Time-Space Tradeoffs in Resolution: Superpolynomial Lower Bounds for Superlinear Space

Chris Beck
Princeton University

Joint work with
Paul Beame & Russell Impagliazzo
SAT

• The satisfiability problem is a central problem in computer science, in theory and in practice.

• Terminology:
  – A *Clause* is a boolean formula which is an OR of variables and negations of variables.
  – A *CNF formula* is an AND of clauses.

• Object of study for this talk:
CNF-SAT: Given a CNF formula, is it satisfiable?
Resolution Proof System

• Lines are clauses, one simple proof step

\[ C \lor x \quad D \lor \neg x \]

\[ \frac{C \lor x \quad D \lor \neg x}{C \lor D} \]

• Proof is a sequence of clauses each of which is
  – an original clause or
  – follows from previous ones via resolution step

• a CNF is UNSAT iff can derive empty clause \( \bot \)
General resolution: Arbitrary DAG
Tree-like resolution: DAG is a tree
SAT Solvers

• Well-known connection between Resolution and SAT solvers based on Backtracking

• These algorithms are very powerful
  – sometimes can quickly handle CNF’s with millions of variables.

• On UNSAT formulas, computation history yields a Resolution proof.
  – Tree-like Resolution $\approx$ DPLL algorithm
  – General Resolution $\succeq$ DPLL + “Clause Learning”
    • Best current SAT solvers use this approach
SAT Solvers

• DPLL algorithm: backtracking search for satisfying assignment
  – Given a formula which is not constant, guess a value for one of its variables $x$, and recurse on the simplified formula.
  – If we don’t find an assignment, set $x$ the other way and recurse again.
  – If one of the recursive calls yields a satisfying assignment, adjoin the good value of $x$ and return, otherwise report fail.
SAT Solvers

• DPLL requires very little memory
• Clause learning adds a new clause to the input CNF every time the search backtracks
  – Uses lots of memory to try to beat DPLL.
  – In practice, must use heuristics to guess which clauses are “important” and store only those. Hard to do well! Memory becomes a bottleneck.

• Question: Is this inherent? Or can the right heuristics avoid the memory bottleneck?
Proof Complexity & Sat Solvers

• Proof Size $\leq$ Time for Ideal SAT Solver
• Proof Space $\leq$ Memory for Ideal SAT Solver

• Many explicit hard UNSAT examples known with exponential lower bounds for Resolution Proof Size.

• **Question:** Is this also true for Proof Space?
Space in Resolution

\[ \text{Clause space} := \max_t (\# \text{ active clauses}) \]

[Esteban, Torán ‘99]

\[ C_1 \quad C_2 \quad xyv \quad \overline{xvz} \quad \ldots \quad yvz \quad \perp \]

Time step \( t \)  

Must be in memory
Lower Bounds on Space?

- Considering Space alone, tight lower and upper bounds of $\theta(n)$, for explicit tautologies of size $n$.
- Lower Bounds: [ET’99, ABRW’00, T’01, AD’03]
- Upper Bound: All UNSAT formulas on $n$ variables have tree-like refutation of space $\leq n$. [Esteban, Torán ‘99]
  - For a tree-like proof, Space is at most the height of the tree which is the stack height of DPLL search.
- **But**, these tree-like proofs are typically too large to be practical, so we should instead ask if both small space and time are simultaneously feasible.
Size-Space Tradeoffs for Resolution

• [Ben-Sasson ‘01] Pebbling formulas with linear size refutations, but for which all proofs have $\text{Space} \cdot \log \text{Size} = \Omega(n/\log n)$.

• [Ben-Sasson, Nordström ‘10] Pebbling formulas which can be refuted in $\text{Size} \in O(n)$, $\text{Space} \in O(n)$, but $\text{Space} \in O(n/\log n) \Rightarrow \text{Size} \in \exp(n^{\Omega(1)})$.

But, these are all for $\text{Space} < n$, and SAT solvers generally can afford to store the input formula in memory. Can we break the linear space barrier?
Size-Space Tradeoffs

• **Informal Question:** Can we formally show that memory rather than time can be a real bottleneck for resolution proofs and SAT solvers?

• **Formal Question (Ben-Sasson):**
  “Does there exist a $c$ such that any CNF with a refutation of size $T^c$ has a refutation of size $T^c$ in space $O(n)$?"

