

Pourquoi est-il si difficile de prouver qu'un problème a une borne inférieure de complexité ?

Un point de vue en logique

Etienne Grandjean, GREYC,

Université de Caen



Rétrospective

1971-1973: Cook, Levin, Karp, etc

- élaborent la notion de problème NP
- montrent que pratiquement tous les problèmes combinatoires « difficiles » sont NP-durs ou NP-complets
- énoncent la conjecture $P \neq NP$ équivalente à $SAT \notin P$ que certains renforcent en SAT est « exponentiel »



Rétrospective

- 1982: Cook énonce:
« Pour aucun problème NP-complet classique, on n'a encore prouvé aucune borne inférieure de complexité *non triviale* sur un modèle de calcul de portée générale »
- Quelle est la situation actuelle?



De nombreux résultats de complexité structurelle ont été prouvés

- **Des théorèmes d'inclusion**

Exemple: $\text{NSPACE}(S) = \text{co-NSPACE}(S)$
pour toute fonction $S(n) \geq \log n$
(Immerman et Szelepcsényi, 1987)



Des théorèmes de séparation

- **Hiérarchie stricte sur le temps:**

$$\text{DTIME}(T) \subset \text{DTIME}(T')$$

$$\text{pour } T(n) = o(T'(n))$$

- **Apport strict du non déterminisme:**

$$\text{DTIME}(n) \subset \text{NTIME}(n)$$

pour le **temps linéaire des MT** [PPST 1983]



Des « résultats potentiels »

- Si $\text{DTIME}(n^2) \leq_{\text{lin}} A$
(= A « dur » pour le temps quadratique par réductions **linéaires**)
alors $A \notin \text{DTIME}(o(n^2))$
- Idem pour $\text{NTIME}(n^2)$

Pourtant, aucun résultat prouvé de ce genre pour aucun problème naturel dans NP



Quelques constatations

- Les problèmes dans P sont **souvent dans DLIN**: connexité, planarité, Horn-satisfaisabilité, etc.
- Les problèmes NP sont **presque toujours NLIN**: les 21 problèmes NP-complets de [Karp 1972]

DLIN, NLIN: temps linéaire des RAMs



Existence de problèmes NLIN-complets

Problème RISA (Reduction of Incompletely Specified Automata, problème du [Garey-Johnson 1979])

Donnée: un automate fini déterministe A incomplètement spécifié (= de fonction de transition partielle) et un entier K

Question: peut-on compléter la fonction de transition de A pour que l'automate réduit équivalent ait moins de K états ?



Existence de problèmes NLIN-complets

RISA \in NLIN et on prouve

- NLIN \leq_{lin} RISA [G 1990]
- $\text{DTIME}_{\text{MT}}(n) \subset \text{NTIME}_{\text{MT}}(n) \subseteq \text{NLIN}$

D'où on tire la borne inférieure sur MT

$$\text{RISA} \notin \text{DTIME}_{\text{MT}}(n)$$



Proof that RISA is NLIN-complete

Three steps:

1. a **logical characterization** of NLIN
2. **normalize** this logic
3. **unfold** the normalized logical formula along the input



Step 1: logical characterization of NLIN

- Describe a computation of linear time on an input of length n

$([n], \text{input}) = (n, \text{input}(0), \dots, \text{input}(n-1))$

where $[n] = \{0, 1, \dots, n-1\}$ and $\text{input}: [n] \rightarrow [n]$



Step 1: logical characterization of NLIN

Essential points:

- Do **not** describe **completely** the diagram *time x space* of size $\Theta(n^2)$
- Only describe at each instant the part of the memory **of size $\Theta(1)$ that is modified**
- So the description of the computation has size $\Theta(n)$



Step 1: logical characterization of NLIN

- Encode each linear computation by a fixed number k of functions $\mathbf{f} = (f_1, \dots, f_k)$,

$f_i: [n] \rightarrow [n]$, of global size $kn = \Theta(n)$

- Describe an accepting computation by a one-variable FO formula

$\forall x \Psi(\text{input}, \mathbf{f}, x)$

where Ψ is quantifier-free: the unique variable $x \in [n]$ essentially represents the n or $O(n)$ instants

