P=NP ? Casser la barrière de la complexité ?

Une éloge de l'approximation et du hasard

Richard Lassaigne
IMJ/Equipe de Logique mathématique,
CNRS-Université Paris Diderot

- Comment casser la barrière de la complexité
  si **P≠NP** ?

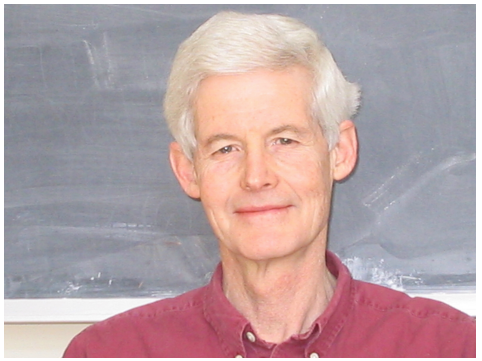
- L'**approximation** permet-elle d'obtenir
  des algorithmes plus **efficaces** ?

- Comment **approximer** les problèmes de **comptage** ?

- Peut-on utiliser des algorithmes **probabilistes** ?

- Et la complexité en **espace** ?
  Un exemple où elle est primordiale : la **vérification**

- S. Cook (1971)
  Le problème SAT est NP-complet



- R. Karp (1972) :
  Réduction entre problèmes combinatoires
  "Computational intractability is the rule rather than the exception"

- Lettre de K. Gödel
  à J. Von Neumann (1956) :
  Formule une question sur la possibilité de calculer une borne linéaire ou quadratique pour la longueur des preuves dans la logique du premier ordre

Classes de complexité et Logique

Peut-on refléter la complexité (structurelle) d'un problème par la complexité logique de la formule définissant ce problème ?

Exemple : 3-coloriage d'un graphe

il existe une partition des sommets en 3 sous-ensembles t.q. les extrémités de toute arête soient de 2 couleurs différentes

Théorème (R. Fagin, 74) : Pour toute propriété sur les structures finies, le problème de décision associé est dans NP ssi il est définissable par une formule existentielle du 2e ordre

Problème : caractérisation logique de le classe P (Immerman, Vardi, etc...)

Différents types de problèmes

- Décision : Existe-t-il une solution ?
- Recherche : Trouver une solution
- Comptage : Compter les solutions
- Génération : Engendrer les solutions (de manière uniforme)

Voyageur de commerce (TSP)

- Entrée : Un graphe valué, un entier k
- Question : Existe-t-il un circuit hamiltonien dont le coût global soit $\leq k$?

Recouvrement de sommets (VC)

- Entrée : Un graphe , un entier k
- Question : Existe-t-il un recouvrement de taille $\leq k$?

Sous-graphe complet (CLIQUE)

- Entrée : Un graphe , un entier k
- Question : Existe-t-il un sous-graphe complet de taille $\geq k$?

Colorabilité d'un graphe (COL)

- Entrée : Un graphe , un entier $k \geq 3$
- Question : Existe-t-il un k-coloriage des sommets du graphe ?

Satisfaisabilité (SAT)

- Entrée : Une formule propositionnelle sous forme CNF
- Question : Existe-t-il une valuation satisfaisant la formule ?

- P classe des problèmes décidables en temps polynomial

- NP classe des problèmes vérifiables en temps polynomial

$A \prec B$ s'il existe une fonction f calculable en temps polynomial
t.q. $x \in A$ ssi $f(x) \in B$

A est NP-complet si $A \in NP$ et
pour tout $B \in NP$, $B \prec A$ (A est NP-dur)

Exemples :
- SAT $\prec$ HAM (circuit hamiltonien), COL (colorabilité d'un graphe), CLIQUE (ss-graphe complet)
- VC (recouvrement de sommets), TSP (voyageur de commerce), KNAPSACK (sac à dos),...

