PTAS et sac-à-dos

12 octobre 2021 12:56

PTAS : polynomial time approximation scheme

Schéma d'approx en temps poly.

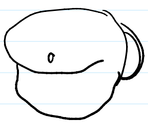
Un algo d'approx. qui reçoit en param une valeur $\varepsilon > 0$ tel que

- $APP \geq (1-\varepsilon) \cdot OPT$ (si pb maximisation)
- $APP \leq (1+\varepsilon) \cdot OPT$ (si pb minimisation)

et qui roule en temps polynomial si on traite ε comme une constante.

$O(\frac{1}{\varepsilon} \cdot n)$ $\quad O(2^{2^{2^{2^{1/\varepsilon}}}} \cdot n^2)$ $\quad O(n^{10/\varepsilon})$

SAC-A-DOS (knapsack)



But : remplir le sac avec des objets de val. max sans dépasser la capacité

objets  $W = 16$

poids	3	8	4	12
valeur	10	20	4	12

OPT = 34 (3 premiers objets)

p	8	8	9
v	7	7	9

$\leftarrow W = 16$ glouton = non

Entrée : capacité $W \in \mathbb{N}$, ens. d'objets paires poids/valeur
$$R = \{(w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)\}$$

Sortie : $R' \subseteq R$ t.q. $\sum_{(w_i, v_i) \in R'} w_i \leq W$ qui

maximise $\sum_{(w_i, v_i) \in R'} v_i$.

Algo exact de prog. dynamique par fct de récurrence

$\hookrightarrow$ qui retourne tjrs une sol. opt.

- Soit $v_{max} = \max\{v_1, v_2, \ldots, v_n\}$
- Observations : $OPT \leq n \cdot v_{max}$
  $OPT \geq v_{max}$
- Pour $i \in \{1, 2, \ldots, n\}$ et $p \in \{1, 2, \ldots, n \cdot v_{max}\}$, on définit :

- Pour $i \in \{1, 2, \dots, n\}$ et $p \in \{1, 2, \dots, n \cdot v_{max}\}$,

on définit :

$B(i,p)$ = poids minimum possible pour avoir un profit de p en choisissant un sous-ensemble de $\{(w_1, v_1), \dots, (w_i, v_i)\}$, ou ∞ si impossible

$$B(i,p) = \min_{\{R' \subseteq \{(w_1,v_1),\dots,(w_i,v_i)\} : \sum_{(w_j,v_j) \in R'} v_j = p\}} \left(\sum_{(w_j,v_j) \in R'} w_j \right)$$

On cherche le p maximum tel que $B(n,p) \leq W$

On définit $B(i,p)$ avec une récurrence

Base : $B(0,p) = \begin{cases} 0 & \text{si } p = 0 \\ \infty & \text{sinon} \end{cases}$

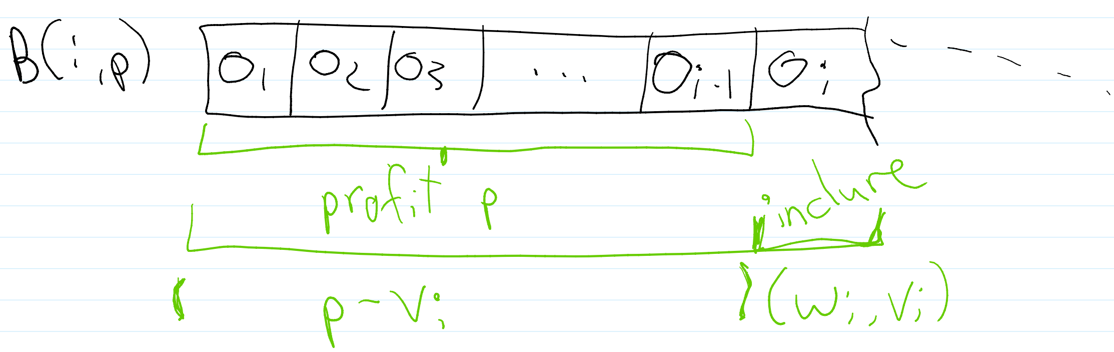
$B(i, 0) = 0 \quad \forall i$

Récurrence :

$$B(i,p) = \min \begin{cases} B(i-1, p) \\ B(i-1, p - v_i) + w_i \end{cases}$$

atteindre profit p avec les $i-1$ premiers elts

atteindre p en incluant le i-ème elt



exactSacADos$(W, \{(w_1,v_1),\dots,(w_n,v_n)\})$

B = table de dimension $n \times (n \cdot v_{max})$

// case de base de B

$n \cdot V_{max}$ | n

pour $i = 1..n$

  pour $p = 1..n \cdot V_{max}$

    $B[i,p] = \min(B[i-1,p], B[i-1,p-v_i] + w_i)$

maxp = 0

pour $p = 1..n \cdot V_{max}$

  si $B[n,p] \leq W$

    maxp = p

return maxp

$O(n^2 V_{max})$

Cet algo roule en temps pseudo-polynomial.
Sa complexité dépend des valeurs numériques de l'entrée et non de leur quantité (n).

