Algorithmique Avancée
Approximation garantie – Le sac-à-dos

Denis TRYSTRAM
Notes de cours ENSIMAG Alternants 2A

Oct. 2023

Résumé des épisodes précédents...

- On cible la résolution de problèmes d'optimisation d'un certain objectif (discret) sous contraintes.
- Nous savons caractériser la difficulté de tels problèmes.
- Nous avons également étudié plusieurs techniques de conception d'algorithmes.
  Pour les problèmes difficiles, viser la solution exacte n'est pas toujours possible pour les instances les plus grandes.
- La question du jour est :
  Peut-on concevoir des heuristiques[1] *dont la qualité des solutions vis-à-vis de l'optimum est garantie ?*
- Analyse d'un exemple : Le sac-à-dos

[1] par opposition aux méthodes exactes, implicites ou non.

Un exemple préliminaire

Rappel du problème du sac-à-dos.

- KnapSack
- **Entrées** :
  Un ensemble S de n couples de nombres entiers (w_i, b_i)
  Un entier K (capacité du sac)
- **Question** : Remplir le sac en maximisant le bénéfice total

  c'est-à-dire, sélectionner des objets dans S tels que $\Sigma_{i \in S} w_i \leq K$

Une instance de KnapSack

Un sac de capacité 8 et 7 objets (poids,bénéfice)

index	1	2	3	4	5	6	7
	(2,5)	(3,8)	(6,11)	(4,9)	(5,12)	(8,17)	(4,8)

- Une borne supérieure naturelle d'un sac-à-dos partiellement rempli est son bénéfice plus le bénéfice restant que l'on obtiendrait si on remplissait le reste avec le meilleur bénéfice possible.

On introduit la densité de l'objet i comme $\lambda_i = \frac{b_i}{w_i}$

Densité

index	1	2	3	4	5	6	7
	(2,5)	(3,8)	(6,11)	(4,9)	(5,12)	(8,17)	(4,8)
densité	2,50	2,67	1,83	2,25	2,40	2,12	2,00

On trie les objets par densité décroissante

index	2	1	5	4	6	7	3
	(3,8)	(2,5)	(5,12)	(4,9)	(8,17)	(4,8)	(6,11)

- Mettre les objets dans cet ordre l'un après l'autre tant qu'ils rentrent dans le sac.
- L'heuristique *Best Density* (BD) est simple, peu coûteuse (linéaire en le nombre d'objets une fois que le tri a été fait).

Application de l'heuristique à l'instance

index	2	1	5	4	6	7	3
	(3,8)	(2,5)	(5,12)	(4,9)	(8,17)	(4,8)	(6,11)

- On sélectionne le premier objet :
  le bénéfice est de 8 et il reste une capacité de $8 - 3 = 5$

Application de l'heuristique à l'instance

index	2	1	5	4	6	7	3
	(3,8)	(2,5)	(5,12)	(4,9)	(8,17)	(4,8)	(6,11)

- On sélectionne le premier objet :
  le bénéfice est de 8 et il reste une capacité de $8 - 3 = 5$
- On sélectionne le second objet :
  bénéfice $8 + 5 = 13$, il reste une capacité $5 - 2 = 3$

Application de l'heuristique à l'instance

index	2	1	5	4	6	7	3
	(3,8)	(2,5)	(5,12)	(4,9)	(8,17)	(4,8)	(6,11)

- On sélectionne le premier objet :
  le bénéfice est de 8 et il reste une capacité de $8 - 3 = 5$
- On sélectionne le second objet :
  bénéfice $8 + 5 = 13$, il reste une capacité $5 - 2 = 3$
- Aucun autre objet ne peut rentrer dans le sac sans remettre en cause un objet déjà placé...

Cette solution n'est clairement pas la meilleure possible...
On aurait par exemple pu mettre l'objet 6 dans le sac pour un bénéfice de 17.

Peut-on évaluer la qualité de cette heuristique ?

- Cette question porte sur l'ensemble de toutes les instances possibles...
- Ici, on peut juste montrer que l'heuristique ne construit pas la meilleure solution (qui aurait été de remplacer le second objet par le cinquième menant à un bénéfice de 20).

Rapport d'approximation pour l'heuristique BD sur l'instance $\mathcal{I}$ du problème.

Défini comme $Inf[\frac{f_{BD}(\mathcal{I})}{f^*(\mathcal{I})}]$

où f est la somme des objets séléctionnés.

Peut-on évaluer la qualité de cette heuristique ?

- Cette question porte sur l'ensemble de toutes les instances possibles...
- Ici, on peut juste montrer que l'heuristique ne construit pas la meilleure solution (qui aurait été de remplacer le second objet par le cinquième menant à un bénéfice de 20).

