Compressibility of TFNP Reductions

Nathan Acheampong and Jake Gameroff
Supervised by Professor Robert Robere
School of Computer Science, McGill University

Introduction & Background

TFNP is the class of all search problems satisfying:

- Totality: Every instance has a solution.
- Polynomial verification: Solutions can be checked efficiently.

The class has many natural problems such as the following.

1. Given an integer N, find one of its prime factors.
2. Given a graph G and an odd-degree vertex, find another one.
3. Given $f : [n] \to [n-1]$, find $x \neq y$ with $f(x) = f(y)$.

As we can see, the totality is often obtained from combinatorial lemmas.

Alternative model of computation

Black-box TFNP, or TFNP^{dt}, is the analogue of TFNP where the model of computation is the *decision tree*. Efficient verification of solutions is done by shallow decision trees – those of depth polylog$n$ – where n is the length of the string encoding the instance of the problem at hand.

Reductions

Reductions capture the idea of using one problem to solve another.

Decision-tree reduction: Given two TFNP^{dt} problems A_n and B_m, we say A_n *decision-tree reduces* to B_m *in depth* d if there are the following two families of decision trees of depth at most $d \leq$ polylog$n$:

- **Forward maps:** For every index i of the input string to B_m, there is a tree f_i which takes the input x to A_n and calculates the bit $f_i(x)$ at that index.
- **Backward maps:** For every solution o of B_m, there is a tree T_o transforming o into a solution $T_o(x)$ of A_n.

The complexity of the reduction is $\log m + d$.

In this case, we write or $\mathrm{A}_n \leq_d \mathrm{B}_m$.

Completion

A problem P is complete for a class $\mathcal{C}$ if $P \in \mathcal{C}$ and all problems in $\mathcal{C}$ decision-tree reduce to P. In order to study TFNP^{dt}, subclasses have been defined by considering sets for which important problems are complete, since TFNP^{dt} is suspected to have no complete problem. Among them is PLS^{dt}, for which the Sink-of-DAG problem is complete.

Connection to Proof Complexity

A *propositional proof system* is a polynomial time algorithm which, given a formula F and a proof P, verifies that (i) F is unsatisfiable, and (ii) that P is a proof that F is unsatisfiable.

The *Resolution Proof System* is defined by the following two rules and acts only on clauses.

$$\text{Resolution}: \quad C \vee x, D \vee \bar{x} \vdash C \vee D$$

$$\text{Weakening}: \quad C \vdash C \vee D$$

Here, we have clauses C, D and literals $x, \bar{x}$ not in C and D respectively. The width of a clause is number of literals in it. A Resolution proof of size m and width d is a sequence of m clauses, whose widest clause has d literals. The complexity of the proof is $\log m + d$.

The connection: It is important to note that for all total search problems A, there is an unsatisfiable CNF formula F that claims the A is not total. A subclass $\mathcal{C}^{dt}$ of TFNP^{dt} is *characterised* by a proof system P if the complexity of the reduction of A to the $\mathcal{C}^{dt}$-complete problem is in the same order as the complexity of proving $F \in$ UNSAT. PLS^{dt} is the first subclass characterised by a proof system [2]; this proof system is Resolution.

Main Results

In this section, we give a proof-sketch that the Sink-of-DAG (SoD) problem has the compression property. We define the following search problems:

- SoD$_N$ – Given an $N \times N$ grid where each node either has a self-loop or points to a node one level down. Return $(1, 1)$ if it has a self-loop, or any other node with a self-loop and a predecessor.
- Iter$_N$ – This is the same as SoD but on an $N^2 \times 1$ grid and with the additional property that each node can point to any node under it (as opposed to just one level down).

It is an interesting exercise to check the following:

- **Lemma 1.** If $\mathrm{A} \leq_d \mathrm{SoD}_N$ then $A \leq_d \mathrm{Iter}_{N^2}$;
- **Lemma 2.** If $\mathrm{A} \leq_d \mathrm{Iter}_N$ then $A \leq_d \mathrm{SoD}_N$.

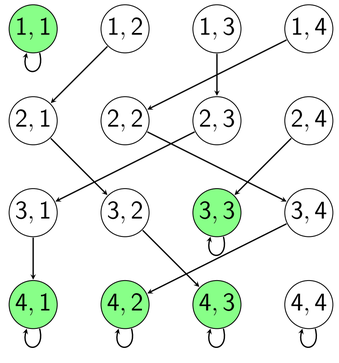

Figure: Example SoD$_4$ instance. Green nodes are solutions.

Theorem 1. Let $\mathrm{A} \in$ TFNP^{dt} be a search problem taking inputs of size n. If A reduces to SoD$_N$ in depth d, then A reduces to SoD$_m$ in depth $O(d)$, where $m = n^{O(d)}$.

Proof Sketch

Setup: Consider the reduction $\mathrm{A} \leq_d \mathrm{Iter}_N$ with forward maps (f_{ij}). We may assume all decision paths have length d, since we can add in dummy queries from unused variables.

Observation: We will call two decision tree paths *identical* if they query the same set of literals. A critical observation is that there are at most $\binom{n}{d}2^d = n^{O(d)}$ non-identical paths.

Compression: For any pair of identical paths f_u, f_v with outputs $o_u < o_v$ (i.e. o_u is above o_v on the grid), rewire the output of f_u to be o_v. This bounds the total number of distinct outputs by $n^{O(d)}$. See the figure below for an illustration.

Cleanup: Delete each tree f_o for any unused output o and keep exactly one tree per used output, giving $n^{O(d)}$ trees. The compressed trees yield $\mathrm{A} \leq_d \mathrm{Iter}_{n^{O(d)}}$, since no solution is introduced (they are only lost).

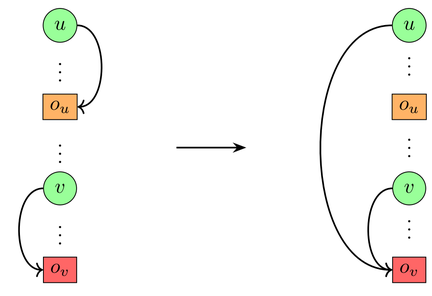

Figure: Compression step: rewiring o_u to o_v.

This shows that Iter has the compression property. So if $\mathrm{A} \leq_d \mathrm{SoD}_N$, Lemma 1 implies that $\mathrm{A} \leq_d \mathrm{Iter}_{N^2}$ and so $\mathrm{A} \leq_{O(d)} \mathrm{Iter}_{n^{O(d)}}$. Then we invoke Lemma 2 to conclude that $\mathrm{A} \leq_{O(d)} \mathrm{SoD}_{n^{O(d)}}$. □

References & Acknowledgements

[1] Sam Buss, Noah Fleming, and Russell Impagliazzo. TFNP Characterizations of Proof Systems and Monotone Circuits. In *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of LIPIcs, pages 30:1–30:40. Schloss Dagstuhl, 2023.
[2] David Johnson, Christos Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988.

This research was supported by two NSERC Undergraduate Student Research Awards.