Problème KNAPSACK :

- Entrée : Suite de "valeurs" entières $(v_i)_{1 \leq i \leq n}$
  Suite de "volumes" entiers $(w_i)_{1 \leq i \leq n}$, Entier B
- Sortie : Un ensemble $S \subseteq \{1, 2, \ldots, n\}$ t.q.
  $\sum_{i \in S} v_i$ soit maximal
  sous les contraintes $\sum_{i \in S} w_i \leq B$

Problème d'optimisation : Triplet (Sol, val, but)

- pour l'entrée x, $Sol(x)$ est l'ensemble des solutions possibles,
- $val : Sol(x) \longrightarrow \mathbb{N}$
- $but = max$ ou $but = min$

Soit $V = max\{v_i \ / \ 1 \leq i \leq n\}$
Lemme : Le problème $KNAPSACK$ peut être résolu par un algorithme pseudo-polynomial en temps $O(n^2.V)$

Idée : Pour $1 \leq i \leq n$ et $1 \leq v \leq nV$
Soit $W(i, v)$ le volume total minimum obtenu en sélectionnant des objets dans $\{1, \ldots, i\}$ dont la valeur totale est v

- $W(v, 0) = 0$
- $W(v, i+1) = min\{W(v, i), W(v - v_{i+1}, i) + w_{i+1}\}$

La valeur maximale est : $max\{v \ / \ W(v, n) \leq B\}$
Temps de calcul : $O(n^2.V)$

Algorithme A d'ε-approximation :
pour toute entrée x, la solution fournie $y = A(x)$ est t.q.

$$|val(y) - opt(x)| \leq \varepsilon.max\{opt(x), val(y)\}$$

Schéma d'approximation :

- Algorithme d'ε-approximation pour tout ε
- PTAS si son temps de calcul est poly($|x|$)
- FPTAS si son temps de calcul est poly($|x|, 1/\varepsilon$)

Théorème (Ibarra et Kim, 75) :
Schéma d'approximation pleinement polynomial (FPTAS) pour le problème KNAPSACK.

Idée : Si les "valeurs" v_i étaient bornées par un polynôme en n, on obtiendrait un algorithme en temps polynomial

Algorithme :

- $b := \lceil log \frac{\varepsilon.V}{n} \rceil$
- Pour $i = 1, \ldots, n$, $u_i := \lfloor \frac{v_i}{2^b} \rfloor$
- Utiliser l'algorithme précédent sur l'entrée $x' = (u_1, \ldots, u_n, w_1, \ldots, w_n, B)$

Temps de calcul : $O(n^2.\frac{V}{2^b}) = O(n^3.1/\varepsilon)$

Problème du voyageur de commerce MinTSP :

- Entrée : Un graphe $G = (E, R)$ complet (sur $\{1, 2, \ldots, n)$
  Coûts (c_{ij}) sur les arêtes
- Sortie : Un circuit hamiltonien de coût global minimum

Théorème : Pour tout $0 \leq \varepsilon < 1$,
s'il existe un algorithme d'ε-approximation pour MinTSP,
alors $P = NP$

Réduction du pb du circuit hamiltonien au problème MinTSP :
Etant donné un graphe $G = (E, R)$ à n sommets, on lui associe un graphe $G' = (E, R')$ complet avec des coûts définis par :

$$c_{ij} = \begin{cases} 1 & \text{if } (i,j) \in R \\ \frac{n}{1-\varepsilon} & \text{if } (i,j) \notin R \end{cases}$$

Pb	SAT	VC	CLIQUE	TSP	KNAPSACK
Déci.	NP-com.	NP-com.	NP-com.	NP-com.	NP-com.
Opt.					Pseudo-Poly
Approx. (ε-appr.)	MaxSAT $\varepsilon = 1/2$	MinVC $\varepsilon = 1/2$	MaxCLIQ.	MinTSP	FPTAS
Non Approx.	FPTAS $\Rightarrow$ P=NP		ε-approx. $\Rightarrow$P=NP	ε-approx. $\Rightarrow$P=NP	

Fully Polynomial-Time Approximation Scheme (FPTAS) : poly($|x|$, $1/\varepsilon$)

Problèmes de **comptage** : L. Valiant [Val79]

- $\sharp P$ classe des problèmes de comptage associés aux problèmes de décision NP
- $\sharp SAT$ est un problème $\sharp P$-complet

Schéma probabiliste d'approximation :

(R. Karp et M. Luby, 85)

Algorithme probabiliste A

- Entrée : instance x d'un problème de comptage, $\varepsilon, \delta > 0$
- Sortie : valeur $A(x, \varepsilon, \delta)$ telle que

$$Pr[(1-\varepsilon)\#(x) \leq A(x,\varepsilon,\delta) \leq (1+\varepsilon)\#(x)] \geq 1-\delta$$

