

Algebraic versions of “P=NP ?”

Pascal Koiran

Laboratoire de l'Informatique du Parallélisme

Ecole Normale Supérieure de Lyon

Modnet Antalya Conference - November 4-11, 2006.

Valiant's model : $VP_K = VNP_K$?

- Complexity of a polynomial f measured by number $L(f)$ of arithmetic operations $(+,-,\times)$ needed to evaluate f :

$L(f)$ = size of smallest arithmetic circuit computing f .

- $(f_n) \in VP$ if number of variables, $\deg(f_n)$ and $L(f_n)$ are polynomially bounded. For instance, $(X^{2^n}) \notin VP$.
- $(f_n) \in VNP$ if $f_n(\bar{x}) = \sum_{\bar{y}} g_n(\bar{x}, \bar{y})$

for some $(g_n) \in VP$

(sum ranges over all boolean values of \bar{y}).

A typical VNP family : the permanent.

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

It is VNP-complete if $\text{char}(K) \neq 2$.

VP and VNP are almost the only classes studied in Valiant's framework.

Sharp contrast with the “complexity theory zoo” of discrete classes (> 400 classes at www.complexityzoo.com).

Some exceptions :

- VQP : $\deg(f_n)$ polynomially bounded and $L(f_n) \leq n^{\text{poly}(\log n)}$.
- Malod (2003) has studied versions of VP and VNP without bound on $\deg(f_n)$: VP_{nb} , VNP_{nb} ; and constant-free classes : VP^0 , VNP^0 , VP_{nb}^0 , VNP_{nb}^0 .
- We will define a class VPSPACE (or VPAR ?) which contains VNP_{nb} .

Blum-Shub-Smale model (as presented by Poizat) :

$$P_K = NP_K ?$$

- Computation model is richer : in addition to $+$, $-$, \times gates, circuits may use $=$ and (if K ordered) \leq gates.

Selection gates :

$$s(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{if } x = 1 \end{cases}$$

We may assume that $x \in \{0, 1\}$.

For instance, $s(x, y, z) = xz + (1 - x)y$.

- Focus on decision problems.

Complexity classes

- A problem : $X \subseteq K^\infty = \bigcup_{n \geq 1} K^n$.
- X is P_K if for all $x \in K^n$,

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

with C_n constructed in polynomial time by a Turing machine.

- X is NP_K if for all $x \in K^n$,

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

with $Y \in P_K$.

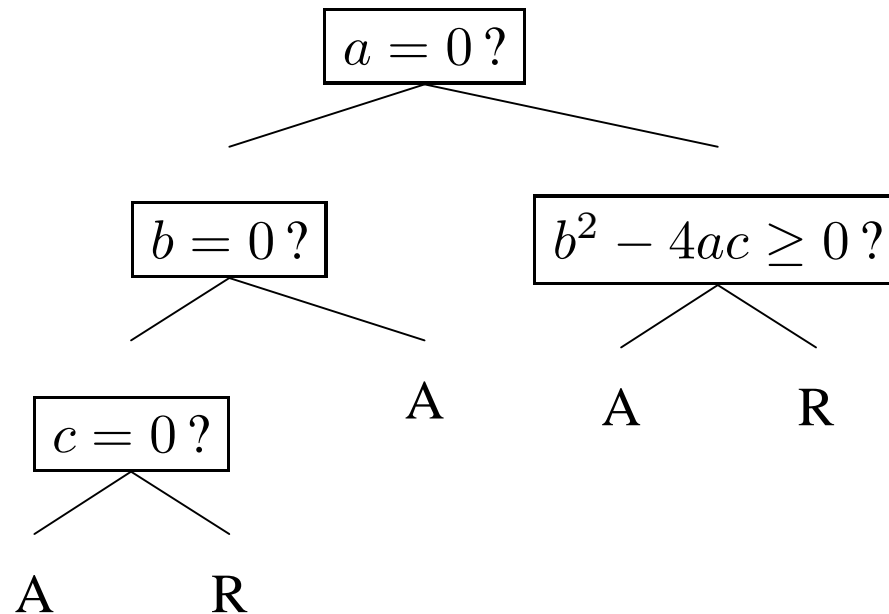
A typical $NP_{\mathbb{R}}$ -complete problem :

decide whether a polynomial of degree 4 in n variables has a real root.

Best algorithms to this day are of complexity exponential in n .

Decision trees

$$\exists x \in \mathbb{R} \ ax^2 + bx + c = 0 ?$$



Internal nodes labeled by *arbitrary* polynomials.

