Algorithmes d'approximation

François Schwarzentruber

Définition 1 Un *problème de minimisation* est de la forme

Problème de minimisation
entrée : instance x
sortie : une solution $sol^* \in Sol_x$ avec $sol^* = argmin_{sol \in Sol_x} c(x, sol)$

Problème de minimisation
entrée : instance x
sortie : $min_{sol \in Sol_x} c(x, sol)$

où Sol_x est l'espace des solutions de l'instance x et c est une fonction de coût.

Quand on dit qu'un problème d'optimisation est NP-complet, cela signifie que le problème de décision associé, est NP-complet :

Problème de décision associé
entrée : instance x, un nombre λ
sortie : oui s'il existe une solution $sol \in Sol_x$ avec $c(x, sol) \leq \lambda$

Un problème de maximisation, c'est la même chose mais avec *argmax* et *max*. Un problème d'optimisation désigne à la fois un problème de minimisation ou de maximisation.

Définition 2 Étant donné un problème d'optimisation, un algorithme d'approximation calcule une solution quasi optimale.

Définition 3 Un algorithme d'approximation est de ratio ρ si, sur toute entrée de taille n,

$$\frac{\text{cout(solution calculée)}}{\text{cout(solution optimale)}} \leq \rho(n) \qquad\qquad \rho(n) \leq \frac{\text{cout(solution calculée)}}{\text{cout(solution optimale)}}$$

si problème de minimisation — si problème de maximisation

La quantité $\rho(n)$ s'appelle ratio, ou facteur d'approximation : c'est une garantie de performance de l'algorithme d'approximation.

1 Classification

Définition 4 (PTAS) Un *polynomial time approximation scheme* est un algorithme d'approximation qui prend en entrée une instance I et un nombre ϵ, et qui calcule une solution de ratio $\begin{cases} 1+\epsilon & \text{pour pb de min} \\ 1-\epsilon & \text{pour pb de max} \end{cases}$, et tel qu'en fixant ϵ, l'algorithme est en temps $poly(|I|)$.

Définition 5 (FPTAS) Un *fully polynomial time approximation scheme* est un PTAS en temps $poly(|I|, \frac{1}{\epsilon})$.

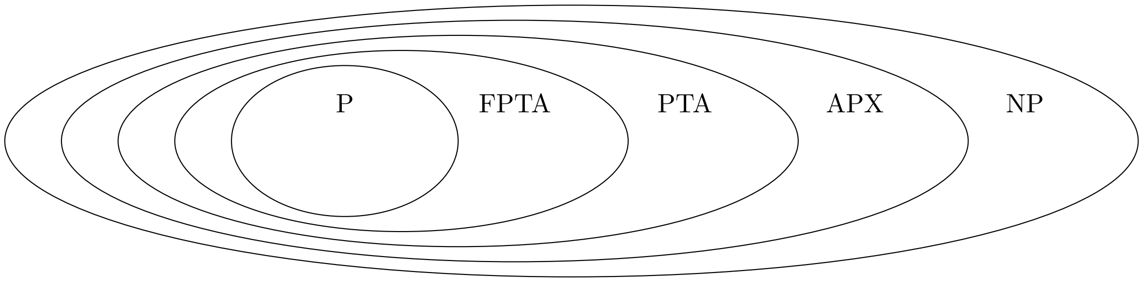
Remarque 6
- Avec PTAS, on peut avoir une complexité $2^{1/\epsilon} n^{1/\epsilon}$.
- Avec FPTAS, on ne peut pas mais on peut avoir $n^2 \frac{1}{\epsilon^3}$.

Définition 7 (Classes de complexité)

APX = problèmes d'optimisation admettant un algo d'approximation poly avec ratio constant

PTA = problèmes d'optimisation admettant un PTAS

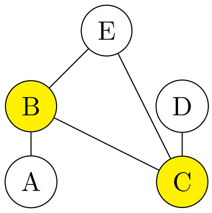
FPTA = problème d'optimisation admettant un FPTAS



2 Couverture de sommets

Définition 8 (couverture de sommets) Soit un graphe $G = (V, E)$. Une couverture de sommet est un ensemble $C \subseteq V$ tel que pour toute arête $(u, v) \in E$, $u \in C$ ou $v \in C$.

Exemple 9



Définition 10 (problème d'optimisation) **VERTEX COVER**

entrée : un graphe $G = (V, E)$ non orienté
sortie : une couverture de sommets de cardinal minimal.

