
Proceedings Paper:
Scheder, D. and Talebanfard, N. (2020) Super strong ETH is true for PPSZ with small resolution width. In: Saraf, S., (ed.) 35th Computational Complexity Conference (CCC 2020). 35th Computational Complexity Conference (CCC 2020), 28-31 Jul 2020, Saarbrücken, Germany (Virtual). Leibniz International Proceedings in Informatics (LIPIcs), 169 . Schloss Dagstuhl - Leibniz-Zentrum für Informatik , 3:1-3:12. ISBN 9783959771566

https://doi.org/10.4230/LIPIcs.CCC.2020.3

Super Strong ETH Is True for PPSZ with Small Resolution Width

Dominik Scheder
Shanghai Jiaotong University, China
dominik.scheder@gmail.com

Navid Talebanfard
Institute of Mathematics, The Czech Academy of Sciences, Prague, Czech Republic
talebanfard@math.cas.cz

Abstract

We construct k-CNFs with m variables on which the strong version of PPSZ k-SAT algorithm, which uses resolution of width bounded by $O(\sqrt{\log \log m})$, has success probability at most $2^{-(1-(1+\epsilon)2/k)m}$ for every $\epsilon > 0$. Previously such a bound was known only for the weak PPSZ algorithm which exhaustively searches through small subformulas of the CNF to see if any of them forces the value of a given variable, and for strong PPSZ the best known previous upper bound was $2^{-(1-O(\log(k)/k))m}$ (Pudlák et al., ICALP 2017).

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases k-SAT, PPSZ, Resolution

Digital Object Identifier 10.4230/LIPIcs.CCC.2020.3

Funding *Dominik Scheder*: Supported by the National Natural Science Foundation of China under grant 61502300 and 11671258.
Navid Talebanfard: Supported by GAČR grant 19-27871X.

1 Introduction

The PPSZ algorithm for k-SAT by Paturi, Pudlák, Saks, and Zane [7] is simple to state but famously difficult to analyze. Given a k-CNF formula Φ as input, it first chooses a random ordering π of its variables $x_1, \ldots, x_m$. It goes through them one by one, in the order given by π. For each variable x, it tries to derive the correct value using a certain *proof heuristic* P. P takes as input a k-CNF formula Φ and a variable x and returns a value in $\{0, 1, ?\}$. P must be *sound*, meaning if $P(\Phi, x) = b \in \{0, 1\}$ then $\Phi \models (x = b)$, i.e., every satisfying assignment of Φ sets x to b; however, we allow P to be *incomplete*, i.e., it may answer "?", meaning "I don't know". If $P(\Phi, x) = b \in \{0, 1\}$, then PPSZ sets x to b; otherwise it sets x to some $b \in \{0, 1\}$ chosen uniformly at random. In either case, it simplifies Φ to $\Phi|_{x \mapsto b}$. Once all variables have been processed, the resulting formula either contains the empty clause $\square$, and we declare this run of PPSZ a failure; or it does not, in which case PPSZ has found a satisfying assignment.

If PPSZ has success probability p then we can repeat it $1/p$ times, obtaining a constant success probability. As long as P runs in subexponential time, the overall running time of this Monte Carlo algorithm is dominated by $1/p$ (which will, most likely, be exponential in n). Which proof heuristics P should one consider? There are currently just two on the market. The first one is P_w, which checks whether $(x = b)$ is implied by a set of up to w clauses of Φ. The second one is R_w, which tries to derive the clause $(x = b)$ by resolution, bounded by width w. Obviously they both can be implemented in time $O^*\left(\binom{|\Phi|}{w}\right) \leq O^*\left(\binom{m^k}{w}\right)$, which

is subexponential as long as $w \in o\left(\frac{m}{\log m}\right)$. It is easy to see that $R_{w \cdot k}$ is at least as strong as P_w. We also speak of *weak PPSZ* when it uses P_w and *strong PPSZ* when it uses R_w (ignoring the concrete values of w).

Proving positive results, i.e., lower bounds on the success probability, seems remarkably insensitive to our choice of P. In fact, all lower bounds we currently know work for P_w, for any $w \in \omega(1)$:

▶ **Theorem 1** (Paturi, Pudlák, Saks, and Zane [7] and Hertli [6]). *On k-CNF formulas with m variables, the success probability of PPSZ using the heuristic P_w is at least $2^{-m(1-s_k)+o(m)}$, where $\lim_{k\to\infty} k s_k = \frac{\pi^2}{6}$, provided that $w = w(m) \in \omega(1)$.*

Originally, Paturi, Pudlák, Saks, and Zane stated their algorithm as using R_w, i.e., width-bounded resolution; however, it is easy to see that their analysis works for the weaker heuristic P_w as well, see for example [10] for a formal proof. We do not know any better bound for PPSZ using R_w, for any $w \in o(m)$.

The parameter s_k in the theorem is called the *savings* of the algorithm. Ignoring constant factors, the theorem shows that the savings of PPSZ are at least $\Omega(1/k)$. Other algorithms, arguably much simpler, such as PPZ [8] and Schöning's Random Walk [11] have smaller savings than PPSZ, but also of order $\Omega(1/k)$. In general, let σ_k be the supremum of all σ such that there is a randomized algorithm for k-SAT running in time $O\left(2^{m(1-\sigma)}\right)$. There is a whole hierarchy of conjectures about how large the savings for k-SAT can be. Here is a list, sorted from weak to strong.

1. P $\neq$ NP: k-SAT has no polynomial time algorithm.
2. ETH (exponential time hypothesis): $\sigma_3 < 1$.
3. SETH (strong exponential time hypothesis): $\lim_{k\to\infty} \sigma_k = 0$, i.e., as k grows, the advantage over brute force shrinks to nil.
4. SSETH (super strong exponential time hypothesis): $\sigma_k \in O(1/k)$.