- This gives the equivalence:

$([n], \text{input})$ is accepted by a computation

iff

$([n], \text{input})$ satisfies $\exists \mathbf{f} \forall x \Psi(\text{input}, \mathbf{f}, x)$



Step 1: logical characterization of NLIN

We have proved that each problem PB in NLIN is definable by a formula in ESO($\forall 1$):

$([n], \text{input}) \in \text{PB}$

iff

$([n], \text{input})$ satisfies $\exists \mathbf{f} \forall x \Psi(\text{input}, \mathbf{f}, x)$



Step 2:

Normalization of the formula $ESO(\forall 1)$

Theorem (F. Olive '97): Any formula in $ESO(\forall 1)$

$$\exists \mathbf{f} \forall x \Psi(\text{input}, \mathbf{f}, x)$$

can be normalized in a purely conjunctive form, i.e., is equivalent to a similar formula where Ψ is a conjunction of equalities (involving the arithmetical functions *successor* and *zero*)

$$\exists \mathbf{f} \forall x \bigwedge_i s_i(x) = t_i(x)$$

where s_i and t_i are compositions of functions



Step 3: Reduce the problem to RISA

Essentially unfold the formula on its domain
 $[n] = \{0, \dots, n-1\}$:

$$\forall x \bigwedge_i s_i(x) = t_i(x)$$

becomes

$$\bigwedge_{a < n} \bigwedge_i s_i(a) = t_i(a)$$



Step 3: Reduce the problem to RISA

- So, $([n], \text{input})$ satisfies $\exists f \forall x \bigwedge_i s_i(x) = t_i(x)$

iff the following conjunction of equalities (of length $O(n)$) is satisfiable:

$$\bigwedge_{a < n} \bigwedge_i s_i(a) = t_i(a) \wedge \bigwedge_{a < n} \text{input}(a) = v_a$$

where v_a is the a^{th} value of the input.

- This satisfiability problem can be linearly reduced to an instance (A, K) of problem RISA with $K = n+1$



Conséquence « potentielle »

Puisque RISA est NLIN-complet, on a

$NLIN \neq DLIN$ (temps linéaire des RAMs)

ssi $RISA \notin DLIN$

Mais on ne sait pas prouver la conjecture $NLIN \neq DLIN$

(analogue affaibli de $NP \neq P$)



Schéma de preuve pour borne inférieure de complexité

1) Un résultat de séparation de classes de complexité:

$$C \not\subseteq C' \text{ (ou } C' \subset C \text{ strictement)}$$

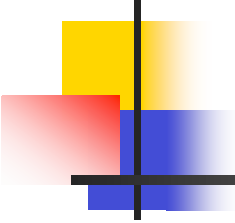
2) Un résultat de « dureté »:

Pb est C -difficile:

$$C \leq Pb \text{ par une réduction appropriée}$$

On en déduit:

$$Pb \notin C'$$



Pourquoi ça ne marche pas souvent?

On n'a presque jamais (1) et (2)
ensemble

On ne sait pas prouver

1) Soit le **résultat de séparation**:

Exemples: $NP \neq P$ ou $NLIN \neq DLIN$?

Mais on a $NTIME_{MT}(n) \neq DTIME_{MT}(n)$

2) Soit le **résultat de dureté**:

Pas de candidat naturel dans NP qui soit « dur » pour
 $NTIME(n^2)$

ou même pour $DTIME(n^2)$



What about SAT?

Hardness result

Theorem (Schnorr, Cook, Robson and al.)

