Interpolation in Valiant’s theory

Pascal Koiran    Sylvain Perifel

LIP, ENS Lyon

Paderborn, October 30, 2007
Introduction

Two ways of computing a polynomial with integer coefficients

- Algorithm that evaluates the polynomial at an integer point.
  Example: \( P(x, y) = (x + y)^2 \) on input \((1, 3) \rightarrow 16\).
Two ways of computing a polynomial with integer coefficients

- Algorithm that evaluates the polynomial at an integer point.
  Example: $P(x, y) = (x + y)^2$ on input $(1, 3) \rightarrow 16$.

- Arithmetic circuit that computes the polynomial.
  Example:

```
x y
  |  |
  |  +
  |   |
  |   x
```

If a polynomial $P$ can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?
Question ♠

If a polynomial $P$ can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?

In other words, does the use of boolean operations other than $+$ and $\times$ enable a superpolynomial speed-up in the computation?
If a polynomial $P$ can be evaluated by a polynomial-time algorithm, is it true that it is computable by an arithmetic circuit of polynomial size?

In other words, does the use of boolean operations other than $+$ and $\times$ enable a superpolynomial speed-up in the computation?

The use of families of polynomials makes these questions meaningful.
Divisions

- Strassen: positive answer for divisions if the polynomial has a polynomial degree.
- Idea: replace $\frac{1}{1-x}$ by $1 + x + x^2 + \cdots + x^{p(n)}$. 
Divisions

- Strassen: positive answer for divisions if the polynomial has a polynomial degree.
- Idea: replace $\frac{1}{1-x}$ by $1 + x + x^2 + \cdots + x^{p(n)}$.
- What if the degree is not polynomial?
In order to show that question ♣ has a negative answer, one looks for a polynomial $P$ that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.
In order to show that question ♣ has a negative answer, one looks for a polynomial $P$ that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.

But lack of candidates (usual examples don’t work: determinant, permanent, etc.).
In order to show that question ♣ has a negative answer, one looks for a polynomial $P$ that can be evaluated in polynomial time but cannot be computed by polynomial-size circuits.

But lack of candidates (usual examples don’t work: determinant, permanent, etc.).

In order to show that question ♣ has a positive answer, one wants to transform an evaluation algorithm into an arithmetic circuit.
If question ♣ has a negative answer, then $\text{VP} \neq \text{VNP}$.
1. Valiant’s classes

2. The counting hierarchy

3. Interpolation

4. Consequences
Arithmetic circuits:

- gates $+$ and $\times$
- inputs $x_1, \ldots, x_n$ and the constant $-1$
- $\rightarrow$ multivariate polynomials with integer coefficients.
We will skip the problem of constants and of uniformity...
P and NP in Valiant’s model

- **Family of polynomials** \((f_n)\): one circuit \(C_n\) per polynomial \(f_n \in \mathbb{Z}[x_1, \ldots, x_{u(n)}]\).
P and NP in Valiant’s model

- **Family of polynomials** \((f_n)\): one circuit \(C_n\) per polynomial \(f_n \in \mathbb{Z}[x_1, \ldots, x_{u(n)}]\).

- **VP**: families of polynomials of polynomial degree computed by arithmetic circuits of polynomial size. Example: the determinant

\[
\det_n(x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{n,n}) = \sum_{\sigma \in S_n} \varepsilon(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)}.
\]
P and NP in Valiant’s model

- **VNP**: exponential sum of a VP family. If $(f_n(x_1, \ldots, x_{u(n)}, y_1, \ldots, y_{p(n)})) \in \text{VP}$,

$$
\sum_{\bar{\epsilon} \in \{0, 1\}^{p(n)}} f_n(\bar{x}, \bar{\epsilon})
$$

Example: the permanent (VNP-complete)

$$
\text{per}_n(x_{1,1}, \ldots, x_{1,n}, x_{2,1}, \ldots, x_{n,n}) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}.
$$
Languages (PP) or functions (♯P). We will focus on languages.
Counting classes

- Languages (PP) or functions (♯P). We will focus on languages.

- A language $A$ is in PP if there exists a polynomial-time nondeterministic Turing machine such that $x \in A$ iff more than half of the computation paths are accepting.
Counting classes

- Languages (PP) or functions (♯P). We will focus on languages.

- A language $A$ is in PP if there exists a polynomial-time nondeterministic Turing machine such that $x \in A$ iff more than half of the computation paths are accepting.

- A function $f : \{0, 1\}^* \rightarrow \mathbb{N}$ is in #P if it counts the number of accepting paths of a polynomial-time nondeterministic Turing machine.
Counting hierarchy

Counting hierarchy: $\text{CH} = \text{PP} \cup \text{PP}^{\text{PP}} \cup \text{PP}^{\text{PP}^{\text{PP}}} \cup \ldots$ (similarity with the polynomial hierarchy).
Counting hierarchy

- Counting hierarchy: \( \text{CH} = \text{PP} \cup \text{PP}^{\text{PP}} \cup \text{PP}^{\text{PP}^{\text{PP}}} \cup \ldots \) (similarity with the polynomial hierarchy).