We already know (as shown by PPZ, Schöning's and PPSZ algorithms) that $\sigma_k \in \Omega(1/k)$, so Point 4 actually conjectures that $\sigma_k \in \Theta(1/k)$. Of course proving an unconditional upper bound on σ_k is far out of reach for now. However one could try to prove such upper bounds on the savings of specific algorithms. This would then shed light on the difficulty of improving k-SAT algorithms. In this paper we prove close to tight upper bounds on the savings of the strong PPSZ algorithm showing that its running time is consistent with SSETH, that is the worst case running time of PPSZ is as predicted by SSETH. This is in contrast to a recent result of Vyas and Williams [12] who showed that SSETH is false for random k-SAT.

1.1 Previous Results: Hard Instances

The first hard instances for PPSZ were given by the authors together with Chen and Tang [3]. That work constructed k-CNFs based on a random distribution of linear systems and showed that PPSZ using R_w, that is resolution of bounded width, succeeds with probability at most $2^{-m(1-O(\log^2(k)/k))}$ on these formulas, as long as $w \leq \frac{\ln(k)n}{k}$ (Theorem 1.2 in [3]). Together with Pudlák [9] we then improved this lower bound to $2^{-m(1-O(\log(k)/k))}$, which holds as long as $w \leq n/k$ (Theorem 6 in [9]). This improvement came mainly from clarifying and sharpening a union bound in [3]. However based on a completely different construction, it gave an upper bound of $2^{-m(1-2(1+\epsilon)/k)}$ for the "weak" heuristic P_w, for some $w = n^{\Theta(\epsilon)}$ (Theorem 5 in [9]). This construction is based on Tseitin formulas defined on large girth graphs. For R_w, it was left open whether one can obtain the same bound.

2 Our Results

▶ **Theorem 2** (SSETH Holds for PPSZ). *For every $k \in \mathbb{N}$, there is a polynomial p and a sequence $(F_m)_{m\in\mathbb{N}}$ of satisfiable k-CNF formulas F_m on m variables, such that for every $\epsilon > 0$ and $w \leq \sqrt{\epsilon \cdot \frac{\log\log m}{2\log k}} - 3$, it holds that* $\Pr[\text{ppsz}(F_m, R_w) \text{ succeeds}] \leq p(m)2^{-m(1-2(1+\epsilon)/k)}$.

Thus, the super strong exponential time hypothesis is true for Strong PPSZ, provided that we do not make it too strong, i.e., keep w fairly small. Note that this gives an upper bound on the savings of PPSZ by $2/k$, which is quite close to the currently best lower bound of $(\pi^2/6 + o(1))/k$ [7].

Our result is incomparable to the previous ones. We feel that the "super strong ETH bounds" of $2(1+\epsilon)/k$ in the exponent make this result much stronger than its predecessors. However, the doubly-logarithmic upper bound on w is, of course, much more restrictive than the $w \leq m/k$ bound of Theorem 6 in [9]. Might it be that super strong ETH fails for $w = m/k$? Maybe even for $w = \log(m)$? If we could rule out this possibility, we would have done so in this paper. However, remember that the lower bound on the success probability (Paturi, Pudlák, Saks, and Zane [7]) holds for $P_{\omega(1)}$, which is arguably the weakest possible non-trivial proof heuristic. At the moment, there are no better lower bounds for $R_{o(m)}$, which is much stronger than $P_{\omega(1)}$. Thus, we feel that the parameter w is not as relevant as the savings.

▶ **Conjecture 3.** *Super Strong ETH holds for PPSZ using R_w, as long as $w = o(m)$.*

To be honest, the only supporting evidence we have for this conjecture is the lack of progress in analyzing the success probability of PPSZ. If this conjecture is true, the hard instances proving it might use a very different construction from those in Theorem 2. Thus, we further conjecture:

▶ **Conjecture 4.** *Theorem 2 holds for some $w = \Theta(\log m)$, with the same formulas F_m.*

We have a little bit more evidence supporting the second conjecture: our constructions are based on Tseitin formulas, and our bound on w is related to the girth of a graph H; the graph H has $\Theta(\log m)$ vertices and girth $\Theta(\log\log m)$. However, the resolution width of Tseitin formulas is usually governed by the expansion properties of the underlying graph, not its girth, and thus we hope that some proof also works for $w = \Theta(\log m)$.

Recently, Hansen, Kaplan, Zamir, and Zwick [5] published an improved version of PPSZ, called biased-PPSZ. Roughly stated, the idea of their improvement is that, looking at a formula F with a unique satisfying assignment α, we can identify a set a set $X \subseteq V$ of variables on which α is *biased*, i.e., the number of $x \in X$ set to 1 by α deviates from $|X|/2$ significantly. Thus, setting those variables to 1 with some probability $p \neq 1/2$ gives a higher success probability. We have not checked whether the bounds of Theorem 2 also hold for biased-PPSZ.

2.1 Notation

Given a set of variables X, a *partial assignment* is a function $\alpha : X \to \{0, 1, *\}$, that is an assignment of 0-1 values to some of the variables with $*$ intended to mean unset by α. We denote the set of variables to which α gives a value by $\text{var}(\alpha) := \{x \in X : \alpha(x) \in \{0, 1\}\}$. For two partial assignments α and β we write $\alpha \subseteq \beta$ to mean that for every $x \in \text{var}(\alpha)$, it holds that $\beta(x) = \alpha(x)$. Naturally, $\alpha \subset \beta$ means that $\alpha \subseteq \beta$ and $|\text{var}(\alpha)| < |\text{var}(\beta)|$. Given a

variable x and $b \in \{0,1\}$, $x \mapsto b$ is the partial assignment which sets x to b. The assignment which sets every variable to 0 is denoted by $\mathbf{0}$. For $Y \subseteq X$, we write $Y \mapsto \mathbf{0}$ to denote the partial assignment which sets all variables in Y to 0. If $\mathrm{var}(\alpha) \cap \mathrm{var}(\beta) = \emptyset$, we define $\alpha \cup \beta$ to be the partial assignment which sets all $x \in \mathrm{var}(\alpha)$ to $\alpha(x)$, all $x \in \mathrm{var}(\beta)$ to $\beta(x)$, and all other variables to $*$. Finally the restriction of a formula Φ by α is denoted by $\Phi|_\alpha$.

