Approximation non polynomiale

Bruno Escoffier

Journées Franciliennes de Recherche Opérationnelle

Paris - 20 Novembre 2012

1. A toutes fins utiles
2. Algorithmes approchés exponentiels
3. Quelques mots sur les algorithmes FPT approchés
4. Résultats négatifs?

A toutes fins utiles...

Algorithmes approchés

A est r-approché si pour toute instance I:

$$val(\mathtt{A}(I)) \leq r \times \mathtt{opt}(I)$$

$r \geq 1$ ($r = 1$: algorithme exact).

Algorithmes approchés

$\mathtt{A}$ est r-approché si pour toute instance I:

$$val(\mathtt{A}(I)) \leq r \times \mathtt{opt}(I)$$

$r \geq 1$ ($r = 1$: algorithme exact).

Définition analogue pour un problème de maximisation:

$$val(\mathtt{A}(I)) \geq r \times \mathtt{opt}(I)$$

$r \leq 1$.

Des résultats positifs...

- min vertex cover est approximable à rapport 2
- min tsp est approximable à rapport 3/2 dans le cas métrique
- min set cover est approximable à rapport $O(\log n)$
- ...

Algorithmes approchés polynomiaux

... et négatifs

Si $P \neq NP$:

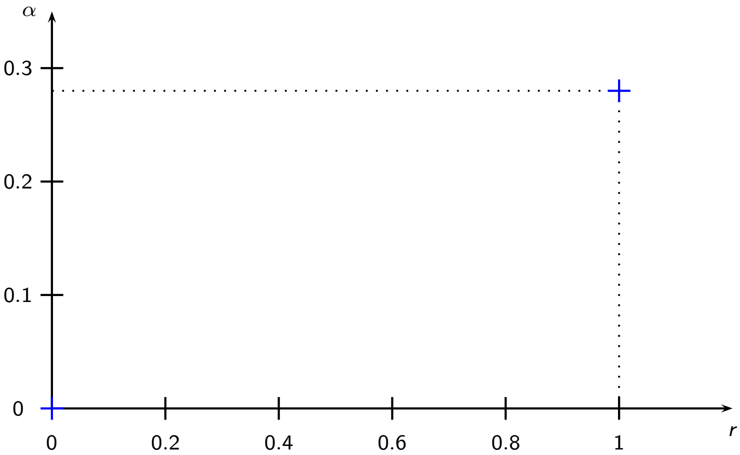
- max independent set inapproximable avec un rapport constant
- min coloring avec un rapport constant
- min set cover avec un rapport constant
- min vertex cover avec un rapport meilleur que 1.36 (meilleur que 2 sous UGC)
- ...

Algorithmes approchés exponentiels

En quelques mots

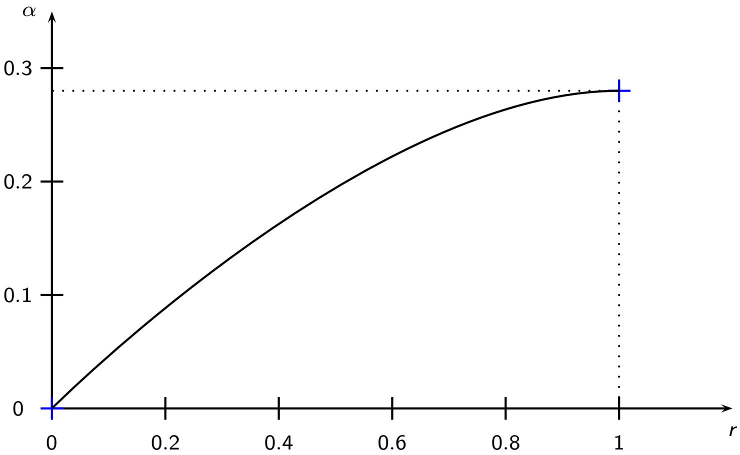
Le problème max independent set

Un algorithme en $O^*(2^{\alpha n})$?



