Près de 40 ans après le théorème de Cook, où en est la complexité algorithmique ?

(Une histoire très partielle du problème «P=NP ?»)

Pascal Koiran

LIP

Ecole Normale Supérieure de Lyon

Colloquium Jacques Morgenstern, 14 mai 2009.

Le théorème de Cook : SAT est NP-complet

(The Complexity of Theorem-Proving Procedures, STOC'71)

Les notations «modernes» (SAT, P, NP, NP-complet) n'existent pas :

– Cook s'intéresse aux tautologies, pas aux formules satisfaisables (problème équivalent par négation).
– P s'appelle $\mathcal{L}^*$.

Il n'y a pas que le théorème de Cook…

- Autres problèmes NP-complets :
  CNF-SAT, 3CNF-SAT, isomorphisme de sous-graphe.
- Primalité et isomorphisme de graphe ?
- Peut-on montrer que SAT n'est pas résoluble en temps polynomial ?
  «Such a proof would be a major breakthrough in complexity theory.»
- Complexité des preuves par résolution ?
- Borne inférieure sur la complexité du calcul des prédicats
  (il y a d'illustres prédécesseurs).

Les 21 problèmes de Karp

Reducibility among combinatorial problems.

In *Complexity of Computer Computations,*

Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights.

Plenum, p.85-103, 1972 (http ://www.cs.berkeley.edu/ luca/cs172/karp.pdf).

- La NP-complétude n'est pas un phénomène isolé.
- Définition de NP à base de certificat (*guess*) à vérifier en temps polynomial.
- Réductions many-one : $A \leq_m B$ si $x \in A \Leftrightarrow f(x) \in B$, avec f calculable en temps polynomial.
- Les notations «modernes» apparaissent : P, NP, P=NP.
- Statut de la programmation linéaire ?

La force brute selon Levin

(Universal Search Problems, Problemy Peredachi Informatsii, 1973)

- Article soumis en 1972, résultat présenté en séminaire dès 1971. Thèse dirigée par Kolmogorov.
- S'inscrit dans la tradition soviétique de l'étude de la recherche exhaustive (brute-force search, ou *perebor*).
  Voir *A Survey of Russian Approaches to* Perebor *(Brute-Force Search) Algorithms* par B. A. Trakhtenbrot, 1984.
- Notion de réduction encore plus restrictive que celle de Karp (correspondance entre instances mais aussi entre certificats).
  Proche des *relations universelles* d'Agrawal et Biswas (1992).

Lettre de Kurt Gödel à John von Neumann

(http ://rjlipton.wordpress.com/the-gdel-letter/)

Princeton, 20 March 1956

Dear Mr. von Neumann :

With the greatest sorrow I have learned of your illness. The news came to me as quite unexpected. Morgenstern already last summer told me of a bout of weakness you once had, but at that time he thought that this was not of any greater significance. As I hear, in the last months you have undergone a radical treatment and I am happy that this treatment was successful as desired, and that you are now doing better. I hope and wish for you that your condition will soon improve even more and that the newest medical discoveries, if possible, will lead to a complete recovery.

Since you now, as I hear, are feeling stronger, I would like to allow myself to write you about a mathematical problem, of which your opinion would very much interest me :

One can obviously easily construct a Turing machine, which for every formula F in first order predicate logic and every natural number n, allows one to decide if there is a proof of F of length n (length = number of symbols).

Let $\psi(F, n)$ be the number of steps the machine requires for this and let $\phi(n) = \max_F\,(F, n)$. The question is how fast $\phi(n)$ grows for an optimal machine. One can show that $\phi(n) \geq k \cdot n$. If there really were a machine with $\phi(n) \sim k \cdot n$ (or even $\sim k \cdot n^2$) this would have consequences of the greatest importance.

Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine. After all, one would simply have to choose the natural number n so large that when the machine does not deliver a result, it makes no sense to think more about the problem. Now it seems to me, however, to be completely within the realm of possibility that $\phi(n)$ grows that slowly.

Since it seems that $\phi(n) \geq k \cdot n$ is the only estimation which one can obtain by a generalization of the proof of the undecidability of the Entscheidungsproblem and after all $\phi(n) \sim k \cdot n$ (or $\phi(n) \sim k \cdot n$) only means that the number of steps as opposed to trial and error can be reduced from N to $\log N$ (or $(\log N)^2$). However, such strong reductions appear in other finite problems, for example in the computation of the quadratic residue symbol using repeated application of the law of reciprocity.