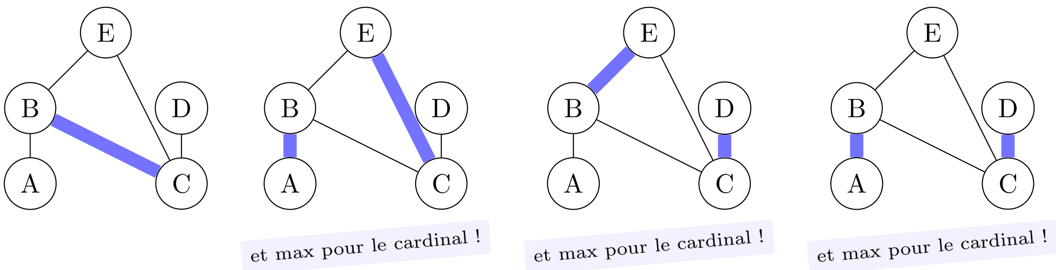
Un sommet n'est marié à ≤ 1 autre sommet.

Définition 11 (couplage) Soit $G = (V, E)$ un graphe non orienté. Un ensemble $M \subseteq E$ est un couplage si pour tout sommet $u \in V$, il existe au plus une arête $(a, b) \in M$ avec $a = u$ ou $b = u$.

Définition 12 (couplage maximum) Un couplage M est maximum si pour tout couplage M' on a $|M'| \leq |M|$.

Définition 13 (couplage maximal pour l'inclusion) Un couplage maximal pour l'inclusion est un couplage M tel que tout pour tout $M' \subseteq E$, $M \subsetneq M'$ implique M' n'est pas un couplage.

Exemple 14 (couplages maximaux pour l'inclusion)



Proposition 15 VERTEX COVER admet un algorithme d'approximation est de ratio 2.

3 Couverture de sommets pondérée et relaxation linéaire

Définition 16 **problème de la couverture de sommets pondérée**

entrée : Un graphe non orienté $G = (V, E)$, des poids positifs $w(u)$ à chaque sommet $u \in V$;
sortie : une couverture de sommets $C^* \subseteq V$ tel que son poids $\sum_{v \in C^*} w(v)$ soit minimal.

Proposition 17 C est une couverture de sommets.

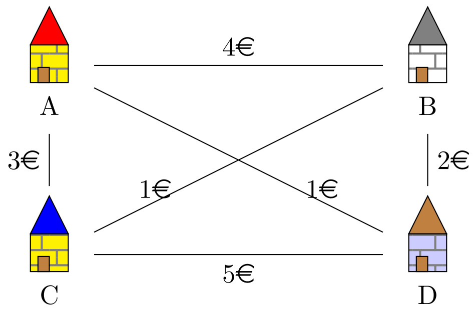
Théorème 18 WEIGHTED VERTEX COVER admet un algorithme 2-approximant.

4 Inapproximabilité du voyageur de commerce (TSP)

Définition 19 (problème d'optimisation) **TSP**

entrée : un graphe non orienté complet pondéré $G = (V, d)$ où $d : V \times V \to \mathbb{N}$ avec $d(i, j) = d(j, i)$
sortie : un tour dans G de poids minimal

Exemple 20



Théorème 21 Si P $\neq$ NP, alors TSP n'admet pas d'algo d'approximation en temps poly de ratio constant.

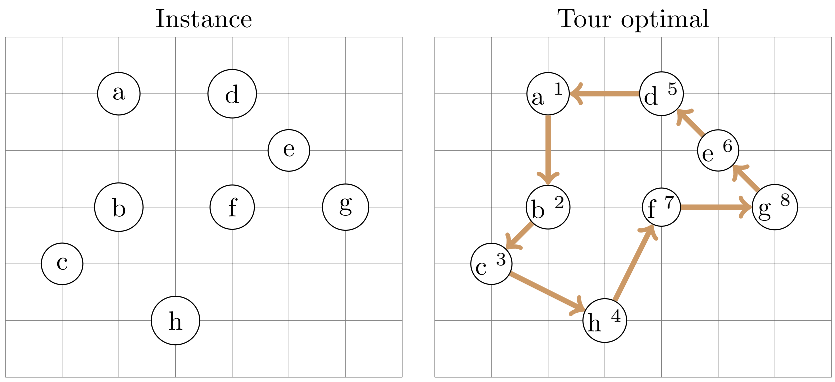
5 Voyageur de commerce avec inégalité triangulaire

Définition 22 Un graphe non orienté complet pondéré $G = (V, d)$ vérifie l'inégalité triangulaire si $d(u, w) \leq d(u, v) + d(v, w)$ pour tous les sommets $u, v, w \in V$.

Définition 23 **problème du voyageur de commerce avec inégalité triangulaire**

entrée : un graphe non orienté complet pondéré $G = (V, d)$ avec inégalité triangulaire ;
sortie : un tour de poids minimal.