- Majority operator \( \mathbf{C} \): if \( C \) is a complexity class, \( \mathbf{C}.C \) is the set of languages \( A \) such that there exists a language \( B \in C \) satisfying:

\[
x \in A \iff \#\{y \in \{0, 1\}^{p(|x|)} \mid (x, y) \in B\} \geq 2^{p(|x|) - 1}.
\]

- \( C_0 P = P \) et \( C_{i+1} P = \mathbf{C}.C_i P \). Then \( \text{CH} = \cup_i C_i P \).
Some inclusions

PSPACE

CH

$P^{PP}$

PH

NP

BPP

P
A central lemma

Lemma

If $VP = VNP$ then $CH = P$.

Proof (idea)
If $VP = VNP$ then the permanent has polynomial-size arithmetic circuits. Then it can be evaluated in polynomial time. Since the permanent is $\#P$-complete, it yields $PP = P$, hence $CH = P$. □
Sequences of integers

**Definition**

A sequence of integers \((a_{n,k})_{k \leq 2^p(n)}\) of exponential bitsize is computable in CH if

\[\{(1^n, k, j, b) \mid \text{the } j\text{-th bit of } a_{n,k} \text{ is } b\} \in \text{CH}.\]
Some results of Bürgisser

Theorem (Bürgisser)

If \((a_{n,k})\) is computable in \(\mathsf{CH}\), then it is also the case of

\[
\begin{align*}
c_n &= \sum_{k=0}^{2^{\mathcal{P}(n)}} a(n, k) \quad \text{and} \quad d_n = \prod_{k=0}^{2^{\mathcal{P}(n)}} a(n, k).
\end{align*}
\]

Proof (idea)

Key ingredient: iterated addition and multiplication are in \(\mathsf{LOGTIME}\)-uniform \(\mathsf{TC}^0\) (recent result of Hesse, Allender and Barrington for the multiplication). Then scaling up to obtain the result on the counting hierarchy.

\(\mathsf{TC}^0\): polynomial-size circuits of constant depth with majority gates.

\(\mathsf{LOGTIME}\)-uniform: very strong uniformity condition.
Main result (bis)

If question ♣ has a negative answer, then $VP \neq VNP$. 

In other words, if $VP = VNP$ then question ♣ has a positive answer: we know how to transform an evaluation algorithm into an arithmetic circuit.
If question ♠ has a negative answer, then \( VP \neq VNP \).

In other words, if \( VP = VNP \) then question ♠ has a positive answer: \textit{we know how to transform an evaluation algorithm into an arithmetic circuit.}
Some tools from Lagrange

Going from the evaluation at integer points to the computation:

Lagrange interpolation.
Some tools from Lagrange

Going from the evaluation at integer points to the computation: Lagrange interpolation.

Lemma (Lagrange interpolation)

Let \( p(x) \) be a polynomial in one variable and of degree \( \leq d \). Then

\[
p(x) = \sum_{i=0}^{d} p(i) \prod_{j \neq i} \frac{x - j}{i - j},
\]

where the integer \( j \) ranges from 0 to \( d \).

Proof

Both polynomials are of degree \( \leq d \) and coincide on \( d + 1 \) points.
Lemma

Let $p(x_1, \ldots, x_n)$ be a polynomial of degree $\leq d$. Then

$$p(x_1, \ldots, x_n) = \sum_{0 \leq i_1, \ldots, i_n \leq d} p(i_1, \ldots, i_n) \prod_{k=1}^{n} \left( \prod_{j_k \neq i_k} \frac{x_k - j_k}{i_k - j_k} \right),$$

where the integers $j_k$ range from 0 to $d$. 
Main result (ter)

<table>
<thead>
<tr>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let ((f_n(x_1, \ldots, x_{u(n)}))) be a family of polynomials. We say that ((f_n)) can be evaluated in (\text{CH}) at integer points if</td>
</tr>
<tr>
<td>[{(1^n, i_1, \ldots, i_{u(n)}, j, b) \mid \text{the } j\text{-th bit of } f_n(i_1, \ldots, i_{u(n)}) \text{ is } b} \in \text{CH}.]</td>
</tr>
</tbody>
</table>
Main result (ter)

Definition

Let \((f_n(x_1, \ldots, x_{u(n)}))\) be a family of polynomials. We say that \((f_n)\) can be evaluated in \(\text{CH}\) at integer points if

\[
\{(1^n, i_1, \ldots, i_{u(n)}, j, b) \mid \text{the } j\text{-th bit of } f_n(i_1, \ldots, i_{u(n)}) \text{ is } b\} \in \text{CH}.
\]

What we will show:

(if \(\text{VP} = \text{VNP}\) and \(f\) can be evaluated in \(\text{CH}\) at integer points)
then \(f\) has a polynomial-size circuit.
Valiant’s criterion

**Definition of $\text{VP}_{\text{nb}}$:** idem $\text{VP}$ but without the polynomial constraint on the degree

$\longrightarrow$ families of polynomials computed by arithmetic circuits of polynomial size.
Valiant’s criterion

**Definition of $\text{VP}_{\text{nb}}$:** idem $\text{VP}$ but without the polynomial constraint on the degree → families of polynomials computed by arithmetic circuits of polynomial size.

