Complexity of Circuit Satisfiability

Ramamohan Paturi

University of California, San Diego
jointly with Pavel Pudlák, Czech Academy of Sciences

November 9, 2009

Overview

- Exact Algorithms
- Examples of Recent Progress
- Complexity Theory of Exact Algorithms
- Circuit Satisfiability Resource Trade-offs

Exact Algorithms for **NP**-complete Problems

- Exact solutions, worst-case complexity

Exact Algorithms for **NP**-complete Problems

- Exact solutions, worst-case complexity
- Exponential-time algorithms, an active field of research

Exact Algorithms for **NP**-complete Problems

- Exact solutions, worst-case complexity
- Exponential-time algorithms, an active field of research
- Improvements over exhaustive search

Exact Algorithms for **NP**-complete Problems

- Exact solutions, worst-case complexity
- Exponential-time algorithms, an active field of research
- Improvements over exhaustive search
- Goal: Limitations

Exact Complexity — **NP** Parameterization

- Two parameters with each instance: size of input and a complexity parameter

Exact Complexity — **NP** Parameterization

- Two parameters with each instance: size of input and a complexity parameter
- Natural and robust complexity parameters
  1. Satisfiability: n, the number of variables and m, input size
  2. Hamiltonian path: n, the number of vertices and m, size of the graph

Exact Complexity — **NP** Parameterization

- Two parameters with each instance: size of input and a complexity parameter
- Natural and robust complexity parameters
  1. Satisfiability: n, the number of variables and m, input size
  2. Hamiltonian path: n, the number of vertices and m, size of the graph
- **NP**:
  - $L \in \mathbf{NP}$ if $\exists p(.), \Phi(., .)$ such that
    $x \in L$ iff $\exists y, |y| \leq p(x), \Phi(x, y)$
  - where $\Phi(x, y)$ is a poly-time decidable relation. and $p(x)$ is poly-time computable and polynomially bounded

Exact Complexity — **NP** Parameterization

- Two parameters with each instance: size of input and a complexity parameter
- Natural and robust complexity parameters
  1. Satisfiability: n, the number of variables and m, input size
  2. Hamiltonian path: n, the number of vertices and m, size of the graph
- **NP**:
  - $L \in \mathbf{NP}$ if $\exists p(.), \Phi(.,.)$ such that $x \in L$ iff $\exists y, |y| \leq p(x), \Phi(x, y)$
  - where $\Phi(x, y)$ is a poly-time decidable relation. and $p(x)$ is poly-time computable and polynomially bounded
- Canonical parameterization for **NP**: $\mathbf{NP}(n, m)$
  - $|x|$, size of the input and $p(x)$, the complexity parameter

Nontrivial Exact Algorithms

- **NP**(n, m)
- Trivial exact algorithms: worst-case time complexity — $O(\mathrm{poly}(m)2^n)$
- Nontrivial exact algorithms: worst-case time complexity — $O(\mathrm{poly}(m)2^{\mu n})$, $\mu < 1$ may depend on the class of instances.
- Also known as moderately exponential-time or improved exponential-time algorithms

Examples

- Example 1: TSP
  - Input $G = (V, E, W)$, $|V| = n, |E| = m$, with $p(G) = \log n!$
  - Held-Karp dynamic programming algorithm with $O(n^2 2^n)$ is nontrivial.

Examples

- Example 1: TSP
  - Input $G = (V, E, W)$, $|V| = n, |E| = m$, with $p(G) = \log n!$
  - Held-Karp dynamic programming algorithm with $O(n^2 2^n)$ is nontrivial.
- Open Problem: Find a nontrivial exact algorithm for TSP (or Hamiltonian path) under the complexity parameter, n, the number of vertices.

Examples

- Example 1: TSP
  - Input $G = (V, E, W)$, $|V| = n, |E| = m$, with $p(G) = \log n!$
  - Held-Karp dynamic programming algorithm with $O(n^2 2^n)$ is nontrivial.
- Open Problem: Find a nontrivial exact algorithm for TSP (or Hamiltonian path) under the complexity parameter, n, the number of vertices.
- Example 2: k-SAT
  - Input CNF F where each clause has at most k literals. $|F| = m, p(F) = n$, the number of variables
  - Best-known algorithms with nontrivial upper bounds of the form $2^{n(1-c/k)}$ for $c > 1$.

