

Approximation et Probabilités

MAX-CUT

Entrée: Un graphe G non orienté pondéré (V, E, w)

Sortie: Une coupe maximale de G .

On considère l'algorithme suivant :

Chaque sommet $v \in V$ est placé indépendamment dans S avec probabilité $1/2$.

1. Pour une arête $e = \{u, v\}$, on définit une variable aléatoire X_e qui vaut 1 si l'arête est coupée et 0 sinon. Calculer $\mathbb{E}[X_e]$.
2. Montrer que l'espérance de la valeur de la coupe produite par l'algorithme vaut

$$\mathbb{E}[\text{CUT}] = \frac{1}{2} \sum_{e \in E} w(e).$$

3. En déduire que cet algorithme est un algorithme d'approximation randomisé de rapport $\frac{1}{2}$ pour le problème MAX-CUT pondéré.
4. Montrer que ce facteur $\frac{1}{2}$ est optimal pour cet algorithme.
5. On fixe désormais les sommets un par un. Montrer qu'à chaque étape, il est possible de choisir l'affectation d'un sommet de manière à ne pas faire décroître l'espérance conditionnelle de C par rapport au nouveau choix.
6. En déduire un algorithme déterministe polynomial calculant une coupe de poids au moins $\frac{1}{2} \sum_{e \in E} w(e)$.

On pose

$$C = \sum_{e \in E} w(e) X_e.$$

7. Les variables $(X_e)_{e \in E}$ sont-elles indépendantes ? Justifier.
8. On modélise l'algorithme par des variables aléatoires $(Y_v)_{v \in V}$, où $Y_v = 1$ si $v \in S$ et 0 sinon. Montrer que C est une fonction Lipschitz des variables (Y_v) .
9. En déduire une inégalité de concentration de type Azuma–Hoeffding :

$$\mathbb{P}(|C - \mathbb{E}[C]| \geq t) \leq 2 \exp \left(- \frac{2t^2}{\sum_{v \in V} D(v)^2} \right),$$

où $D(v)$ désigne la somme des poids des arêtes incidentes à v .

10. Interpréter cette inégalité selon la structure du graphe.

On associe à chaque sommet $v \in V$ un vecteur unitaire $x_v \in \mathbb{R}^n$. On choisit un vecteur aléatoire r uniformément sur la sphère unité et on place v dans S si $\langle r, x_v \rangle \geq 0$.

11. Montrer que la probabilité qu'une arête $\{u, v\}$ soit coupée ne dépend que de l'angle θ_{uv} entre x_u et x_v .
12. On admet que cette probabilité vaut θ_{uv}/π . Comparer ce résultat avec l'algorithme de la partie I.
13. Expliquer pourquoi ce principe permet d'obtenir un facteur d'approximation strictement supérieur à $\frac{1}{2}$.

Solution: Approximation et Probabilités

1. Soit $e = \{u, v\}$. L'arête est coupée si et seulement si u et v sont placés dans des parties différentes. Les placements étant indépendants,

$$\mathbb{P}(X_e = 1) = \mathbb{P}(u \in S, v \notin S) + \mathbb{P}(u \notin S, v \in S) = \frac{1}{2}.$$

Ainsi $\mathbb{E}[X_e] = \frac{1}{2}$.

2. Par linéarité de l'espérance,

$$\mathbb{E}[\text{CUT}] = \mathbb{E}\left[\sum_{e \in E} w(e)X_e\right] = \sum_{e \in E} w(e)\mathbb{E}[X_e] = \frac{1}{2} \sum_{e \in E} w(e).$$

3. Pour toute coupe, on a $\text{CUT} \leq \text{OPT} \leq \sum_{e \in E} w(e)$. Ainsi

$$\mathbb{E}[\text{CUT}] \geq \frac{1}{2} \text{OPT},$$

ce qui montre que l'algorithme est une approximation randomisée de facteur $\frac{1}{2}$.

4. Considérons un graphe biparti complet équilibré avec poids uniformes. Dans ce cas, $\text{OPT} = \sum_{e \in E} w(e)$, mais l'algorithme randomisé coupe chaque arête avec probabilité $\frac{1}{2}$. Le facteur $\frac{1}{2}$ est donc optimal pour cet algorithme.
5. Supposons les sommets $v_1, \dots, v_{k-1}$ déjà fixés. Pour v_k , on compare les deux choix possibles ($v_k \in S$ ou non) et leurs espérances conditionnelles. La moyenne de ces deux valeurs est l'espérance conditionnelle actuelle, donc au moins l'un des deux choix préserve cette valeur.
6. En procédant ainsi pour tous les sommets, on obtient une coupe déterministe telle que

$$\text{CUT} \geq \frac{1}{2} \sum_{e \in E} w(e).$$

L'algorithme est polynomial.