2.2 The Formula

Let $G = (V, E)$ be a graph. For every $e \in E(G)$ we introduce a variable x_e. Given a *charge* $c : V \to \{0,1\}$, the *Tseitin formula on G with charge c* is the Boolean formula

$$\mathrm{Tseitin}(G, c) := \bigwedge_{u \in V(G)} \left(\sum_{e \in E(G) : u \in e} x_e \equiv c(u) \mod 2 \right) \tag{1}$$

If G has maximum degree k then this can be expressed as a k-CNF formula on $m = |E(G)|$ variables and $|V(G)|2^{k-1}$ clauses. Usually in proof complexity, the charge c is chosen so that $\mathrm{Tseitin}(G, c)$ is unsatisfiable. In this paper, all charges will be 0, and $\mathrm{Tseitin}(G, \mathbf{0})$ is obviously satisfiable: set all variables to 0. We will hence drop c from the notation and simply write $\mathrm{Tseitin}(G)$ to denote this formula. The constraint $\sum_{e \in E(G) : u \in e} x_e \equiv 0 \mod 2$ is called the *Tseitin constraint* of vertex u. Given a set $\mathcal{B}$ of pairs of edges in G consider the following formula

$$\mathrm{Tseitin}(G) \wedge \bigwedge_{\{e,f\} \in \mathcal{B}} (\bar{x}_e \vee \bar{x}_f).$$

The constraint $(\bar{x}_e \vee \bar{x}_f)$ is called a *bridge constraint*. It is easy to see that $\mathbf{0}$ is the unique satisfying assignment of this formula if and only if every cycle in G contains a bridge in $\mathcal{B}$. We will consider a particular instantiation of bridges given by graph homomorphisms. A *graph homomorphism* from a graph G to a graph H is a function $\varphi : V(G) \to V(H)$ such that $\{\varphi(u), \varphi(v)\} \in E(H)$ whenever $\{u, v\} \in E(G)$. Thus, φ also induces a function from $E(G)$ to $E(H)$; $\varphi(\{u, v\}) := \{\varphi(u), \varphi(v)\}$. Given G, H, and a homomorphism φ from G to H, we define a Tseitin formula with bridges on the variable set $\{x_e \mid e \in E(G)\}$:

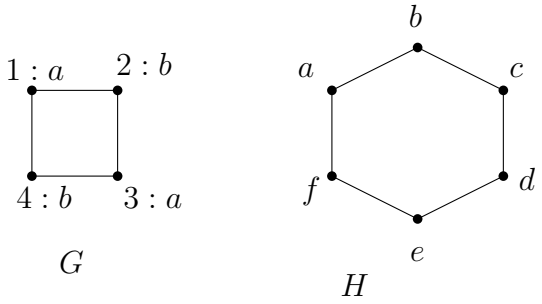
$$\mathrm{TseitinBridge}(G, H, \varphi) := \mathrm{Tseitin}(G) \wedge \bigwedge_{\substack{e,f \in E(G) \\ e \neq f, \varphi(e) = \varphi(f)}} (\bar{x}_e \vee \bar{x}_f)\ . \tag{2}$$

For brevity, we write $V = V(G)$ and $E = E(G)$.

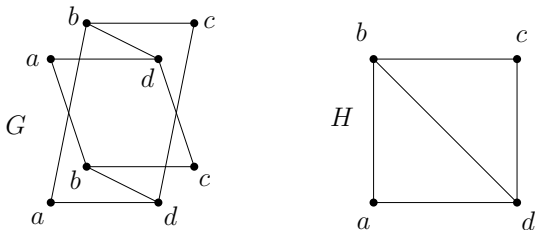
▶ **Observation 5.** *If* $\mathrm{girth}(G) > |E(H)|$ *then* $\mathrm{TseitinBridge}(G, H, \varphi)$ *is uniquely satisfiable by* $\mathbf{0}$.

Proof. Let $\alpha \neq \mathbf{0}$ be a total assignment. Let $F := \{e \in E(G) \mid \alpha(x_e) = 1\}$. If some vertex u has degree 1 in (V, F), then α violates its Tseitin constraint. Otherwise, (V, F) has a cycle, which has length at least $\mathrm{girth}(G)$. By the pigeonhole principle, this cycle contains two edges e, f such that $\varphi(e) = \varphi(f)$, and thus α violates their bridge constraint. ◀

Locally Injective Homomorphisms. A homomorphism φ is called *locally injective* if for every $u \in V(G)$ and any two of its neighbors v_1 and v_2, it holds that $\varphi(v_1) \neq \varphi(v_2)$. Note that $\varphi : G \to H$ being locally injective immediately implies that $\deg_G(u) \leq \deg_H(\varphi(u))$. We call φ *locally bijective* if, additionally, $\deg_G(u) = \deg_H(\varphi(u))$ for all vertices u of G. Note that a locally bijective homomorphism bijectively maps the neighborhood of u to the neighborhood of $\varphi(u)$. The graph G is called a *covering graph* of H or a *lift* of H.



Example of a homomorphism that is not locally injective. The two neighbors of 1 are both mapped to b.



Example of a locally bijective homomorphism. The letters next to the vertices of G are not their names but rather their images under φ.

