Total NP Search Problems, Resolution, PLS, and Wrong Proof

Sam Buss
U.C. San Diego

FOCS '2021 Workshop:
Reflections on
Propositional Proofs in Algorithms and Complexity

February 7, 2022

Talk outline

- Total NP Search Problems (TFNP)
- Resolution and PLS
  - The direct connection
  - The bounded arithmetic connection
- The Wrong-Proof problem
  - Small width resolution and PLS
- Res(small) and CPLS
- Concluding comments

Some results are "folklore";
New results: joint with N. Fleming & R. Impagliazzo ([BFI'ip]).

Total NP Search Problems — TFNP

Definition (Meggido-Papadimitriou'91; Papadimitriou'94)

A Total NP Search Problem (TFNP) is a polynomial time relation $R(x, y)$ so that R is

- *Total*: For all x, there exists y s.t. $R(x, y)$,
- *Honest (poly growth rate)*:
  If $R(x, y)$, then $|y| \leq p(|x|)$ for some polynomial p.

The TFNP Problem is:

Given an input x, output a y s.t. $R(x, y)$.

TFNP is intermediate between P (polynomial time) and NP (non-deterministic polynomial time).

Polynomial Local Search (PLS)

Inspired by Dantzig's algorithm and other local search algorithms:

Definition ([JPY'88].)

A PLS problem consists of polynomial time functions: $N(x,s)$ and $i(x)$, and a polynomial time predicate $F(x,s)$ s.t.

- $\forall x(F(x,i(x)))$.
- $\forall x,s(F(x,s) \rightarrow F(x,N(x,s)))$.

A solution is a point s such that $F(x,s)$ and $N(x,s) \geq s$.

$F(x,s)$ means "s is a feasible solution for x".
$i(\cdot)$ gives an initial feasible solution.
$s' = N(x,s)$ means "s' is the neighbor of s"

The input is x.
A solution to the PLS problem is any local minimum s.
Clearly, a PLS problem is in TFNP.

For many TFNP classes, it is useful to let the polynomial-time computations be relative to an oracle Ω:
("black-box" versus "white-box")

Definition (Meggido-Papadimitriou'91; Papadimitriou'94)

A Total NP Search Problem (TFNP) is a polynomial time relation $R(x, y, \Omega)$ so that R is

- *Total*: For all x, Ω, there exists y s.t. $R(x, y, \Omega)$,
- *Honest (poly growth rate)*:
  If $R(x, y, \Omega)$, then $|y| \leq p(|x|)$ for some polynomial p.

The TFNP Problem is:

Given an input x, output a y s.t. $R(x, y, \Omega)$.

W.l.o.g., $x = 1^n$ is a size parameter and Ω codes everything else.
The size of Ω is $N = 2^{n^{O(1)}}$. Queries $\Omega(z)$ have $|z| = n^{O(1)}$.

For PLS relative to an oracle, F and N can access the oracle.

CNF Search Problem:

"**CNF formula**" means a propositional formula in conjunctive normal form.

"**Width** $w(n)$ **CNF**" means a CNF in which all clauses have width $\leq w(n)$, where n is the size of the CNF.

Definition

A **CNF Search Problem** is the problem of: given an unsatisfiable CNF formula and a truth assignment τ, find a clause that is falsified by τ.

Observation

A CNF Search Problem for a sufficiently uniform family of (exponentially large) unsatisfiable polylog-width CNF formulas is the same thing as an oracle TFNP problem.

"Exponentially large" is $N = 2^{n^{O(1)}}$. "Polylog" in N is $n^{O(1)}$.

Observation

A CNF Search Problem for a sufficiently uniform family of (exponentially large) unsatisfiable polylog-width CNF formulas is the same thing as an oracle TFNP problem.