7. Les variables X_e ne sont pas indépendantes en général : deux arêtes partageant un sommet dépendent de la position de ce sommet.
8. La variable C dépend des affectations $(Y_v)_{v \in V}$. Modifier la valeur d'un sommet v n'affecte que les arêtes incidentes à v . La variation maximale de C est donc majorée par

$$D(v) = \sum_{e \ni v} w(e).$$

Ainsi C est une fonction Lipschitz des Y_v .

9. En ordonnant les sommets et en révélant leur affectation un à un, on obtient une martingale (M_k) . L'inégalité d'Azuma-Hoeffding donne

$$\mathbb{P}(|C - \mathbb{E}[C]| \geq t) \leq 2 \exp\left(-\frac{2t^2}{\sum_{v \in V} D(v)^2}\right).$$

10. Si les degrés pondérés sont bien répartis (par exemple graphe presque régulier), la somme $\sum_v D(v)^2$ est modérée et la concentration est forte. À l'inverse, un sommet de très fort degré dégrade la borne.
11. La probabilité que $\{u, v\}$ soit coupée correspond à la probabilité que $\langle r, x_u \rangle$ et $\langle r, x_v \rangle$ aient des signes opposés. Cette probabilité ne dépend que de l'angle θ_{uv} entre x_u et x_v .
12. On admet

$$\mathbb{P}(\{u, v\} \text{ coupée}) = \frac{\theta_{uv}}{\pi}.$$

Cette quantité peut être strictement supérieure à $\frac{1}{2}$ lorsque $\theta_{uv} > \frac{\pi}{2}$.

13. En choisissant les vecteurs x_v de manière optimale (via une relaxation semi-définie), on maximise l'espérance de la coupe, ce qui permet de dépasser le facteur $\frac{1}{2}$. L'algorithme de Goemans–Williamson atteint un facteur $\simeq 0,878$.
14. La randomisation permet d'éviter les pires cas déterministes, de simplifier l'analyse par linéarité de l'espérance et de produire des solutions de bonne qualité en moyenne.
15. Exemples classiques : MAX-SAT, Set Cover, Vertex Cover, ou encore l'algorithme randomisé pour l'arrondi de relaxations linéaires.

Coloriages

On s'intéresse au problème suivant :

k -COL

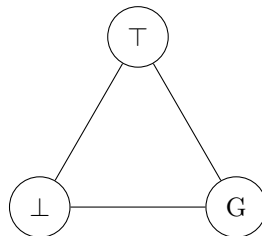
Entrée: Un graphe non orienté $G = (V, E)$.

Sortie: Décider si G est k -coloriable.

1. Montrer que pour tout $k \in \mathbb{N}^*$, si $(k+1)$ -COL $\in P$, alors k -COL $\in P$.

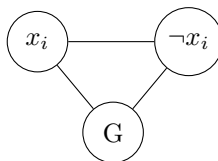
Nous montrons maintenant que 3-SAT se réduit polynomialement à 3-COL. On introduit plusieurs « gadgets » pour faciliter la réduction.

Gadget de couleurs : Les trois couleurs représentent *vrai* ($\top$), *faux* ($\perp$) et une couleur auxiliaire G :

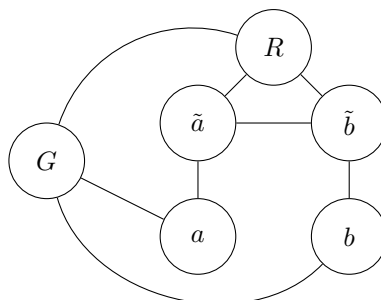


2. Que fait ce gadget dans un graphe ?
3. Trouver un gadget imposant qu'un sommet s soit colorié comme $\top$.

Gadget de littéraux : Pour chaque variable x_i , on crée deux sommets x_i et $\neg x_i$, reliés entre eux et à G . Ainsi, exactement l'un des deux littéraux est colorié en $\top$.



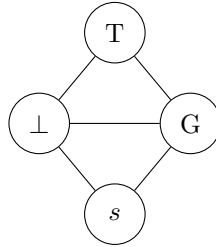
Gadget OR : Supposons a et b coloriés en $\top$ ou $\perp$. Le sommet R doit représenter $a \vee b$.