▶ **Theorem 6.** *Let G be a graph on n vertices and m edges. Suppose there is a locally injective graph homomorphism $\varphi : G \to H$ for some graph H with $|E(H)| < \mathrm{girth}(G)$. Then for all $\epsilon > 0$ and $w := \sqrt{\frac{\epsilon \cdot \mathrm{girth}(H)}{2}} - 3$, the success probability of PPSZ with heuristic R_w on $\Phi := \mathrm{TseitinBridge}(G, H, \varphi)$ is at most*

$$\Pr[\mathrm{ppsz}(\Phi, R_w)] \leq 2^{-m+(1+\epsilon)n} .$$

Proof of Theorem 2 using Theorem 6. We first show how to construct F_m for infinitely many m. Let n_0 be some given, sufficiently large even integer. A well-known fact, first proven by Erdős and Sachs [4], is that there is a k-regular graph G_0 on n_0 vertices having girth at least $g_0 := \frac{\log n_0}{\log(k-1)}$. Set $n_1 := \left\lfloor \frac{2(g_0-1)}{k} \right\rfloor$ or $n_1 := \left\lfloor \frac{2(g_0-1)}{k} \right\rfloor - 1$, whichever is even, and let G_1 be a k-regular graph on n_1 vertices, such that $\mathrm{girth}(G_1) \geq g_1 := \frac{\log n_1}{\log(k-1)}$. This exists, provided that n_0 is sufficiently large. Note that G_1 has at most $g_0 - 1 < \mathrm{girth}(G_0)$ edges.

A result by Angluin and Gardiner [1] states that there is a common lift G of G_0 and G_1. That is, G is a covering graph of G_0 and of G_1. Being a lift of a k-regular graph, G is k-regular as well. A closer inspection of their proof reveals that $n := |V(G)| \leq 4n_0n_1$.

Let $m := \frac{kn}{2}$ be the number of edges in G. We set $\Phi_m := \mathrm{TseitinBridge}(G, G_1, \varphi_1)$, where φ_1 is the locally bijective homomorphism from G to G_1.

It is not difficult to see that lifting cannot decrease the girth, and thus $\mathrm{girth}(G) \geq \mathrm{girth}(G_0) > |E(G_1)|$. Thus, we can apply Theorem 6 to G, G_1, and φ_1, and conclude that the success probability of PPSZ on Φ_m is at most $2^{-m+(1+\epsilon)n}$ when using heuristic R_w. A quick calculation shows that $g_1 \geq \frac{\log\log m}{\log k}$ if n_0 is sufficiently large, and thus $w \geq \sqrt{\epsilon \cdot \frac{\log\log m}{2\log k}} - 3$.

This construction gives us an infinite set $M \subseteq \mathbb{N}$ and, for each $m \in M$, a satisfiable k-CNF formula F_m on m variables for which the claimed hardness result holds. By a simple tweaking of the construction, we can ensure that M is "reasonably dense", meaning that there is some $m^* \in M \cap [m - \log m, m]$ for all sufficiently large m. We then let F_{m^*} be F_m, plus $m - m^*$ dummy variables. The success probability is then at most $2^{-m^*(1-2(1+\epsilon)/k)} \leq \mathrm{poly}(m)2^{-m(1-2(1+\epsilon)/k)}$. We leave the details to the reader. ◀

3 All You Need to Know About PPSZ: Proof of Theorem 6

We will explain the connection between PPSZ and width-bounded resolution lower bounds. After this section, the reader can forget everything about PPSZ and think of this paper as proving a certain resolution width lower bound. If $C = (C' \vee x)$ and $D = (D' \vee \bar{x})$ are clauses, then $(C' \vee D')$ is called the *resolvent* of C and D. It is clear that $C \wedge D$ logically implies $C' \vee D'$. Let Φ be a CNF formula. A *resolution derivation from* Φ is a sequence of clauses $C_1, \ldots, C_t$ such that every C_i is (1) a clause of Φ or (2) the resolvent of two earlier clauses. The *width* of the derivation is $\max_{1 \le i \le t} |C_i|$. For a clause C, we denote by $\text{width}(\Phi \vdash C)$ the minimum width of a resolution derivation from Φ that contains C. Resolution is complete for refutations, that is, Φ is unsatisfiable if and only if there is a derivation of the empty clause, denoted by $\square$, from Φ.

Proof of Theorem 6. Let G, H, φ be as in Theorem 6, and let $\Phi := \text{TseitinBridge}(G, H, \varphi)$. The only satisfying assignment of Φ is $\mathbf{0}$. Consider a variant of PPSZ run on Φ such that whenever it has to pick a random value for a variable, it correctly sets it to 0. Fix a permutation π. Let $Y(\pi)$ be the set of variables for which this variant of PPSZ under π could not derive the value using R_w, and let $Z(\pi) := \text{var}(\Phi) \setminus Y(\pi)$ be the rest, i.e., all variables whose value can be derived using R_w once all variables before them in π are set to 0. It is not difficult to see that the success probability of the actual PPSZ on Φ is exactly $\mathbb{E}_\pi \left[2^{-|Y(\pi)|}\right]$.

Suppose, for the sake of contradiction, that PPSZ using heuristic R_w has success probability greater than $2^{-m+(1+\epsilon)n}$. Then there is some π for which $Z(\pi) \ge (1+\epsilon)n$. Fix this π and set $Z := Z(\pi)$ and $Y := Y(\pi)$. The set of variables Z corresponds to a set F of edges, $F = \{e \in E(G) \ : \ x_e \in Z\}$. Set $G' = (V, F)$. Note that G' has n vertices and at least $(1+\epsilon)n$ edges. Setting a variable in Φ to 0 corresponds to simply deleting the corresponding edge in G, and therefore

$$\Phi|_{Y \mapsto \mathbf{0}} = \text{TseitinBridge}(G', H, \varphi) \ .$$

For a graph $G = (V, E)$ and a set $X \subseteq V$, define the edge boundary $\partial(X) := \{e \in E \ : \ |e \cap X| = 1\}$. Call G an (a, b)-expander if $|\partial(X)| \ge b$ for all sets X of exactly a vertices. The next lemma is basically Lemma 17 from [9], adapted for our purposes. We give a proof for completeness.