Examples

- Example 1: TSP
  - Input $G = (V, E, W)$, $|V| = n, |E| = m$, with $p(G) = \log n!$
  - Held-Karp dynamic programming algorithm with $O(n^2 2^n)$ is nontrivial.
- Open Problem: Find a nontrivial exact algorithm for TSP (or Hamiltonian path) under the complexity parameter, n, the number of vertices.
- Example 2: k-SAT
  - Input CNF F where each clause has at most k literals. $|F| = m, p(F) = n$, the number of variables
  - Best-known algorithms with nontrivial upper bounds of the form $2^{n(1-c/k)}$ for $c > 1$.
- **SUBEXP**: for every $\epsilon > 0$, $\exists$ algorithm with time complexity $O(\mathrm{poly}(|x|)2^{\epsilon p(x)})$
- Open Problem: Does there exist a **SUBEXP** algorithm for k-SAT?
- If not, what are the best possible exponents?

Why Exact Algorithms?

- Certain applications will benefit from exact solutions even for moderate size parameters.
- Approximation algorithms are not always satisfactory. Moreover, it is hard to approximate for some problems.
- Constant factor improvements in the exponent will lead to similar improvements in the size of computationally feasible inputs
- Designing improved exact algorithms is leading to new algorithmic techniques and analyses
- Refined understanding of the complexity relationships among **NP**-hard problems
- Much work has been on heuristic algorithms for 3-SAT and other problems which can solve fairly large instances.
  - Rigorous analysis of heuristics
  - What are the hard instances?

Maximum Independent Set

- Given $G = (V, E)$, find a maximum size independent set with the number of vertices as the complexity parameter
- $2^{0.334n}$ algorithm in polynomial space — Tarjan and Trojanowski 1977
- $2^{0.304n}$ algorithm in polynomial space — T. Jian 1986
- $2^{0.296n}$ in polynomial space and $2^{0.276n}$ in exponential space — Robson 1986
- $2^{0.25n}$ — Robson 2001, relatively long, partially computer-generated proof in a technical report
- $2^{0.287n}$ in polynomial space using measure and conquer analysis technique — Fomin, Grandoni, and Kratsch 2006
- Better bounds are known for sparse graphs.

k-SAT

- Decide if given a k-CNF Φ is satisfiable. n, the number of variables is the complexity parameter
- Best known bounds for small values of k: $2^{?n}$

k	unique-k–SAT	k–SAT	k-SAT	k-SAT	k-SAT
3	0.386...	0.521...	0.415...	0.409...	0.404...
4	0.554...	0.562...	0.584...		0.559...
5	0.650...		0.678...		
6	0.711...		0.736...		
	Paturi,Pudlák,Saks,Zane		Schöning	Rolf, ...	Iwama,Tamaki

- Best bound for $k \geq 5$: $2^{(1-\mu_k/(k-1))n}$ with $\mu_k \approx 1.6$ for large k.

Graph Coloring to Tutte Polynomial

- Dramatic progress on k-colorability, chromatic number, and Tuttle polynomial — the power of inclusion-exclusion
- All can be solved in 2^n time and in 2^n space — Björklund, Husfeldt, Kaski, Koivisto 2006-2008
- Tutte polynomial can also be solved in 3^n time and polynomial space
- Chromatic number can be computed in $2^{1.167n}$ time in polynomial space
- 3-colorability: $2^{0.41n}$ in polynomial space — Beigel and Eppstein, 2005
- 4-colorability: $2^{0.807n}$ in polynomial space — Byskov, 2004

Other Problems and Techniques

- Minimum dominating set, treewidth, maximum cut, minimum feedback vertex set, . . .
- Pruning the search tree (Davis-Putnam, Branch and Reduce)
- Dynamic Programming
- Local search
- Measure and conquer
- Inclusion-exclusion, Fourier transform, Möbius inversion
- Color coding
- Group algebra
- Matrix multiplication
- Exponential-time divide-and-conquer
- Sieve algorithms

Exact Complexity

- Which problems have such improved algorithms?
  Is there a c^n algorithm for TSP with $c < 2$?

