On the Average-Case Hardness of Total Search Problems

Chethan Kamath, Pietrzak Group

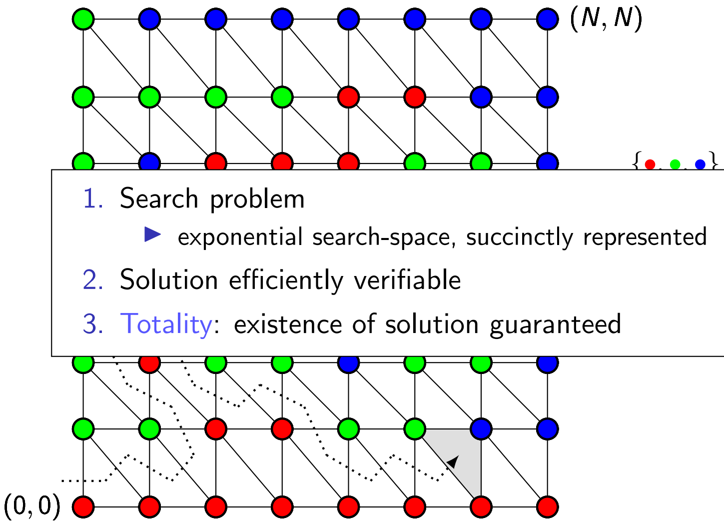
IST AUSTRIA

Thesis Defence, February 3rd, 2020

Outline

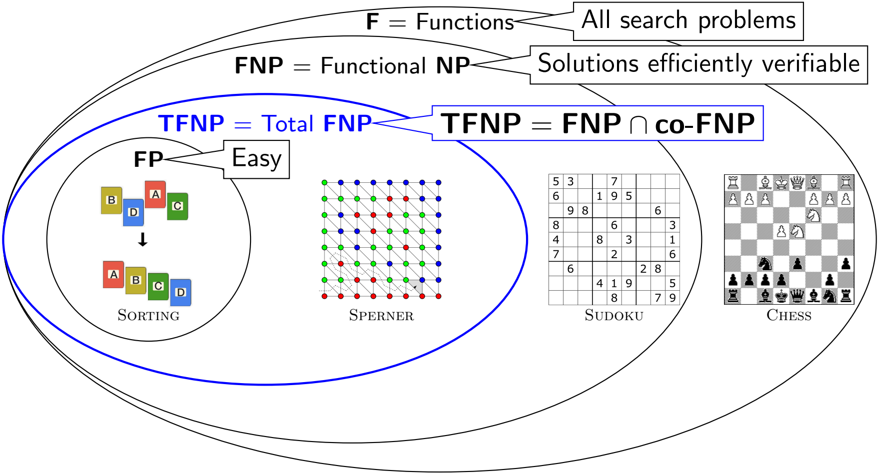

ENGLISH IDIOM:
A needle in a haystack
something that is very difficult to find
OysterEnglish.com

Needle in a Haystack, Guaranteed: Sperner's Lemma

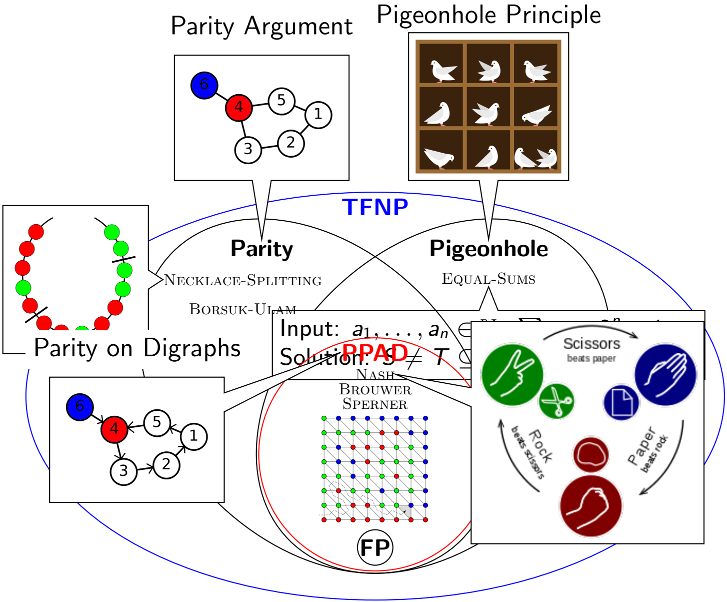


1. Search problem
   - exponential search-space, succinctly represented
2. Solution efficiently verifiable
3. Totality: existence of solution guaranteed

