Complexité
et
Approximation

Richard Lassaigne
Logique mathématique,
CNRS-Université Paris Diderot

Complexité des problèmes

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

Question : L'approximation permet-elle d'améliorer
l'efficacité des algorithmes ?

Problèmes d'Optimisation

Schémas d'Approximation

Problèmes de Comptage

Schémas Probabilistes d'Approximation

Vérification Probabiliste et Approximation

Conclusion

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, $Sol(x)$ 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 : Il existe un 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 $MaxGSAT_k$:

- Entrée : n variables propositionnelles
  m formules propositionnelles $F_1, \ldots, F_m$
  (à au plus k variables chacune)
- Sortie : Un ensemble maximal de formules satisfaisables

Théorème : Le problème $MaxGSAT_k$ est ε-approximable avec $\varepsilon = 1 - 2^{-k}$

Corollaire :

- Le problème $MaxSAT$ est $1/2$-approximable
- Le problème $MaxSAT_k$ est ε-approximable
  avec $\varepsilon = 2^{-k}$

Problème Recouvrement de sommets minimun MinVC :

- Entrée : Un graphe $G = (E, R)$
- Sortie : Un ensemble $C \subseteq E$ de taille minimum t.q.
  toute arête de G ait l'une des ses extrémités dans C

Algorithme :

- $C := \emptyset$
- Tant que $E \neq \emptyset$:
  si $R \neq \emptyset$, choisir une arête $(u, v) \in R$ et la retirer de R
  ajouter u, v à C et les retirer de E
  $G := (E, R)$

Soit C l'ensemble ainsi obtenu et C_{min} un optimum
Alors : $|C_{min}| \geq |C|/2$ et $(|C| - |C_{min}|) \leq |C|/2$

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 :

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

p-Prédicat : R relation binaire sur Σ^* t.q.

- pour tout $x, y \in \Sigma^*$, $R(x, y)$ entraîne $|y| = \text{poly}(|x|)$
- pour tout $x, y \in \Sigma^*$, $R(x, y)$ est décidable
  en temps poly $(|x|)$

Fonction F_A associée à un pb A de la classe NP :
fonction de comptage $F_A(x) = |\{y \in \Sigma^* \ / \ R(x, y)\}|$

Classe $\#P$: Fonctions $F : \Sigma^* \longrightarrow \mathbb{N}$ t.q.
il existe un p-Prédicat R avec $F(x) = |\{y \in \Sigma^* \ / \ R(x, y)\}$

Exemples : $\#SAT$, $\#HAM$, $\#COL$

Réduction entre fonctions $\#P$: $f, g : \Sigma^* \longrightarrow \mathbb{N}$
$g \prec_P f$ s'il existe 2 fonctions calculables en temps poly
$h : \Sigma^* \longrightarrow \Sigma^*$ et $\alpha : \mathbb{N} \longrightarrow \mathbb{N}$ t.q. $g(x) = \alpha(f(h(x)))$

Soit f une fonction de la classe $\#P$

Schéma Probabiliste d'Approximation (RAS) pour f :

Algorithme probabiliste A t.q. pour toute entrée x,
pour tout $\varepsilon > 0$ (paramètre d'approximation),
pour tout $\delta > 0$ (paramètre de confiance),
A fournit une valeur $A(x, \varepsilon, \delta)$ satisfaisant

$$Prob_{\Omega}(|A(x, \varepsilon, \delta) - f(x)| \leq \varepsilon . f(x)) \geq 1 - \delta$$

Ω est l'espace probabiliste des tirages de l'algorithme

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

Problème $\#DNF$: $\#P$-complet

Théorème : (R. Karp, M. Luby et N. Madras, 89)

La méthode de Monte-Carlo avec borne (Chernoff-Hoeffding) sur l'échantillon est un FPRAS pour le problème $\#DNF$