Complexity \equiv tree depth.

Model is unrealistic :

the complexity of polynomial evaluation should be taken into account !

Circuits versus trees

Circuit with T test ($=, \leq$) or selection gates \rightarrow tree of depth T .

Can $\text{NP}_{\mathbb{R}}$ problems be solved by decision trees of polynomial depth?

If not, $\text{P}_{\mathbb{R}} \neq \text{NP}_{\mathbb{R}}$!

Similar questions for various structures M , for instance,

$M = (\mathbb{C}, +, -, \times, =), (\mathbb{R}, +, -, \leq), (\mathbb{R}, +, -, =), \{0, 1\}$.

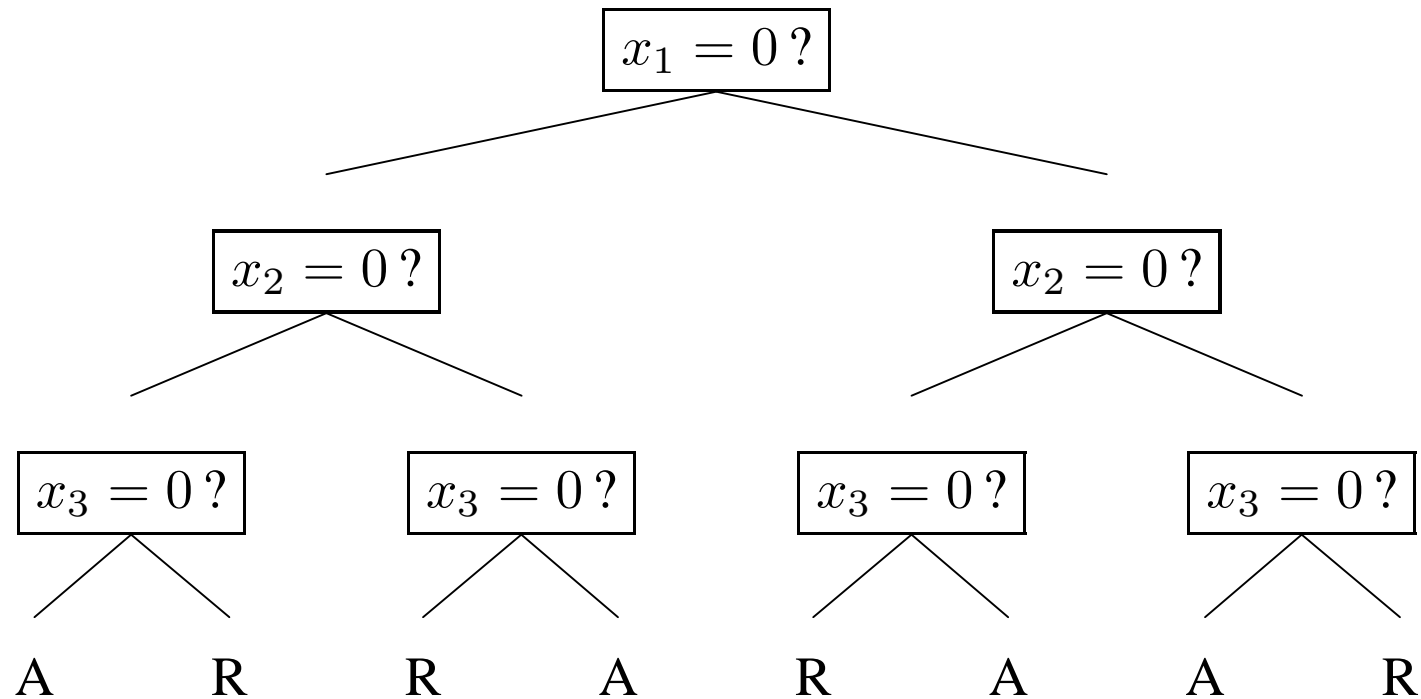
Do NP_M problems have polynomial depth decision trees ?

For $M = \{0, 1\}$, the answer is...

Labels of internal nodes are of the form “ $x_i = 0 ?$ ”.

Do NP_M problems have polynomial depth decision trees ?

For $M = \{0, 1\}$, **Yes.**



Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, =)$, the answer is...

Internal nodes are of the form :

$$a_1x_1 + \cdots + a_nx_n + b = 0?$$

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, =)$, **No.**

Twenty Questions :

INPUT : x_1, \dots, x_n .

QUESTION : $x_1 \in \{0, 1, 2, \dots, 2^n - 1\}$?