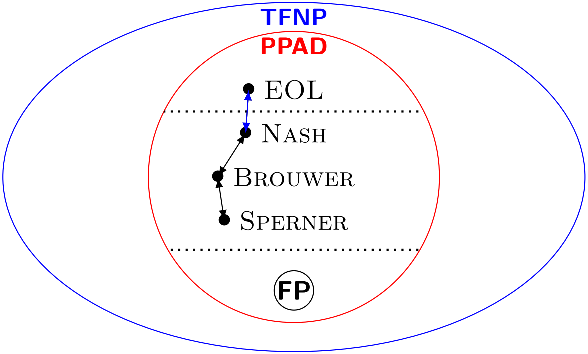
Search Problems vs. Total Search Problems [MP91,P94]



Subclasses of **TFNP**: Arguments of Existence

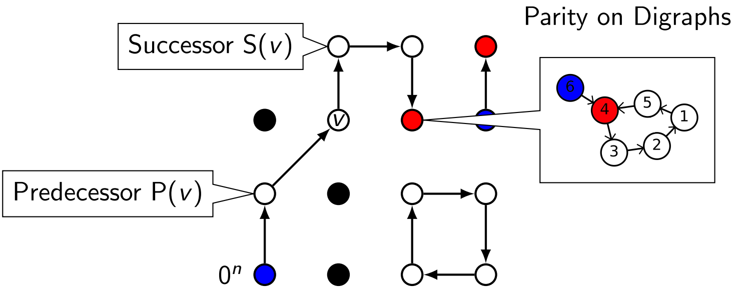


PPAD: Polynomial Parity Argument on Digraphs



- Compete problem for **PPAD**: END-OF-LINE (EOL)
- NASH, BROUWER, SPERNER $\in$ **PPAD** [P94]
- They are also **PPAD**-complete [DGP05,CDT09]

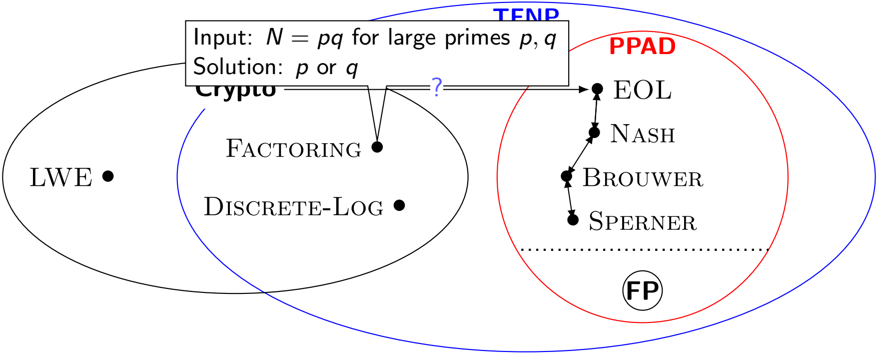
End-of-Line



- Input: A digraph on $\{0,1\}^n$ with in-/out-degree ≤ 1
- Guarantee: 0^n is a source •
- Solution: Any sink •
- Problem: Instance easy if the whole digraph given as input
- Succinct representation: Circuits $\mathsf{S},\mathsf{P} : \{0,1\}^n \to \{0,1\}^n$