▶ **Lemma 7.** *Let $\epsilon > 0$ and let G' be a graph on n vertices with at least $(1+\epsilon)n$ edges. Let $\ell \in \mathbb{N}$ and $h = \ell/\epsilon$. If $h < \text{girth}(G')$ then G' contains a non-empty subgraph G'' that has minimum degree at least 2 and is an $(h, \ell+1)$-expander,*

Proof. Start with $G'' = G'$. If G'' has a vertex of degree 0 or 1, delete it. If G'' contains a set X of h vertices with $|\partial(X)| \le \ell$, delete X from G'', along with all incident edges.

The first type of deletion removes one vertex and at most one edge. The second type removes exactly h vertices. There are at most ℓ edges in the boundary of X; since $|X| < \text{girth}(G')$, the graph $G''[X]$ is a forest, and thus there are at most $h-1$ edges within X. Thus, removing X removes at most $\ell + h - 1 < (1+\epsilon)h$ edges.

We see that a step that removes a vertices removes fewer than $(1+\epsilon)a$ edges. Suppose the process terminates with t vertices deleted. Trivially $t \le n$. Fewer than $(1+\epsilon)n$ edges have been deleted, so G'' is non-empty. ◀

Let G'' be given by Lemma 7 with $\ell := w+1$. We will further restrict Φ so that only edges of G'' remain unset. Let $F'' := E(G) \setminus E(G'')$, $Y'' := \{x_e : e \in F''\}$, and $\Phi'' := \Phi|_{Y'' \mapsto \mathbf{0}}$. Note that $\Phi'' = \text{TseitinBridge}(G'', H, \varphi)$. Recall that all edges of G'' are mentioned in Z and since $Y'' \supseteq Y$ and restricting additional variables cannot increase the resolution width, we conclude that there exists $e \in E(G'')$ such that

$$\text{width}(\Phi'' \vdash \bar{x}_e) \leq w. \tag{3}$$

Towards a contradiction, we claim that in fact this resolution width is *large* for all variables x_e where $e \in E(G'')$. Indeed, we have the following theorem:

▶ **Theorem 8** (Resolution Lower Bound). *Let G be a graph of minimum degree 2 that is an $(h, \ell+1)$-expander. Suppose there is a locally injective homomorphism $\varphi : G \to H$ into some graph H. Then*

$$\text{width}(\text{TseitinBridge}(G, H, \varphi) \vdash \bar{x}_e) > \ell - 1 \ , \tag{4}$$

for all edges e of G, provided that $2h\ell + 5h + \ell < \text{girth}(H)$.

Note that G'' has minimum degree 2 and is a $(h, \ell+1)$-expander for $\ell = w+1$ and $h = \frac{w+1}{\epsilon}$. Also note that $\varphi : V(G'') \to V(H)$ (or rather, the restriction of φ to $V(G'')$) is still a locally injective homomorphism. Recall that $w = \sqrt{\frac{\epsilon \cdot \text{girth}(H)}{2}} - 3$ and hence $2h\ell + 5h + \ell = 2(w+1)^2/\epsilon + 5(w+1)/\epsilon + w + 1 < \text{girth}(H)$, and thus Theorem 8 applies to G''. This contradicts (3) and finishes the proof of Theorem 6. ◀

4 Proof of Theorem 8

Let $\Phi = \text{TseitinBridge}(G, H, \varphi)$ and let e^* be an edge of G. We will show that $\text{width}(\Phi \vdash \bar{x}_{e^*}) > \ell - 1$ for all such edges e^*. In fact, we will prove $\text{width}(\Phi|_{x_{e^*} \mapsto 1} \vdash \square) > \ell - 1$, which is a slightly stronger statement.

We will use the game characterization of resolution width due to Atserias and Dalmau [2]. Given a CNF formula F, the ℓ-bounded Atserias-Dalmau game played by two players, Prover and Delayer is defined as follows. A *position* in this game is a partial assignment α setting up to ℓ variables. The start position is the empty assignment. At position α, Prover can either (1) forget some variables, i.e., replace α by some $\beta \subset \alpha$. Or, (2), if $|\text{var}(\alpha)| \leq \ell - 1$, pick a variable $x \notin \text{var}(\alpha)$ and *query it*; Delayer has to respond with a truth value $b \in \{0, 1\}$, and α is updated to $\alpha \cup (x \mapsto b)$. The game ends if α violates a clause of F, in which case Prover wins. Delayer wins if she has a strategy to play indefinitely.

▶ **Theorem 9** (Atserias and Dalmau [2]). *Let F be an unsatisfiable CNF formula. If Delayer has a winning strategy for the $\ell+1$-bounded game then there is no width-ℓ resolution refutation of F.*

To show that $\text{width}(\Phi|_{x_{e^*} \mapsto 1} \vdash \square) > \ell - 1$ we define a winning strategy for Delayer for the ℓ-bounded game that ensures she never loses. Indeed, we will modify the game a bit: it is now played on Φ instead of $\Phi|_{x_{e^*} \mapsto 1}$; the starting position is the partial assignment $x_{e^*} \mapsto 1$; Prover can never forget x_{e^*} but is now allowed partial assignments up to size $\ell + 1$. That is, he can query a new variable provided $|\text{var}(\alpha)| \leq \ell$. It is easy to see that if Delayer wins this modified game, she wins the original one, too. Since $\Phi = \text{TseitinBridge}(G, H, \varphi)$, we can easily rephrase the rules of the game in terms of sets of edges instead of partial assignments:

The Atserias-Dalmau, Graph View. A position of the game is described by two disjoint set $F_0, F_1 \subseteq E(G)$. F_0 and F_1 correspond to the variables of Φ that the current partial assignment sets to 0 and 1, respectively. The start position is $F_0 = \emptyset$ and $F_1 = \{e^*\}$.