Schéma probabiliste d'approximation pleinement polynomial : **FPRAS**

Le temps de calcul est $poly(|x|, (1/\varepsilon), \log(1/\delta))$

Schémas probabilistes d'approximation classiques

- Approximation de $\sharp DNF$: Karp, Luby, et Madras [KLM89]

Entrée : Formule propositionnelle sous forme normale disjonctive Φ
Sortie : Nombre de valuations satisfaisant Φ

- Approximation de la fiabilité d'un réseau : D. Karger [K99]

Entrée : Graphe dont les arêtes ont une probabilité de disparition
Sortie : La probabilité que le graphe reste connexe

Problème $\#SAT$: $\#P$-complet

Théorème : S'il existe un FPRAS pour le problème $\#SAT$
alors $RP = NP$

Classe RP : Problèmes de décision pour lesquels il existe un algorithme Monte-Carlo (avec erreur d'1 seul côté)
en temps poly

Exemple : Le problème de primalité est dans coRP

Classe RP (Randomised Polynomial Time) :
Problèmes P pour lesquels il existe un algorithme probabiliste A fonctionnant en temps polynomial t.q. pour toute entrée x

- si $P(x)$, alors $Prob_\Omega(A \text{ accepte } x) \geq \frac{1}{2}$
- si $\neg P(x)$, alors $Prob_\Omega(A \text{ accepte } x) = 0$

Ω est l' espace probabiliste des tirages de la machine

Classe $coRP$:
Problèmes P dont le complémentaire est dans RP

Exemple : Le problème de primalité est dans $coRP$

Classe RP

- Algorithmes de Monte-Carlo (avec erreur d'1 seul côté) :

$$Prob_{\Omega}(P(x) \text{ et } A \text{ rejette } x) < \frac{1}{2}$$

- Réduction (exponentielle) de la probabilité d'erreur :

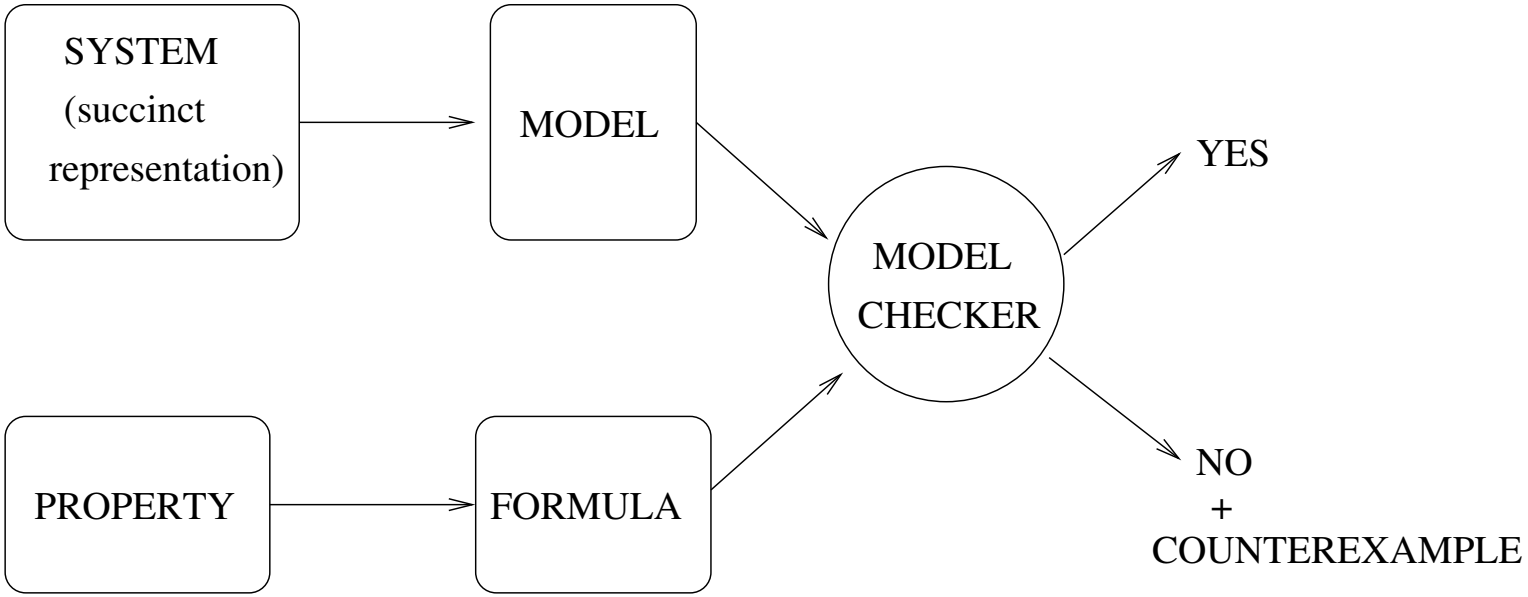
En itérant k fois l'algorithme A, on obtient un algorithme A' t.q.