F formule DNF : $\bigvee_{i=1}^{m} G_i$

S_i ensemble des valuations satisfaisant G_i $(i = 1, \ldots, n)$

$c(v) =$ nombre de formules G_i que la valuation v satisfait

Idée : On échantillonne dans le multi-ensemble $M = \biguplus_{i=1}^{m} S_i$

- choisir une formule G_i avec probabilité $|S_i|/|M|$
- choisir une valuation satisfaisant G_i de manière uniforme

Lemme 1 : $X(v) = |M|/c(v)$ est un bon estimateur pour $\#F$

Lemme 2 : $\mu = \#F/|M| \geq 1/m$

Taille de l'échantillon (polynomiale) : $N = \frac{4m}{\varepsilon^2} ln(2/\delta)$

Problème Network Reliability :

- Entrée : Un graphe $G = (E, R)$ connexe
  avec probabilité de disparition p pour chaque arête
- Sortie : La probabilité que le graphe G devienne non connexe

Résultat (D. Karger, 94) : Conception d'un FPRAS

Méthode :

- Dénombrement des coupures αminimales dans un graphe
- Utilisation du FPRAS pour la version probabiliste de $\#DNF$

Problème $\#SAT$: $\#P$-complet

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

ClasseRP : 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

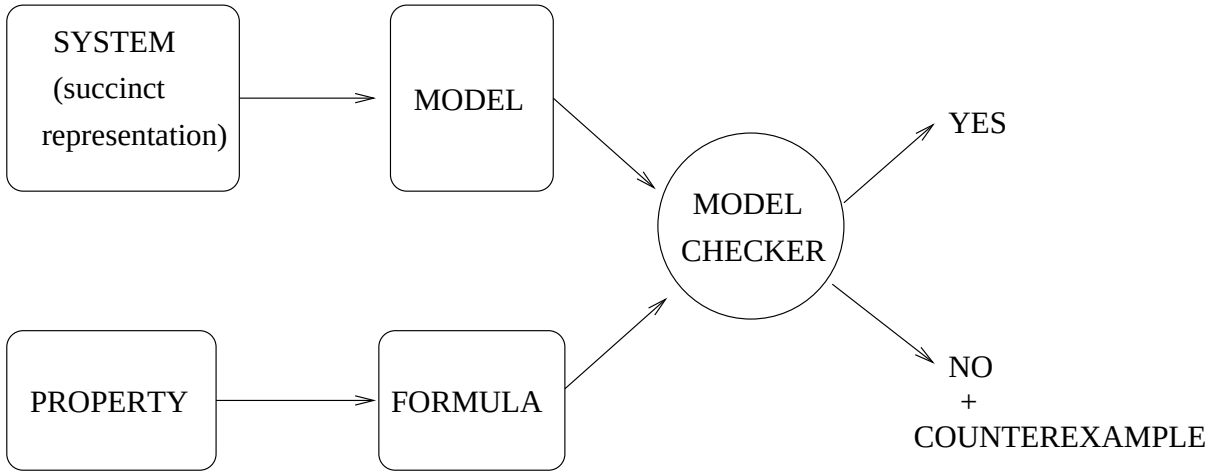
Idée de la preuve :

- Un FPRAS pour $\#SAT$ permettrait de distinguer, en temps polynomial, pour l'entrée x, entre $\#(x) = 0$ et $\#(x) > 0$
- Alors il existerait un algorithme probabiliste en temps polynomial pour SAT et $NP \subseteq BPP$

BPP classe des algorithmes Monte-Carlo en temps poly (avec erreur des 2 côtés)

Remarque : M. Jerrum et A. Sinclair

$$NP \subseteq BPP \Rightarrow RP = NP$$



Entrée :

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

Sortie :

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

Logique Temporelle Linéaire (LTL)

Les formules de LTL sont construites à l'aide de

- propositions atomiques : $P = \{p_1, p_2, \ldots, p_n\}$
- connecteurs propositionnels : $\neg, \vee$ $(\wedge, \rightarrow)$
- opérateurs temporels : X (Next) U (Until)

L'ensemble des formules LTL est défini par induction :

- toute proposition atomique p_i est une formule,
- si φ et ψ sont des formules,
- alors $\neg\varphi$, $\varphi \vee \psi$, $X\varphi$ et $\varphi U \psi$ sont des formules