In every step, Prover either (1) removes one edge e from F_0 or F_1 (but never removes e^*). Or (2) he *queries* an edge $e \in E(G) \setminus (F_0 \cup F_1)$, provided $|F_0| + |F_1| \leq \ell$. Delayer can then decide whether to add e to F_0 or F_1.

Prover wins if there is a vertex u in G such that all edges incident to u are in $F_0 \cup F_1$ but $\deg_{F_1}(u)$ is odd (then the partial assignment α violates the Tseitin constraint of u); or if there are two edges $e, f \in F_1$ with $\varphi(e) = \varphi(f)$ (then α violates a bridge constraint).

We will now describe a winning strategy for Delayer. Throughout the game, she maintains a set $\tilde{F}_1$ such that $F_1 \subseteq \tilde{F}_1 \subseteq E \setminus F_0$. Let $V(\tilde{F}_1)$ denote the set of vertices incident to at least one edge of $\tilde{F}_1$. She makes sure $\tilde{F}_1$ satisfies certain invariants:

1. Every connected component of $(V, \tilde{F}_1)$ is a path; a path of positive length (i.e., a path that is not an isolated vertex) is called an $\tilde{F}_1$-path.
2. Every $\tilde{F}_1$-path contains at least one edge of F_1.
3. φ is injective on $V(\tilde{F}_1)$.
4. Each $\tilde{F}_1$-path has length at least $2h + 1$, and the first and last h edges of every $\tilde{F}_1$-path are not in F_1.
5. Each $\tilde{F}_1$-path has length at most $2h\ell + 2h + \ell$.

▶ **Observation 10.** *If $\tilde{F}_1$ satisfies the invariants, then no constraint is violated.*

Proof. In fact we show that invariants 1-4 already give the result. First, consider a Tseitin constraint of a vertex u. Since $\tilde{F}_1$ consists of disjoint paths, so does F_1. Thus, u is incident to 0, 1, or 2 edges of F_1. If it is incident to 0 or 2 edges of F_1, the Tseitin constraint of u is clearly not falsified. If it is incident to exactly one edge of F_1, then it is the endpoint of some path of F_1-edges. By Invariant 4, u is incident to some other edge $f \in \tilde{F}_1 \setminus F_1$. Thus, f is neither in F_0 nor in F_1, and the Tseitin constraint of u is not violated.

Next, consider a bridge constraint $(\bar{x}_e \vee \bar{x}_f)$. By construction we have $\varphi(e) = \varphi(f)$. By Invariant 3, φ is injective on $\tilde{F}_1$, and thus e, f cannot both be in F_1, and the bridge constraint is not violated. ◀

We will use the following property of φ.

▶ **Proposition 11.** *Let G' be a connected subgraph of G of diameter less than* girth(H)*. Then φ is injective on $V(G')$, and thus $\varphi(G')$ is isomorphic to G'.*

Proof. For the sake of contradiction, suppose $u, v \in V(G')$ are two vertices with $\varphi(u) = \varphi(v)$. Let p be a shortest path from u to v in G'. Write p as $u = u_0, u_1, \ldots, u_t = v$. By assumption, $t <$ girth(H). Under φ, the path p is mapped to a reduced walk in H, *reduced* meaning that $\varphi(u_{i-1}) \neq \varphi(u_{i+1})$ for all $1 \leq i \leq t - 1$. Since $\varphi(u) = \varphi(v)$, this is a closed walk and thus contains a cycle. The cycle has length at most $t <$ girth(H), a contradiction. ◀

How to initialize $\tilde{F}_1$. Delayer can easily initialize $\tilde{F}_1$. Write $e^* = \{u^*, v^*\}$. Since G has minimum degree 2, Delayer can start a reduced walk from u^* of length h, and also from v^* and add this to $\tilde{F}_1$. Since $2h + 1 <$ girth(H), this is a path; by Proposition 11, φ is injective on its vertices.

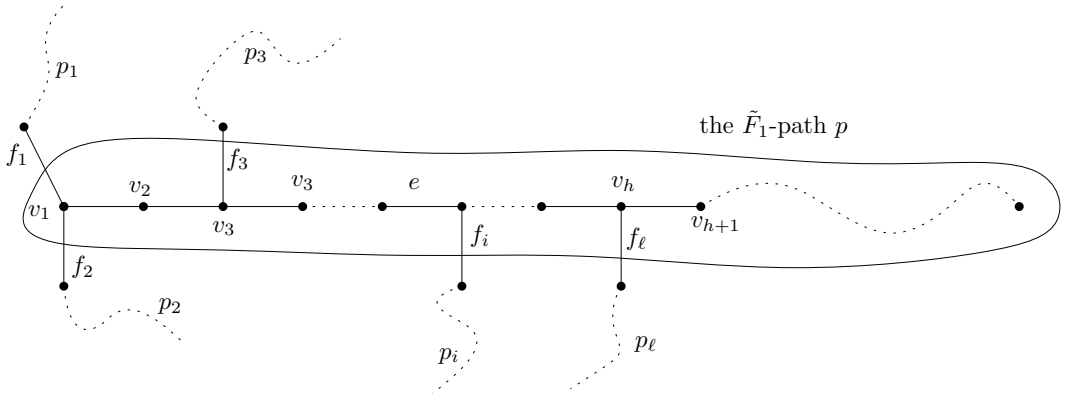
How to handle a Forget Step. Suppose Prover forgets some edge $e \in F_0 \cup F_1$. If $e \in F_0$, Delayer leaves $\tilde{F}_1$ unchanged. If $e \in F_1$, let p be the $\tilde{F}_1$-path containing e. If p contains some other F_1-edge besides e, Delayer does not change $\tilde{F}_1$; otherwise it simply removes all of p from $\tilde{F}_1$. All invariants stay satisfied.