• **Our results:** Families of formulas of size $n$ having refutations in Time, Space $n^{\log n}$, but all resolution refutations have $T > (n^{0.58 \log n} / S)^{\log \log n / \log \log \log n}$.
Tseitin Tautologies

Given an undirected graph \( G = (V, E) \) and \( \chi : V \to \mathbb{F}_2 \), define a **CSP**:

**Boolean variables:** \( \forall e \in E \quad x_e \)

**Parity constraints:** \( \forall v \in V \quad \oplus_{e \sim v} x_e = \chi(v) \) (linear equations)

When \( \chi \) has **odd** total parity, CSP is UNSAT.
Tseitin Tautologies

• When $\chi$ odd, $G$ connected, corresponding CNF is called a Tseitin tautology. [Tseitin ‘68]

• Only total parity of $\chi$ matters

• Hard when $G$ is a constant degree expander:
  [Urquhart 87]: Resolution size $= 2^{\Omega(E)}$
  [Torán 99]: Resolution space $= \Omega(E)$

• This work: Tradeoffs on $n \times l$ grid, $l \gg n$, and similar graphs, using isoperimetry.
Graph for our result

• Take as our graph $G := K_n \otimes P_l$ and form the Tseitin tautology, $\tau$.

• For now we’ll take $l = 2n^4$, but it’s only important that it is $\gg n$, and not too large.

• Formula size $N = 2^{2n} \cdot nl = 2^{2n+o(1)}$. 
A Refutation

• A Tseitin formula can be viewed as a system of inconsistent $\mathbb{F}_2$-linear equations. If we add them all together, get $1=0$, contradiction.

• If we order the vertices (equations) intelligently (column-wise), then when we add them one at a time, never have more than $n^2$ variables at once

• Resolution can simulate this, with some blowup:
  – yields Size, Space $\approx 2^{n^2}$ (for a $2^{2n}$ sized formula)
A Different Refutation

• Can also do a “divide & conquer” DPLL refutation. Idea is to repeatedly bisect the graph and branch on the variables in the cut.

• Once we split the CNF into two pieces, can discard one of them based on parity of assignment to cut.

• After $n^2 \log |V|$ queries, we’ve found a violated clause – idea yields tree-like proof with Space $\approx n^2 \log n$, Size $n^{n^2}$. (Savitch-like savings in Space)
Complexity Measure

• To measure progress as it occurs in the proof, want to define a *complexity measure*, which assigns a value to each clause in the proof.

• Wish list:
  – Input clauses have small value
  – Final clause has large value
  – Doesn’t grow quickly in any one resolution step
Complexity Measure for Tseitin

• Say an assignment to an (unsat) CNF is a critical if it violates only one constraint.
• For \( \pi \) critical to Tseitin formula \( \tau \), “\( \pi \)’s vertex”:
  \[
  \nu_\tau(\pi) := \text{vertex of that constraint}
  \]
• For any Clause \( C \), define the “critical vertex set”:
  \[
  \text{crit}_\tau(C) := \{ \nu_\tau(\pi) : \pi \text{ critical to } \tau, \text{ falsifies } C \}
  \]
Critical Set Examples

Blue  = 0
Red   = 1

In these examples, Graph is a Grid.
For a clause that doesn’t cut the graph, **Critical set** is ... still everything.
Critical Set Examples

Blue = 0
Red = 1

For this clause, several components.
Parity mismatch in only one, **Upper left is critical**.
Critical Set Examples

Blue  = 0
Red   = 1
Complexity Measure

• Define $\mu(C) := |crit_{\tau}(C)|$. Then $\mu$ is a sub-additive complexity measure:
  ▪ $\mu($clause of $\tau) = 1$,
  ▪ $\mu(\bot) = \# vertices$,
  ▪ $\mu(C) \leq \mu(C_1) + \mu(C_2)$, when $C_1, C_2 \vdash C$.

• Useful property: Every edge on the boundary of $crit_{\tau}(C)$ is assigned by $C$. 
Complexity vs. Time

• Consider the time ordering of any proof, and plot complexity of clauses in memory v. time

• Only constraints – start low, end high, and because of sub-additivity, cannot skip over any \([t, 2t]\) window of \(\mu\)-values on the way up.
A Classic Application

• A classic application in the spirit of [BP’96], to show Proof Size lower bounds w/ these ideas

• A *restriction* $\rho$ to a formula $\varphi$ is a partial assignment of *truth values* to *variables*, resulting in some *simplification*. We denote the restricted formula by $\varphi|_\rho$.

• If $\Pi$ is a proof of $\varphi$, $\Pi|_\rho$ is a proof of $\varphi|_\rho$. 
A Classic Application

• Consider a random restriction $\rho$ to formula $\varphi$. Prove:
  – Any wide clause $C$ is likely to be killed  
    ($C|_\rho$ becomes trivial and disappears)
  – Any refutation of $\varphi|_\rho$ contains a wide clause

• Then, if there was a short proof of $\varphi$, by a union bound over its clauses we would get a short proof of $\varphi|_\rho$, a contradiction.
• What happens to our formula $\tau$ when we apply a random restriction?