SAT is **weakly NLIN-hard**:

$$\text{NLIN} \leq_{n(\log n)^2} \text{SAT}$$

Precisely, each problem in NLIN is reducible to SAT in **quasi-linear time** $O(n(\log n)^2)$ and space $O(\log n)$



Separation results and consequences

- **Theorems** (Fortnow, Lipton et al. 1997-2000)

- $\text{NLIN} \not\subseteq \text{co-NTIMESPACE}(n^{1+o(1)}, n^{1-\varepsilon})$

for any $\varepsilon > 0$

- $\text{NLIN} \not\subseteq \text{DTIMESPACE}(n^a, n^{o(1)})$

for any $a < \text{Golden Ratio} \approx 1.618$

- **Consequences:**

- $\text{SAT} \not\subseteq \text{co-NTIMESPACE}(n^{1+o(1)}, n^{1-\varepsilon})$

for any $\varepsilon > 0$

- $\text{SAT} \not\subseteq \text{DTIMESPACE}(n^a, n^{o(1)})$

for any $a < \text{Golden Ratio}$



Examples

- $\text{SAT} \notin \text{co-NTIMESPACE}(n(\log n)^k, n^{0.9})$
for each k
- $\text{SAT} \notin \text{DTIMESPACE}(n^{1.6}, n(\log n)^k)$
for each k



Method of proof for separation results

Proof by contradiction using:

- **Classical hierarchy results** (proved by **diagonalization**)
- **Padding** arguments
- **Improvements of a divide-and-conquer method** (the computation is divided into intervals) initiated by [Nepomnjascii '70] and [Kannan '84]: simulations of **deterministic** computations of **sublinear space** by **nondeterministic or alternating machines**



Critique de la méthode de preuve de borne inférieure de complexité

La méthode de « dureté » ou « complétude » est-elle la seule possible?

Rappel de la méthode:

1) Un résultat de séparation de classes de complexité:

$$C \not\subseteq C' \text{ (ou } C' \subset C \text{ strictement)}$$

2) Un résultat de « dureté »:

Pb est C -difficile (ou C -complet) :

$C \leq Pb$ par une réduction appropriée

On en déduit: $Pb \notin C'$



Le théorème de Ladner et ses extensions

- Si $NP \neq P$ alors il existe des problèmes dans $NP - P$ qui ne sont pas NP-complets
- *Variante linéaire* [Chapdelaine '04]:
Si $NLIN \neq DLIN$ alors il existe des problèmes dans $NLIN - DLIN$ (donc avec borne inférieure de complexité non triviale) et qui ne sont pas NLIN-complets



What about natural problems?

- It seems that *almost* each **natural problem is complete in some natural complexity class** (for appropriate reductions):
 - DIRECTED-GRAPH-REACHABILITY is NL-complete
 - UNDIRECTED-GRAPH-REACHABILITY (UGR) and TREE-REACHABILITY are both L-complete (see [Reingold '04] for UGR)



What about natural problems?

Almost every natural NP problem

- either is known to belong to P
- or is known to be NP-complete

Notable exception: GRAPH-
ISOMORPHISM

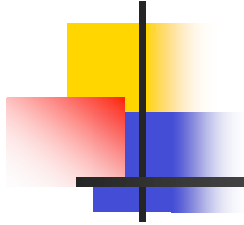


What about linear reductions?

There seems to be only a few equivalence classes for \equiv_{lin} (equivalence classes of problems for linear time bounded reductions) among natural NP-complete problems:

- NLIN-complete problems (few)
- Problems linearly equivalent to SAT (many): VERTEX-COVER, etc
- Problems linearly equivalent to PLAN-SAT: PLAN-HAMILTON, etc.

(by results of [Barbanchon 2002-2004])



In particular, any lower bound proved for SAT holds similarly for all the many problems linearly equivalent to SAT:

VERTEX-COVER, 3-COLORABILITY,
DIGRAPH-KERNEL, etc.



Retour sur les problèmes NP-complets « naturels »

Ils sont NLIN:

- leurs témoins sont « linéaires »
- leur vérification est aussi « linéaire » (= dans DLIN).

En fait, beaucoup sont « encore plus faciles ».