How to handle a Query from Prover. Suppose Prover queries an edge e. Delayer has now to choose whether to include e into F_0 or F_1, and potentially update $\tilde{F}_1$

Case 1: e is not in $\tilde{F}_1$. Then Delayer adds e to F_0 and leaves $\tilde{F}_1$ unchanged. All invariants still hold. This includes the case that e is incident to some vertex on a $\tilde{F}_1$-path, but is not itself inside this path.

Case 2: e is in some $\tilde{F}_1$-path p but not among its first or last h edges. Delayer adds e to F_1 and leaves $\tilde{F}_1$ unchanged. All invariants still hold.

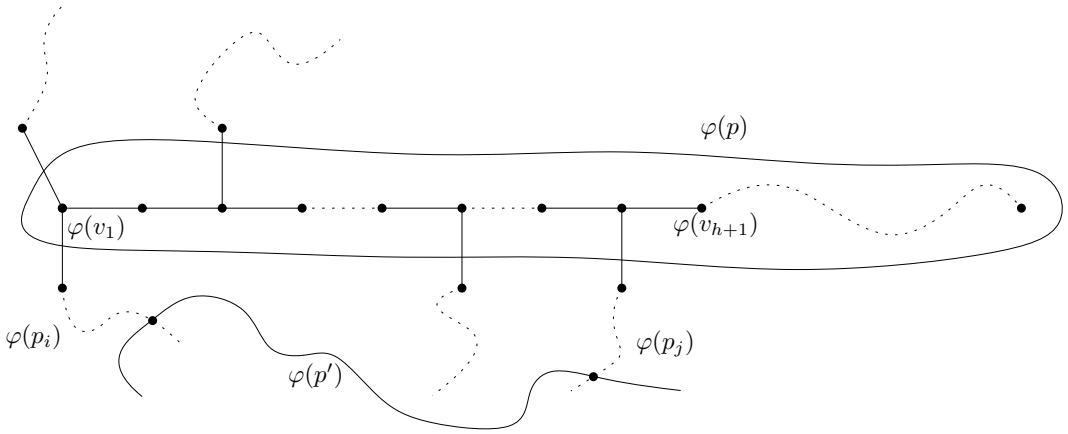
Case 3: e is among the first or last h edges of some $\tilde{F}_1$-path p. Let $v_1, \dots, v_{h+1}$ be the first $h+1$ vertices of p, and let q denote the length-h-path $v_1, \dots, v_{h+1}$. By assumption, e lies on the path q. Since G is an $(h, \ell+1)$-expander, there are edges $f_1, \dots, f_{\ell+1}$, each incident to exactly one vertex in $\{v_1, \dots, v_h\}$. One of those edges could be $\{v_h, v_{h+1}\}$, but without loss of generality, for $1 \le i \le \ell$, edge f_i connects some $a_i \in \{v_1, \dots, v_h\}$ to some b_i outside $\{v_1, \dots, v_{h+1}\}$. Since G has minimum degree 2 and girth larger than h, we can find paths $p_1, \dots, p_\ell$ such that each p_i has length h and starts with a_i as its first and b_i as its second vertex. Since $3h < \text{girth}(H) \le \text{girth}(G)$, the p_i are vertex-disjoint. Since $h + |p| \le h + 2h\ell + 2h + \ell < \text{girth}(H) \le \text{girth}(G)$, the path p_i intersects p only in vertex a_i. Thus, $C := p \cup p_1 \cup \dots \cup p_\ell$ is a tree, and its diameter is at most $h + 2h\ell + 2h + \ell$. This figure shows how C could look like:



Call p_i *blocked by* F_0 if it contains some edge from F_0; at most $|F_0|$ of the ℓ paths are blocked by F_0. Let p' be an $\tilde{F}_1$-path different from p. We say p' *blocks* p_i if the vertex sets of $\varphi(p')$ and $\varphi(p_i)$ intersect.

▶ **Proposition 12.** *Let path p' in $\tilde{F}_1$ be different from p. Then p' blocks at most one of the paths $p_1, \dots, p_\ell$.*

Proof. Let $C = p \cup p_1 \cup \dots \cup p_\ell$. As argued above, this is a tree in G and its diameter is less than girth(H). By Proposition 11, its image $\varphi(C)$ is a tree in H, isomorphic to C. Suppose, for the sake of contradiction, that $\varphi(p')$ intersects $\varphi(p_i)$ and $\varphi(p_j)$. This scenario would look like this:



Since $\varphi(p')$ and $\varphi(p)$ do not share any vertex (by Invariant 3), the subgraph $\varphi(p') \cup \varphi(p_i) \cup \varphi(p_j) \cup \varphi(p)$ contains a cycle. This cycle has size at most $|p'|+|p_i|+|p_j|+|q| \leq 2h\ell+2h+\ell+3h$, a contradiction. ◀

Call p_i *blocked by* $\tilde{F}_1$ if there is some $\tilde{F}_1$-path different from p that blocks p_i. By Proposition 12, at most $|F_1| - 1$ paths p_i are blocked by $\tilde{F}_1$. Thus, a total of at most $|F_0| + |F_1| - 1 \leq \ell - 1$ of the paths p_i are blocked by F_0 or $\tilde{F}_1$. Thus, there exists some path p_i, $1 \leq i \leq \ell$, that is not blocked. We now modify p by removing the edges on the path $v_1, v_2, \ldots, a_i$ and adding p_i. Let $\hat{p}$ denote the new version of p and $\hat{F}_1$ the new version of $\tilde{F}_1$. Note that $F_1 \subseteq \hat{F}_1$ still holds, since we only modify the set $\tilde{F}_1 \setminus F_1$. Obviously, $\hat{F}$ satisfies Invariants 1, 2, and 4. Since p_i is not blocked by F_0, $\hat{F}$ is disjoint from F_0; because p_i is not blocked by $\tilde{F}_1$, Invariant 3 still holds. Invariant 5 might be violated: $\hat{p}$ might be too long. We will deal with this in a minute.

