Exemples d'algorithmes d'approximations et d'algorithmes probabilistes

Définition 1 [COR3 34.1] Un **problème abstrait** Q est une relation binaire sur un ensemble I d'instances de problème, et un ensemble S de solutions.

Définition 2 Un **problème de décision** est un problème abstrait où $S = \{0, 1\}$.

Définition 3 Un **problème d'optimisation** est un problème abstrait dans lequel une certaine valeur doit « être minimisée ou maximisée ».

Exemple 4 Dans un graphe, le problème du plus court chemin est un problème d'optimisation.

Remarque 5 On peut associer un problème de décision à un problème d'optimisation. Par exemple, le problème de décision associé à PLUS-COURT-CHEMIN est le problème dont les instances sont des graphes et des chemins, de solution 1 s'il s'agit d'un plus court chemin dans le graphe, et 0 sinon.

I. Algorithmes d'approximation

Définition 6 **Un algorithme d'approximation** est un algorithme qui fournit une solution approchée à un problème d'optimisation.

A. Facteur d'approximation

Définition 7 **Facteur d'approximation** [PAPA 9.2] Soit un problème d'optimisation, et un algorithme $\mathcal{A}$ qui, étant donnée une instance I renvoie une solution de valeur $\mathcal{A}(I)$. En notant $\mathrm{OPT}(I)$ la valeur de la solution optimale. Le facteur d'approximation de l'algorithme $\mathcal{A}$ est défini comme

$$\alpha_{\mathcal{A}} = \max_{I} \frac{\mathcal{A}(I)}{\mathrm{OPT}(I)}$$

Remarque 8 Un algorithme est une α-approximation d'un problème, si pour ce problème donné, son facteur d'approximation vaut α.

Exemple 9 **Couverture de sommets** [PAPA 9.2.1]

- **Entrée :** Un graphe non-orienté $G = (S, A)$
- **Sortie :** Un sous-ensemble $S' \subseteq S$ qui touche toutes les arêtes
- **Minimise** $|S'|$

Définition 10 [PAPA 9.2.1] Un **couplage** dans un graphe est un ensemble $A' \subseteq A$ d'arêtes telles qu'elles n'aient aucun sommet en commun. Un couplage est **maximal** si on ne peut rajouter aucune arête.

Propriété 11 [PAPA 9.2.1] Soit un couplage maximal A' d'un graphe, une couverture de sommet minimale contient au moins $|A'|$ sommets.

Algorithme 12 [COR3 35.1]

```
Couverture-Sommet-Approchée(A,S):
  C = {}
  A' = A
  tant que (A' non vide):
    sout (u,v) une arête de A
    C = C union {u,v}
    supprimer de A' toute arête incidente à u ou v
  renvoyer C
```

Propriété 13 [PAPA 9.2.1] Cet algorithme est une 2-approximation du problème de couverture de sommets.

B. Algorithmes Gloutons

Définition 14 [PAPA 5] Un **algorithme glouton** construit une solution, morceau par morceau, en choisissant systématiquement le prochain morceau qui donne le plus grand bénéfice immédiat.

Exemple 15 **Couverture d'ensemble** [PAPA 5.4] Soit un ensemble B, le problème de couverture d'ensemble est un problème d'optimisation défini comme :

- **Entrée:** Une liste d'ensemble $S_1, \ldots, S_m$, avec $\forall i S_i \subseteq B$
- **Sortie:** Une liste de S_i telle que leur union soit B.
- **Coût:** Le nombre d'ensembles choisis

Propriété 16 [PAPA 5.4] S B contiens n éléments, et que la solution optimale utilise k ensembles, alors l'algorithme glouton consistant à prendre à chaque étape l'ensemble qui couvre le plus d'éléments restant utilise au plus $k \ln(n)$ ensembles.

Exemple 17 **Voyageur de commerce métrique** En se basant sur un arbre couvrant minimal, il est possible de calculer une 2-approximation pour le voyageur de commerce dans le cas métrique.

Remarque 18 Sans l'hypothèse métrique, si P != NP, il n'existe aucune α-approximation du problème du voyageur de commerce.

Exemple 19 **Coloration de graphe** [TOR 9.3] Étant donné un graphe non orienté et non pondéré $G = (V, E)$, on appelle k -coloration de G, pour $k \in \mathbb{N}^*$, une application c qui associe à chacun des sommets de G un e«ntier de $[0, k-1]$ de sorte que si deux sommets u et v sont voisins alors leur couleur sont différentes.