On va voir trois classes de problèmes « faciles »



Une première classe de problèmes « faciles »

Certains problèmes de graphes NP ou NP-complets sont dans

$$\mathbf{vertexNLIN} =_{\text{déf}} \mathbf{NTIME}_{RAM}(n)$$

où n est le nombre de sommets (« vertices ») du graphe:

Ces problèmes

- ont des **témoins de taille $O(n)$**
- sont **vérifiables en temps déterministe $O(n)$**



Exemples de problèmes dans vertexNLIN

HAMILTON, NON-PLANARITE, de
témoins:

- un circuit hamiltonien de longueur $n = |V|$
- un sous-graphe homéomorphe à $K_{3,3}$ ou K_5 de taille $O(n)$



Outils pour prouver $Pb \in \text{vertexNLIN}$

$G = (V, E)$ a une solution pour Pb

ssi G « satisfait » une formule φ_{Pb} dite **ESO(1 \forall)**

de la forme

$$\exists \mathbf{S} \forall x \psi(E, \mathbf{S}, x)$$

où $E \subseteq V^2$ relation binaire « arête » sur le domaine V ,

S liste de variables du second ordre (variables de fonctions ou de relations)

x est l'unique variable du 1^{er} ordre (sur V)

ψ est une formule sans quantificateur



La formule ESO($1\forall$) pour le problème HAMILTON

$\exists (\leq, succ, min, max)$

[\leq ordre total de fonction successeur *succ*...
 $\wedge \forall x [x \neq max \Rightarrow E(x, succ(x))]$
 $\wedge E(max, min)]$



Caractérisation logique

- *Proposition [G-Olive, 2004]:*

$Pb \in \text{vertexNLIN}$ **ssi** $Pb \in \text{ESO}(1\forall)$

(= définissable par formule ESO à 1 variable \forall du 1^{er} ordre)

- La classe $\text{ESO}(1\forall) = \text{vertexNLIN}$ est **robuste**, i.e. n'est pas augmentée si on admet:

- les notions prédéfinies *d'ordre total*, *d'arbre*, etc
- certains opérateurs de *fermeture transitive*:

ex: $f^*(x,y)$ ssi il existe k tel que $f^k(x) = y$



New results for vertexNLIN

- For each combinatorial graph problem PB one can prove (often easily)
 $PB \in \text{vertexNLIN}$ or $PB \notin \text{vertexNLIN}$
- Structural complexity results:
 - $\text{vertexDLIN} \subset \text{vertexNLIN} \subset \text{NLIN}$ (strictly)
 - $\text{vertexNLIN} \neq \text{co-vertexNLIN}$



Une seconde classe de problèmes « faciles »

La classe mixte temps-espace
 $\text{NTIMESPACE}(n, \sqrt{n})$

à laquelle appartiennent les problèmes
SAC-A-DOS, CLIQUE

Si $G = (V, E)$ possède une k -clique, alors
 $k^2 \leq |E| \leq |G| = n$, donc $k \leq \sqrt{n}$



Algorithme pour CLIQUE

Données: graphe $G = (V, E)$, entier k

- *Deviner* $C \subseteq V$ tel que $|C| = k$
en temps non déterministe $k \leq \sqrt{n}$
- *Vérifier* que C est une clique de G
en espace $O(k) = O(\sqrt{n})$ et temps $O(n)$



Algorithm for KNAPSACK in NTIMESPACE(n, \sqrt{n})

Essentially manage separately

- The « small » summands of length $< \sqrt{n}$ summing them up in space $O(\sqrt{n})$,
- The « big » summands of length $\geq \sqrt{n}$ (there are $\leq \sqrt{n}$ « big » ones): guess some of them and add them (scholar algorithm) in space $O(\sqrt{n})$



D'autres problèmes dans NTIMESPACE(n, \sqrt{n})

Beaucoup de restrictions planaires de
problèmes NP-complets:

PLAN-3-COL, PLAN-SAT, etc

Conséquence du *théorème de
l'ensemble séparateur* dans les
graphes planaires [Lipton-Tarjan 1979]



Planar Separator Theorem

The set of vertices V of each planar graph $G = (V, E)$ can be partitioned (in linear time) into three disjoint sets A, V_1, V_2 so that

- $|V_1| \leq 2/3 |V|, |V_2| \leq 2/3 |V|$
- $|A| \leq \sqrt{|V|}$
- The deletion of A disconnects V_1 from V_2



Un dernier mot sur NTIMESPACE(n, \sqrt{n})

Cette classe est aussi *caractérisable en logique ESO* [Chapdelaine-G 2003]



Le point de vue de la « complexité paramétrée » [Downey-Fellows, années 90]

Point de départ (la « part du démon »):

Beaucoup de problèmes « difficiles » ou NP-complets deviennent « faciles » si une partie de la donnée, le *paramètre* k , est « petit »



Complexité paramétrée: exemple

Problème VERTEX-COVER (VC)

Donnée: graphe $G = (V, E)$, entier k

Paramètre: k

Question: G possède-t-il une k -couverture
 $C \subseteq V$?