It would be interesting to know, for instance, the situation concerning the determination of primality of a number and how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search.

I do not know if you have heard that Post's problem , whether there are degrees of unsolvability among problems of the form $(\exists y)\phi(y, x)$, where ϕ is recursive, has been solved in the positive sense by a very young man by the name of Richard Friedberg. The solution is very elegant. Unfortunately, Friedberg does not intend to study mathematics, but rather medicine (apparently under the influence of his father). By the way, what do you think of the attempts to build the foundations of analysis on ramified type theory, which have recently gained momentum ? You are probably aware that Paul Lorenzen has pushed ahead with this approach to the theory of Lebesgue measure. However, I believe that in important parts of analysis non-eliminable impredicative proof methods do appear.

I would be very happy to hear something from you personally. Please let me know if there is something that I can do for you. With my best greetings and wishes, as well to your wife,

Sincerely yours,

Kurt Gödel

Ce n'est que la partie émergée de l'iceberg...

– Quelques inclusions :

$$\text{LOGSPACE} \subseteq \text{P} \subseteq \text{NP} \subseteq \text{PH} \subseteq \text{PSPACE} \subseteq \text{EXP}$$

– Toutes sont conjecturées strictes.

– On sait quand-même que que $\text{P} \neq \text{EXP}$ et $\text{LOGSPACE} \neq \text{PSPACE}$.
Se démontre par une méthode bien éprouvée : la diagonalisation.

Hiérarchie en temps [Hartmanis-Stearns 1965]

Soient $\mathcal{A}$ et $\mathcal{B}$ des classes de complexité.

Comment montrer que $\mathcal{A} \neq \mathcal{B}$?

- Les problème de $\mathcal{A}$ sont décidés par une famille d'algorithmes (ϕ_a) : ϕ_a accepte l'ensemble A_a des x tels que $\phi_a(x) = 1$.
- On définit un problème B par : $x \in B \Leftrightarrow x \notin A_x$.
- Par construction, $B \notin \mathcal{A}$.
- Reste à montrer que $B \in \mathcal{B}$ (en général par simulation).

Théorème :

Si f et g sont constructibles en temps et $f(n) \log f(n) = o(g(n))$ on a l'inclusion stricte :

$$\mathrm{DTIME}(f(n)) \subset \mathrm{DTIME}(g(n)).$$

Peut-on montrer que $\mathrm{P} \neq \mathrm{NP}$ par diagonalisation ?

L'art de faire des excuses

Complexity theory could be defined as the field concerned with deep, nontrivial, mathematically-sophisticated justifications for failure (Scott Aaronson).

FIG. 1 – I can't find an efficient algorithm, but neither can all these famous people. From *Computers and Intractability* (Garey-Johnson, 1979)

Les méta-excuses

– La relativisation montre que les preuves par diagonalisation ont leurs limites.

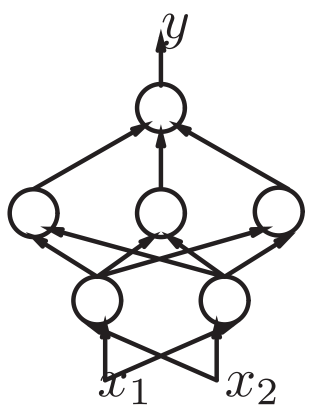
Deux excuses (ou «barrières») découvertes plus récemment :

– Les preuves naturelles [Razborov-Rudich, 1994].
– L'agébrisation, ou *relativisation algébrique* (Aaronson-Wigderson, 2008) : une généralisation de la relativisation qui s'applique aux résultats obtenus par arithmétisation (comme IP $=$ PSPACE).

La relativisation

- Une machine de Turing avec accès à un oracle A peut décider en une étape de calcul si un mot appartient à A.
- Il existe des oracles A et B tels que $\mathrm{P}^A = \mathrm{NP}^A$ et $\mathrm{P}^B \neq \mathrm{NP}^B$ (Baker-Gill-Solvay, 1975).
- Malheureusement, les preuves par diagonalisation relativisent : le résultat reste vrai en présence de tout oracle A.

Changeons de problème : les circuits booléens



Chaque porte calcule une fonction booléenne choisie dans $\mathcal{B}$.
Usuellement : $\mathcal{B} = \{\vee_2, \wedge_2, \neg\}$.