Algorithme 20 [TOR Programme 9.6] On peut colorier un graphe avec un algorithme glouton en $O(|S| + |A|)$, en prenant pour chaque sommet, la plus petite couleur disponible parmi celles des voisins :

```
fonction coloration(S,A):
  c = [0,...,0]
  Pour v dans S:
    c[v] <- min(c, voisins(v))
  renvoyer c
```

DEV

Remarque 21 Il existe des graphes pour lequels l'algorithme glouton renvoie une coloration arbitrairement mauvaise. En revanche, pour certains graphe, cette coloration est optimale (graphe d'intervales), ou une 3-approximation (graphe de disques unitaires).

Algorithme 22 Si un graphe est 3-coloriable, alors on peut construire une $\sqrt{|S|}$-approximation.

C. Schéma d'approximation

Définition 23 **Schéma d'approximation** [VAZ 8] Un schéma d'approximation est un algorithme $\mathcal{A}$ qui, étant donné une entrée I et un seuil $\varepsilon > 0$ vérifie:

$$\mathcal{A}(I, \varepsilon) \leq (1 + \varepsilon)\ \mathrm{OPT}(I)$$

Définition 24 **Schéma d'approximation polynomial** [VAZ 8] Si en plus, pour tout $\varepsilon > 0$ fixé, l'exécution de **cal(A)** est polynomiale en la taille de **I**, nous dirons qu'il s'agit d'un **schéma d'approximation polynomial**.

Définition 25 **Schéma d'approximation totalement polynomial** [VAZ 8] Un algorithme est un **schéma d'approximation totalement polynomial** si en plus, l'exécution de $\mathcal{A}$ est également polynomial en $\frac{1}{\varepsilon}$.

Exemple 26 Le problème du sac à dos admet un schéma d'approximation totalement polynomial

II. Algorithmes probabilistes

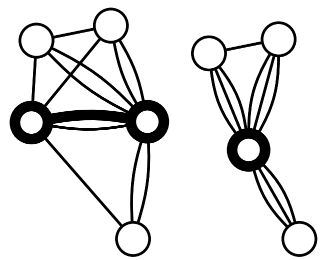
A. Définitions et exemples

Définition 27 **Algorithme probabiliste** [TOR 9.6] est un algorithme qui effectue des choix aléatoires pendant son exécution.

Remarque 28 **Méthode probabiliste** [TOR 13.4.1.2] Certains algorithmes déterministes peuvent répondre de manière probabiliste. C'est le cas, par exemple, si le choix repose sur une espérance ou si la formule utilisée est vraie dans la majorité des cas.

Définition 29 [PAPA 8.1] Une **coupe** d'un graphe est un ensemble d'arête tel que leur retrait sépare le graphe. Le problème MIN-CUT est le problème d'optimisation qui cherche une coupe minimale, c'est à dire avec le moins d'arêtes.

Exemple 30 [PAPA 5.1] L'algorithme de **Karger** cherche une coupe d'un graphe en compressant tour à tour une arête au hasard, jusqu'à ce qu'il ne reste que 2 sommets, et renvoi l'ensemble des arêtes reliant ces deux sommets.



Propriété 31 [PAPA 5.1] L'algorithme de Karger renvoie une coupe minimale avec une probabilité supérieure à $\frac{1}{|S|^2}$.

Définition 32 **Monte-Carlo/Las Vegas** [TOR 9.6] Il y a deux principaux types d'algorithmes probabilistes:

	Réponse	**Temps de calcul**
Las Vegas	Toujours correct	Statistiquement faible
Monte Carlo	Correct avec probabilité	Déterministe

Exemple 33 [TOR 9.6] Le **tri rapide** est un exemple d'algorithme de Las Vegas.

[PAPA 1.3]

Exemple 34 **Test de primalité** Pour tester la primalité d'un nombre, il est possible de se baser sur le petit théorème de Fermat:

$$p \text{ premier} \Rightarrow \forall 1 \leq a < p, a^{p-1} \equiv 1[p]$$

Si la formule n'est pas vérifiée, le nombre p n'est pas premier. S'il l'est, alors p est premier avec une certaine probabilité.

Algorithme 35 **Test de primalité de Fermat**

```
est_premier(N):
  répéter k fois:
    choisir a aléatoirement entre 1 et N-1
    si a**(N-1) % N != 1:
      renvoyer FAUX
  renvoyer VRAI
```

On a alors:

$$P(\text{est-premier}(N) = \text{VRAI} \mid N \text{ non premier}) \leq \frac{1}{2^k}$$

Remarque 36 En réalité, cette majoration n'est pas toujours valide. En effet, le petit théorème de Fermat n'est pas une équivalence: il existe des nombres non premiers N tels que l'exécution de est_premier(N) renverrai toujours VRAI. De tels nombres sont appelés des nombres de **Carmichael**.
Il s'agit donc d'un algorithme de Monte Carlo.