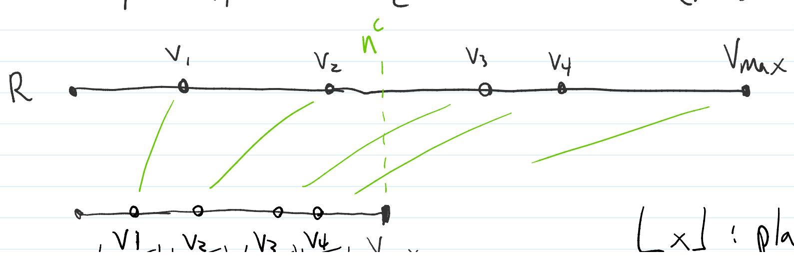
ex: $R = \{(2, 1000000), (1, 20000000)\}$

$O(n^2 v_{max}) \rightarrow 2^2 \cdot 20000000$

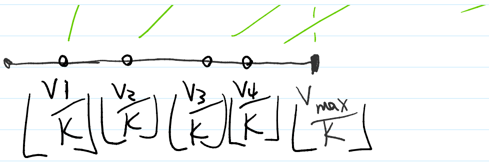
Thm: il existe un algo exact pour SAC-A-DOS en temps $O(n^2 \cdot v_{max})$.

On veut un algo polynomial qui est une $(1-\varepsilon)$-approx, pour tout $\varepsilon > 0$ donné.

Idée: diviser tous les v_i par un certain facteur K pour que $\max\{v_i/K\}$ soit $\in O(n^c)$ constante c



$\lfloor x \rfloor$: plancher de x



$\lfloor x \rfloor$: plancher de x

Problème: l'algo exact ne fonctionne qu'avec des v_i entiers. On prend les planchers.

On perd de la précision avec ces planchers.

approxADos$(W, \{(w_1,v_1),\dots,(w_n,v_n)\}, K)$
  pour $i = 1..n$
    $v_i' = \lfloor \frac{v_i}{K} \rfloor$
  p = exactSacADos$(W, \{(w_1,v_1'),(w_2,v_2'),\dots,(w_n,v_n')\})$
  return p

Soit $X \subseteq R$ la solution optimale de l'algo exact avec les (w_i, v_i').

On a $APP = \sum_{(w_i,v_i)\in X} v_i$

Soit $X^* \subseteq R$ la sol. optimale par rapport aux (w_i, v_i)

On a $OPT = \sum_{(w_i,v_i)\in X^*} v_i$.

<u>Lemme</u>: puisque $v_i' = \lfloor \frac{v_i}{K} \rfloor$, on a

$$v_i/K - 1 \leq v_i' \leq v_i/K$$

En particulier, $v_i \leq K(v_i'+1)$ et $v_i \geq K v_i'$

(1) Donc, $OPT = \sum_{(w_i,v_i)\in X^*} v_i \leq \sum_{(v_i,w_i)\in X^*} K(v_i'+1) = K|X^*| + K \sum_{(v_i,w_i)\in X^*} v_i'$

$$\leq K \cdot n + K \sum_{(v_i, w_i) \in X^*} v_i'$$

② $$APP = \sum_{(w_i, v_i) \in X} v_i \geq \sum_{(w_i, v_i) \in X} K v_i' = K \sum_{(w_i, v_i) \in X} v_i'$$

$$\geq K \sum_{(w_i, v_i) \in X^*} v_i'$$ car X est optimal sur les v_i'

$$OPT \leq Kn + \sum_{(w_i, v_i) \in X^*} v_i' \leq Kn + APP$$

si $K = \frac{V_{max}}{n} \cdot \varepsilon$, on obtient

$$OPT \leq \frac{V_{max}}{\cancel{n}} \cdot \varepsilon \cdot \cancel{n} + APP = \varepsilon V_{max} + APP$$

$$\Rightarrow APP \geq OPT - \varepsilon V_{max}$$ Rappel: $OPT \geq V_{max}$

$$\geq OPT - \varepsilon OPT = OPT(1 - \varepsilon)$$