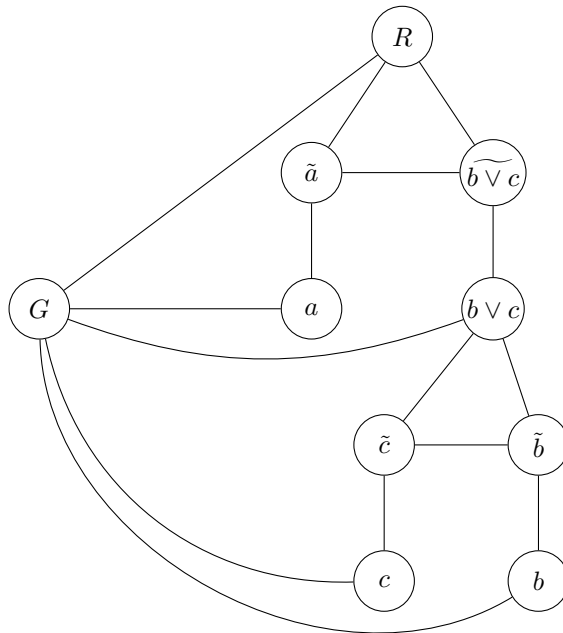
4. Montrer que R a la couleur de a ou celle de b , et que si a et b ont des couleurs différentes, alors R peut être colorié arbitrairement.
5. Construire un gadget OR à trois entrées a, b, c .
6. Construire un gadget vérifiant une clause $(v_1 \vee v_2 \vee v_3)$.
7. Donner une réduction polynomiale de 3-SAT à 3-COL.

Solution: Coloriages

1. Il suffit d'ajouter un sommet relié à tous les autres. Si le graphe obtenu est $(k + 1)$ -coloriable, alors en retirant ce sommet on obtient une k -coloration du graphe initial.
2. Il force l'utilisation de 3 couleurs et empêche la 2-coloration.
3. On relie s à $\perp$ et à G :



4. Si ce n'était pas le cas, a et b sont coloriés en $\top$ et $\perp$. Puisque R est connecté à G , alors il est colorié en $\top$ ou $\perp$, selon le coloriage de $\tilde{a}$ et $\tilde{b}$. Si a et b étaient de la même couleur, alors $\tilde{a}$ et $\tilde{b}$ sont coloriés en les deux couleurs complémentaires et donc R est de cette même couleur de départ.
5. On chaîne deux gadgets OR :



6. On applique le gadget OR à trois entrées, puis on relie le sommet résultat R à G et à $\perp$, ce qui force R à être colorié en $\top$.
7. Chaque variable introduit un gadget de taille constante, chaque clause un gadget OR de taille constante. La taille du graphe est linéaire en la taille de la formule, la réduction est donc polynomiale.

Géographie et PSPACE-complétude

On s'intéresse à la complexité en espace du problème GEOGRAPHY sur un graphe orienté.

GEOGRAPHY

Entrée: Un graphe orienté $G = (V, E)$ et un sommet de départ $s \in V$.
Sortie: Décider si le joueur 1 a une stratégie gagnante.

Quelques définitions :

- $\text{PSPACE} = \bigcup_{k \in \mathbb{N}} \text{SPACE}(n^k)$ est la classe des langages décidables en espace polynomial.
 - QBF est le problème de la validité d'une formule booléenne quantifiée. C'est le problème de référence pour prouver la PSPACE-dureté.
 - Une réduction en espace logarithmique ($A \leq_L B$) est une fonction calculable en espace $O(\log n)$ qui transforme une instance de A en une instance de B . Puisque $L \subseteq P$, toute réduction en espace logarithmique est aussi une réduction polynomiale.
1. Montrer que $\text{GEOGRAPHY} \in \text{PSPACE}$ en proposant un algorithme de type *backtracking*.

On souhaite établir que GEOGRAPHY est PSPACE-difficile via une réduction depuis le problème QBF. Soit φ une formule close de la forme :

$$\varphi = \exists x_1 \forall x_2 \exists x_3 \dots \forall x_n (C_1 \wedge C_2 \wedge \dots \wedge C_m)$$

2. Pourquoi peut-on supposer « sans perte de généralité » que n est pair et que les quantificateurs alternent strictement ?
3. Décrire le mécanisme de la réduction en expliquant comment les choix des joueurs sur les variables influencent l'issue de la partie dans les clauses.
4. Justifier que cette réduction s'effectue en espace polynomial.

Le problème PLANAR-GEOGRAPHY restreint les instances aux graphes orientés planaires.

5. Montrer que PLANAR-GEOGRAPHY est PSPACE-complet.

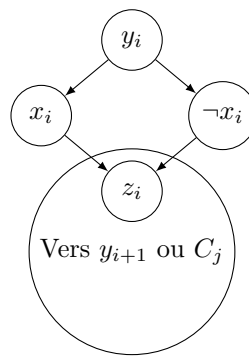
On a vu que GEOGRAPHY sur les graphes orientés est PSPACE-complet. On considère ici la variante UNDIRECTED-GEOGRAPHY où le graphe $G = (V, E)$ est non-orienté.