B. Classes de compléxité probabiliste [VAZ A.4]

Définition 37 On définit la classe de complexité **RP** (*Randomized polynomial*) comme les problèmes qui possèdent un algorithme de Monte Carlo qui rejette toujours une instance négative, et qui accepte au moins 1 fois sur 2 une instance positive.

Propriété 38 P ⊆ RP ⊆ NP

Définition 39 On définit la classe de complexité **ZPP** (*Zero-error probabilistic polynomial*) comme comme les problèmes qui possèdent un algorithme de Las Vegas, dont le temps de calcul est polynomial en espérance.

Propriété 40 ZPP = RP ∩ co − RP.

C. Dérandomisation [TOR 13.4.1.2]

Problème 41 Le problème MAXSAT est un problème d'optimisation associé à SAT. Étant donné une formule propositionnelle en forme normale conjonctive, on cherche une valuation satisfiant un nombre maximal de clauses.

Motivation 42 Le problème MAXSAT est NP-difficile, mais raisonner en termes de probabilités permet de construire un algorithme simple donnant une solution approché au problème.

Théorème 43 [TOR Théorème 13.21] Soit une formule φ en FNC avec c clauses de au moins k littéraux indépendants, alors l'esperance $\mathbb{E}(\varphi)$ du nombre de clauses satisfaites par une valuation aléatoire vérifie :

$$\mathbb{E}(\varphi) \geq c\left(1 - \frac{a}{2^k}\right)$$

DEV

Méthode 44 **Dérandomisation par l'espérance conditionnelle** On peut construire un algorithme glouton dont les choix reposent sur des calculs d'espérances ce qui supprime le caractère aléatoire de l'algorithme.

Propriété 45 Soit φ une formule et opt_φ le nombre maximal de clauses de φ simultanément satisfiable, alors la valuation renvoyé par la méthode 44 satisfait au moins $\frac{\text{opt}_\varphi}{2}$ clauses de φ.

Exemples d'algorithmes d'approximations et d'algorithmes probabilistes

1 Def [COR3 34.1]
2 Def
3 Def
4 Ex
5 Rem

I. Algorithmes d'approximation

6 Def Un algorithme d'approximation [NAN]

A. Facteur d'approximation

7 Def Facteur d'approximation[PAPA 9.2]
8 Rem
9 Ex Couverture de sommets[PAPA 9.2.1]
10 Def [PAPA 9.2.1]
11 Prop [PAPA 9.2.1]
12 Algo [COR3 35.1]
13 Prop [PAPA 9.2.1]

B. Algorithmes Gloutons

14 Def [PAPA 5]
15 Ex Couverture d'ensemble[PAPA 5.4]
16 Prop [PAPA 5.4]
17 Ex Voyageur de commerce métrique
18 Rem
19 Ex Coloration de graphe[TOR 9.3]
20 Algo [TOR Programme 9.6]
21 Rem
22 Algo

C. Schéma d'approximation

23 Def Schéma d'approximation [VAZ 8]
24 Def Schéma d'approximation polynomial [VAZ 8]
25 Def Schéma d'approximation totalement polynomial[VAZ 8]
26 Ex

II. Algorithmes probabilistes

A. Définitions et exemples

27 Def Algorithme probabiliste [TOR 9.6]
28 Rem Méthode probabiliste [TOR 13.4.1.2]
29 Def [PAPA 8.1]
30 Ex [PAPA 5.1]
31 Prop [PAPA 5.1]
32 Def Monte-Carlo/Las Vegas [TOR 9.6]
33 Ex [TOR 9.6]
34 Ex Test de primalité
35 Algo Test de primalité de Fermat
36 Rem

B. Classes de compléxité probabiliste [VAZ A.4]

37 Def
38 Prop
39 Def
40 Prop

C. Dérandomisation[TOR 13.4.1.2]

41 Prob
42 Motiv
43 Thm [TOR Théorème 13.21]
44 Métho Dérandomisation par l'espérance conditionnelle
45 Prop

Remarque

- Dans la tortue, il y a deux sections utiles pour cette leçon : la section 9.6 sur les algorithmes probabilistes, et la section 13.4 sur les algorithmes d'approximation (en particulier pour la section)

Bibliographie

[COR3] T. H. Cormen, *Introduction à l'algorithmique (3rd édition).*

[PAPA] S. Dasgupta & C. Papadimitriou & U. Vazirani, *Algorithms.*

[TOR] T. Balabonski & S. Conchon & J. Filliâtre & K. Nguyen & L. Sartre, *MP2I MPI, Informatique Cours et exercices corrigés.*

[VAZ] V. V. Vazirani, *Algorithmes d'approximation.*