Proof sketch for $\Rightarrow$ direction:
Given a sufficiently uniform family of exponentially large, unsatisfiable, polylog width CNF's, and a truth assignment τ, encode them bitwise with an oracle Ω. The TFNP problem is to find a falsified clause, or to find a place where the CNF is incorrectly encoded by Ω. The solution to the TFNP problem must be verifiable in polynomial time. This is possible since the clauses are polylog-width and since the CNF is sufficiently uniform. □

For the oracle (black-box) version of TFNP: The "sufficient uniformity" does not require a uniform algorithm for generating the CNF instances. It only requires that, for any Ω that does ***not*** correctly encode one of the CNF's, there is a small (size $n^{O(1)}$) witness, verifiable in polynomial time, that it is not a valid instance of the family of CNF's.

Observation

A CNF Search Problem for a sufficiently uniform family of (exponentially large) unsatisfiable polylog-width CNF formulas is the same thing as an oracle TFNP problem.

Proof sketch for $\Leftarrow$ direction:
Given a TFNP problem $R(x, y, \Omega)$, choose the propositional variables p_z to have values given by $\Omega(z)$, and let the CNF be

$$\bigwedge_y \neg [\![R(x, y, \Omega) \text{ accepts}]\!].$$

$[\![R(x, y, \Omega) \text{ accepts}]\!]$ is the DNF of clauses of size $n^{O(1)}$ representing the answers to queries to the oracle Ω by an accepting computation of R.
$[\![R(x, y, \Omega)]\!]$ is expressed as a decision tree of depth $n^{O(1)}$ querying variables p_z for queries "$\Omega(z)$?" made by the computation $R(x, y, \Omega)$.
Note $n^{O(1)} = polylog(2^n) = polylog(N)$.

Equivalence of PLS and polylog-width resolution

Theorem (? — B.-Kołodziejczyk-Thapen'14)

A family of polylog width CNF Search problems is in PLS iff it has (sufficiently uniform) polylog-width resolution refutations.

Proof sketch for $\Leftarrow$ direction: A poly log-width, exponentially long, resolution refutation $\mathcal{R}$ can be converted into a PLS problem, with Ω encoding a propositional truth assignment τ and a resolution refutation $\mathcal{R}$, by

- The nodes of the PLS problem are the lines (clauses) of $\mathcal{R}$.
- A vertex s is feasible (satisfies $F(x, s, \Omega)$ iff $\tau(s) = False$.
- The neighborhood function N maps s to the hypothesis s' used to derive the clause s s.t. $\tau(s') = False$.
- Solutions are falsified input clauses.

Theorem (? — B. Kołodziejczyk-Thapen'14)

A family of polylog width CNF Search problems is in PLS iff it has (sufficiently uniform) polylog-width resolution refutations.

Proof sketch for $\Rightarrow$ direction:

The main conditions for a PLS problem solving a CNF Search problem can restated as:

- $F(x, i(x), \Omega)$
- $F(x, s, \Omega) \land s' := N(x, s, \Omega) < s \to F(x, s', \Omega)$
- $F(x, s, \Omega) \land s' := N(x, s, \Omega) \geq s \to (C_{s'}$ is false), where $C_{s'}$ is the clause that is found to be falsified at the solution s' to the PLS problem.

F and N are computed by polynomial time oracle machines. Queries to the oracle $\Omega(z)$ give values of variables p_z in the CNF Search Problem.

Thus, $\neg F(x,s,\Omega)$ and $N(x,s,\Omega)$ can computed by $n^{O(1)}$ many queries to the values of variables p_x.

- $\neg F(x,s,\Omega)$ is a conjunction of polylog-width clauses.
- $s' := N(x,s,\Omega)$ is determined by a $n^{O(1)}$-depth (polylog-depth) decision tree.
  Let $s_1, s_2, \ldots s_L$ be the possible values for s'

By b. and c., there is a straightforward polylog-width resolution derivation of $[\![\neg F(x,s,\Omega)]\!]$ from the clauses

$$C_{s_1} \;\ldots\; C_{s_{L'}} \quad [\![\neg F(x, s_{L'+1}, \Omega)]\!] \;\ldots\; [\![\neg F(x, s_L, \Omega)]\!].$$

Note $s_{L'+1}, \ldots, s_L < s$.
Combining these derivations for all s, together with $[\![F(x,i(x),\Omega)]\!]$ from condition a., we get a polylog-width resolution refutation of the initial clauses C_s.