Autres opérateurs temporels :

$$F\psi \equiv (trueU\psi) \ (true \equiv (p_1 \vee \neg p_1))$$
$$G\psi \equiv \neg F \neg \psi$$

Logique Temporelle Linéaire (LTL)

On interprète les formules de LTL sur les chemins
Chemin $\sigma = (s_0, s_1, \ldots, s_i, \ldots)$: on note σ^i le chemin $(s_i, s_{i+1}, \ldots)$

La relation de satisfaction pour les formules LTL est définie par induction :

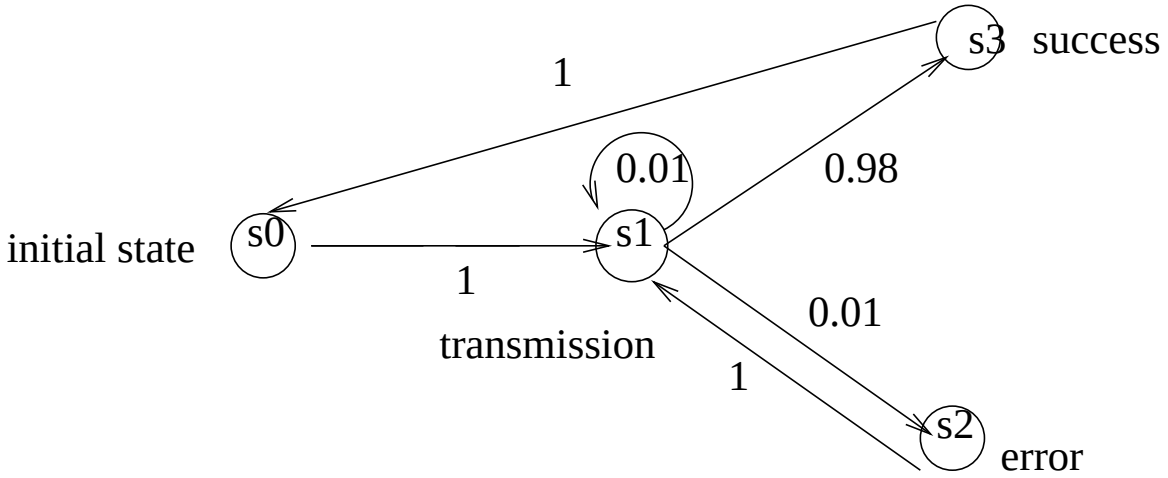
- $\sigma \models p_i$ ssi $p_i \in L(s_0)$
- $\sigma \models \neg\psi$ ssi $\sigma \not\models \psi$
- $\sigma \models \varphi \vee \psi$ ssi $\sigma \models \varphi$ ou $\sigma \models \psi$
- $\sigma \models X\psi$ ssi $\sigma^1 \models \psi$
- $\sigma \models \varphi U \psi$ ssi $(\exists i \geq 0)$ t.q. $\sigma^i \models \psi$ et $(\forall j < i)$ $\sigma^j \models \varphi$

La relation de satisfaction est étendue aux systèmes de transition : $\mathcal{M}_P \models \psi$ ssi pour tout chemin σ de $\mathcal{M}_P$, $\sigma \models \psi$

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)

Espace probabiliste :