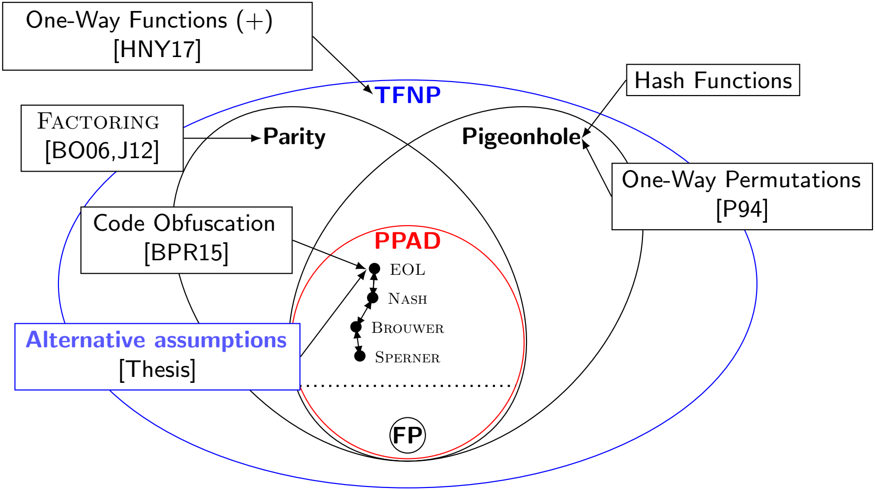

THIS ISN'T GOING TO BE EASY BUT WE'LL START LOOKING
LOST
LOST
NATE FAKES
©2014 Nate Fakes | www.nateramblings.com

Cryptography and **TFNP**

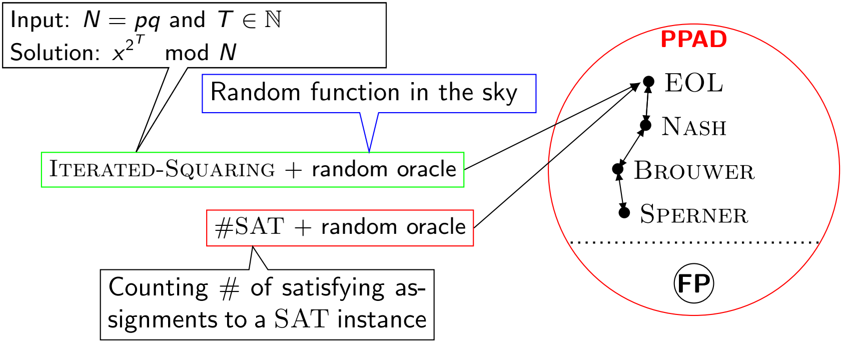


- Goal: Come up with hard distribution on **TFNP** instances.
- Why? To defeat heuristics
  - E.g.: Lemke-Howson algorithm and NASH
- How? Reduce from cryptographic hardness assumptions

Cryptography and **TFNP**: What is known?

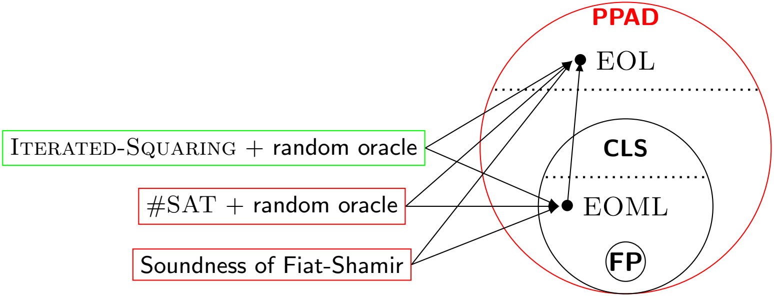


Our Results



- Theorem 1: EOL is hard-on-average relative to a random oracle assuming ITERATED-SQUARING is hard [CHK+19a]
- Theorem 2: EOL is hard-on-average relative to a random oracle assuming #SAT is hard (worst case) [CHK+19b]

Our Results...

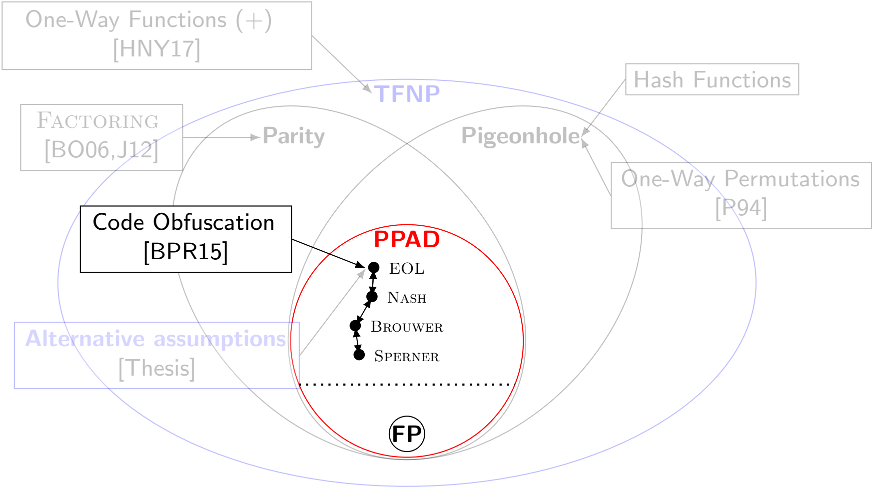


Further strengthenings:

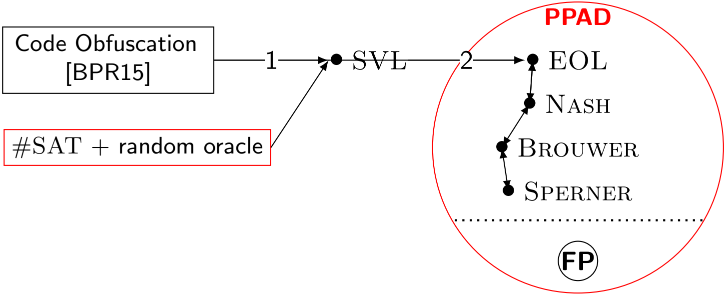
- $\boxed{\text{Theorem } 2+}$: EOL is hard-on-average assuming the soundess of the Fiat-Shamir Transform for Sumcheck Protocol
- Theorems 1 and 2+ apply to **CLS** $\subseteq$ **PPAD** [HY17]
  - Contains interesing problems from game theory (e.g., Simple stochastic games, mean payoff games) [FGMS19]

[3]

Techniques

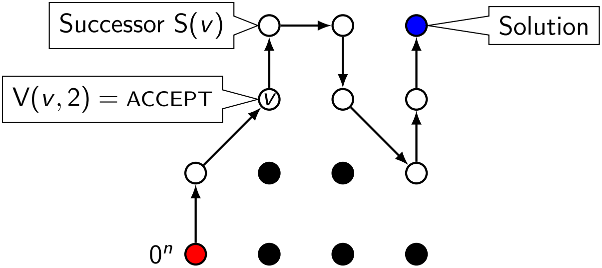

One-Way Functions (+)
[HNY17]
TFNP
Hash Functions
Factoring
[BO06,J12]
Parity
Pigeonhole
One-Way Permutations
[P94]
Code Obfuscation
[BPR15]
PPAD
EOL
Nash
Brouwer
Sperner
Alternative assumptions
[Thesis]
FP

Techniques...



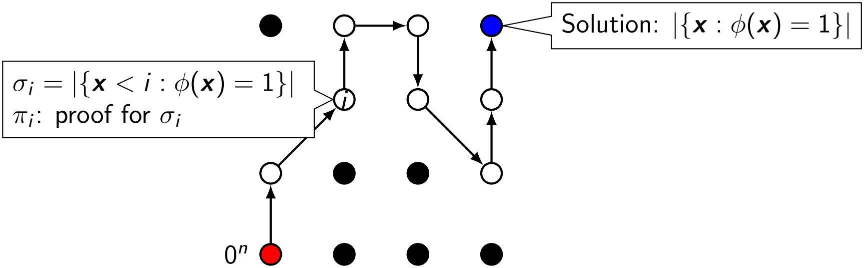
- Intermediate promise problem: Sink-of-Verifiable-Line
  - Step 1: Construct SVL from code obfuscation
  - Step 2: Simulate EOL using reversible pebbling
- Theorem 2: SVL is hard-on-average relative to a random oracle assuming #SAT is hard (worst case) [CHK+19b]

Sink-of-Verifiable-Line (SVL)



- Input: A digraph on $\{0,1\}^n$ with in-/out-degree ≤ 1
- Path starting at 0^n defined by successor $\mathsf{S} : \{0,1\}^n \to \{0,1\}^n$
- Verifier circuit $\mathsf{V} : \{0,1\}^n \times [2^n] \to$ ACCEPT/REJECT
  - Promise Verifier accepts (v, i) iff $v = \mathsf{S}^i(0^n)$
- Solution: L-th vertex •

