

Howto Knapsack problems 0-1

Risoluzione con l' algoritmo branch and bound.

1. Ordino la lista degli oggetti per P_j/W_j decrescente
2. Creo il nodo 0
3. Calcolo il suo Upper Bound come somma dei primi elementi il cui peso rientra nella capacità massima del problema. Questo è il valore massimo che posso trovare all'interno del mio grafico: se lo trovo mi fermo perché ho l'ottimo, altrimenti continuo a esplorare. Se la somma dei pesi $<$ capacità totale, calcolo l'ultimo elemento come: $\lfloor \text{cap_residua} * P_j/W_j \rfloor$
4. Creo il nodo figlio di sinistra (taken), con $x_j=1$ e lo scrivo sull'arco
5. Calcolo l'Upper Bound
6. Per tutti gli elementi che ho preso non frazionati continuo dal punto 4
7. Se la capacità rimasta mi permette di prendere solo una parte dell'oggetto, pongo l'ultimo oggetto $=0$ e provo a vedere utilizzando l'oggetto successivo
8. Scrivo nell'arco $x_{j-1} = 0$
9. Calcolo il nuovo Upper Bound con $x_{j-1}=0$ e $x_j=1$
10. Se l'Upper Bound è minore a quello trovato nel caso $x_{j-1}=1$ e per calcolarlo ho usato tutti gli oggetti rimanenti, allora faccio back-tracking perché non posso ottenere valori migliori
11. Altrimenti continuo dal punto 6 finché non termino tutti gli oggetti in lista e giungo alla soluzione con solo oggetti presi "pienamente". Calcolo z come peso degli oggetti presi
12. Se $z < U_0$ faccio back-tracking e parto a espandere dal primo nodo che posso sdoppiare considerando l'oggetto come non preso, creando il nodo di destra, con $x_j=0$ scritto sull'arco (punto 5)
13. Quando tutto l'albero è stato controllato, definisco come ottimo il valore di Z maggiore.
14. Scrivo il vettore x di oggetti presi: es. $x=(1,0,0,1)$

Ricordati di scrivere su tutti gli archi se l'oggetto i -esimo è taken ($= 1$) o not taken ($= 0$).

Se il testo lo richiede, ricordati di sottolineare sul grafo gli Upper Bound che sono stati ricalcolati perché contengono una combinazione diversa di oggetti taken e not taken.

Risoluzione con la programmazione dinamica

1. Ordino la lista degli oggetti per P_j/W_j decrescente
2. Comincio col creare M_0 che è l'insieme che non prende alcun oggetto: $M_0 = \{((0),0,0)\}$
3. Vado a capo e riportando tutti gli oggetti dell'insieme precedente, inserisco l'oggetto i -esimo ($i =$ pedice di M , ovvero M_1 , aggiungo l'oggetto 1)
 - a) Non devo inserire gli oggetti, o i gruppi di oggetti il cui peso supera la capacità del knapsack
4. Controllo che non ci siano stati dominati, altrimenti li cancello
 - a) Si definiscono stati dominati quelli il cui peso è maggiore o uguale al peso di altri stati e il profitto è minore o uguale.
5. Riprendo dal punto 3 finché non ho finito gli oggetti
6. Al termine scrivo $z =$ e il profitto maggiore che ho trovato
7. Determino gli oggetti che sono stati presi con $x = (...)$ inserendo 1 se ho preso l'oggetto i -esimo, 0 se non l'ho preso, separando le cifre con una virgola

La cosa più difficile è stare ben attenti a non perdere stati dominati, perché potrebbe facilmente scapparne uno che proveniva dallo stato precedente (e nello stato precedente non era dominato, mentre ora ho trovato un gruppo di elementi con profitto maggiore e peso minore).