• Say, each edge is set to 0,1,* with probability $1/3$, independently.
Restricting a single variable

• When a single variable is assigned, its corresponding edge $e$ is removed

• If edge variable is set to 0, restricted tautology is exactly the Tseitin tautology on $G$ with $e$ removed:

$$\tau(G, \chi)|_{\{e=0\}} = \tau(G\setminus e, \chi)$$
Restricting a single variable

- When a single variable is assigned, its corresponding edge $e$ is removed.

- If edge variable is set to 1, the same except both endpoints of $e$ have their parity constraints flipped; total parity is the same:

$$\tau(G, \chi)|_{\{e=1\}} = \tau(G \setminus e, \chi') \cong \tau(G \setminus e, \chi)$$
Applying a random restriction

- Say, each edge is set to 0, 1, * with probability 1/3, independently.

- WHP: Yields a Tseitin tautology on a (highly) connected subgraph of original graph

Each $K_{n,n}$ above becomes a $G(n, n, \frac{1}{3})$. 
Applying a random restriction

• Say, each edge is set to 0, 1, * with probability 1/3, independently.

• WHP: Yields a Tseitin tautology on a (highly) connected subgraph of original graph

Each $K_{n,n}$ above becomes a $G(n, n, \frac{1}{3})$. 
Applying a random restriction

• Before, every approximately balanced cut had $n^2$ edges crossing

• In the subgraph, balanced cuts have at least $\approx \frac{1}{3} n^2$ edges crossing, so for a clause $C|_\rho$:

$$\mu(C|_\rho) \in \left[ \frac{|V|}{3}, \frac{2|V|}{3} \right] \Rightarrow |C|_\rho = \Omega(n^2)$$
A Classic Application

• Suppose original formula $\tau$ has a short proof $\Pi$. By a union bound, except with probability

$$|\Pi| \left(\frac{2}{3}\right)^{\frac{1}{3}n^2}$$

$\Pi|_\rho$ only contains clauses of width $< \frac{1}{3} n^2$.

• But whp, we saw any proof of $\tau|_\rho$ must contain a medium complexity clause of width $\geq \frac{1}{3} n^2$.

• So we have a contradiction unless

$$|\Pi| \geq 2^{\Omega(n^2)}$$
Observation

• To prove a size lower bound, we only really needed a width lower bound. We showed there was at least one clause of complexity between $|V|/3$ and $2|V|/3$.

• To prove a time space tradeoff, we need to track progress occurring in the proof more carefully than this.
Medium complexity clauses

• Fix $t_0 = n^4$ and say that clause $C$ has complexity level $i$ iff
  $$2^i t_0 \leq \mu(C) < 2^{i+1} t_0$$

• Medium complexity clause: complexity level between 0 and $\log n - 1$
  – By choice of parameters $n^4 \leq \mu(C) < |V|/2$

• Subadditivity $\Rightarrow$ can’t skip any level going up.
Main Lemma

- For some very small $p^* = 2^{-\Omega(n^2)}$, for any set $S$ of $M$ clauses,

$$\Pr[S|\rho \text{ contains } \geq k \text{ clauses of distinct medium complexity levels}] \leq (Mp^*)^k$$

(defer proof)
Complexity vs. Time

• Consider the time ordering of any proof, and divide time into $m$ equal epochs (fix $m$ later)
Two Possibilities

• Consider the time ordering of any proof, and divide time into $m$ equal epochs (fix $m$ later)

• Either, a clause of medium complexity appears in memory for at least one of the breakpoints between epochs,
Two Possibilities

- Consider the time ordering of any proof, and divide time into $m$ equal epochs (fix $m$ later)
- Or, all breakpoints only have Hi and Low. Must have an epoch which starts Low ends Hi, and so has clauses of all $\approx \log n$ Medium levels.
Analysis

- Consider the restricted proof. With probability 1, one of the two scenarios applies.
- For first scenario, $M = mS$ clauses,
  $Pr[1st \ scenario] \leq mSp^*$
- For second scenario, $m$ epochs with $M = T/m$ clauses each. Union bound and main lemma:
  $Pr[2nd \ scenario] \leq m \left( \frac{T}{m} p^* \right)^k$, where $k = \log n$
Analysis

• Optimizing $m$ yields something of the form $TS = \Omega(p^{*2})$

• Can get a nontrivial tradeoff this way, but need to do a lot of work optimizing $p^*$.

