

EL PROBLEMA DE LA MOCHILA

Mochila continua (con fraccionamiento).

Es la versión más fácil de resolver desde el punto de vista computacional puesto que un (simple) algoritmo voraz encuentra la solución óptima siempre que se escoja el criterio de selección apropiado (en este caso el del valor específico).

Divide y Vencerás (DV) NO se puede aplicar, pues la casuística de llamadas recursivas sería infinita (no se cumplen las condiciones de aplicación).

Por un motivo parecido, tampoco puede aplicarse un método de búsqueda y enumeración (como vuelta atrás (VA) o ramificación y poda (RP)) en los que tampoco se cumplen las condiciones de aplicación.

Si no se puede aplicar DV, tampoco se puede aplicar programación dinámica (PD).

Por lo tanto, esta versión del problema tiene solución voraz y es la MEJOR.

Mochila Discreta (sin fraccionamiento).

La versión general de este problema es la que no impone ninguna restricción en los pesos de los objetos ni en la capacidad máxima de la mochila.

Según esto se han hecho dos distinciones:

→ Mochila discreta con pesos discretos (para suavizar la dificultad del problema se exige que los pesos - capacidad mochila y objetos- no sean cantidades que pertenecen a un dominio continuo).

En este caso, DV se puede aplicar, pero el resultado es un algoritmo muy poco eficiente, con una complejidad temporal exponencial con el número de objetos a considerar (inviable en la mayoría de casos).

La técnica PD, en sus versiones memoización o iterativa, reduce esa complejidad exponencial en polinómica, por lo tanto, es la MÁS EFICIENTE para resolverlo.

El voraz se puede aplicar, pero no será capaz de encontrar la solución óptima, solo un subóptimo, puesto que no se conoce ningún criterio de selección que lo consiga ante cualquier instancia del problema.

Aun así, PD resulta poco eficiente en cuanto a complejidad espacial y temporal si los pesos son cantidades muy dispersas (el tamaño de los almacenes puede llegar a ser demasiado grande como para poder mantenerlos en memoria y el tiempo necesario para cumplimentarlos excesivo, a pesar de que la complejidad sea cuadrática).

VA y RP se pueden aplicar (si ocurre lo anterior puede que sea la única alternativa). El uso de cotas pesimistas, optimistas y también de estrategias de búsqueda puede ayudar a suavizar la dificultad computacional de este problema. Las heurísticas voraces son imprescindibles para reducir el espacio de búsqueda si el número de objetos a tratar es elevado.

→ Mochila discreta sin restricciones en los pesos (pueden ser continuos o discretos).

DV se puede aplicar (el algoritmo sería exactamente el mismo que el de antes) pero la complejidad resultante sería exponencial.

PD no se puede aplicar pues no es posible indexar tablas con valores continuos. La versión recursiva (memoización) no resulta eficaz pues la probabilidad de que aparezcan problemas repetidos es prácticamente nula (el algoritmo sería igual de malo que el de DV, en realidad sería algo peor al añadir el manejo del almacén).

El voraz se puede aplicar, pero no será capaz de encontrar la solución óptima, solo un subóptimo, puesto que no se conoce ningún criterio de selección que lo consiga ante cualquier instancia del problema. (he pegado lo de antes :)

VA y RP son las ÚNICAS TÉCNICAS que pueden aplicarse si lo que se desea es la solución óptima. El uso de cotas pesimistas, optimistas y también de estrategias de búsqueda puede ayudar a suavizar la dificultad computacional de este problema. Las heurísticas voraces son imprescindibles si el número de objetos a tratar es elevado. (vuelvo a pegar).

*** **

En resumen, el problema (general) de la mochila discreta, es decir, el que no hace concesiones en cuanto al dominio de los pesos, es un problema (muy) difícil desde el punto de vista computacional. La única solución que se le conoce es exponencial. Los mejores algoritmos que se conocen están basados en la técnica Ramificación y poda.