Total NP Functions I: Complexity and Reducibility

Sam Buss (UCSD)
sbuss@math.ucsd.edu

Newton Institute, March 2012

Total NP Functions — TFNP

[Meggido-Papadimitriou'91, Papadimitriou'94].

Definition

TFNP, the class of Total NP Functions, is the set of polynomial time relations $R(x,y)$ such that

- $R(x,y)$ is polynomial time and honest (so, $|y| = |x|^{O(1)}$),
- R is *total*, i.e., for all x, there exists y s.t. $R(x,y)$.

Thm. If TFNP problems are in FP (p-time), then $\mathrm{NP} \cap \mathrm{coNP} = \mathrm{FP}$.

Pf. If $(\exists y \le s)A(x,y) \leftrightarrow (\forall y \le t)B(x,y)$ is in $\mathrm{NP} \cap \mathrm{coNP}$, then $A(x,y) \vee \neg B(x,y)$ defines a TFNP predicate. □

Thus, any $\mathrm{NP} \cap \mathrm{coNP}$ predicate gives a TFNP problem. These are called $F(\mathrm{NP} \cap \mathrm{coNP})$ functions.

Also open: Does $\mathrm{TFNP} = \mathrm{FP}^{\mathrm{NP}}$?

These two papers emphasized especially the following problems:

1. Polynomial Local Search (PLS). Functions based on finding a local minimum. [JPY'88]

2. PPAD. Functions which are guaranteed total by Sperner's Lemma, or Brouwer's Fixpoint Theorem, or similar problems.
"PPAD" = "**Polynomial Parity Argument in Directed graphs**".

3. Nash Equilibrium (Bimatrix Equilibrium) and **Positive Linear Complementarity Problem (P-LCP)**. Solutions can be found with Lemke's algorithm (pivoting).

[DGP'08] says: "Motivated mainly by ... Nash equilibria"

Polynomial Local Search (PLS)

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

Definition (Johnson, Papadimitriou, Yanakakis'88)

A PLS problem, $y = f(x)$, consists of polynomial time functions: initial point $i(x)$, neighboring point $N(x,s)$, and cost function $c(x,s)$, a polynomial time predicate for feasibility $F(x,s)$, and a polynomial bound $b(x)$ such that

0. $\forall x(F(x,s) \to s \le b(x))$.
1. $\forall x(F(x,i(x)))$.
2. $\forall x(N(x,s) = s \vee c(x,N(x,s)) < c(x,s))$.
3. $\forall x(F(x,s) \to F(x,N(x,s)))$.

A solution value for $y = f(x)$ is a point $y = s$ such that $F(x,s)$ and $N(x,s) = s$.

Thus, a solution is a local minimum. Clearly, $\mathrm{PLS} \subseteq \mathrm{TFNP}$.

Examples of PLS problems include:

Dantzig's algorithm for linear programming,
Lin-Kernighan for Traveling Salesman,
Kernighan-Lin for graph partition.

Totality of PLS functions is guaranteed by the principle that

Any finite directed acyclic graph has a sink.

The acyclicity must be witnessed by a cost function. PLS problems use the wlog assumption that each node has outdegree ≤ 1.

Note that the output $y = f(x)$ may not be the local minimum found by following edges starting from x. Indeed, that problem is PSPACE-hard.

PPAD, PPA, PPADS

PPA:

Any undirected graph with degrees ≤ 2 which has a vertex of degree 1 has another vertex of degree 1.

PPAD:

Any directed graph with in-/out-degrees ≤ 1 which has a vertex of total degree 1 has another vertex of total degree 1.

PPADS:

Any directed graph with in-/out-degrees ≤ 1 which has a source, also has a sink.

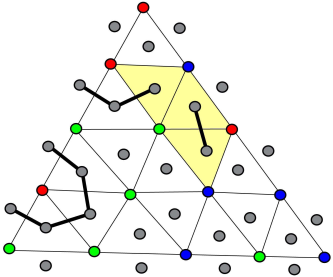
In all cases, the graph is given implicitly by a Turing machine which computes the neighbors of a given vertex.