Twenty Questions is in NP_M : guess $y \in \{0, 1\}^n$,
check that $x_1 = \sum_{j=1}^n 2^{j-1} y_j$.

A *canonical path argument* shows that its decision tree complexity is 2^n .

Therefore, $\text{P}_M \neq \text{NP}_M$ (Meer).

Conjecture (Shub-Smale) : Twenty Questions is not in $\text{P}_{(\mathbb{C}, +, -, \times, =)}$.

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, \leq)$, the answer is...

Internal nodes are of the form :

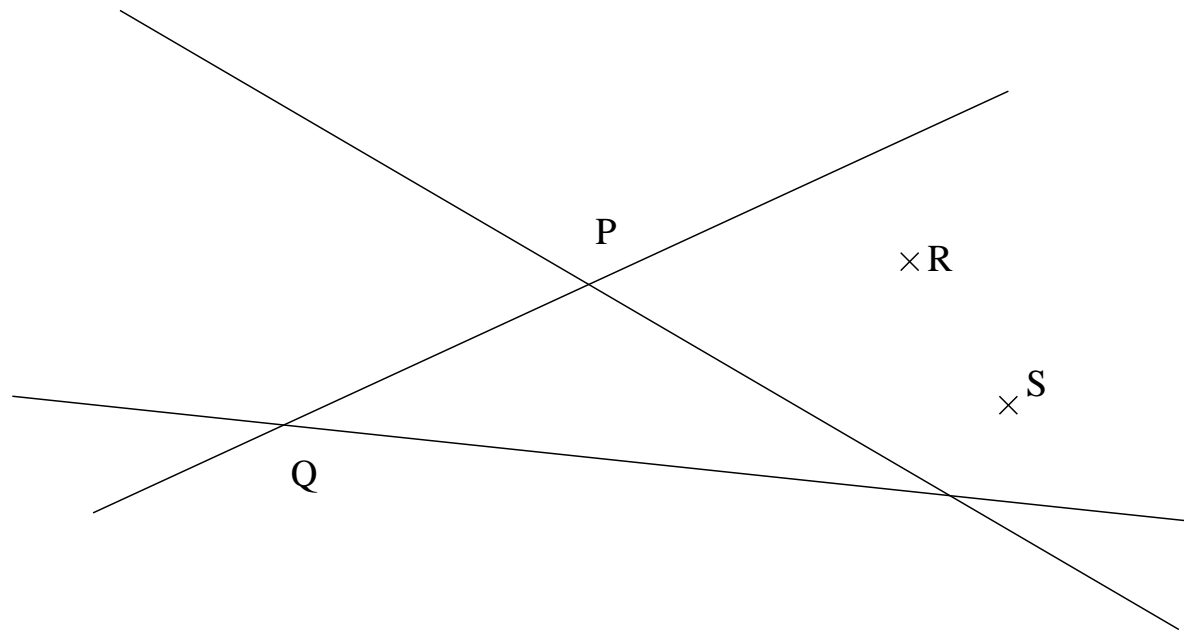
$$a_1x_1 + \cdots + a_nx_n + b \geq 0?$$

Remark : Twenty Questions *is* in P_M by binary search.

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, \leq)$, **Yes.**

Proof based on algorithms for point location in arrangements of hyperplanes.



R, S lie in the same 2-dimensional cell.

$]P, Q[$ is a 1-dimensional cell.

$\{P\}$ and $\{Q\}$ are 0-dimensional cells.

Decision trees for $\text{NP}_{(\mathbb{R},+, -, \leq)}$ problems : the construction

1. $\text{NP}_M \subseteq \text{PAR}_M$: problems solvable in parallel polynomial time (by uniform circuits of possibly exponential size).
2. For inputs in \mathbb{R}^n , any PAR_M problem is a union of *cells* of an arrangement of $2^{n^{O(1)}}$ hyperplanes.
3. In this arrangement, point location can be performed in depth $n^{O(1)}$ (Meiser, Meyer auf der Heide). Now, just label the leaves correctly.

Corollary [Fournier-Koiran] : if $\text{P} = \text{NP}$ then $\text{P}_M = \text{NP}_M$.

Proof sketch : with access to an NP oracle,
one can effectively “run” the tree on any input $x \in \mathbb{R}^n$
(i.e., construct the path followed by x from the root to a leaf).

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{C}, +, -, \times, =)$, the answer is...

Internal nodes are of the form

$$P(x_1, \dots, x_n) = 0?$$

where P is an arbitrary polynomial.

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{C}, +, -, \times, =)$, **Yes**.

Not the topic of this talk...