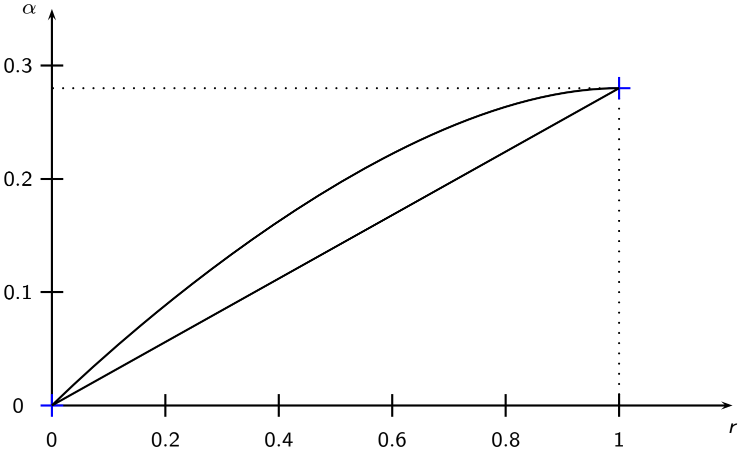
Le problème max independent set

Un algorithme en $O^*(2^{\alpha n})$?

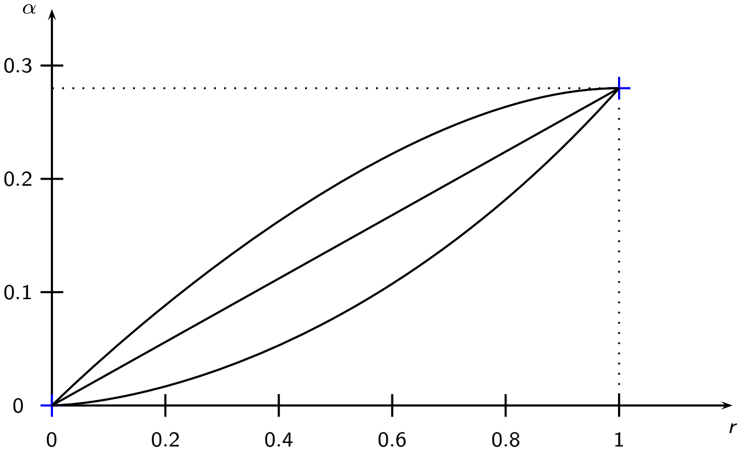


Le problème max independent set

Un algorithme en $O^*(2^{\alpha n})$?



Un algorithme en $O^*(2^{\alpha n})$?



La problématique

Compromis temps/qualité de la solution

Un problème Π

- meilleur algorithme exact: $O^*(\gamma^n)$;
- meilleur rapport d'approximation polynomial connu: r.

La problématique

Compromis temps/qualité de la solution

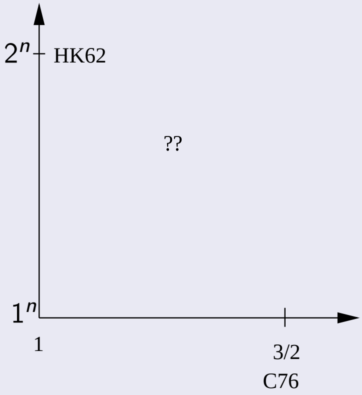
Un problème Π

- meilleur algorithme exact: $O^*(\gamma^n)$;
- meilleur rapport d'approximation polynomial connu: r.

$\rightarrow$ Puis-je résoudre Π:
(1) en temps (significativement) plus faible que $O^*(\gamma^n)$
(2) avec un rapport d'approximation (significativement) meilleur que r?