• One idea to improve this: try to make the events we bound over more balanced, using main lemma for both scenarios
Two Possibilities

• Consider the time ordering of any proof, and divide time into \( m \) equal epochs (fix \( m \) later)

Hi     Med     Low

• Either, \( \sqrt{\log n} \) distinct medium complexities appear in memory among all the clauses at breakpoints between epochs,
Two Possibilities

- Consider the time ordering of any proof, and divide time into \( m \) equal epochs (fix \( m \) later)

- Or, have an epoch with at least \( \sqrt{\log n} \) different complexities
Restating this as a decomposition

• Consider the *tableau* of any small space proof

• At least one set has $\geq \sqrt{\log n}$ complexities

Size is $mS$

$T/m$

$S$

$T$
An even better tradeoff

- Don’t just divide into epochs once
- Recursively divide proof into epochs and sub-epochs where each sub-epoch contains $m$ sub-epochs of the next smaller size

- Prove that, if an epoch does a lot of work, either
  - Breakpoints contain many complexities
  - A sub-epoch does a lot of work
An even better tradeoff

- If an epoch contains a clause at the end of level $b$, but every clause at start is level $\leq b - a$, (so the epoch makes $a$ progress),
- and the breakpoints of its $m$ children epochs contain together $\leq k$ complexity levels,
- then some child epoch makes $a/k$ progress.
Internal Node Size = mS
Leaf Node Size = T/m^(h-1)

Previous Slide shows:
Some set has \( \geq k \) complexities, where \( k^h = \log n \)
An even better tradeoff

• Choose $h = k$, have $k^k = \log n$, so $k = \omega(1)$

• Choose $m$ so all sets are the same size: $T/m^{k-1} \approx mS$, so all events are rare together.

• Have $m^k$ sets in total

Finally, a union bound yields $T \geq \left(\frac{\exp(n^2)}{S}\right)^k$
Final Tradeoff

- Tseitin formulas on $K_n \otimes P_l$ for $l = 2n^4$
  - are of size $2^{2n} \cdot \text{poly}(n)$
  - have resolution refutations in Size, Space $\approx 2^{n^2}$
  - have $n^2 \log n$ space refutations of Size $2^{n^2 \log n}$
  - and any resolution refutation for Size $T$ and Space $S$ requires $T > (2^{0.58n^2 / S})^{\omega(1)}$

If space is at most $2^{n^2 / 2}$ then size blows up by a super-polynomial amount
Medium Sets have large Boundary

Claim:
For any $S \subseteq V$, $n^4 \leq |S| \leq \frac{|V|}{2}$, $|\delta(S)| \geq n^2$. 
Medium Sets have large Boundary

Claim:
For any $S \subseteq V$, $n^4 \leq |S| \leq \frac{|V|}{2}$, $|\delta(S)| \geq n^2$. 
Extended Isoperimetric Inequality

Lemma:
For $S_1, S_2 \subseteq V$, both medium, $2|S_1| < |S_2|$, have $|\delta(S_1) \cup \delta(S_2)| \geq 2n^2$.

Two medium sets of very different sizes $\Rightarrow$ boundary guarantee doubles.

Also: $k$ medium sets of super-increasing sizes $\Rightarrow$ boundary guarantee goes up by factor of $k$
Extended Isoperimetric Inequality

If the sets aren’t essentially blocks, we’re done.

If they are blocks, reduce to the line:
Intervals on the line

• Let \([a_1, b_1], \ldots, [a_k, b_k]\) be intervals on the line, such that \(b_k - a_k \geq 2(b_{k-1} - a_{k-1})\)

• Let \(\mu(k)\) be the minimum number of distinct endpoints of intervals in such a configuration.

• Then, a simple inductive proof shows \(\mu(k) \geq k + 1\)
Ext. Isoperimetry in Random Graph

• Lemma: WHP, $G(n, n, 1/3)$ contains a $r$-regular subgraph, for $r = \left(\frac{1}{3} - o(1)\right)n^2$.

• Corollary: Assuming this event holds, can find $r$ edge disjoint paths between any two columns of the random subgraph.

So the same proofs as before generalize.
“Regular”: On every root to leaf path, no variable resolved more than once.
Tradeoffs for Regular Resolution

• Theorem: For any $k$, 4-CNF formulas (Tseitin formulas on long and skinny grid graphs) of size $n$ with
  – Regular resolution refutations in size $n^{k+1}$, Space $n^k$.
  – But with Space only $n^{k-\varepsilon}$, for any $\varepsilon > 0$, any regular resolution refutation requires size at least $n^{\varepsilon \log \log n / \log \log \log n}$. 
Regular Resolution

• Can define partial information more precisely

• Complexity is monotonic wrt proof DAG edges. This part uses regularity assumption, simplifies arguments with complexity plot.

• **Random Adversary** selects random assignments based on proof
  – No random restrictions, conceptually clean and don’t lose constant factors here and there.
Open Questions

• More than quasi-polynomial separations?
  – For Tseitin formulas upper bound for small space is only a $\log n$ power of the unrestricted size
  – Candidate formulas?
  – Are these even possible?

• Other proof systems?

• Other cases for separating search paradigms: “dynamic programming” vs. “binary search”?
Thanks!