óptima de decisión para cada estado posible en esa etapa. Es común que la decisión de este problema de una etapa sea obvia.

7.- <u>Se dispone de una relación recursiva que identifica la política óptima para la etapa *Op*, dada la política óptima para la etapa (Op + 1).</u>
Para este algoritmo, la relación recursiva que se utiliza es:

$$f^{*}_{op}(m) = \min\{LIGA_{m,m} + f^{*}_{op+10}(m)\}$$

Etapa – operación ✍ op

Decisión óptima ✍ x^*_{op}

Estado – máquina ✍ m

Para encontrar la política óptima de decisión cuando se comienza en el estado *m* de la etapa *op* se necesita encontrar el valor x_{op} que dé un mínimo. La ruta mínima correspondiente se obtiene si se usa este valor de x_{op} y después se sigue la política óptima cuando el proceso se encuentra en el estado x_{op} de la etapa (*op* +1).

Sea x_{op} la variable de decisión de la etapa u operación *Op* (*Op* = 10, 20, 30, ... , *N*). Sea $f_{op}(m, x_{op})$ el valor de la función objetivo que debe minimizarse, dado que el sistema se encuentre en el estado *m* de la etapa *Op* y se elige x_{op}. Sea $f^*_{op}(m)$ el valor mínimo de $f_{op}(m, x_{op})$ sobre todos los valores posibles de x_{op}. La relación recursiva siempre tendrá la forma

$$f^*_{op}(\text{maq}) = \min\{ f_{op}(\text{maq}, x_{op})\}$$

en donde $f_{op}(maq, x_{op})$ se escribe en términos de *m*, x_{op}, $f^*_{op+10}(.)$ y una medida de eficiencia de x_{op} para la primera etapa.

8.- <u>Cuando se usa esta relación recursiva, el procedimiento de solución se mueve hacia atrás etapa por etapa, hasta que encuentra la política óptima en la etapa inicial.</u>

Para la solución del algoritmo, se obtiene una tabla como la siguiente para cada etapa (*op* = *N, N-10, ...,* *10*)

maq	$f^{*}_{op}(maq)$	x^{*}_{op}

Cuando se obtiene esta tabla para la etapa inicial (op = 10), el problema queda resuelto. Como se conoce el estado de la etapa inicial, la primera decisión está especificada por x^{*}_{10} en esta tabla. El valor óptimo de las otras variables de decisión queda a su vez especificado por las otras tablas de acuerdo al estado del sistema que resulta de la decisión anterior.