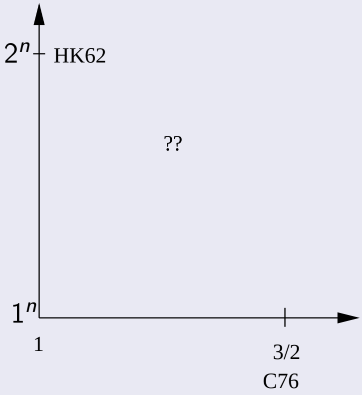
TSP

- Résoluble en $O^*(2^n)$
- 3/2-approximable en temps polynomial



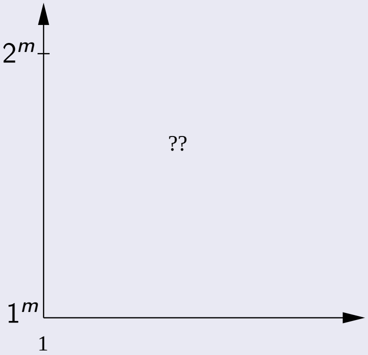
TSP

- Résoluble en $O^*(2^n)$
- 3/2-approximable en temps polynomial



Minimum Set Cover

- Résoluble en $O^*(2^m)$ (recherche exhaustive)
- Inapproximable à rapport constant en temps poly



Approximation algorithms

- Techniques
  - Générer peu de solutions
  - Diviser pour approcher
  - Randomisation
  - Arbre de recherche approché
  - Glouton . . .
- Problems
  - MaxIS, Min VC, Min SC
  - Min Bandwidth
  - Min Coloring
  - TSP, Steiner Tree
  - Max Sat . . .

Quelques techniques illustrées sur le stable maximum

Générer un faible nombre de solutions

Le principe

Générer un faible nombre de solutions (en les complétant polynomialement, si possible et nécessaire) et retourner la meilleure solution calculée.

Générer un faible nombre de solutions

- Générer les sous-ensembles de V de taille $\sqrt{n}$
- Si l'un d'eux est un stable, le renvoyer
- Sinon renvoyer un sommet.

Rapport d'approximation:

Générer un faible nombre de solutions

- Générer les sous-ensembles de V de taille $\sqrt{n}$
- Si l'un d'eux est un stable, le renvoyer
- Sinon renvoyer un sommet.

Rapport d'approximation: $\sqrt{n}$

Générer un faible nombre de solutions

- Générer les sous-ensembles de V de taille $\sqrt{n}$
- Si l'un d'eux est un stable, le renvoyer
- Sinon renvoyer un sommet.

Rapport d'approximation: $\sqrt{n}$
Complexité:

Générer un faible nombre de solutions

- Générer les sous-ensembles de V de taille $\sqrt{n}$
- Si l'un d'eux est un stable, le renvoyer
- Sinon renvoyer un sommet.

Rapport d'approximation: $\sqrt{n}$
Complexité: sous-exponentielle $O(2^{\sqrt{n}\log n})$

Générer un faible nombre de solutions

- Générer les sous-ensembles de V de taille $\sqrt{n}$
- Si l'un d'eux est un stable, le renvoyer
- Sinon renvoyer un sommet.

Rapport d'approximation: $\sqrt{n}$
Complexité: sous-exponentielle $O(2^{\sqrt{n}\log n})$

Appliqué sur:

- capacitated dominating set (Cygan, Pilipczuk & Wojtaszczyk (2010))
- min independent dominating set (Bourgeois, Escoffier & Paschos (2010))

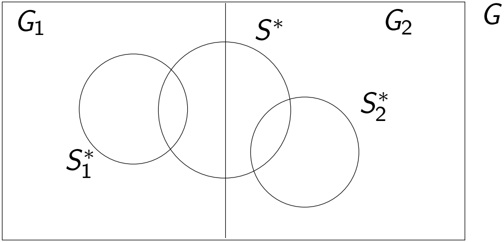
Diviser pour approcher (1)

Le principe

Résoudre optimalement le problème sur une série de ("petites") sous-instances de l'instance initiale

- Couper de manière appropriée l'instance en un ensemble de sous-instances (dont la taille est fonction du rapport que l'on souhaite obtenir)
- Résoudre le problème sur ses sous-instances
- Construire une solution sur l'instance initiale en utilisant les solutions sur les sous-instances

