Méthode probabiliste et dérandomisation

François Schwarzentruber

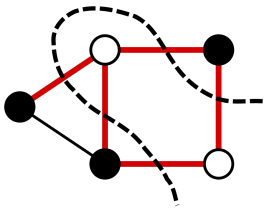
— La *méthode probabiliste* est l'art de démontrer l'existence d'objets par des arguments probabilistes.
— La *dérandomisation* consiste à construire un algorithme déterministe à partir d'un algorithme probabiliste, en gardant les même garanties.

1 MAXCUT

Définition 1 (taille d'une coupe) Soit $G = (V, E)$ un graphe non orienté. Soit $C = \{V_0, V_1\}$ avec $V_0 \sqcup V_1 = V$, une coupe de G. La taille de C, notée $\#C$, est le nombre d'arêtes allant d'un sommet de V_0 à un sommet de V_1.

Définition 2 (coupe maximale) Une coupe maximale est une coupe de taille maximale.

Exemple 3 (coupe maximale de taille 5)



Définition 4 **MAXCUT**
entrée : un graphe G non orienté
sortie : une coupe maximale

1.1 Algorithme naïf

Placer chaque sommet de manière aléatoire uniforme dans V_0 ou dans V_1.

1.2 Existence d'une solution

Théorème 5 Soit $G = (V, E)$ une graphe non orienté. Il existe une coupe de G de taille au moins $\frac{|E|}{2}$.

1.3 Algorithme d'approximation probabiliste en moyenne

Définition 6 Étant donné un problème d'optimisation, un algorithme probabiliste $\mathcal{A}$ est une approximation de facteur $\rho(n)$ *en moyenne* si pour toute entrée de taille n, $\mathcal{A}$ calcule une solution `sol` avec

$$\begin{cases} \frac{\mathbb{E}(\text{coût}(\texttt{sol}))}{\text{coût d'une solution optimale}} \leq \rho(n) & \text{si c'est un problème de minimisation ;} \\ \frac{\mathbb{E}(\text{coût}(\texttt{sol}))}{\text{coût d'une solution optimale}} \geq \rho(n) & \text{si c'est un problème de maximisation.} \end{cases}$$

Proposition 7 L'algorithme probabiliste naïf pour le calcul d'une coupe est une $\frac{1}{2}$-approximation de **MAXCUT** en moyenne.

Proposition 8 La probabilité que l'algorithme naïf renvoie une coupe de taille $\geq \frac{|E|}{2}$ est d'au moins $\frac{1}{1+\frac{|E|}{2}}$.
Autrement dit :

$$\mathbb{P}(\#C \geq \frac{|E|}{2}) \quad \geq \quad \frac{1}{1+\frac{|E|}{2}}.$$

Proposition 9 Le nombre moyen de répétitions de l'algorithme naïf pour trouver une coupe de taille au moins $\frac{|E|}{2}$ est $1 + \frac{|E|}{2}$.

Remarque 10 On obtient donc un algorithme de type Las Vegas pour le calcul d'une coupe contenant au moins la moitié des arêtes.

2 Dérandomisation

Question 11 Les algorithmes probabilistes sont-ils nécessaires ?

2.1 Méthode des probabilités conditionnelles

Cette approche est dûe à Erdös et Selfridge [ES73]. Illustrons la sur l'algorithme naïf probabiliste pour **MAXCUT**. On note C la coupe calculée par l'algorithme probabiliste, et $\#C$ sa taille. Il y a une exécution de l'algo probabiliste naïf avec $\#C \geq \frac{|E|}{2}$.

Idée de l'algorithme déterministe.
Placer de manière déterministe les sommets dans V_0 ou V_1 de façon à conserver l'invariant

$$\mathbb{E}(\#C \mid \text{choix déterministes déjà faits}) \geq \frac{|E|}{2}.$$

🎓écrire le détail en maths

2.2 État de l'art

Le meilleur ratio pour **MAXCUT** est environ 1/0.878, voir [GW95]. C'était la première fois que la programmation semidéfinie est utilisée pour concevoir un algo d'approximation. C'est une forme d'optimisation avec des expressions quadratiques. Une instance se transforme en :

$$\begin{array}{l} \text{maximiser } \sum_{(i,j)\in E} \frac{1-v_i v_j}{2} \\ \left\{ \; v_i \in \{-1,1\} \right. \end{array}$$

où $v_i = -1$ signifie que $i \in X$ et $v_i = 1$ que $i \in Y$. Puis ils font de la relaxation, et algo probabiliste (on fera pareil après pour **MAXSAT**). La borne 1/0.878 est optimale si la *conjecture des jeux uniques* est vraie [KKMO07].

3 Théorème d'Adleman

Théorème 12 (Théorème d'Adleman) [Adl78] Tout problème de décision L dans RP peut se dérandomiser de manière non-uniforme : il existe un polynôme P, et pour tout entier n, il existe un algorithme déterministe qui décide l'ensemble des instances positives de taille n en temps $P(n)$.

Définition 13 P/*poly* est la classe des problèmes de décision L qui admette un algorithme A en temps polynomial, et une suite de mots $\alpha_0, \alpha_1, \ldots$ de longueur $poly(n)$ tels que pour toute instance de taille n,

$$x \in L \text{ ssi } A(x, \alpha_n) \text{ renvoie 'oui'.}$$

Théorème 14 (théorème d'Adleman) RP $\subseteq$ P/*poly*.

Références

[Adl78] Leonard M. Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science, Ann Arbor, Michigan, USA, 16-18 October 1978*, pages 75–83. IEEE Computer Society, 1978.

[ES73] Paul Erdoes and JL Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3) :298–301, 1973.

[GW95] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6) :1115–1145, 1995.

[KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps ? *SIAM Journal on Computing*, 37(1) :319–357, 2007.