352

ASTÉRISQUE

2013

SÉMINAIRE BOURBAKI
VOLUME 2011/2012
EXPOSÉS 1043-1058

(1045) *Difficulté d'approximation*

Pierre PANSU

SOCIÉTÉ MATHÉMATIQUE DE FRANCE
Publié avec le concours du CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE

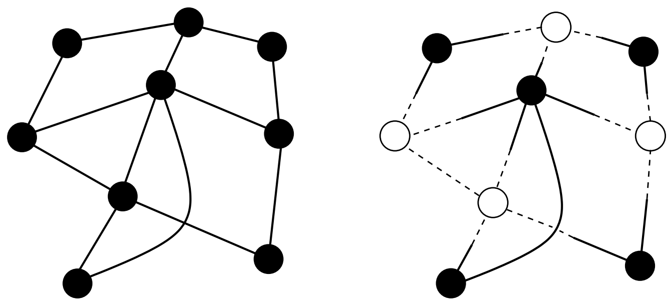
Séminaire BOURBAKI Novembre 2011
64e année, 2011-2012, n° 1045, p. 83 à 120

DIFFICULTÉ D'APPROXIMATION [d'après Khot, Kindler, Mossel, O'Donnell,...]

par **Pierre PANSU**

INTRODUCTION

Ce soir, je dois recevoir n invités. Je dois les répartir sur deux tables. Je les connais bien, je sais quelle solide inimitié certains éprouvent pour d'autres. Par exemple, je dois absolument éviter de placer L... et M... à la même table. Et de même pour N... et P... Mais il y a trop de couples ennemis. Je vais tout de même chercher un plan de table qui maximise le nombre de couples séparés. En termes combinatoires, je forme un graphe dont les sommets sont mes invités et les arêtes les liens d'inimitié. Il s'agit de trouver un coloriage des sommets en deux couleurs (noir et blanc) qui maximise le nombre d'arêtes bicolores. Par exemple, le graphe ci-dessous à gauche a 13 arêtes, le coloriage proposé à droite a 11 arêtes bicolores, on ne peut pas faire mieux. Je dis que la *coupe maximale* du graphe vaut 11.



Le problème MAX CUT consiste à écrire un algorithme qui prend en entrée un graphe à n sommets et un entier k et retourne un coloriage avec au moins k arêtes bicolores, s'il en existe, ou bien s'arrête s'il n'en existe pas.

(*) Ce travail a bénéficié d'une aide de l'Agence Nationale de la Recherche portant la référence ANR-10-BLAN 0116.

C'est le prototype des problèmes de *recherche* (il s'agit de trouver une solution à un ensemble de contraintes). On remarque que, étant donné un coloriage, vérifier s'il y a au moins k arêtes bicolores peut se faire rapidement, en temps polynomial (ici, au plus quadratique) en la taille du coloriage, i.e en le nombre de sommets. Cette propriété caractérise les problèmes de recherche de la *classe* NP. Cette classe est très vaste. Elle contient en particulier le problème de trouver une preuve d'un théorème (vérifier une preuve convenablement écrite peut se faire en temps polynomial en la longueur de la preuve). Néanmoins, MAX CUT est NP-*complet* au sens suivant. Tout problème de la classe NP se ramène à MAX CUT, par un prétraitement qui ne prend qu'un temps polynomial en la taille des données. Si un jour on trouve un algorithme qui résout MAX CUT en temps polynomial, il en sera de même pour tous les problèmes NP. Peu de gens y croient (c'est la célèbre conjecture $P \neq NP$), si bien que la résolution de MAX CUT est considérée comme inaccessible au calcul.

À défaut d'une solution exacte, on se contenterait volontiers d'une solution approchée. Étant donné $\alpha < 1$, une α-*approximation* de MAX CUT est un algorithme polynomial qui retourne un coloriage réalisant α fois la coupe maximale. Cette notion s'étend à tout problème d'optimisation combinatoire, où les instances sont des fonctions à valeurs réelles sur des ensembles finis qu'il s'agit de maximiser (pour les problèmes de minimisation, on garde la même terminologie, avec $\alpha > 1$). On s'autorise des tirages au hasard[(1)]. Dans ce cas, on demande que la solution retournée par l'algorithme soit satisfaisante avec probabilité $> 1/2$.

DÉFINITION 0.1. — *Soit* $\alpha < 1$. *On dit qu'un problème d'approximation combinatoire est* α-approximable *s'il possède une* α-*approximation.*

Par exemple, tirer la couleur de chaque sommet au hasard indépendamment donne une 1/2-approximation de MAX CUT.

On peut faire mieux. En 1994, Michel Goemans et David Williamson [**34**] ont proposé une α-approximation de MAX CUT pour tout $\alpha < \alpha_{GW} = 0.8785672057848516\ldots$ On a de bonnes raisons de penser que cette borne est optimale, i.e. que le problème MAX CUT est NP-difficile à approcher au-delà de la constante α_{GW}. Subhash Khot, Guy Kindler, Elchanan Mossel et Ryan O'Donnell [**43**] l'ont prouvé sous une hypothèse *a priori* plus forte que $P \neq NP$, connue sous le nom de Conjecture des Jeux Uniques (UGC).