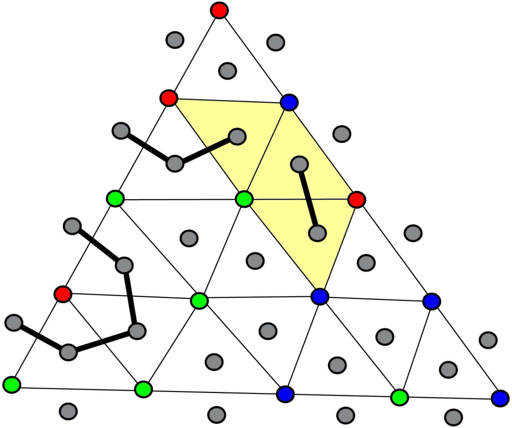
SPERNER is in PPAD

Sperner's Lemma Suppose a triangle is uniformly triangulated, and each vertex is labeled with one of three colors. Also suppose for each color (R,G,B) one of the three sides has no vertex of that color. Then, the triangulation contains a triangle whose vertices are colored with distinct colors.

Associated TFNP problem: Given a polynomial time Turing that computes the color of any vertex, find a pan-chromatic triangle or find a place where the edge color restriction is violated.

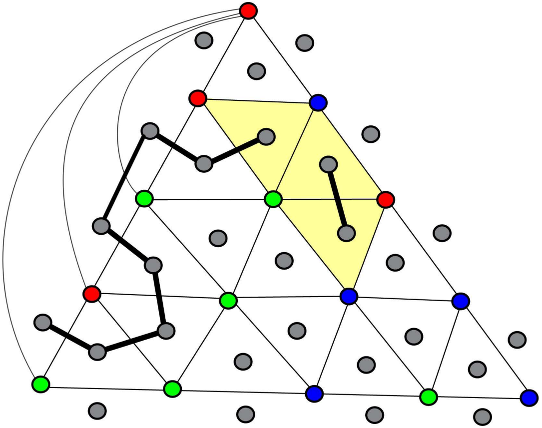
(Wlog, the Turing machine can be given by a Boolean circuit that calculates M's values for inputs of a given length.)





For an “R-G” path: feasible triangles are ones with both Red and Green. Outgoing direction keeps Red to the left and Green to the right. Initially at outer left triangle. All other sources/sinks are panchromatic triangles.

Either a source or sink suffices. (Sperner’s lemma does not care about the order of colors on the panchromatic triangle.)



For an "R-G" path: feasible triangles are ones with both Red and Green. Outgoing direction keeps Red to the left and Green to the right. Initially at outer left triangle. All other sources/sinks are panchromatic triangles.

Either a source or sink suffices. (Sperner's lemma does not care about the order of colors on the panchromatic triangle.)

2-NASH

A two player game is two $m \times n$ matrices A and B representing the payoffs for players A and B, as functions of the m (resp., n) strategies of A (resp., B).

Definition

A *Nash equilibrium* is pair of mixed strategies $\mathbf{x}$, $\mathbf{y}$ (probability distributions on strategies) s.t. the quantities

$$\mathbf{x}^T A\mathbf{y} \qquad \text{and} \qquad \mathbf{x}^T B\mathbf{y}$$

are at (local) maxima relative to $\mathbf{x}$ and to $\mathbf{y}$, respectively, and separately.

Definition

An ϵ-*Nash equilibrium* is similar, but requiring only that the derivative test for local maxima gives values $< \epsilon$.

P-LCP

Definition ($\mathrm{LCP}(M, \mathbf{q})$)

Given $n \times n$ M and vector $\mathbf{q}$, find vectors $\mathbf{x}$, $\mathbf{y}$ s.t.

$$\mathbf{y} = M\mathbf{x} + \mathbf{q}, \qquad \mathbf{x}, \mathbf{y} \geq 0, \qquad \mathbf{x}^T \mathbf{y} = 0.$$

M is a *P-matrix* if its principal minors all have determinant > 0.

The P-LCP problem is the LCP-problem for P-matrices

Given a $n \times n$ matrix M and $\mathbf{q}$ either find a solution $\mathrm{LCP}(M, \mathbf{q})$, or find a principal minor with determinant ≤ 0.

[Papadimitriou'94] 2-Nash and P-LCP are in PPAD. (Based on Lemke's algorithm.)

More TFNP classes and problems

Polynomial Pigeonhole Principle, PIGEON [P'84, Cook]

Give polynomial time $f : [m] \to [m-1]$, find $i \neq j$ such that $f(i) = f(j)$.

Factoring integers

Given non-prime $m \in \mathbb{N}$, find a factor of m.

Smith's problem

Given an odd-degree graph G and a Hamiltonian cycle, find another Hamiltonian cycle of G.

SMITH is in PPA, but not known to be PPA-complete.
FACTORING does not belong to any "named" TFNP subclass.