Exact Complexity

- Which problems have such improved algorithms?
  Is there a c^n algorithm for TSP with $c < 2$?
- Can these improvements extend to arbitrarily small exponents?
  Is 3-SAT in **SUBEXP**? How about 3-coloring?

Exact Complexity

- Which problems have such improved algorithms?
  Is there a c^n algorithm for TSP with $c < 2$?
- Can these improvements extend to arbitrarily small exponents?
  Is 3-SAT in **SUBEXP**? How about 3-coloring?
- Can we prove improvements beyond a certain point are not possible (at least under some complexity assumption)?
  Lower bounding the exponent for 3-SAT under suitable complexity assumptions?

Exact Complexity

- Which problems have such improved algorithms?
  Is there a c^n algorithm for TSP with $c < 2$?
- Can these improvements extend to arbitrarily small exponents?
  Is 3-SAT in **SUBEXP**? How about 3-coloring?
- Can we prove improvements beyond a certain point are not possible (at least under some complexity assumption)?
  Lower bounding the exponent for 3-SAT under suitable complexity assumptions?
- Is progress on different problems connected? If k-coloring has a c^n algorithm, can we prove k-SAT has a d^n algorithm? c and d are independent of k.

Approach

- Consider natural, though restricted, models of computations
- Limitations
- **CircuitSat**

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms
- Several algorithms couched as exponential-time can in fact be seen as OPP algorithms based on an observation by Eppstein

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms
- Several algorithms couched as exponential-time can in fact be seen as OPP algorithms based on an observation by Eppstein
- OPP: space efficiency, parallelization, speed-up by quantum computation

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms
- Several algorithms couched as exponential-time can in fact be seen as OPP algorithms based on an observation by Eppstein
- OPP: space efficiency, parallelization, speed-up by quantum computation
- What is the best success probability achievable in OPP?

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms
- Several algorithms couched as exponential-time can in fact be seen as OPP algorithms based on an observation by Eppstein
- OPP: space efficiency, parallelization, speed-up by quantum computation
- What is the best success probability achievable in OPP?
- SAT problems can be solved with probability $2^{-n+O(\lg n)}$ in OPP.

OPP: Two Resource Computational Model

- OPP: one-sided error probabilistic polynomial-time algorithms
- Includes several Davis-Putnam style backtracking algorithms, local search algorithms
- Several algorithms couched as exponential-time can in fact be seen as OPP algorithms based on an observation by Eppstein
- OPP: space efficiency, parallelization, speed-up by quantum computation
- What is the best success probability achievable in OPP?
- SAT problems can be solved with probability $2^{-n+O(\lg n)}$ in OPP.
- Hamiltonian path problem can be solved with probability $1/n!$ in OPP, whereas it can be solved in $n^2 2^n$ time using the inclusion-exclusion principle.

Time and Success Probability

- Consider $\lg t + \lg 1/p$ for time t and success probability p.
- For what problems, does this quantity decrease with time?
- If one can present evidence that Hamiltonian path cannot achieve c^{-n} success probability in OPP, then we provide evidence for the relative power of algorithmic paradigms — for example, exponential-time may be strictly advantageous
- On the other hand, c^{-n} OPP algorithm for Hamiltonian path would be exciting.

Prior Work

- Possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same.

Prior Work

- Possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same.
- **SNP** $\subseteq$ **SUBEXP** if any **SERF**-complete problem for **SNP** is in **SUBEXP** — Impagliazzo, Paturi and Zane 1998

Prior Work

- Possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same.
- **SNP** $\subseteq$ **SUBEXP** if any **SERF**-complete problem for **SNP** is in **SUBEXP** — Impagliazzo, Paturi and Zane 1998
- Some **SERF**-complete languages for **SNP**: k-SAT , k-coloring

Prior Work

- Possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same.
- **SNP** $\subseteq$ **SUBEXP** if any **SERF**-complete problem for **SNP** is in **SUBEXP** — Impagliazzo, Paturi and Zane 1998
- Some **SERF**-complete languages for **SNP**: k-SAT , k-coloring
- All are equivalent as far as the existence of subexponential time algorithms is concerned.