Dans cet exposé, on donnera un aperçu :

(1) C'est une commodité. Cela ne joue un rôle essentiel dans aucun des algorithmes présentés ci-dessous. Tous peuvent être transformés en algorithmes déterministes.

– de l'étonnante efficacité des relaxations semi-définies, méthode de construction d'algorithmes inaugurée par M. Goemans et D. Williamson (d'aucuns la font remonter à [**55**]) ;
– des questions mathématiques qui surgissent de l'analyse de ces relaxations semi-définies ;
– de l'émergence de résultats de difficulté d'approximation, depuis le théorème PCP.

Le présent texte constitue une suite de l'exposé de Bernard Chazelle dans ce séminaire, [**20**], dont la lecture est chaudement recommandée.

Je remercie les participants du groupe de lecture de complexité algorithmique (ENS, 2009-2010) et du trimestre « Géométrie métrique, groupes et algorithmes » (IHP, janvier-mars 2011) pour leur aide au fil des mois, et Assaf Naor pour sa vision d'ensemble du sujet.

1. UN ALGORITHME D'APPROXIMATION POUR MAX CUT

1.1. Relaxations semi-définies

On se propose de donner une idée de la méthode de relaxation semi-définie. Il s'agit d'un procédé de construction d'algorithmes d'approximation. Il n'est pas systématique (il n'y a pas une relaxation semi-définie canonique pour chaque problème combinatoire). On peut seulement présenter la démarche sur des exemples. Nous avons choisi trois problèmes d'optimisation qui ont donné lieu à des développements mathématiques intéressants : MAX CUT, SPARSEST CUT et MAX ACYCLIC SUBGRAPH. On commence par MAX CUT. Ce problème a tout pour plaire : un intérêt historique, c'est le premier succès de la méthode ; un saut d'intégralité intéressant, une étude d'inapproximabilité spectaculaire.

1.2. L'algorithme de Goemans et Williamson pour MAX CUT

THÉORÈME 1.1 (M. Goemans et D. Williamson, [**34**]). — *MAX CUT est α-approximable pour tout $\alpha < \alpha_{GW} = 0.878\ldots$.*

1.2.1. *Arithmétisation.* — On commence par formuler algébriquement le problème combinatoire. Soient G un graphe, V l'ensemble de ses sommets, $E \subset V \times V$ l'ensemble de ses arêtes. On choisit de représenter un coloriage de G par une fonction $x : V \to \{-1, 1\}$. On choisit d'exprimer le nombre d'arêtes bicolores par la fonction

$$OBJ(x) = \sum_{(u,v)\in E} \frac{1}{2}(1 - x_u x_v).$$

Calculer la coupe maximale de G, c'est maximiser la fonction OBJ sur le cube discret $\{-1,1\}^V$, i.e. sur l'ensemble des fonctions $x : V \to \mathbb{R}$ qui satisfont aux contraintes

$$\forall v \in V,\ x_v^2 = 1.$$

Notons $OBJ(G)$ le maximum.

1.2.2. *Relaxation.* — On plonge le cube discret dans un espace plus vaste. On se donne un espace euclidien ℓ_2 et on considère les applications $y : V \to \ell_2$. On remplace chaque produit $x_u x_v$ par un produit scalaire $y_u \cdot y_v$. Les contraintes deviennent

$$\forall v \in V,\ y_v \cdot y_v = 1.$$

La fonction objectif devient

$$SDP(y) = \sum_{(u,v)\in E} \frac{1}{2}(1 - y_u \cdot y_v).$$

Remarquer que la restriction de SDP aux applications y qui prennent leurs valeurs dans une droite coïncide avec OBJ. Il en est de même des contraintes, donc

$$\max SDP \geq \max OBJ.$$

Admettons pour l'instant qu'il existe un algorithme qui, avec une précision arbitraire, détermine en temps polynomial l'application $y_{\max} : V \to \ell_2$ qui maximise SDP sous les contraintes imposées. Notons $SDP(G)$ le maximum. La fonction

$$n \mapsto \min_{|V|=n} \frac{OBJ(G)}{SDP(G)}$$

s'appelle le saut d'intégralité *(integrality gap)* de la relaxation choisie.

1.2.3. *Procédure d'arrondi.* — Pour compléter la méthode en un algorithme d'approximation, on construit maintenant un coloriage x à partir de la solution $y_{\max}$ du problème continu. Cela donnera simultanément une minoration du saut d'intégralité, et donc du facteur d'approximation réalisé par l'algorithme.

On voit $y_{\max}$ comme un plongement du graphe G dans la sphère unité. On tire un hyperplan vectoriel H uniformément au hasard. Il sépare les sommets en deux parties, voilà le coloriage x_H cherché.

1.2.4. *Analyse.* — Pour chaque arête (u,v), la probabilité que H sépare y_u de y_v vaut $\frac{1}{\pi}\arccos(y_u \cdot y_v)$. En effet, l'intersection de l'hyperplan avec le plan engendré par y_u de y_v est une droite vectorielle tirée uniformément au hasard dans ce plan, la probabilité qu'elle sépare y_u de y_v est proportionnelle à l'angle entre y_u et y_v. Par conséquent, l'espérance du nombre d'arêtes bicolores dans le coloriage aléatoire obtenu est

$$\mathbb{E}_H(OBJ(x_H)) = \sum_{(u,v)\in E} \frac{1}{\pi} \arccos(y_u \cdot y_v).$$