On peut résoudre VC par un algorithme
déterministe (avec backtrack) de temps
 $O(2^k |G|)$



Complexité paramétrée: Algo pour VC

Principe récursif:

Soit (a,b) une arête de G .

Toute k -couverture de G

- soit est de la forme $\{a\} \cup C$ où C est une $(k-1)$ -couverture de $G - \{a\}$
- soit est de la forme $\{b\} \cup C$ où C est une $(k-1)$ -couverture de $G - \{b\}$



Complexité paramétrée

Définition: un problème paramétré est **FPT** (« **Fixed Parameter Tractable** ») s'il est résoluble par un algorithme déterministe en temps

$$f(k) n^c = O_k(n^c)$$

où f est une fonction quelconque, k est le paramètre, n est la taille de la donnée et c est une constante.

Typiquement: $f(k) = O(2^k)$



Complexité paramétrée: bilan

Depuis 1990, on a montré que de nombreuses versions paramétrées de problèmes NP-complets sont FPT et très souvent FPLIN:

- VERTEX-COVER, etc.



Complexité paramétrée: exemple

Problème Multidimensional-Matching (MM)

Données: une relation $M \subseteq X_1 \times \dots \times X_r$ d'arité r et un entier k

Paramètres: r, k

Question: existe-t-il $M' \subseteq M$ avec $\text{card}(M') = k$, tel que deux éléments distincts de M' ne coïncident jamais sur aucune de leurs r coordonnées?

**MM est FPLIN,
i.e. est résoluble par algorithme en temps linéaire
 $O_{r,k}(|M|)$**



Outil de preuve: une formulation logique

$(M, r, k) \in MM$ **ssi** la structure associée
 F_M satisfait la formule

$$\exists x_1 \dots \exists x_k \bigwedge_{1 \leq i < j \leq k} \bigwedge_{1 \leq h \leq r} f_h(x_i) \neq f_h(x_j)$$

où la structure associée

$$F_M = \langle M; f_1, \dots, f_r \rangle$$

est définie par

$$f_i(y) = y_i \text{ pour } y = (y_1, \dots, y_r) \in M$$



Complexité paramétrée et logique

Proposition [Durand-G 05]: Si un problème paramétré Pb s'exprime en « **Logique Acyclique** » (logique à définir)
alors $Pb \in \mathbf{DTIME}_{\text{RAM}}(\mathbf{O}_k(n))$.



Essential tool: coverings of tables


A *covering* of a table

$$G = (g_1, \dots, g_k), g_i: D \rightarrow D',$$

is a k -tuple (c_1, \dots, c_k) such that

$$\forall x \in G \quad g_1(x) = c_1 \vee \dots \vee g_k(x) = c_k$$

Coverings of tables generalize vertex-coverings of graphs



A nontrivial application of acyclic logic

The **acyclic subgraph problem**

Input: graph G and acyclic graph H

Parameter: H

Question: Is H a subgraph of G ?

Can be computed by an algorithm in
linear time $O_H(|G|)$



Conclusion

Les problèmes NP « naturels »

- sont **de petite complexité non déterministe**: temps linéaire ou moins
- **sont souvent résolubles par algorithmes avec « backtrack limité »**: $O(k)$ pas non déterministes, et sont typiquement de complexité $O(2^k n^c)$, donc sont « Fixed Parameter » polynomiaux $O_k(n^c)$ et souvent linéaires $O_k(n)$.

Ce qui « explique » qu'on ne trouve pas de problème NP « naturel »

qui soit « dur » pour $DTIME(n^{1+\varepsilon})$ tel que $\varepsilon > 0$