Prior Work

- Possibility of arbitrarily small exponents for various **NP**-complete problems is one and the same.
- **SNP** $\subseteq$ **SUBEXP** if any **SERF**-complete problem for **SNP** is in **SUBEXP** — Impagliazzo, Paturi and Zane 1998
- Some **SERF**-complete languages for **SNP**: k-SAT , k-coloring
- All are equivalent as far as the existence of subexponential time algorithms is concerned.
- Key tool: complexity parameter preserving reductions via Sparsification Lemma

Exponential Time Hypothesis

- $s_k = \inf\{\delta | \exists 2^{\delta n} \text{ algorithm for } k\text{-CNF SAT}\}$
- $s_\infty = \lim_{k\to\infty} s_k$

Exponential Time Hypothesis

- $s_k = \inf\{\delta | \exists 2^{\delta n}$ algorithm for k-CNF SAT$\}$
- $s_\infty = \lim_{k \to \infty} s_k$
- **ETH** — Exponential Time Hypothesis: $s_3 > 0$

Exponential Time Hypothesis

- $s_k = \inf\{\delta | \exists 2^{\delta n}$ algorithm for k-CNF SAT$\}$
- $s_\infty = \lim_{k\to\infty} s_k$
- **ETH** — Exponential Time Hypothesis: $s_3 > 0$
- **ETH** implies that s_k increases infinitely often — Impagliazzo and Paturi, 1999
- In other words, $\forall k, \exists k' > k, s_{k'} > s_k$.

Exponential Time Hypothesis

- $s_k = \inf\{\delta | \exists 2^{\delta n}$ algorithm for k-CNF SAT$\}$
- $s_\infty = \lim_{k\to\infty} s_k$
- **ETH** — Exponential Time Hypothesis: $s_3 > 0$
- **ETH** implies that s_k increases infinitely often — Impagliazzo and Paturi, 1999
- In other words, $\forall k, \exists k' > k, s_{k'} > s_k$.
- **ETH** implies that $(d, 2)$-CSP takes d^{cn} time where c is an absolute constant. Traxler 2008
- Other similar conditional lower bounds by Marx, Williams, Patrascu

Exponential Time Hypothesis

- $s_k = \inf\{\delta | \exists 2^{\delta n}$ algorithm for k-CNF SAT$\}$
- $s_\infty = \lim_{k\to\infty} s_k$
- **ETH** — Exponential Time Hypothesis: $s_3 > 0$
- **ETH** implies that s_k increases infinitely often — Impagliazzo and Paturi, 1999
- In other words, $\forall k, \exists k' > k, s_{k'} > s_k$.
- **ETH** implies that $(d, 2)$-CSP takes d^{cn} time where c is an absolute constant. Traxler 2008
- Other similar conditional lower bounds by Marx, Williams, Patrascu
- Open Problems: Assuming **ETH** or other suitable assumption, prove
  - a specific lower bound on s_3
  - $s_\infty = 1$
  - Assuming $s_\infty = 1$, can we prove a 2^n lower bound on k-coloring?

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates
- $\mathbf{Pr}[C(y, *) = b]$ — probability C outputs b for the input y

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates
- $\mathbf{Pr}[C(y,*) = b]$ — probability C outputs b for the input y
- **CircuitSat** — the *circuit satisfiability* problem: given an encoding of $D \in \mathcal{C}$, does there exist a $y \in \{0,1\}^{n(D)}$ such that D on variable setting y outputs 1.

Probabilistic Circuits and Circuit Satisfiability

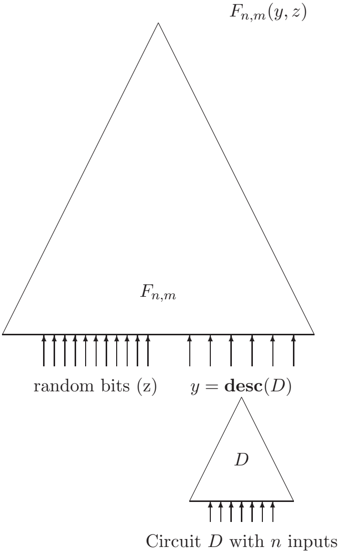
- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates
- $\mathbf{Pr}[C(y, *) = b]$ — probability C outputs b for the input y
- **CircuitSat** — the *circuit satisfiability* problem: given an encoding of $D \in \mathcal{C}$, does there exist a $y \in \{0,1\}^{n(D)}$ such that D on variable setting y outputs 1.
- Family $\mathcal{F}$ of circuits for deciding **CircuitSat** — $\{F_{n,m} | n, m \geq 1\}$, indexed by size of the input circuit and the number of its variables