**Lemma**

Let

$$f_n(x_1, \ldots, x_n) = \sum_{\alpha(1), \ldots, \alpha(n)} a(n, \alpha(1), \ldots, \alpha(n))x_1^{\alpha(1)} \cdots x_n^{\alpha(n)},$$

where $a(n, \alpha(1), \ldots, \alpha(n))$ is a sequence of integers computable in CH.

If $\text{VP} = \text{VNP}$ then $(f_n) \in \text{VP}_{\text{nb}}$. 
Main theorem

**Theorem**

Let \((f_n(x_1, \ldots, x_{u(n)})\) be a family of multivariate polynomials. Suppose \((f_n)\) can be evaluated in \(\text{CH}\) at integer points. If \(\text{VP} = \text{VNP}\) then \((f_n) \in \text{VP}_{nb}\).
**Theorem**

Let \((f_n(x_1, \ldots, x_{u(n)}))\) be a family of multivariate polynomials. Suppose \((f_n)\) can be evaluated in \(\text{CH}\) at integer points. If \(\text{VP} = \text{VNP}\) then \((f_n) \in \text{VP}_{\text{nb}}\).

**Proof (idea)**

- By the results of Bürgisser, the coefficients of the interpolation polynomial are computable in \(\text{CH}\).
- By Valiant’s criterion, if \(\text{VP} = \text{VNP}\) then \((f_n) \in \text{VP}_{\text{nb}}\). □
Under the hypothesis $VP = VNP$, we aim at showing that a family of polynomials that can be “easily evaluated” has polynomial-size circuits.
Under the hypothesis $\text{VP} = \text{VNP}$, we aim at showing that a family of polynomials that can be “easily evaluated” has polynomial-size circuits.

Idea: use Lagrange interpolation (enables to go from the evaluation to the polynomial itself).
Under the hypothesis $\text{VP} = \text{VNP}$, we aim at showing that a family of polynomials that can be “easily evaluated” has polynomial-size circuits.

Idea: use Lagrange interpolation (enables to go from the evaluation to the polynomial itself).

Technical points:

Valiant’s criterion: if the coefficients are computable in CH, then the polynomial has polynomial-size circuits (under the hypothesis that $\text{VP} = \text{VNP}$)

the results of Bürgisser enable to compute in CH the coefficients of the interpolation polynomial.
Consequence for question ♣

Theorem

If question ♣ has a negative answer, then $VP \neq VNP$. 

Remark: if question ♣ has a positive answer, then $P = \text{PP} \Rightarrow VP = VNP$. 
Consequence for question ♣

**Theorem**

*If question ♣ has a negative answer, then VP ≠ VNP.*

**Remark:** if question ♣ has a *positive* answer, then $P = PP \Rightarrow VP = VNP$. 
Bounded and unbounded versions

Theorem

(In a constant-free context)

$$\text{VP} = \text{VNP} \Rightarrow \text{VP}_{\text{nb}} = \text{VNP}_{\text{nb}}.$$ 

Remark: on fields of positive characteristic, this result was shown by Malod (2003).
Transfer toward BSS

- Algebraic versions of P and NP: Blum-Shub-Smale model.
- On a field $K$ of characteristic zero, operations $+$, $\times$ and $=$.
Transfer toward BSS

- Algebraic versions of P and NP: Blum-Shub-Smale model.
- On a field $K$ of characteristic zero, operations $+$, $\times$ and $=$.
- Separation of $P^K$ and $NP^K$ thanks to problems in $NP(K,+,=)$? (Twenty Questions, Subset Sum, . . . )
Transfer toward BSS

- Algebraic versions of \( P \) and \( NP \): Blum-Shub-Smale model.
- On a field \( K \) of characteristic zero, operations \( +, \times \) and \( = \).
- Separation of \( P_K \) and \( NP_K \) thanks to problems in \( NP_K(\,+,=) \)?
  (Twenty Questions, Subset Sum, . . . )

**Theorem**

\[
VP = VNP \Rightarrow NP(\,+,=) \subseteq P(\,+,\times,=).
\]

We use exponential-size products as an intermediate step.
Question ♣ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.
Conclusion

▶ Question ♠ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.

▶ Little intuition on the answer.
Conclusion

- Question ♠ is central but difficult: if the answer is positive, we obtain a transfer result; otherwise we obtain the separation of VP and VNP.

- Little intuition on the answer.

- Candidates for a negative answer? (polynomials that can be easily evaluated but that do not have polynomial-size circuits)
Outline

1. Valiant’s classes
2. The counting hierarchy
3. Interpolation
4. Consequences