Rapport d'approximation pour l'heuristique BD sur l'instance $\mathcal{I}$ du problème.

Défini comme $Inf[\frac{f_{BD}(\mathcal{I})}{f^*(\mathcal{I})}]$

où f est la somme des objets séléctionnés.

- On peut construire une famille d'instances qui conduit à des solutions arbitrairement mauvaises pour cette heuristique !

Instances arbitrairement mauvaises

On note $f_{BD}(\mathcal{I})$ le profit d'une solution générée par l'heuristique *BestDensity* sur l'instance $\mathcal{I}$, et $f^*(\mathcal{I})$ le profit optimal pour cette instance.

Instances arbitrairement mauvaises

On note $f_{BD}(\mathcal{I})$ le profit d'une solution générée par l'heuristique *BestDensity* sur l'instance $\mathcal{I}$, et $f^*(\mathcal{I})$ le profit optimal pour cette instance.

Considérons l'instance suivante avec $n = 2$ objets et $K = k$.

- $(w_1 = 1, b_1 = 1)$
- $(w_2 = k, b_2 = k - 1)$

Preuve

- L'objet 1 a une densité égale à 1
- l'objet 2 a une densité $\frac{k-1}{k} = 1 - \frac{1}{k}$

La densité du premier est plus grande que celle du second, donc c'est celui-là qui est sélectionné en premier, l'autre ne peut plus rentrer dans le sac !
Ainsi, le bénéfice obtenu est de 1.
Si l'on avait mis le second, le bénéfice aurait été de $k - 1$ (en $\Theta(k)$)

Sur cette instance, **l'heuristique est arbitrairement mauvaise** car on peut prendre k aussi grand que l'on souhaite.

Amélioration (simple)

Il est possible de modifier l'heuristique précédente pour obtenir une garantie de performance.

- Trier les objets par densités décroissantes.
- Construire de proche en proche une solution glouton S_1 jusqu'à ce qu'un dernier objet ne rentre plus dans le sac.
- Construire une seconde solution S_2 avec l'objet qui a le plus grand profit.
- Prendre la solution qui réalise le maximum entre S_1 et S_2.

Sur l'exemple

index	2	1	5	4	6	7	3
	(3,8)	(2,5)	(5,12)	(4,9)	(8,17)	(4,8)	(6,11)

- On sélectionne le premier objet :
  le bénéfice est de 8 et il reste une capacité de 5
- Le second objet est le dernier à pouvoir être sélectionné :
  bénéfice de 13, plus aucun objet ne rentre.
- On compare cette solution avec celle obtenue par l'objet de plus grand bénéfice :
  c'est le troisième avec 17 qui est supérieur.
- Cette solution n'est pas optimale, mais elle est *garantie*.

Preuve

Propriété générale

Cet algorithme BestDensity amélioré, noté A, n'est jamais plus que deux fois pire que l'optimal.

C'est-à-dire, pour chaque instance $\mathcal{I}$:

$$2 \cdot f_A(\mathcal{I}) \geq f^*(\mathcal{I})$$

où $f_A(\mathcal{I})$ est le profit de la solution générée par l'algorithme.

Preuve

- On note i_T l'indice du premier objet non sélectionné dans S_1
- On note $\mathcal{I}'$ le sous-ensemble des objets de $S_1 \cup i_T$

On a la propriété de borne supérieure suivante :
$f^* \leq \Sigma_{i \in S_1} b_i + b_{i_T}$

Preuve

- On note i_T l'indice du premier objet non sélectionné dans S_1
- On note $\mathcal{I}'$ le sous-ensemble des objets de $S_1 \cup i_T$

On a la propriété de borne supérieure suivante :
$f^* \leq \Sigma_{i \in S_1} b_i + b_{i_T}$
En effet, $f^* \leq \Sigma_{i \in S_1} b_i + b'_{i_T}$ où b'_{i_T} est la partie tronquée de i_T dont le bénéfice est bien entendu inférieur à b_{i_T} (objet complet).

La preuve de la $\frac{1}{2}$-approximation découle directement de l'inégalité précédente.

- On considère les objets de $S_1 \cup S_2$
  $f(S_1 \cup S_2) \geq b(\mathcal{I}') \geq f^*$ car $b_{S_2} \geq b_{i_T}$
- Le coût de cette politique est $f_A = max(b_{S_2}, \Sigma_{i \in S_1} b_i)$
- comme $2.max(x, y) \geq x + y$, on a bien :
  $2f_A \geq \Sigma_{i \in S_1} b_i + b_{S_2} \geq f^*$.