Un langage A est dans P/poly si $A \cap \{0,1\}^n$ est reconnu par un circuit C_n de taille polynomiale en n.

Il est *plus difficile* de montrer que NP$\not\subseteq$P/poly puisque P $\subset$ P/poly :
– Argument intuitif : à l'intérieur de votre ordinateur il y a des circuits.
– Argument formel : simulation des machines de Turing par des circuits.
En fait, on ne sait même pas montrer que EXP$\not\subseteq$P/poly !

Tout n'est pas perdu

Presque toute les fonctions booléennes nécessitent des circuits de taille $\geq 2^n/n$ [Shannon, 1949] :

– Il y a 2^{2^n} fonctions de n variables.

– Le nombre de cicuits de taille s est simplement exponentiel en s.

Malheureusement, on veut une **construction explicite**.

La meilleure borne inférieure connue devient alors $4.5n - o(n)$ [Lachish-Raz, 2001] (leur fonction est dans P).

Bornes inférieures pour des classes restreintes de circuits : le grand succès des années 1980

Circuits monotones (sans $\neg$) : bornes inférieures exponentielles
pour des problèmes NP-complets (CLIQUE)
ou même polynomiaux (COUPLAGE PARFAIT)
dues à Razborov ; Andreev ; Alon et Boppana…

Circuits de profondeur constante (fan-in non borné) : bornes infs dues à
Ajtai ; Furst-Saxe-Sipser ; Hastad ; Razborov ; Smolensky ; Yao…

- La fonction parité n'est pas dans AC^0.
- Si p premier et m n'est pas une puissance de p,
  MOD m n'est pas dans $\mathrm{AC}^0(p)$.
- borne inférieure pour les circuits avec des portes MOD 6 ?

Très peu de progrès en 20 ans ; stagnation largement expliquée
par la barrière des preuves naturelles.

Changeons encore de problème : les circuits arithmétiques

- On se place sur un corps K :
  les entrées sont des constantes de K, ou des variables.
- Portes $\vee$, $\wedge$, $\neg$ remplacées par $+$, $\times$.
- Un circuit calcule un polynôme de $K[X_1, \ldots, X_n]$.
- Pour $K = \mathbb{Z}/2\mathbb{Z}$, on retrouve les circuits boolens :
  $x \wedge y = x \times y$, $x \vee y = x + y + x \times y$, $\neg x = x + 1$.
- Espoir : bornes inférieures plus faciles pour des corps comme $\mathbb{Q}$ ou $\mathbb{C}$?
- Nombreuses variantes possibles :
  circuits multilinéaires, coefficents bornés [Morgenstern 1973]…

Modèle de Valiant (1979) : $\mathrm{VP}_K = \mathrm{VNP}_K$?

– Complexité d'un polynôme f mesurée par :

$\boxed{\text{L(f) = taille du plus petit circuit arithmétique pour } f.}$

– $(f_n) \in$ VP si le nombre de variables, $\deg(f_n)$ et $L(f_n)$ sont polynomialement bornés. Par exemple, $(X^{2^n}) \notin$VP.

– $(f_n) \in$ VNP si $f_n(\overline{x}) = \sum_{\overline{y}} g_n(\overline{x}, \overline{y})$

pour $(g_n) \in$ VP

(la somme porte sur toutes les valeurs booléennes de $\overline{y}$).

Une famille VNP typique : le permanent.

$$\mathrm{per}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i\sigma(i)}.$$

Elle est VNP-complète si $\mathrm{char}(K) \neq 2$.

Le problème du déterminant et du permanent

Un vieux problème (Polya, Szegö, 1913) :
exprimer un permanent $n \times n$ comme un déterminant.

Meilleure borne inférieure connue : $\Omega(n^2)$ [Mignon-Ressayre, 2004].

On peut exprimer un déterminant $n \times n$
comme un permanent de taille polynomiale en n [Valiant].

Remarques :

– Mignon et Ressayre sont des spécialistes de géométrie algébrique et de théorie des représentations.
– La *Geometric Complexity Theory* de Mulmuley est basée sur la théorie des représentations.
– Mais la preuve du théorème de Mignon et Ressayre est «élémentaire».

Modèle de Blum-Shub-Smale (1989) : $\mathrm{P}_K = \mathrm{NP}_K$?