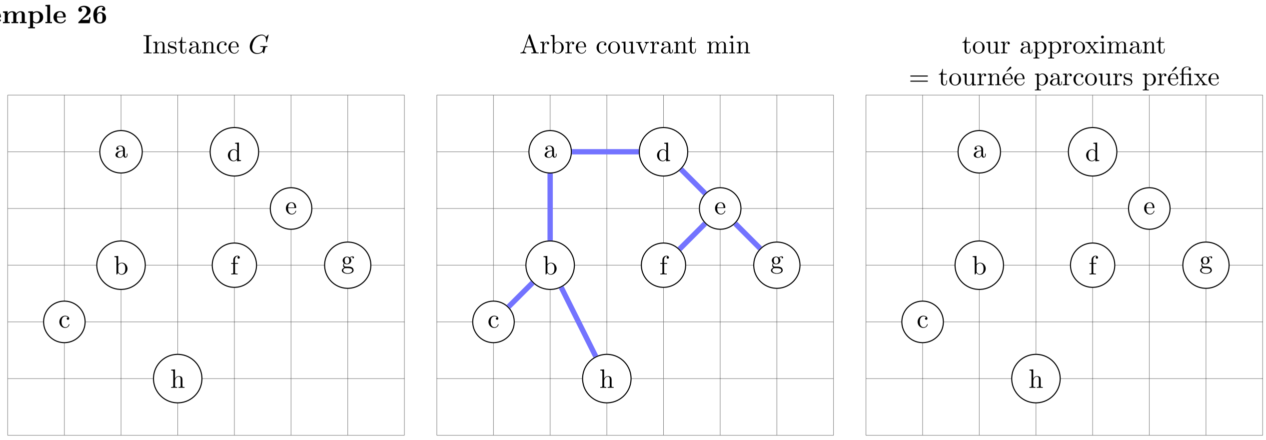
Exemple 24



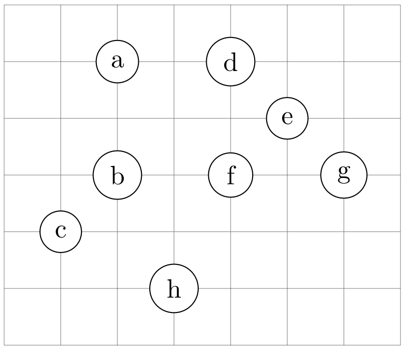
5.1 Algorithme 2-approximant (super simple à expliquer)

Théorème 25 METRIC TSP admet un algorithme 2-approximant.

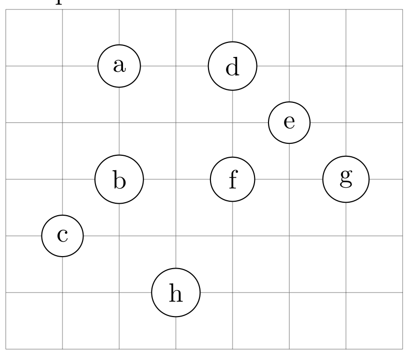
Exemple 26



Arbre couvrant min où on a doublé les arêtes



tour optimal où on a enlevé une arête



6 Voyage de commerce euclidien

Définition 27 **problème du voyageur de commerce euclidien**

entrée : un ensemble P de n points de $\mathbb{R}^d$ où le poids entre points est donné par la distance euclidienne ;
sortie : un tour sur P de poids minimal.

Théorème 28 (admis) EUCLIDEAN TSP admet un PTAS [Aro98].

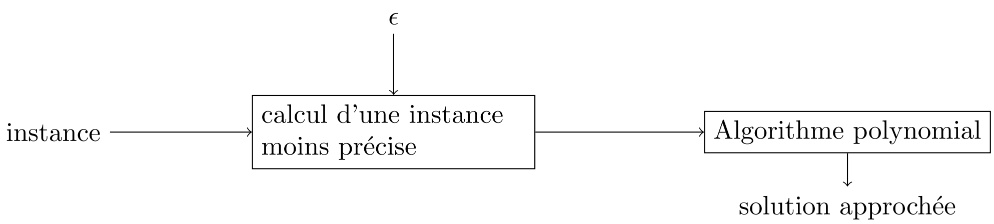
7 Sac à dos

Ref : [TK06][11.8, p. 644], [DPV06][p. 283], [Vaz04][p. 68]

SAC A DOS

entrée : n objets $((w_i, v_i))_{i=1..n}$, un poids W
sortie : un sous-ensemble $J \subseteq \{1, \ldots, n\}$ tel que $\sum_{i \in J} w_i \leq W$ et $\sum_{i \in J} v_i$ maximal.

Théorème 29 SAC A DOS admet un FPTAS.



Références

[Aro98] Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5) :753–782, 1998.

[DPV06] S. Dasgupta, C.H. Papadimitriou, and U.V. Vazirani. Algorithms. 2006.

[TK06] É. Tardos and J. Kleinberg. Algorithm design, 2006.

[Vaz04] V.V. Vazirani. *Approximation algorithms*. springer, 2004.