$$Prob_{\Omega}(P(x) \text{ et } A' \text{ rejette } x) < \frac{1}{2^k}$$

Type \ Pb	♯SAT	♯DNF	Network Reliability
Comptage	♯P-complet	♯P-complet	
Approximation (probabiliste)		FPRAS	FPRAS
Non Approximation	FPRAS $\Rightarrow$ RP=NP		

Fully Polynomial-Time Randomised Approximation Scheme (FPRAS) : poly($|x|$, $1/\varepsilon$, $log(1/\delta)$)

RP : problèmes décidables par un algorithme probabiliste (avec erreur d'un seul côté) en temps polynomial



Entrée :

- Modèle $\mathcal{M} = (S, R)$ $R \subseteq S^2$ (relation de transition)
- Etat initial s_0
- Formule φ (Logique Temporelle Linéaire : **LTL**)

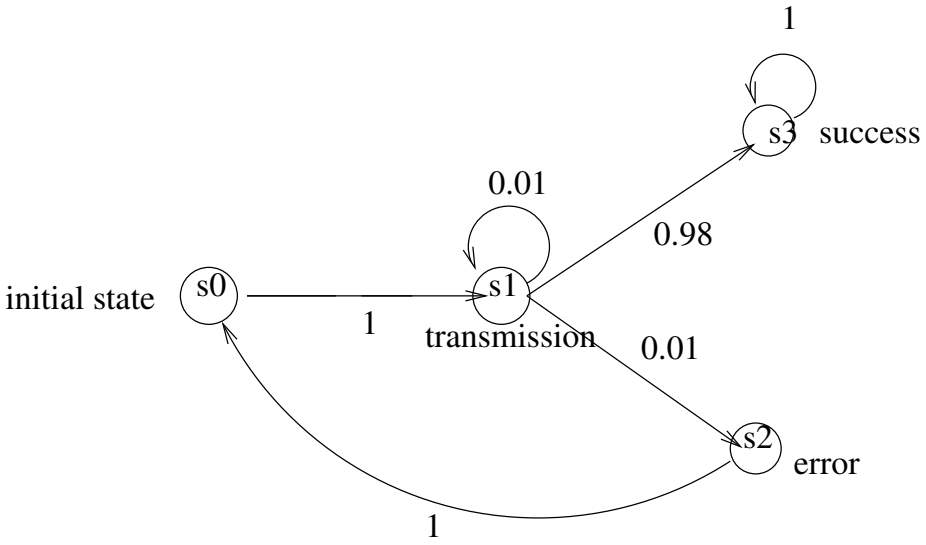
Sortie :

- OUI si $(\mathcal{M}, s_0) \models \varphi$
- NON avec trace d'erreur si $(\mathcal{M}, s_0) \not\models \varphi$

Systèmes de Transition Probabilistes

Entrée :

- Modèle $\mathcal{M} = (S, P, L)$ et état initial s_0
- $P : S^2 \longrightarrow [0, 1]$ Fonction de probabilité
- $L : S \longrightarrow 2^{AP}$ (étiquetage des états)
- Formula ψ (**LTL**)



Sortie : $Prob_{\Omega}[\psi]$

Exemple : $\psi \equiv transmission$ Until $success$

(Ω espace probabiliste des chemins d'exécution d'origine s_0)

Complexité : (Courcoubetis et Yannakakis, 1995)

Vérification qualitative (i.e. $prob > 0$?)