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates
- $\mathbf{Pr}[C(y,*) = b]$ — probability C outputs b for the input y
- **CircuitSat** — the *circuit satisfiability* problem: given an encoding of $D \in \mathcal{C}$, does there exist a $y \in \{0,1\}^{n(D)}$ such that D on variable setting y outputs 1.
- Family $\mathcal{F}$ of circuits for deciding **CircuitSat** — $\{F_{n,m} | n, m \geq 1\}$, indexed by size of the input circuit and the number of its variables
- A circuit family $\mathcal{F} = \{C_{n,m}\}$ *decides* **CircuitSat** with *success probability* $p(n)$ — for all input circuits D such that $n(D) = n$ and $y = \mathbf{desc}(D)$
  - $\mathbf{Pr}[F_{n,m}(y,*) = 1] \geq p(n)$ if D is satisfiable
  - $\mathbf{Pr}[F_{n,m}(y,*) = 0] = 1$ otherwise

Probabilistic Circuits and Circuit Satisfiability

- $\mathcal{C}$ — the family of non-uniform probabilistic circuits
- For $C \in \mathcal{C}$: n — number of variables; complexity parameter, partitioned as input and random variables, size — counts of gates
- $\mathbf{Pr}[C(y,*) = b]$ — probability C outputs b for the input y
- **CircuitSat** — the *circuit satisfiability* problem: given an encoding of $D \in \mathcal{C}$, does there exist a $y \in \{0,1\}^{n(D)}$ such that D on variable setting y outputs 1.
- Family $\mathcal{F}$ of circuits for deciding **CircuitSat** — $\{F_{n,m} | n, m \geq 1\}$, indexed by size of the input circuit and the number of its variables
- A circuit family $\mathcal{F} = \{C_{n,m}\}$ *decides* **CircuitSat** with *success probability* $p(n)$ — for all input circuits D such that $n(D) = n$ and $y = \mathbf{desc}(D)$
  - $\mathbf{Pr}[F_{n,m}(y,*) = 1] \geq p(n)$ if D is satisfiable
  - $\mathbf{Pr}[F_{n,m}(y,*) = 0] = 1$ otherwise
- Success probability of $\mathcal{F}$: $p(n) \geq \inf_{m,y} \mathbf{Pr}[F^y_{n,m}(z) = 1]$.



Probabilistic Circuit for **CircuitSat**

Complexity of Circuit Satisfiability

- Complexity of $\mathcal{F}$ for deciding **CircuitSat** for circuits with n inputs — $\lg(1/p(n))/n$

Complexity of Circuit Satisfiability

- Complexity of $\mathcal{F}$ for deciding **CircuitSat** for circuits with n inputs — $\lg(1/p(n))/n$
- The complexity of $\mathcal{F}$ for deciding **CircuitSat** — $E_{\textbf{CircuitSat}}(\mathcal{F}) = \limsup \lg(1/p(n))/n$

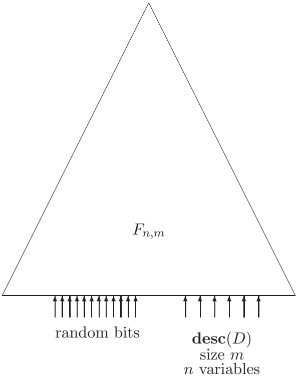
Complexity of Circuit Satisfiability

- Complexity of $\mathcal{F}$ for deciding **CircuitSat** for circuits with n inputs — $\lg(1/p(n))/n$
- The complexity of $\mathcal{F}$ for deciding **CircuitSat** — $E_{\textbf{CircuitSat}}(\mathcal{F}) = \limsup \lg(1/p(n))/n$
- The complexity of deciding **CircuitSat** by $f(n,m)$-bounded probabilistic circuit families —
  $\inf\{\varepsilon | \exists$ a f-bounded $\mathcal{F}$ such that $E_{\textbf{CircuitSat}}(\mathcal{F}) \leq \varepsilon\}$.