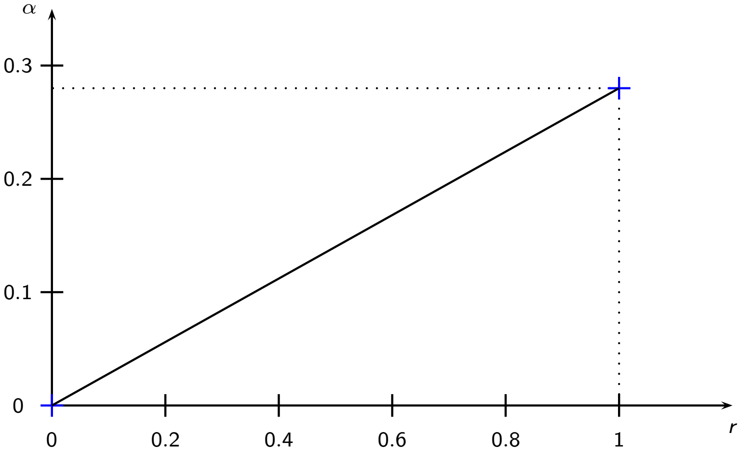
Diviser pour approcher (2)



$$|S^*| \leqslant |S_1^*| + |S_2^*| \leqslant 2 \max\{|S_1^*|, |S_2^*|\} \Longrightarrow \frac{\max\{|S_1^*|, |S_2^*|\}}{|S^*|} \geqslant \frac{1}{2}$$

Complexité: $O\left(\gamma^{n/2}\right)$, si $O\left(\gamma^{n}\right)$ est la complexité exacte de max independent set

Un algorithme en $O^*(2^{\alpha n})$ pour max independent set?

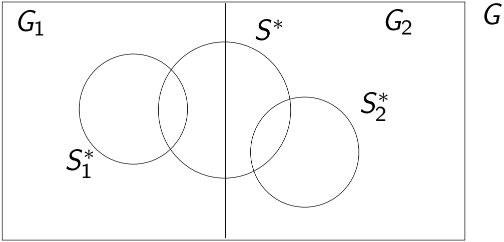


Diviser pour approcher (4)

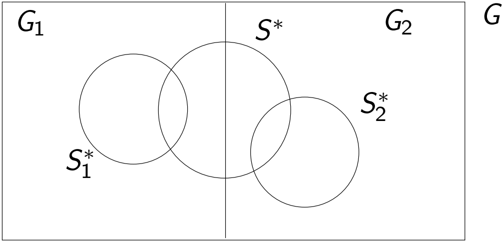
Applicable

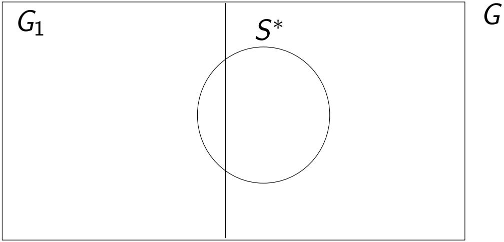
- aux problèmes héréditaires
- par extension à min vertex cover
- à min set cover (idée: grouper les ensembles pour réduire leur nombre)

Randomisation



Randomisation





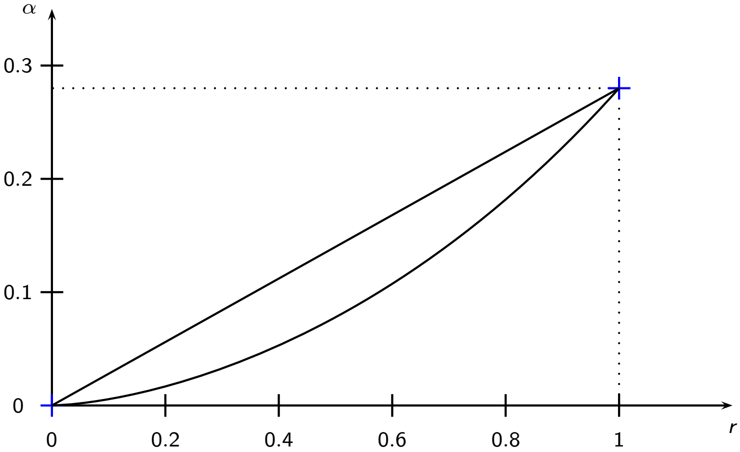
Randomisation

Le principe

1. Répéter le partitionnement un nombre exponentiel de fois: β^n fois.
2. Résoudre chacune des β^n instances
3. Renvoyer la meilleure solution.