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, \times, \leq)$, the answer is...

Internal nodes are of the form

$$P(x_1, \dots, x_n) \geq 0?$$

where P is an arbitrary polynomial.

Do NP_M problems have polynomial depth decision trees ?

For $M = (\mathbb{R}, +, -, \times, \leq)$, **Yes**.

1. $\text{NP}_{\mathbb{R}} \subseteq \text{PAR}_{\mathbb{R}}$: problems solvable in parallel polynomial time (by uniform circuits of possibly exponential size).
2. For inputs in \mathbb{R}^n , any $\text{PAR}_{\mathbb{R}}$ problem is a union of *cells* of an arrangement of $2^{n^{O(1)}}$ hypersurfaces of degree $2^{n^{O(1)}}$.

Fix polynomials P_1, \dots, P_s .

Two points x and y are in the same cell if $\text{sign}(P_i(x)) = \text{sign}(P_i(y))$ for all $i = 1, \dots, s$.

Here, $\text{sign}(a) \in \{-1, 0, 1\}$.

3. In this arrangement, point location can be performed in depth $n^{O(1)}$.
Now, just label the leaves correctly.

Point location in arrangements of real hypersurfaces

Theorem [Grigoriev] : Point location can be done in depth $O(\log N)$, where N is the number of nonempty cells.

Remark : $N \leq (sd)^{O(n)}$ where $d = \max_{i=1, \dots, s} \deg(P_i)$.

Hence $\log N = n^{O(1)}$.

Consider inputs x with $P_i(x) \neq 0$ for all i .

Nodes are of the form “ $\prod_{j \in F} P_j(x) > 0$?”, where F is as follows.

Divide and Conquer Lemma :

Let $X = \{1, \dots, s\}$ and F_1, \dots, F_N nonempty subsets of X .

There exists $F \subseteq X$ such that $N/3 \leq |\{F_x; |F \cap F_x| \text{ even}\}| \leq 2N/3$.

Apply to sets F_x defined by conditions of the form :

$$j \in F_x \Leftrightarrow P_j(x) < 0.$$

Then $\prod_{j \in F} P_j(x) > 0 \Leftrightarrow |F \cap F_x| \text{ even}$.

Improved version of divide and conquer lemma

Theorem [Charbit, Jeandel, Koiran, Périfel, Thomassé] :

The range $[\frac{N}{3}, \frac{2N}{3}]$ can be replaced by $[\frac{N}{2} - \alpha, \frac{N}{2} + \alpha]$ where $\alpha = \sqrt{N}/2$.

Remark : One must have $\alpha = \Omega(\sqrt{N}/(\log N)^{1/4})$.

Probabilistic proof : for a random subset F , let

$Y_i = 1$ if $|F \cap F_i|$ is even, and $Y_i = -1$ otherwise.

Need to show that there exists F such that $Y^2 \leq N$, where $Y = \sum_{i=1}^N Y_i$.

This follows from $E[Y^2] = N$:

$$E[Y^2] = E\left[\sum_{i=1}^N Y_i^2 + 2 \sum_{i < j} Y_i Y_j\right]$$

but $E[Y_i^2] = 1$ and for $i \neq j$, by pairwise independence :

$$E[Y_i Y_j] = E[Y_i]E[Y_j] = 0.$$

This can be turned into a deterministic logspace algorithm.

A remark on derandomization

From Motwani, Naor and Naor 1994 :

“A natural approach towards de-randomizing algorithms is to find a method for searching the associated sample Ω for a good point w with respect to a given input instance I . Given such a point w , the algorithm $\mathcal{A}(I, w)$ is now a deterministic algorithm and it is guaranteed to find a correct solution. The problem faced in searching the sample space is that it is generally exponential in size. The result of Adleman showing that $RP \subseteq P/poly$ implies that the sample space Ω associated with a randomized algorithm always contains a polynomial-sized subspace which has a good point for each possible input instance. However, this result is highly non-constructive and it appears that it cannot be used to actually de-randomize algorithms.”

Adleman strikes back

Given s and N , our deterministic logspace algorithm constructs a list of $s^2 N^2 (N + 1)^2$ subsets of $X = \{1, \dots, s\}$ such that for any input F_1, \dots, F_N :

$$-\frac{\sqrt{N}}{2} \leq |\{F_x; |F \cap F_x| \text{ even}\}| - \frac{N}{2} \leq \frac{\sqrt{N}}{2}.$$

holds for some element F of the list.

The deterministic algorithm then performs an exhaustive search in this list.