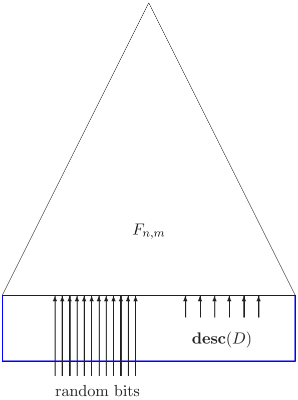
Exponential Amplification Lemma

Lemma

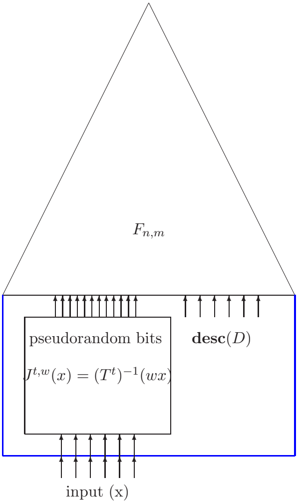
Exponential Amplification Lemma: *Let $\mathcal{F}$ be an f-bounded family for some $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$ such that the success probability is $2^{-\delta n}$ for $0 < \delta < 1$. Then there exists a g-bounded circuit family $\mathcal{G}$ such that $E_{\mathbf{CircuitSat}}(\mathcal{G}) < \delta^2$ where $g(n, m) = O(f(\lceil \delta n \rceil + 5, \tilde{O}(f(n, m))))$.*



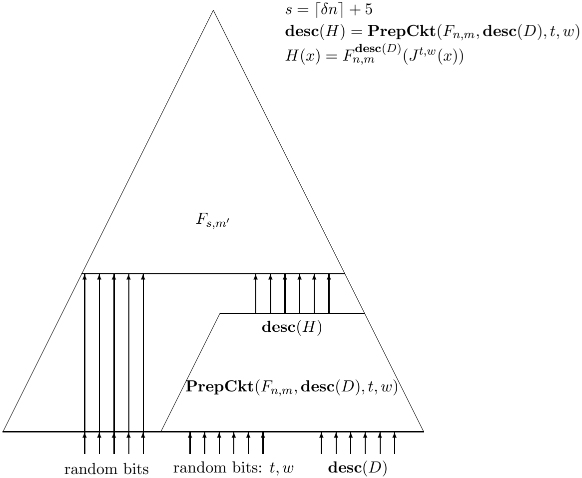
Picture 1: Probabilistic Circuit $F_{n,m}$



Picture 2: Specialization of $F_{n,m}$



Picture 3: $H(x) = F_{n,m}^{\mathbf{desc}(D)}(J^{t,w}(x))$



Picture 2: Circuit $G_{n,m}$

Results: Polynomial Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size* m^k *for some* k *with success probability* $2^{-\delta n}$ *for* $\delta < 1$*, then there exists a* $\mu < 1$ *depending on* k *and* δ *such that* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size* $2^{O(n^{\mu} \lg^{1-\mu} m)}$*.*

Results: Polynomial Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size* m^k *for some* k *with success probability* $2^{-\delta n}$ *for* $\delta < 1$*, then there exists a* $\mu < 1$ *depending on* k *and* δ *such that* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size* $2^{O(n^{\mu} \lg^{1-\mu} m)}$.

- The consequence amounts to $2^{n^{\mu}}$ size deterministic circuits for **CircuitSat** for polynomial size circuits.

Results: Polynomial Size Circuits

Theorem

*If **CircuitSat** can be decided with probabilistic circuits of size m^k for some k with success probability $2^{-\delta n}$ for $\delta < 1$, then there exists a $\mu < 1$ depending on k and δ such that **CircuitSat**(n, m) (and consequently **NP**(n, m)) can be decided by deterministic circuits of size $2^{O(n^{\mu} \lg^{1-\mu} m)}$.*

- The consequence amounts to $2^{n^{\mu}}$ size deterministic circuits for **CircuitSat** for polynomial size circuits.
- If $m = 2^{o(n)}$, **CircuitSat** can be decided by deterministic circuits of size $2^{o(n)}$ — considered implausible — contradicts **ETH**.
- Also implies that $W[P]$ is fixed parameter tractable.