$\rightarrow$ On obtient un rapport r en un temps inférieur à γ^{rn} (avec une proba ~ 1).

Un algorithme en $O^*(2^{\alpha n})$?



Elagage approché d'un arbre de recherche (1)

Le principe

Concevoir un algorithme de branchement en élaguant des branches tout en maîtrisant l'erreur par rapport à l'optimum

Elagage approché d'un arbre de recherche (2)

Algorithme exact

1. Si $\Delta(G) \leq 2$, résoudre le problème optimalement;
2. sinon, brancher sur un sommet v de degré ≥ 3: soit prendre v dans la solution (et retirer $N[v]$), soit ne pas prendre v (et retirer v).

$$T(n) \leq T(n-4) + T(n-1)$$

Elagage approché d'un arbre de recherche (2)

Algorithme exact

1. Si $\Delta(G) \leq 2$, résoudre le problème optimalement;
2. sinon, brancher sur un sommet v de degré ≥ 3: soit prendre v dans la solution (et retirer $N[v]$), soit ne pas prendre v (et retirer v).

$$T(n) \leq T(n-4) + T(n-1)$$

Algorithme 1/2-approché

1. Si $\Delta(G) \leq 7$, trouver une solution 0.5-approchée*;
2. sinon, brancher sur un sommet v de degré ≥ 8: soit prendre v dans la solution (et retirer $N[v]$ et une clique C), soit ne pas prendre v.

$$T(n) \leq T(n-11) + T(n-1)$$

(* rapport $\frac{5}{\Delta(G)+3}$ en temps polynomial, (Berman & Fujito (1985)))

Elagage approché d'un arbre de recherche (3)

- Amélioration quel que soit le rapport désiré.
- Applicable à de nombreux problèmes
  min bandwidth (Cygan & Pilipczuk (2010))
  min and max sat (Danstsin & al. (2001), Escoffier, Paschos & Tourniaire (2012a))
  min set cover (Bourgeois, Escoffier & Paschos (2009))
  ...
  conditions générales d'applications (Escoffier, Paschos & Tourniaire (2012b))

Quelques mots sur les algorithmes FPT approchés

FPT et approximation

Un problème Π, un entier k. Existe-t-il une solution de valeur $\leq k$ ($\geq k$)?

- FPT: algorithme en $f(k)poly(n)$
- W[1]-difficile: pas FPT si FPT$\neq$W[1]

FPT et approximation

Un problème Π, un entier k. Existe-t-il une solution de valeur $\leq k$ ($\geq k$)?

- FPT: algorithme en $f(k)poly(n)$
  Algorithme FPT r-approché en $g(k)poly(n)$ avec g (significativement) plus petite que f?
- W[1]-difficile: pas FPT si FPT$\neq$W[1]

FPT et approximation

Un problème Π, un entier k. Existe-t-il une solution de valeur $\leq k$ ($\geq k$)?

- FPT: algorithme en $f(k)poly(n)$
  Algorithme FPT r-approché en $g(k)poly(n)$ avec g (significativement) plus petite que f?
- W[1]-difficile: pas FPT si FPT$\neq$W[1]
  Algorithme FPT r-approché?

Algorithme FPT r-approché

Un problème Π, un entier k.
$\rightarrow$ S'il existe une solution de valeur $\leq k$, renvoyer une solution de valeur $\leq rk$.
$\rightarrow$ Complexité FPT ($g(k)poly(n)$)

FPT et approximation

min vertex cover

FPT: $\alpha^k poly(n)$

2-approché en temps polynomial

FPT et approximation

min vertex cover

FPT: $\alpha^k poly(n)$
2-approché en temps polynomial
$\rightarrow$ pour tout $r \in [1, 2]$, algorithme FPT r-approché en $\alpha_r^k poly(n)$
avec: $\alpha_1 = \alpha$, $\alpha_2 = 0$:

FPT et approximation

min vertex cover

FPT: $\alpha^k poly(n)$
2-approché en temps polynomial
$\rightarrow$ pour tout $r \in [1, 2]$, algorithme FPT r-approché en $\alpha_r^k poly(n)$
avec: $\alpha_1 = \alpha$, $\alpha_2 = 0$:

- diviser pour approcher (Bourgeois, Escoffier & Paschos (2008))
- élagage approché de l'arbre de recherche (Brankovic & Fernau (2010))
- une réduction mêlant FPT et approximation (Fellows, Kulik, Rosamond & Shachnai (2012))

D'autres problèmes étudiés: Edge dominating set, Steiner tree,...

FPT et approximation

Un problème Π, un entier k. Existe-t-il une solution de valeur $\leq k$ ($\geq k$)?

- FPT: algorithme en $f(k)poly(n)$
  Algorithme FPT r-approché en $g(k)poly(n)$ avec g (significativement) plus petite que f?
- W[1]-difficile: pas FPT si FPT$\neq$W[1]
  Algorithme FPT r-approché?

Deuxième question: aucune réponse (positive) - pour la paramétrisation standard.

Résultats négatifs?

Résultats négatifs?

min independent dominating set n'est pas $g(k)$-approximable en temps FPT si FPT$\neq$W[2] (Downey, Fellows & McCartin (2006))

Résultats négatifs?

min independent dominating set n'est pas $g(k)$-approximable en temps FPT si FPT$\neq$W[2] (Downey, Fellows & McCartin (2006))

Qu'en est-il de max independent set? de min dominating set?

Résultats négatifs?

Compromis temps/qualité de la solution

Un problème Π

- meilleur algorithme exact: $O^*(\gamma^n)$;
- meilleur rapport d'approximation polynomial connu: r.

Résultats négatifs?

Compromis temps/qualité de la solution

Un problème Π

- meilleur algorithme exact: $O^*(\gamma^n)$;
- meilleur rapport d'approximation polynomial connu: r.

$\rightarrow$ Puis-je résoudre Π:
(1) en temps (significativement) plus faible que $O^*(\gamma^n)$
(2) avec un rapport d'approximation (significativement) meilleur que r?

Résultats négatifs?

Compromis temps/qualité de la solution

Un problème Π

- meilleur algorithme exact: $O^*(\gamma^n)$;
- meilleur rapport d'approximation polynomial connu: r.

$\rightarrow$ Puis-je résoudre Π:
(1) en temps (significativement) plus faible que $O^*(\gamma^n)$
(2) avec un rapport d'approximation (significativement) meilleur que r?

(1): en temps sous-exponentiel??

Résultats négatifs?

Réponse: oui (rapport $\sqrt{n}$ pour le stable par exemple)

Résultats négatifs?

Réponse: oui (rapport $\sqrt{n}$ pour le stable par exemple)

mais...

Résultats négatifs?

Réponse: oui (rapport $\sqrt{n}$ pour le stable par exemple)

mais...

Sous **ETH**, pour tout $r > 0$ et tout $\delta > 0$, il n'y a pas d'algorithme r-approché en temps $O(2^{n^{1-\delta}})$ pour max independent set (Escoffier, Kim & Paschos (2012))

Résultats similaires:

- Pas de $7/8 + \epsilon$-approximation pour Max 3-sat
- Pas d'approximation constante pour min coloring
- Pas de $7/6 - \epsilon$-approximation pour min vertex cover

Résultats négatifs?

- Peut-on étendre ces résultats à $2^{o(n)}$?

Résultats négatifs?

- Peut-on étendre ces résultats à $2^{o(n)}$?
- Peut-on trouver des bornes d'inapproximabilité en temps γ^n? (sous quelle hypothèse)?
- Peut-on développer des réductions préservant l'approximation adapté à cette problématique?
- Qu'en est-il des noyaux approchés?
- ...

Validation expérimentale???

Validation expérimentale???

Exemple: Algorithme 5-approché pour min set cover en $O(1.005^{m+n})$