Connection via Bounded Arithmetic

Definition

T^1_2 (resp. S^2_2) is the theory of bounded arithmetic with induction on NP-predicates (and length induction, PIND, on Σ^b_2 predicates).

Theorem (B.-Krajíček'94, Krajíček'94)

- *The provably total functions of T^1_2 (and S^2_2) are the functions many-one reducible to PLS.*
- *The $\forall\Pi^b_1$ (coNP) consequences of T^1_2 (and S^2_2) have straightforward propositional translations which have polylog-width resolution refutations.*

The first item is a witnessing theorem for T^1_2.
The second item is the Paris-Wilkie translation from bounded arithmetic to propositional logic.
These results hold also for the relativized (black box) setting, corresponding to TFNP with an oracle.

Wrong-Proof / Proof Consistency Search Problem

[Beckmann-B.'17] and [Goldberg-Papadimitriou'17,'18]
also [Krajíček'16]

Definition (Wrong-Proof Search Problem)

Let T be a proof system. An instance of Wrong-Proof for T is an (exponentially large) purported T-proof of a contradiction.
A solution to the Wrong-Proof problem is the identification of a syntactic error in the T-proof.

- [Beckmann-B.; Krajíček]: Wrong-Proof for
  Frege and extended-Frege.
- [Goldberg-Papadimitriou]: Wrong-Proof for
  Q-EFF (QBF + extended Frege functions)
- This talk: Wrong-Proof for
  (a) log-width resolution and constant-width resolution and
  (b) Resolution and Res(log).

Wrong-Proof for Resolution Refutations as a TFNP problem

An exponentially large ($2^{n^{O(1)}}$ size) instance is encoded by Ω describing:

- A truth assignment τ.
- For each clause, the presence or absence of each literal. In limited width resolution, the identities of the i-th literals.
- Some clauses are initial clauses; each has a designated literal which is true under τ. (Optional for polylog width.)
- Other clauses are listed with the resolution variable and pointers to their parent clauses (their hypotheses). Parent clauses precede the clause (so the proof is a dag).
- The final clause is the empty clause.
- A solution is either
  - A falsified input clause, or
  - An error in an inference.

Theorem

PLS is many-one equivalent to the Wrong-Proof Problem for polylog-width resolution.

Proof idea: By the previous construction, PLS instances can be converted to instances of the Wrong-Proof for polylog-width resolution, and vice-versa.

Theorem (BFI'ip)

The Wrong-Proof Problem for width 3 resolution is many-one equivalent to the Wrong-Proof Problem for polylog-width resolution.

Proof idea: We need to show how to convert a polylog width resolution derivation to a width 3 resolution refutation. In the TFNP setting, this means converting a width $n^{O(1)}$ resolution refutation to a width 3 resolution refutation.

The idea is to introduce new variables that stand for all possible disjunctions of $n^{O(1)}$ many literals. This is essentially the same as introducing these variables by extension, which can be done with width 3 clauses. With the new variables, any width $n^{O(1)}$ refutation can be converted to a width 3 refutation. □

Restatement as Effective Quasi-P Simulation

Definition (see Pitassi-Santhanan'10)

A proof system P (strongly) effectively p-simulates a proof system Q if there is a truth-preserving polynomial time transformation f such for all φ, an Q-proof of $f(\varphi)$ can be converted (in polynomial time) to a polynomial size P proof of φ.

Define "effectively quasi-p simulates" similarly with quasipolynomial in place of polynomial.

Theorem

Width 3 resolution strongly effectively quasi-p simulates polylog-width resolution.

Proof idea: The same proof idea works; however, now we are converting arbitrary proofs from width 3 resolution to polylog-width resolution. □

Note: For simplicity, the definition of "(strongly) effective p-simulation" is slightly strengthened from the usual one.

Resolution and Res(polylog)

Definition

- A t-conjunction is a conjunction of $\leq t$ literals.
- $\mathbf{Res}(\boldsymbol{f(S)})$ means a propositional refutation system in which lines are permitted to be disjunctions of $f(S)$-conjunctions, where S is the size of the refutation.