Note that e is now either outside $\hat{F}_1$, and Delayer can include it into F_0; or it is inside $\hat{p}$, but then it is not among the first or last h edges of $\hat{p}$, and Delayer can include it into F_1.

It remains to address the possibility that $\hat{p}$ is too long, violating Invariant 5. If indeed $\hat{p}$ has more than $2h\ell + 2h + \ell$ edges, then it must somewhere contain $2h + 1$ consecutive edges that are not in F_1 (note that $|F_1| \leq \ell$). Let $e_0, \ldots, e_{2h}$ be these edges. Define $\hat{F}_1 := \tilde{F}_1 \setminus \{e_h\}$. That is, we split $\hat{p}$ into two parts, the first ending in $e_0, \ldots, e_{h-1}$, the second starting with $e_{h+1}, \ldots, e_{2h}$. Note that this satisfies Invariant 4. If one of these paths contains no edge from F_1 at all, we delete it from $\hat{F}$. We continue this process until all paths in $\hat{F}$ have size at most $2h\ell + 2h + \ell$. The final $\hat{F}$ satisfies all invariants.

5 Conclusion

We constructed close to tight hard instances for the PPSZ algorithm which uses bounded width resolution to derive values and showed that the savings can be at most $\frac{(1+\epsilon)2}{k}$. Several questions of various levels interest remain open. The first one is to obtain Super Strong ETH hard instance for resolution of larger width, ideally as close to $m/\log(m)$ as possible. Even for the weak heuristic, the hard instances from [9] hold for subformulas of size up to $m^{O(1)}$. The next problem is determining the precise constant in the savings of PPSZ.

References

1 Dana Angluin and A Gardiner. Finite common coverings of pairs of regular graphs. *Journal of Combinatorial Theory, Series B*, 30(2):184–187, 1981. doi:10.1016/0095-8956(81)90062-9.

2 Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008. doi:10.1016/j.jcss.2007.06.025.

3 Shiteng Chen, Dominik Scheder, Navid Talebanfard, and Bangsheng Tang. Exponential lower bounds for the PPSZ k-SAT algorithm. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1253–1263. SIAM, 2013. doi:10.1137/1.9781611973105.91.

4 Paul Erdős and Horst Sachs. Reguläre graphen gegebener taillenweite mit minimaler knotenzahl. (regular graphs with given girth and minimal number of knots.). *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg, Math.-Naturwiss.*, 12:251–258, 1963.

5 Thomas Dueholm Hansen, Haim Kaplan, Or Zamir, and Uri Zwick. Faster k-sat algorithms using biased-ppsz. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 578–589. ACM, 2019. doi:10.1145/3313276.3316359.

6 Timon Hertli. 3-SAT faster and simpler - unique-SAT bounds for PPSZ hold in general. *SIAM J. Comput.*, 43(2):718–729, 2014. doi:10.1137/120868177.

7 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005. doi:10.1145/1066100.1066101.

8 Ramamohan Paturi, Pavel Pudlák, and Francis Zane. Satisfiability coding lemma. *Chicago J. Theor. Comput. Sci.*, 1999, 1999.

9 Pavel Pudlák, Dominik Scheder, and Navid Talebanfard. Tighter hard instances for PPSZ. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 85:1–85:13, 2017. doi:10.4230/LIPIcs.ICALP.2017.85.

10 Dominik Scheder. PPSZ for $k \geq 5$: More is better. *TOCT*, 11(4):25:1–25:22, 2019. doi:10.1145/3349613.

11 Uwe Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 410–414. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814612.

12 Nikhil Vyas and R. Ryan Williams. On super strong ETH. In *Theory and Applications of Satisfiability Testing - SAT 2019 - 22nd International Conference, SAT 2019, Lisbon, Portugal, July 9-12, 2019, Proceedings*, pages 406–423, 2019. doi:10.1007/978-3-030-24258-9_28.

A Existence of Common Lift

▶ **Theorem 13** (Angluin and Gardiner [1]). *Let G and H be k-regular graphs. Then there exists a k-regular graph L that is a common lift of both G and H. Furthermore, $|V(L)| \leq 4|V(G)| \cdot |V(H)|$.*

Proof. Suppose first that both $G = (U, E)$ and $H = (V, F)$ are bipartite. By Hall's Theorem, each has a perfect matching, and in fact, we can partition E and F into k perfect matchings each: $E = E_1 \uplus \cdots \uplus E_k$ and $F = F_1 \uplus \cdots \uplus F_k$. The common lift L has vertex set $U \times V$ and edge set

$$\bigcup_{i=1}^{k} \left\{ \{(u,v),(u',v')\} \in \binom{U \times V}{2} \mid \{u,u'\} \in E_i, \{v,v'\} \in F_i \right\} .$$

It is not difficult to see that the projections $\varphi_G : (u,v) \mapsto u$ and $\varphi_H(u,v) \mapsto v$ are locally bijective homomorphisms from L into G and H, respectively.

If G (or H or both) fails to be bipartite, we first replace it by its 2-*lift* G_2. The vertex set of G_2 is $U \times \{0,1\}$, and we form its edge set by creating, for each $\{u,v\} \in E$, two edges $\{(u,0),(v,1)\}$ and $\{(u,1),(v,0)\}$. The graph G_2 is bipartite, and projection to the first coordinate is a locally bijective homomorphism. Finally, observe that the composition of locally bijective homomorphisms is again a locally bijective homomorphism. Altogether, we can replace G and H by their respective 2-lifts G_2 and H_2; these are bipartite graphs, so we find a common lift L on $4|U| \cdot |V|$ vertices. ◀