Effective point location :

Taking the complexity of polynomials into account

For a problem $A \in \text{PAR}_{\mathbb{R}}$, hypersurfaces of the arrangement are defined by polynomials P_i in P-uniform VPAR :

Families of polynomials computed by uniform arithmetic circuits of polynomial depth.

Nodes of the tree of the form “ $\prod_{i \in F} P_i(x) > 0$?” where $F \in \text{PSPACE}$:
in P-uniform VPAR.

Labels of leaves can be computed in PSPACE.

Theorem [Koiran-Périfel] : If VPAR families have polynomial size circuits, then $\text{PAR}_{\mathbb{R}}$ problems have polynomial size circuits.

Can VPAR families have polynomial size circuits ?

- Very strong hypothesis.
- Admits several versions (6 ?), depending on uniformity conditions and role of constants.

With P/poly-uniformity and Valiant's convention for constants :

$$\begin{array}{l} \text{(i) } \text{VPAR} = \text{VP}_{nb}. \\ \Updownarrow \\ \text{(ii) } \text{VP} = \text{VNP} \text{ and } \text{PSPACE} \subseteq \text{P/poly}. \end{array}$$

$\text{VPAR} = \text{VP}_{nb} \Rightarrow \text{PSPACE} \subseteq \text{P/poly}$ assumes GRH
(seems necessary to handle arbitrary constants).

Can we refute $[VP = VNP \text{ and } PSPACE \subseteq P/\text{poly}]$?

To prove that $\neg(A \wedge B)$, one does not always have to prove $\neg A$ or $\neg B$.

For instance, we know that $LOGSPACE \neq P$ or $P \neq PSPACE$.

It was shown by Bürgisser that (under GRH),

$VP = VNP \Rightarrow NP \subseteq NC/\text{poly}$ (problems recognized by polynomial size boolean circuits of polylogarithmic depth).

Hence, assuming GRH, the hypothesis implies that $PSPACE \subseteq NC/\text{poly}$.

Most uniform version of this hypothesis

P-uniform $\text{VPAR}^0 = \text{P-uniform VP}_{nb}^0 \Rightarrow \text{P-uniform NC} = \text{PSPACE}.$

Proof is in two steps. Hypothesis implies :

(i) $\text{P} = \text{PSPACE}.$

(ii) $\text{P-uniform NC} = \bigoplus \text{P}.$

Proof of (ii) based on $\bigoplus \text{P}$ -completeness of $\bigoplus \text{HAMILTONIAN PATHS}.$

Note that $\# \text{HAMILTONIAN PATHS}$ is of the form

$$\sum_{\sigma: n\text{-cycle}} \prod_{i \neq \text{end}(\sigma)} a_{i\sigma(i)}$$

where (a_{ij}) is the graph's adjacency matrix.

Remark : It is known that $\text{LOGSPACE-uniform NC} \neq \text{PSPACE}.$

VPSPACE

Theorem :

A polynomial family $f_n \in \mathbb{Z}[X_1, \dots, X_{p(n)}]$ is in P-uniform VPAR^0 iff :

- (i) $p(n)$ is polynomially bounded.
- (ii) $\deg(f_n)$ is exponentially bounded.
- (iii) The bit size of the coefficients of f_n is exponentially bounded.
- (iv) The map $(1^n, \bar{\alpha}) \mapsto a_{n, \bar{\alpha}}$ is PSPACE computable, where

$$f_n(\bar{X}) = \sum_{\bar{\alpha}} a_{n, \bar{\alpha}} \bar{X}^{\bar{\alpha}}.$$

This characterization is useful in the proof that

$$[\text{VP} = \text{VNP} \text{ and } \text{PSPACE} \subseteq \text{P/poly}] \Rightarrow \text{VPAR} = \text{VP}_{nb}.$$

A natural example of a VPAR family

Resultants of multivariate polynomial systems form a VPAR family.

Proof sketch :

- (i) The *Macaulay matrix* is an exponential size matrix whose non-zero entries are coefficients of the polynomial system.
- (ii) Determinants can be computed by arithmetic circuits of polylogarithmic depth.

Outcome of this work

- Focus put back on evaluation problems :
to show that certain decision problems (in $\text{NP}_{\mathbb{R}}$, or $\text{PAR}_{\mathbb{R}}$) are hard, one must first be able to show that certain evaluation problems (in VPAR) are hard.
- Suggestion of new lower bound problems :
various versions of “ $\text{VP}_{nb} = \text{VPAR} ?$ ”.
- Other natural (complete ?) polynomial families in $\text{VPAR} ?$