We will discuss resolution (that is, $\mathrm{Res}(1)$) and $\mathrm{Res}(\mathit{polylog})$.

The next level of Bounded Arithmetic

Definition

T^2_2 (resp. S^3_2) is the theory of bounded arithmetic with induction on Σ^p_2-predicates (and length induction, PIND, on Σ^p_3 predicates).

Theorem (Krajíček-Skelley-Thapen'07, Krajíček'94, ...)

- *The provably total functions of T^2_2 (and S^3_2) are the functions many-one reducible to CPLS (Colored-PLS).*
- *The $\forall\Pi^b_1$ (coNP) consequences of T^2_2 (and S^3_2) have straightforward propositional translations which have Res(polylog) refutations.*

The first item is an NP-witnessing theorem for T^2_2.
The second item is the Paris-Wilkie translation from bounded arithmetic to propositional logic.
These results hold also for the relativized (black box) setting, corresponding to TFNP with an oracle.

Colored PLS (CPLS) [Krajíček-Skelley-Thapen'07]

Simlar to PLS: With $C(x, s, y)$ expressing that node s has color y and $c(x, s)$ giving a color to terminal nodes s.

Definition (Modified from Krajíček-Skelley-Thapen'07)

A CPLS problem has polynomial time functions $N(x, s)$, $i(x)$ and $c(x, y)$, and polynomial time predicates $F(x, s)$ and $C(x, s, y)$ s.t.:

- $\forall x \forall y (F(x, i(x)) \wedge \neg C(x, i(x), y))$.
  "Initial node (root) has no color".
- $\forall x, s (F(x, s) \rightarrow F(x, N(x, s)))$.
- $\forall x, s, y (F(x, s) \wedge N(x, s) < s \wedge C(x, N(x, s), y) \rightarrow C(x, s, y))$.
  "Colors propagate from neighbors".

A solution to the CPLS problem is a point the following fails.

- $\forall x, s (F(x, s) \wedge N(x, s) \geq s \rightarrow C(x, s, c(x, s)))$.
  "Leaf nodes have a (known) color."

CPLS relativizes to an oracle Ω similarly to PLS.

Theorem (BFI'ip)

A family of CNF Search problems is in CPLS iff it has (sufficiently uniform) resolution refutations.

Proof idea: Similar in spirit to before. For the conversion from CPLS to a resolution refutation, clauses are the disjunctions of the possible colors of the node. □

Theorem (BFI'ip)

The CPLS Search Problem is many-one equivalent to the Wrong-Proof Search problem for Resolution.

Theorem (BFI'ip)

The Wrong-Proof Search problem for Resolution (i.e., $\mathrm{Res}(1)$) is many-one equivalent to the Wrong-Proof Search problem for $\mathrm{Res}(polylog)$.

Theorem (BFI'ip; c.f. Pitassi-Santhanan'10, Atserias-Bonet'04)

Resolution (i.e., $\mathrm{Res}(1)$) strongly effectively quasi-p simulates $\mathrm{Res}(polylog)$.

Concluding comments

- Many-one equivalence of Wrong-Proof Search problem is not always equivalent to Strongly Effective Quasi-P Equivalence. E.g., Pitassi-Santhanan show a quantified propositional logic is complete for effective p-simulation, but their method does not work to give a complete Wrong-Proof Search problem.
- The Wrong-Proof Search problem Frege encompasses all provably total functions of U_2^1, and thus all "usual" TFNP problems [B.-Beckmann]. What can be said about stronger classes, such as for extended Frege or Q-EFF or even stronger? Is there a natural stopping point? (c.f. [Goldberg-Papadimtriou]).
- Is there a better generalization of CPLS for higher levels the of Bounded Arithmetic theories? (Compare to the Game Induction Principles of [Skelley-Thapen'11].)
- What about Wrong-Proof Search for other weak propositional proof systems (cutting planes, SOS, etc.)? [BFI'ip]

Thank you!