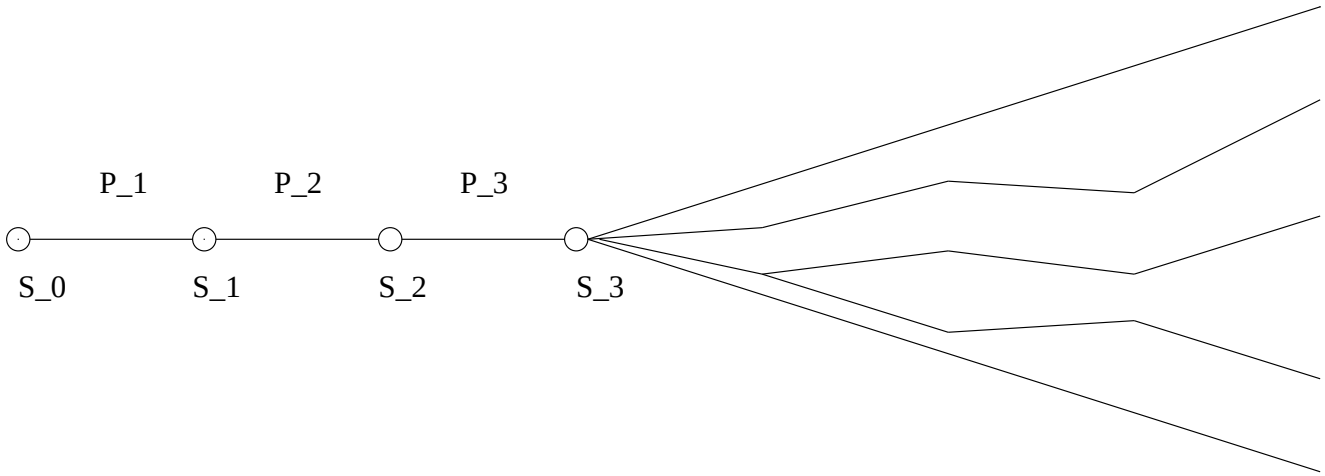
Cône des extensions d'un chemin fini $\rho = (s_0, s_1, \ldots, s_n)$:

$$Prob(\{\sigma/\sigma \text{ est un chemin et } (s_0, s_1, \ldots, s_n) \text{ est un préfixe of } \sigma\}) =$$

$$\prod_{i=1}^{n} P(s_{i-1}, s_i)$$

La mesure peut être définie sur la famille borélienne engendrée par les ensembles $\{\sigma/\rho$ préfixe de $\sigma\}$ où ρ est un chemin fini.

L'ensemble des chemins $\{\sigma/\sigma(0) = s$ and $\mathcal{M}, \sigma \models \psi\}$ est mesurable (Vardi) et $Prob_\Omega[\psi]$ est bien définie



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 = ?)

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

Méthode : Calculer $Prob_{\Omega}[\psi]$

- Transformer étape par étape la formule et la chaîne de Markov $\mathcal{M}$
- Eliminer les connecteurs temporels un par un
- En préservant la probabilité de satisfaction
- En résolvant un système d'équations linéaires de taille $|M|$.

Théorème : R. Lassaigne et S. Peyronnet
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 $\#SAT$

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.		FPTAS
Non Approx.	FPTAS $\Rightarrow$ P=NP		ε-approx. $\Rightarrow$P=NP	ε-approx. $\Rightarrow$P=NP	

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

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

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

Restrictions sur la Vérification Probabiliste :

- approximation absolue
- propriétés monotones ou anti-monotones

$\Longrightarrow$

Schéma probabiliste d'approximation en espace logarithmique pour le calcul de $Prob[\Phi]$

- [CY95] C. Courcoubetis and M. Yannakakis. *The complexity of probabilistic verification*. JACM, 24(4), p. 857-907, 1995.
- [K99] D.R. Karger. *A Randomized Fully Polynomial Approximation Scheme for the All Terminal Network Reliability Problem*. SIAM Journal on Computing 29(2), p.492-514, 1999.
- [KLM89] R. Karp, M. Luby and N. Madras. *Monte-Carlo Approximation Algorithms for Enumeration Problems.* Journal of Algorithms 10, 429-448, 1989.
- [APAL08] R. Lassaigne et S. Peyronnet. *Probabilistic Verification and Approximation*. Journal of Pure and Applied Logic, vol. 152 (1-3), p. 122-131, 2008.
- [LR04] R. Lassaigne et M. de Rougemont : *Logic and Complexity.* Springer-Verlag, 360 p., 2004.
- [Val79] L. Valiant. *The complexity of enumeration and reliability problems*. SIAM Journal of Computing 8, p.410-421, 1979.