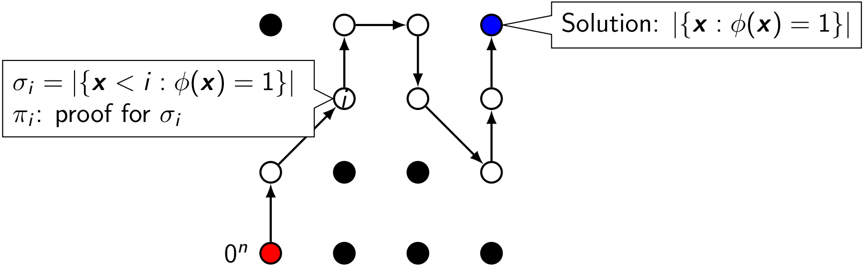
From #SAT to SVL: Verifiable Counting



- Goal: reduce #SAT instance $\phi(x_1, \ldots, x_n)$ to SVL $(\mathsf{S}, \mathsf{V}, L)$
- Attempt 1: Set i-th vertex as σ_i: # satisfying assignments $\leq i$
- Problem: No way to efficiently verify intermediate count
- Attempt 2: Append a proof π_i
- Problem: getting π_i to be small (i.e., **poly**(n))

  ...

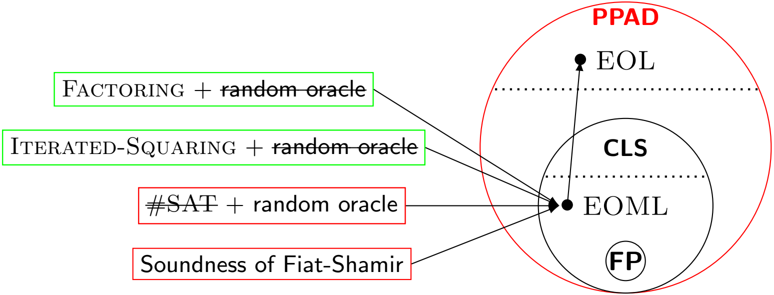
From #SAT to SVL: Verifiable Counting...



...

- Problem: getting π_i to be small (i.e., **poly**(n))
- Solution: use the Sumcheck Protocol [LFKN92]
- Problem: Sumcheck Protocol is interactive
- Solution: use Fiat-Shamir Transform [FS86]
- Problem: next proof $\mathsf{S}(i, \sigma_i, \pi_i) = (i + 1, \sigma_{i+1}, \pi_{i+1})$
- Solution: recursive proof-merging

Conclusion



- Theorem 1: FACTORING instead of ITERATED-SQUARING
- Theorem 1: Removing random oracle
- Theorem 2: Hardness in **CLS**/**PPAD** relative to random oracle

Hunts Needle in a Haystack

HOW LONG does it take to find a needle in a haystack? Jim Moran, Washington, D. C., publicity man, recently dropped a needle into a convenient pile of hay, hopped in after it, and began an intensive search for (a) some publicity and (b) the needle. Having found the former, Moran abandoned the needle hunt.

Desperate junkies search for an alleged "needle in the haystack."

Thank you!

References

[BO06] Buresh-Oppenheim. *On the TFNP complexity of factoring.* Unpublished

[BPR15] Bitansky, Paneth and Rosen. *On the cryptographic hardness of finding a Nash equilibrium.* FOCS'15

[CDT09] Chen, Deng and Teng. *Settling the complexity of computing two-player Nash equilibria.* JACM'09

[CHK+19a] Choudhuri et al.. *PPAD-hardness via iterated squaring modulo a composite.* Unpublished.

[CHK+19b] Choudhuri et al.. *Finding a nash equilibrium is no easier than breaking Fiat- Shamir.* STOC'19

[DGP05] Daskalakis, Goldberg and Papadimitrou. *The complexity of computing a Nash equilibrium.* SICOMP'09

[FGMS19] Fearnley et al.. *Unique end of poten-tial line.* ICALP'19

[FS86] Fiat and Shamir. *How to prove yourself: Practical solutions to identification and signature problems.* Crypto'86

[HNY17] Hubáček, Naor and Yogev. *The journey from NP to TFNP hardness.* ITCS'17

[HY17] Hubáček and Yogev.*Hardness of continuous local search: Query com- plexity and cryptographic lower bounds.* SODA'17

[J12] Jeřabek. *Integer factoring and modular square roots.* JCSS'16

[LFKN92] Lund et al.. *Algebraic methods for interactive proof systems* JACM'92

[MP91] Megiddo and Papadimitrou. *On total functions, existence theorems and computational complexity.* TCS'91

[P94] Papadimitrou. *On the complexity of the parity argument and other inefficient proofs of existence.* JCSS'94