Même **complexité** que **model checking pour LTL**

$$O(|M|.2^{|\psi|})$$

Vérification Quantitative (i.e. $prob =$?)

$$O(poly(|M|).2^{|\psi|})$$

Problème :

- Phénomène d'explosion combinatoire (dû à la modélisation)
- Le problème n'est pas le temps, mais l'espace utilisé pour la représentation explicite du modèle

Exemple : PRISM (Probabilistic Model Checker)

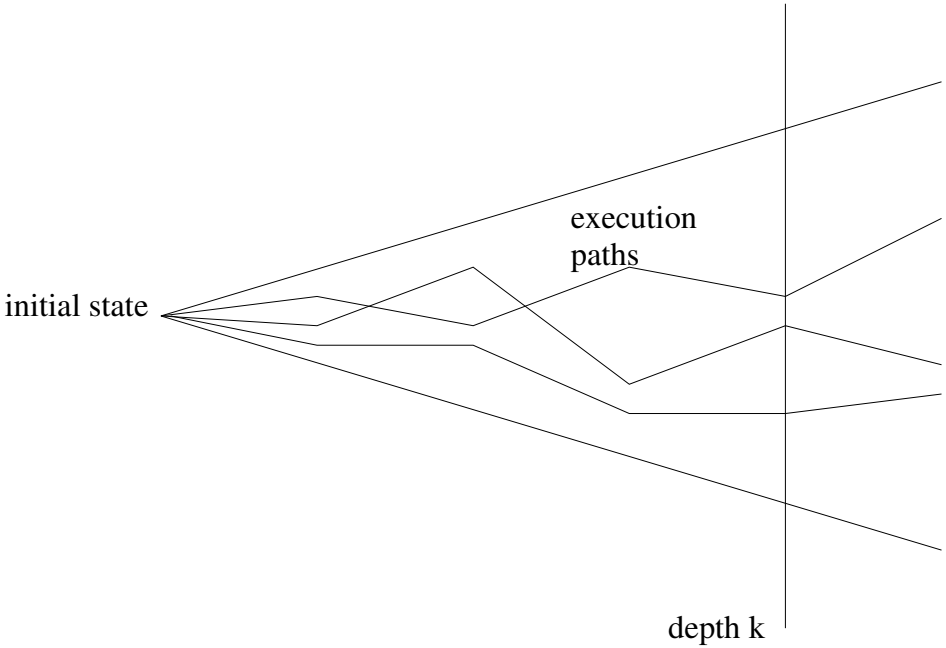
Protocole (probabiliste) du diner des philosophes

Une solution :

- Méthode d'approximation probabiliste
- Utilisation d'une représentation succincte du modèle

On considère $Prob_k(\psi)$ où :

- l'espace probabiliste est l'espace des chemins de longueur $\leq k$



- ψ exprime une propriété monotone (accessibilité)

$$\lim_{k \to \infty} Prob_k(\psi) = Prob_\Omega(\psi)$$

- si ψ exprime une propriété anti-monotone (sûreté), on considère $\neg\psi$

Algorithme générique d'approximation $\mathcal{GAA}$

entrée : $\phi, diagramme, \varepsilon, \delta$

sortie : ε-**approximation** de $Prob_k(\psi)$

$A := 0$

$N := \log(\frac{2}{\delta})/2\varepsilon^2$

Pour $i := 1$ à N

- Engendrer de manière aléatoire un chemin σ de longueur k
- Si ψ est vraie sur σ alors $A := A + 1$

Retourner (A/N)

Algorithme basé sur une estimation de type Monte-Carlo et la borne de Chernoff-Hoeffding

Diagramme : représentation succincte du système
(par exemple programme dans le langage d'entrée de PRISM)

Théorème :

$\mathcal{GAA}$ est un FPRAS pour $Prob_k(\psi)$

Méthodologie : Pour approximer $Prob_\Omega(\psi)$

- Choisir $k \approx log|M| \cdot ln(1/\varepsilon)$
- Itérer l'approximation de $Prob_k(\psi)$

Corollaire :

L'algorithme de point fixe obtenu en itérant $\mathcal{GAA}$ est un schéma probabiliste d'approximation en espace logarithmique pour $Prob(\psi)$

Remarque :

- La complexité en espace est logarithmique...
- La vitesse de convergence est régie par l'ordre de multiplicité de la 2e valeur propre (thm de Perron-Frobenius)

Théorème : R. Lassaigne et S. Peyronnet [APAL08]
L'existence d'un FPRAS pour calculer $Prob_\Omega(\psi)$ ($\psi \in LTL$)
entrainerait $RP = NP$

Idée :

- Réduction de $\#SAT$ au comptage des chemins de longueur $2n$, dont les extensions infinies satisfont une formule LTL.