Results: Quasilinear Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size $\tilde{O}(m)$ with success probability $2^{-\delta n}$ for $\delta < 1$, then* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size $O(\mathrm{poly}(m) n^{O(\lg \lg m)})$.*

- The consequence is very close to the statement $\mathbf{NP} \subseteq \mathbf{P}/\mathbf{poly}$.

Results: Subexponential Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size* $2^{o(n)}\tilde{O}(m)$ *with success probability* $2^{-\delta n}$ *for* $\delta < 1$*, then* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size* $2^{o(n)}\mathrm{poly}(m)$*.*

Results: Subexponential Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size $2^{o(n)}\tilde{O}(m)$ with success probability $2^{-\delta n}$ for $\delta < 1$, then* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size $2^{o(n)}\mathrm{poly}(m)$.*

- Apply the Exponential Amplification Lemma a number of times that grows with n.

Results: Subexponential Size Circuits

Theorem

If **CircuitSat** *can be decided with probabilistic circuits of size* $2^{o(n)}\tilde{O}(m)$ *with success probability* $2^{-\delta n}$ *for* $\delta < 1$*, then* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by deterministic circuits of size* $2^{o(n)}\mathrm{poly}(m)$*.*

- Apply the Exponential Amplification Lemma a number of times that grows with n.
- The consequence of the theorem implies that **CircuitSat** can be solved in $2^{o(n)}\mathrm{poly}(m)$ size deterministic circuits for polynomial size circuits (m is polynomial in n), which contradicts **ETH**.

Results: Small Exponential Size Circuits

Theorem

For every $\alpha, \varepsilon > 0$*, either* $E_{\mathbf{CircuitSat}}(\mathbf{explinear}) \geq 1 - \alpha - \varepsilon$ *or* **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by circuits of size* $2^{n/(1+\varepsilon/\alpha)}\mathrm{poly}(m)$*.*

Results: Small Exponential Size Circuits

Theorem

For every $\alpha, \varepsilon > 0$, either $E_{\textbf{CircuitSat}}(\textbf{explinear}) \geq 1 - \alpha - \varepsilon$ or **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by circuits of size $2^{n/(1+\varepsilon/\alpha)}\mathrm{poly}(m)$.*

- If success probability for **CircuitSat** is better than $2^{-(1-\alpha)n+o(n)}$, then **CircuitSat** can be decided by circuits of size $2^{cn}\mathrm{poly}(m)$ with $c = 1/(1 + \frac{\varepsilon}{\alpha}) < 1$.
- Standard correctness probability boosting would give circuits of size $2^{(1-\varepsilon)n}\mathrm{poly}(m)$ size.

Results: Small Exponential Size Circuits

Theorem

For every $\alpha, \varepsilon > 0$, either $E_{\textbf{CircuitSat}}(\textbf{explinear}) \geq 1 - \alpha - \varepsilon$ or **CircuitSat**(n, m) *(and consequently* **NP**(n, m)*) can be decided by circuits of size $2^{n/(1+\varepsilon/\alpha)}\mathrm{poly}(m)$.*

- If success probability for **CircuitSat** is better than $2^{-(1-\alpha)n+o(n)}$, then **CircuitSat** can be decided by circuits of size $2^{cn}\mathrm{poly}(m)$ with $c = 1/(1 + \frac{\varepsilon}{\alpha}) < 1$.
- Standard correctness probability boosting would give circuits of size $2^{(1-\varepsilon)n}\mathrm{poly}(m)$ size.

Open Problems

- Weaken the hypotheses for **CircuitSat** resource trade-off bounds to $\mathbf{NP} \not\subseteq \mathbf{P}/\mathbf{poly}$.

Open Problems

- Weaken the hypotheses for **CircuitSat** resource trade-off bounds to $\mathbf{NP} \not\subseteq \mathbf{P}/\mathbf{poly}$.
- Does graph coloring or the Hamiltonian path problem have probabilistic polynomial time algorithms with success probability c^{-n}?

Open Problems

- Weaken the hypotheses for **CircuitSat** resource trade-off bounds to $\mathbf{NP} \not\subseteq \mathbf{P}/\mathbf{poly}$.
- Does graph coloring or the Hamiltonian path problem have probabilistic polynomial time algorithms with success probability c^{-n}?
- Prove resource trade-off bounds for linear-size **CircuitSat** in polynomial size models under suitable complexity assumptions.

Thank You