6. Rappeler pourquoi la règle « un sommet ne peut être visité qu'une fois » transforme un jeu sur un graphe non-orienté en un problème de recherche de chemins simples.
7. Soit M un couplage maximum de G . Montrer que si le sommet de départ s est couvert par **tous** les couplages maximums de G , alors le premier joueur possède une stratégie gagnante.
8. En déduire que $\text{UNDIRECTED-GEOGRAPHY} \in \text{P}$.

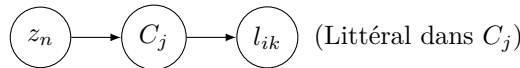
Solution: Géographie et PSPACE-complétude

1. On définit une fonction récursive $Gagnant(u, V_{visites})$. Le joueur 1 gagne en u s'il existe un voisin v tel que $v \notin V_{visites}$ et $Gagnant(v, V_{visites} \cup \{v\})$ est faux. Comme un sommet n'est visité qu'une fois, la profondeur de la pile est au plus $|V|$. L'espace utilisé est donc $O(|V|^2)$, ce qui est polynomial.
2. Si n est impair, on ajoute une variable $\forall x_{n+1}$ muette. Si deux quantificateurs identiques se suivent ($\exists x_i \exists x_{i+1}$), on peut insérer un quantificateur opposé portant sur une variable factice pour forcer l'alternance des tours de jeu.
3. On construit un bout de graphe. Le sommet y_i appartient au joueur $i \bmod 2$. S'il choisit le sommet x_i , le littéral est vrai. S'il choisit $\neg x_i$, le littéral est faux.

Gadget Variable :



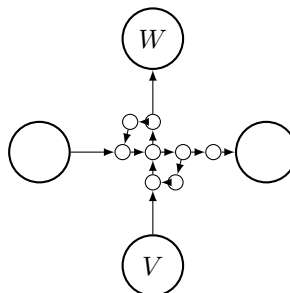
Gadget Clause : Après z_n , le Joueur 2 (qui vient de perdre son tour sur $\forall x_n$) choisit une clause C_j .



Si le littéral l_{ik} a été choisi (sommet visité), le Joueur 2 ne peut plus bouger de l_{ik} et perd. Ainsi, le Joueur 1 gagne si φ est vraie.

4. La construction produit $4n + m$ sommets. La description de V et E se fait par un parcours simple de la formule, nécessitant un espace logarithmique $O(\log |\varphi|)$, ce qui est bien dans PSPACE.
5. La difficulté principale pour prouver la PSPACE-complétude sur graphes planaires est l'élimination des croisements d'arêtes.

L'objectif est de permettre à deux chemins de se croiser (par exemple une arête horizontale $A \rightarrow B$ et une arête verticale $V \rightarrow W$) sans qu'un joueur ne puisse « dévier » sur le chemin de l'autre.



- Si un joueur entre dans le gadget par le sommet V , il doit ressortir par W sous peine de perdre immédiatement ou de donner l'avantage à l'adversaire.
 - Le jeu sur le graphe transformé (planaire) a exactement la même issue que sur le graphe original avec croisements.
 - Comme chaque croisement est remplacé par un nombre constant de sommets et d'arêtes, la transformation reste polynomiale en temps et en espace.
6. Dans un graphe non-orienté, sans la règle de visite unique, les joueurs pourraient faire des allers-retours infinis sur une même arête. La règle garantit que la partie s'arrête en au plus $|V|$ coups, rendant le jeu fini et sans cycle.
7. Supposons que s appartienne à tous les couplages maximums.
- Le Joueur 1 choisit un couplage maximum M . Puisque s est couvert, il existe une arête $\{s, v\} \in M$. Le Joueur 1 joue v .
 - À chaque coup du Joueur 2 vers un sommet w , le Joueur 1 répond en suivant l'arête du couplage M contenant w .
 - Si le Joueur 1 était bloqué, cela signifierait qu'il a parcouru un chemin alternant commençant par s et se terminant sur un sommet non couvert par M . En inversant les arêtes le long de ce chemin, on obtiendrait un couplage maximum ne contenant pas s , ce qui contredit l'hypothèse.
8. Pour décider si le Joueur 1 gagne, il suffit de vérifier si s est dans tous les couplages maximums :
- (a) Calculer la taille k d'un couplage maximum de G .
 - (b) Calculer la taille k' d'un couplage maximum de $G \setminus \{s\}$.
 - (c) Le Joueur 1 gagne si et seulement si $k' < k$.

Le calcul d'un couplage maximum se fait en temps polynomial via l'algorithme d'Edmonds (*Blossom algorithm*) en $O(|E|\sqrt{|V|})$. Le problème est donc dans P.