- Compter le nombre de ces chemins fournit $Prob_\Omega(\psi)$
- S'il existe un FPRAS pour calculer $Prob_\Omega(\psi)$,
  alors il existe un FPRAS pour $\#SAT$

Type \ Pb	Model Checking	Vérification	Vérification
	(Φ formule LTL)	qualitative	quantitative
	$\mathcal{M} \models \Phi$?	$Prob[\Phi] > 0$?	$Prob[\Phi] =$?
Complexité	PSPACE-complet	$O(\|\mathcal{M}\|.2^{\|\Phi\|})$	$O(poly(\|\mathcal{M}\|).2^{\|\Phi\|})$
Non Approximation			FPRAS $\Rightarrow$ RP=NP

Restrictions sur la Vérification Probabiliste :

- approximation absolue
- propriétés monotones (ex : accessibilité) ou anti-monotones (ex : sûreté)

Schéma probabiliste d'approximation en espace logarithmique

Test de propriétés

- Satisfaction classique : $\mathcal{M} \models \mathbf{P}$
- Décider si $\mathcal{M}$ satisfait la propriété $\mathbf{P}$
- Satisfaction approchée : $\mathcal{M} \models_\varepsilon \mathbf{P}$

  $\mathcal{M} \models_\varepsilon \mathbf{P}$ si $\mathcal{M}$ est ε-proche de $\mathcal{M}'$ tel que $\mathcal{M}' \models \mathbf{P}$

  Modèle = **Automate** $\mathcal{A}$

  Propriété = un mot $w \in_\varepsilon \mathcal{L}(\mathcal{A})$

Testeur : Algorithme probabiliste A

- si $\mathcal{M} \models \mathbf{P}$, alors A accepte
- si $\mathcal{M}$ est ε-loin de $\mathbf{P}$, alors A rejette avec grande probabilité

  Le temps de calcul est indépendant de $|\mathcal{M}|$

  (mais dépend de $1/\varepsilon$)

Distance entre mots : distance d'édition avec déplacement

- Approximation de la satisfaction et de l'équivalence logiques
- Testeur pour la distance sur les mots
- Testeur pour l'appartenance à un langage régulier
- Extensions aux langages context-free et aux langages d'arbres
- Algorithmes probabilistes simples, mais preuves difficilles

Gain en Complexité

- Reconnaissance d'une entrée
  Temps linéaire $\longrightarrow$ Temps indépendant de la longueur du mot
- Appartenance à un langage régulier
  Temps exponentiel $\longrightarrow$ Temps polynomial
- Equivalence de 2 automates
  PSPACE-complet $\longrightarrow$ Temps polynomial
- Equivalence de 2 automates à pile
  Indécidable $\longrightarrow$ Temps exponentiel

- Si **P≠NP**, il est utile de pouvoir
  classifier les problèmes **NP**-complets

- L'**approximation** permet de distinguer
  ceux dont l'approximation est plus **facile**

- Les schémas probabilistes d'approximation permettent
  d'**approximer** certains problèmes de **comptage**

- Les algorithmes **probabilistes** sont plus efficaces
  même pour des problèmes de la classe **P**

- Les algorithmes **probabilistes** sont très utiles
  pour réduire (exponentiellement) la complexité en **espace**

- [GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability : A Guide to the theory of NP-completeness*. W.H. Freeman and co, 338 p., 1979.

- [LR04] R. Lassaigne et M. de Rougemont : *Logic and Complexity.* Springer-Verlag, 360 p., 2004.

- [MR95] R. Motwani and P. Rhagavan. *Randomized Algorithms*. Cambridge University Press, 483 P., 1995.

- [Pap94] C. Papadimitriou : *Computational Complexity.* Addison Wesley, 523 p., 1994.

- [Vaz91] V.V. Vazirani. *Approximation Algorithms*. Springer Verlag, 380 p., 2001.