Many-one and Turing reductions

Let $R(x,y)$ and $Q(x,y)$ be TFNP problems.

Definition ($\preccurlyeq$)

A polynomial time many-one reduction from R to Q (denoted $R \preccurlyeq Q$) is a pair of polynomial time functions $f(x)$ and $g(x,y)$ so that, for all x, if y is a solution to $Q(f(x),y)$, then $g(x,y)$ is a solution to R, namely $R(x,g(x,y))$.

Definition ($\preccurlyeq_T$)

A polynomial time Turing reduction $R \preccurlyeq_T Q$ is a polynomial time Turing machine that solves R making (multiple) invocations of Q. It must succeed no matter which solutions y are returned in response to its queries $Q(-,-)$.

Some Complete ($\preccurlyeq$) Problems

$\preccurlyeq$-Complete for PLS:

- FLIP: local maximum output for Boolean circuit [JPY'88]
- Kernighan-Lin for minimum weight graph partition [JPY'88]
- Lin-Kernighan algorithm heuristic for TSP. [PSY'90]

$\preccurlyeq$-Complete for PPA:

- LEAF - find another leaf in an undirected graph of degree ≤ 2.

Open. Are SMITH, various other graph problems (Hamilton Decomposition, TSP Nonadjacency, Chévalley Mod 2), or other combinatorial principles PPA-complete? Or even $\preccurlyeq$-reducible to each other?

The following are $\preccurlyeq$-complete for PPAD:

- SINK.OR.SOURCE - find another, (in-/out-degrees ≤ 1)
- 3D-SPERNER [P'94]
- 2D-SPERNER [Chen-Deng'06a]
- 4-NASH [Daskalakis-Goldberg-P'06]
- 2-NASH [Chen-Deng'06b, Chen-Deng-Teng'09]
- Brouwer, Borsuk-Ulam, Tucker. [P'94]
- 2D-BROUWER [Chen-Deng'06a]

Open: Is P-LCP complete for PPAD?
[Adler-Verma'07/'11] conjecture not.

Complete for PPADS:

- SINK - find a sink, given a source (in-/out-degrees ≤ 1)

Question: It is likely that suitable restrictions on 2D-SPERNER, etc., are $\preccurlyeq$-complete for PPADS. Is there a natural restriction of 2-NASH that is complete for PPADS?

Relativized (type 2) TFNP problems

Recall that many natural TFNP problems have their input implicitly specified by a Turing machine (or circuit) that computes a function f.

Definition

A *relativized* or *Type 2* TFNP problem $R(1^n, f, y)$ is a (oracle) polynomial time predicate that takes as input a size bound 1^n and one or more functions $f : [2^n] \to [2^n]$. A solution is any value such that $R(1^n, f, y)$.

Definition: A many-one reduction from $R \preccurlyeq Q$ is a triple of oracle polynomial time functions:

- a mapping $1^n \mapsto 1^m$, that is computing $m = m(n)$.
- a function $\beta^f : [2^m] \to [2^m]$,
- a function γ^f such that if y is a solution to $Q(1^m, \beta^f, y)$, then γ^f computes a solution to $R(1^n, f, \gamma^f(1^n, y))$.

Definition

The relativized classes PPAD, PPA, PPADS, PPP, etc., are defined as the $\preccurlyeq$ closure of the canonical complete problems for these classes. E.g.,

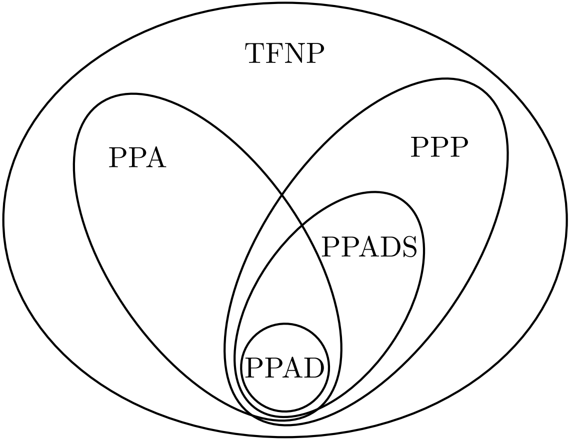
$$\preccurlyeq(\mathit{SOURCE.OR.SINK}) \qquad \preccurlyeq(\mathit{LEAF})$$

$$\preccurlyeq(\mathit{SINK}) \qquad \preccurlyeq(\mathit{PIGEON}).$$

These classes are often denoted "PPAD^G", etc. but we refer to them as just "relativized PPAD", or even just "PPAD", etc.

