

Ejercicios NP

Alberto Benavides
26 de febrero de 2020

Knapsack problem o **problema de la mochila** es un problema en que, dados n elementos con pesos $\omega_j, j \in [1, \dots, n]$ y valores $c_j, j \in [1, \dots, n]$ junto a un entero B , se desea obtener un subconjunto de dichos elementos cuya suma de pesos sea como máximo B y la suma de sus valores sea, a su vez, la máxima.

El problema de decisión correspondiente al problema de la mochila consiste en determinar si existe (o no) un conjunto de soluciones cuya suma de pesos esté por encima de un entero $y < B$ definido y siga siendo menor o igual a B .

Ahora se procede a mostrar que el problema de la mochila pertenece a NP. Los problemas NP son aquellos que son verificables en tiempo polinomial, es decir en tiempo $\mathcal{O}(n^k)$ donde n es el número de elementos y k es una constante entera. Que un problema sea verificable quiere decir que existe un algoritmo capaz de verificar una solución dada para dicho problema. El problema de la mochila requiere tiempo $\mathcal{O}(n)$ para elegir un subconjunto de elementos asignando pertenencia o ausencia de cada uno (n) de ellos y, de manera simultánea, sumar los pesos de los que pertenezcan al subconjunto para determinar si se encuentran en el rango mayor a y y menor o igual a B .

Por otro lado, un problema NP-completo es aquel que no puede ser resuelto en tiempo polinomial pero que, dada una solución, ésta puede ser comprobada en tiempo polinomial. Que el problema de la mochila sea resuelto en tiempo polinomial dada una solución es evidente dado que bastaría con comprobar justamente lo declarado en el párrafo anterior, a saber, que la suma de los pesos de los elementos incluidos se encuentre en el umbral definido $(y, B]$. Por otro lado, encontrar el subconjunto que resuelva el problema de la mochila requeriría un tiempo $\mathcal{O}(2^n)$ dado que habría que comprobar todos los subconjuntos cuya suma de pesos se encuentre en el rango $(y, B]$ y cada vez que el valor de un subconjunto sea superior a un valor $V = 0$, actualizar V y almacenar en

un subconjunto S ese subconjunto. De esta manera, tras recorrer todos los posibles subconjuntos, podría determinarse el mayor.

En el problema de **programación de tareas** (o *task scheduling problem* en inglés) se tiene un conjunto de n tareas cada una con tiempos de procesamiento $p_j, j \in [1, \dots, n]$, ganancias $w_j, j \in [1, \dots, n]$ y tiempos límites $d_j, j \in [1, \dots, n]$. Este problema es una variante del problema de la mochila en la que las ganancias equivalen al valor definido para aquellos elementos y el tamaño de la mochila B corresponde a cada tiempo límite d_j . La diferencia aquí es que se quiere superar una ganancia W , pero el tiempo de procesamiento para encontrar estas programaciones de tareas sigue requiriendo un tiempo $\mathcal{O}(2^n)$, por lo que se trata de un problema NP-completo.