

Zaino 0-1 con backtracking

A.A. 2016-2017

Backtracking: zaino

Zaino 0-1 : Dato uno zaino di “capacità” C ed n articoli O_1, \dots, O_n , ciascuno con un “peso” p_i ed un “valore” v_i , riempire lo zaino in modo da massimizzare il valore degli oggetti inseriti.

Usiamo

- un vettore S come vettore in cui si costruisce della soluzione,
- una vettore Sol come la soluzione migliore
- una variabile $Vmax$ con il valore in ogni momento durante la visita. $Vmax$ e` una variabile globale alla quale possiamo dare come valore iniziale $-\infty$.

Come funzione bounding

- per *inserire* un oggetto nello zaino (potare i sottoalberi sinistri) si puo` richiedere che il peso degli oggetti già nello zaino, sommato al peso dell'oggetto considerato, non superi la capacità dello zaino;
- per *scartare* un oggetto (potare i sottoalberi destri) si può cercare un limite superiore al valore raggiungibile non considerando l'oggetto stesso.

Backtracking: zaino

Per definire la funzione bounding possiamo sfruttare l'ipotesi che i vettori peso e valore siano *ordinati in modo non crescente* rispetto a v_i/p_i ?

L'algoritmo **BOUND** determina un upper bound sulla migliore soluzione ottenibile espandendo il nodo X nell'albero degli stati, trattando gli oggetti come "continui".

```
BOUND (vv, pp, k, C)
  val ← vv
  peso ← pp
  for i ← k+1 to n
    if (peso + P[i] < C)
      peso ← peso + P[i]
      val ← val + V[i]
    else return (val + (|C – peso|)/P[i] · V[i])
  return (val)
```

Backtracking: zaino

Sono considerate globali P, V, C, n, Vmax

```
Zaino (k, S, valore, peso, Sol)
  if (k > n and valore > Vmax)  \\ stampa una soluzione
    for i ← 1 to n
      Sol[i] ← S[i]
      Vmax ← valore
    else
      if (peso + P[k]) <= C)
        S[k] ← 1
        Zaino (k+1, S, valore+V[k], peso+P[k], Sol)
      y ← BOUND (valore, peso, k, C)
      if (y > Vmax)
        S[k] ← 0
        Zaino(k+1, S, valore, peso, Sol)
    return
```

Chiamata esterna: *Zaino* (1, S, 0, 0, Sol)

Nota: è una ricerca in profondità con una funzione di bounding

Backtracking: zaino (esempio)

V = 11, 21, 31, 33, 43, 53, 55, 65

P = 1, 11, 21, 23, 33, 43, 45, 55

M = 110 n=8