Theorem (BCEIP'98)

In the relativized (oracle) setting, we have:
$\mathrm{PPAD} \subset \mathrm{PPADS} \subset \mathrm{PPP}$ *and* $\mathrm{PPAD} \subset \mathrm{PPA}$
No other inclusions hold (even under Turing reductions).



What about PLS? (Relativized setting)

Theorem (Morioka'01, Buresh-Oppenheim–Morioka'04, Buss-Johnson'12)

- PPAD, PPA, PPADS, PPA $\not\preccurlyeq_T$ PLS. *[M]*
- PLS $\not\preccurlyeq_T$ PPA, PPAD. *[BO-M, BJ]*

Open: Do PLS $\preccurlyeq$ PPADS or PLS $\preccurlyeq$ PPP hold?

Proof technique for second result: Translate, á la Paris-Wilkie, the complete problems LEAF (or, SOURCE.OR.SINK) and ITER, for PPA (or, PPAD) and PLS, to propositional statements in either Boolean logic or the nullstellensatz system. Prove a Turing reduction gives rise to a reduction of proposition proofs in these systems: these reductions give either constant depth proofs, quasipolynomial size proofs, or quasipolylogarithmic nullstellensatz refutations. Known upper and lower bounds for these propositional principles imply that no such Turing reduction can exist.

More details: Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be sequents expressing the totality of relativized TFNP problems Q_1 and Q_2.

Thm: If $Q_1 \preccurlyeq_T Q_2$, there is a sequent calculus proof of the sequent $\mathcal{S}_1$ from instances of $\mathcal{S}_2$ which has quasipolynomial size, polylogarithmic height, and constant depth. [BJ'12, based on BO-M'04].

Thm: If $Q_1 \preccurlyeq_T Q_2$, and $\mathcal{S}_2$ has polylogarithmic degree nullstellensatz refutations, then $\mathcal{S}_1$ has polylogarithmic degree nullstellensatz refutations. [BJ'12, based on BO-M'04]

Thm. LEAF and SOURCE.OR.SINK have polylogarithmic degree nullstellensatz refutations. [BCEIP'98]

Thm. ITER does not have polylogarithmic degree nullstellensatz refutations. [BO-M'04]
□

Turing reductions versus many-one reductions

Theorem (Buss-Johnson'12)

Let Q be any complete problem for PPA, PPAD, PPADS, PLS *(LEAF, SOURCE.OR.SINK, SINK, ITER). Then $R \preccurlyeq_T Q$ iff $R \preccurlyeq Q$.*

Proof idea: combine the results of multiple calls to Q into a single call to Q.

Open Question: Does the same for the class PPP (that is, $Q =$PIGEON? (In the relativized setting.)

Some first classes partly inspired by Bounded Arithmetic

1. α-PLS, for α an ordinal ϵ_0 or Γ_0. PLS is modified so that costs are notations for ordinals $< \alpha$. [Beckmann-Buss-Pollett'02].
2. Colored PLS [Krajíček-Skelley-Thapen'07]
3. RAMSEY. Input: an undirected graph G on N nodes. Output: a homogeneous set of size $\frac{1}{2}\log N$.
4. Weak PHP, PHP_n^{2n}, etc.

Thm: α-PLS is many-one equivalent to PLS for $\alpha = \epsilon_0$ and $\alpha = \Gamma_0$. [BBP'02]

Thm: Colored PLS is strictly stronger than PLS. [KST'07]

General Question: What is the strength of RAMSEY? Of PHP_n^{2n}?

Definition (Colored PLS [Krajíček-Skelley-Thapen'07)

Given input x and polynomial time computable functions: neighbor N, initial point $i(x)$, color predicate $C(y,\alpha)$ for node y and color α, end node color assignment function $e(y)$. Find a witness that one of the following is false:

- $\forall\alpha\,[\neg C(i(x),\alpha)]$.
- $\forall y\,\forall\alpha\,[N(y) < y \wedge C(N(y),e) \to C(y,e)]$.
- $\forall y\,[N(y) < y \vee C(y,e(y))]$.

Values of y decrease under N until reaching a leaf. The initial point has no color. Any color of a neighbor of y is a color of y. Every local minimum has a color.

Next talk: More NP search problems from bounded arithmetic!