- Présentation à base de circuits due à Poizat
  (similaire aux circuits arithmético-booléens de von zur Gathen).
  Le modèle de calcul est plus riche : en plus des portes $+$, $\times$
  les circuits ont droit à $=$ et à $\leq$ (si K est ordonné).
- BSS utilisent des machines uniformes.
- On s'intéresse ici à des problèmes de décision.

$\mathrm{P}_K = \mathrm{NP}_K$?
Définition des classes et contenu algorithmique

- X est dans P_K si pour tout $x \in K^n$,

$$x \in X \Leftrightarrow C_n(x_1, \ldots x_n, a_1, \ldots, a_k) = 1$$

  avec C_n construit en temps polynomial par une machine de Turing.
- X est dans NP_K si pour tout $x \in K^n$,

$$x \in X \Leftrightarrow \exists y \in K^{p(n)} \langle x, y \rangle \in Y$$

  avec $Y \in \mathrm{P}_K$.
  Un problème $\mathrm{NP}_{\mathbb{R}}$ typique :
  décider si un polynôme de degré 4 en n variables a une racine réelle .

  C'est une version algorithmique de l'élimination des quantificateurs.
  Les algorithmes connus sur $\mathbb{C}$ [Fichtas-Galligo-Morgenstern, 1990] ou $\mathbb{R}$ sont de complexité exponentielle en n.

Quelques théorèmes de transfert

Corps algébriquement clos [BCSS]

Corps différentiellement clos [Portier]

Réels avec addition et ordre [Fournier-Koiran] :
$P_{\mathbb{R}_{evo}} = NP_{\mathbb{R}_{evo}} \Leftrightarrow NP \subseteq P/poly$.

$VP = VNP \Rightarrow NP \subseteq P/poly$ [Bürgisser] ;
utilise l'hypothèse de Riemann généralisée.

$(VP = VNP$ et $PSPACE \subseteq P/poly) \Rightarrow P_{\mathbb{R}} = PAR_{\mathbb{R}}$ [Koiran-Périfel].
On a $P_{\mathbb{R}} \subseteq NP_{\mathbb{R}} \subseteq PAR_{\mathbb{R}}$.

Algorithmes probabilistes : ce qu'on s'autorise à penser dans les milieux autorisés

Exemples :

- ACIT, un problème de RP : décider si un circuit arithmétique calcule un polynôme non identiquement nul.
- Le test de primalité avant Agrawal-Kayal-Saxena (2004).

Une intution «naturelle» : la randomisation accélère exponentiellement certains problèmes ($P \neq RP$).

Mais cette intuition est en contradiction avec d'autres intuitions...

Hardness versus randomness tradeoffs

Deux problèmes plus ou moins équivalents :

– dérandomiser

– obtenir des bornes inférieures

Point commun : on a besoin de **constructions explicites**.

Par exemple, d'après Kabanets-Impagliazzo (2004) :

– Si on peut dérandomiser ACIT,
  NEXP$\not\subseteq$P/poly ou le permanent n'est pas dans VP.

– Si le permanent n'est pas dans VP, on peut dérandomiser ACIT
  en temps sous-exponentiel dans un cas particulier
  (circuits de profondeur logarithmique).

Une approche possible pour obtenir des bornes inférieures ?
(Agrawal, 2005)

Le modèle des boîtes noires

Le seul moyen d'accéder au polynôme f :

$$x \mapsto \boxed{\textbf{boîte noire}} \to f(x).$$

Nombreux problèmes étudiés : factorisation, PGCD, interpolation...

Deux problèmes équivalents dans ce modèle :

- dérandomiser ACIT.
- Construire un *hitting set*. (ensemble intersectant ?)

Un ensemble intersectant H_s doit contenir un point x tel que $f(x) \neq 0$ pour tout $f \not\equiv 0$ tel que $L(f) \leq s$.

D’un ensemble intersectant à une borne inférieure, ou : le retour de la diagonalisation

On sait qu’il existe H_s de taille polynomiale en $s\ldots$
par un argument probabiliste !

Soit $H_s = \{a_1, \ldots, a_k\}$, $f(X) = \prod_{i=1}^{k}(X - a_i)$.

On a $L(f) > s$ mais f est de degré $s^{O(1)}$:
la borne inférieure «évidente» n’est que $\Omega(\log s)$.

Si H_s est explicite, f est explicite aussi.

Un problème ouvert bien connu : complexité de $\prod_{i=1}^{k}(X - i)$?