The Computational Complexity of Computing Nash Equilibrium

Andreas-Nikolas Göbel

NTUA

Algorithmic Game Theory, June 2011

Summary

- John Nash: Every Game has a (Mixed) Nash Equilibrium.
- C.S. : How hard is to compute a Nash Equilibrium?
- Outline of the talk:
  - Complexity Classes for Search Problems vs Complexity Classes for Decision Problems (NP vs FNP).
  - The class TFNP and its problems that can be grouped according to a non-constructive existence proof:

    PPA and PPAD
    PLS
    PPP
  - The problem NASH and its computational complexity. (will be left for next Thursday).

Decision vs Search Problems

- A language $L \subseteq \Sigma^\star$ is in NP if and only if there is a polynomially decidable and polynomially balanced relation R_L such that $L = \{x : (x, y) \in R_L \text{ for some } y\}$.
- Each decision problem L, Has a corresponding search problem, S_L.
- Given input $x \in \Sigma^\star$, return a $y \in \Sigma^\star$, such that $(x, y) \in R_L$, if such a y exists, otherwise return the string "no".
- For NP-problems we define the class containing their search version to be FNP.
- FP contains the problems of FNP for which a poly-time algorithm is known.

Reductions for search problems

- Polynomial time reductions between two search problems A, B: $\exists f, g$ s.t. $R_A(x, g(y)) \iff R_B(f(x), y)$.
- f produces an instance $f(x)$ of the function probelm B such that we can construct an output $g(y)$ for x from any correct output y of $f(x)$.

Which one is harder?

- SAT is complete for NP.
- FSAT is complete for FNP (a-la-cook proof), That is find a satisfying assignment.
- Solving FSAT immediately solves SAT.
- There exist a poly-time algorithm for FSAT based on the self reducibility of SAT whith n-calls to SAT.

Theorem

$\mathrm{FP} = \mathrm{FNP}$ *if and only if* $\mathrm{P} = \mathrm{NP}$

The class TFNP

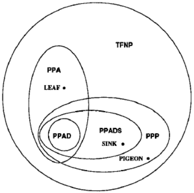
- Contains the search problems for which R_L is total.
- Telling whether R_L is total is undecidable $\leftarrow$ TFNP is a semantic class.
- Semantic Classes are not known to contain complete problems.
- FP $\subseteq$ TFNP $\subseteq$ FNP.

TFNP separation results

- $\mathrm{TFNP} = \mathrm{F}(\mathrm{NP} \cap \mathrm{coNP})$.
  - NP contains problems with "yes" certificate y s.t. $R_1(x, y)$.
  - coNP contains problems with "no" certificate z s.t. $R_2(x, z)$
  - For $\mathrm{TFNP} \subseteq \mathrm{F}(\mathrm{NP} \cap \mathrm{coNP})$ let $R_1 = R$ and $R_2 = \emptyset$.
  - For $\mathrm{F}(\mathrm{NP} \cap \mathrm{coNP})] \subseteq \mathrm{TFNP}$ let $R = R_1 \cup R_2$.
- An FNP-complete problem is in TFNP if and only if $\mathrm{NP} = \mathrm{coNP}$.
  - "if" part comes from the above
  - "only if": Assume that FSAT reduces to the FNP-complete problem S_R, and that $S_R \in \mathrm{TFNP}$
    Any unsat formula φ has a "no" certificate y, s.t. $R(f(\varphi), y)$, and $g(y) =$"no"
    y is guaranteed to exist as R is total, thus $\overline{\mathrm{SAT}}$ is in NP.

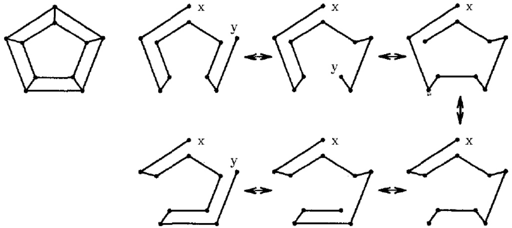
Proofs of existence and the subclasses of TFNP

- **Polynomial Local Search** (PLS): Every directed acyclic graph has a sink.
- **Polynomial Pidgeonhole Principle** (PPP): If a function maps n elements to $n-1$ elements, then there is a collision.
- **Polynomial Parity Argument** (PPA): If an undirected graph has an odd degree node, then it must have another odd degree node.
- **Polynomial Parity Argument Directed** (PPAD): If a directed graph has an unbalanced node (indegree$\neq$ outdegree), then it must have another unbalanced node.



The class PPA (An example)

- ANOTHER HAMILTON CYCLE
- Any graph with odd degree nodes has an even number of hamilton cycles through edge xy
  - Take a HC and remove edge xy to obtain a Hamilton Path
  - Fix x as an endpoint and start rotating from the other end
  - Each hamilton path has two valid neighbours ($d = 3$) except the paths with endpoints x, y
  - Parity Argument: At least two such HP's must exist
  - Add edge x, y

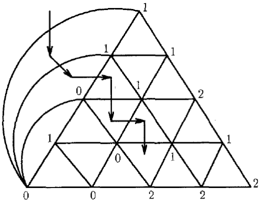


The class PPAD (An example)

- 2DSPERNER
- Sperner's Lemma on triangulated plannar graphs: Consider an arbitrary coloring $(0, 1, 2)$ on a plannar triangulated graph, then there exists an even number of trichromatic triangles.
- The computational problem that arrises for this lemma: The outer face is one trichromatic triangle, can u fing a secong one?
- When the graph coloring is given explicitly it is easy
- Instead we have an algorithm (cirquit) that given a vertice (v), after poly-time it returns the color of v.

The class PPAD (An example) II

- Each original vertex obtains each own color
- No vertex on the edge uv contains the color $3 - c(u) - c(v)$, where c is the coloring function.
- Add external 0-1 edges
- Traverse the triangles using 0-1 edges (doors)
- A room w/o a door must be trichromatic
- The process cannot exit the triangle and cannot fold upon itself
- The tour leaves 0 nodes to the right, hence we have direction.



The class PPA (Formal Definition)

- Let A be a problem and M the associated poly-time TM.
- Let x be an input of A.
- The configuration space $C(x)$ of M on input x is $\Sigma^{p(|x|)}$.
- For $c \in C(x)$, M outputs in $\mathcal{O}(p(n))$ time a set of at most two configurations, namely $M(x, c)$.
  - $M(x, c)$ may be empty.
- c, c' are neibours, $[c, c'] \in G(x)$, iff $c \in M(c', x)$ and $c' \in M(c, x)$.
  - $G(x)$ is a symmetric graph of degree at most two.
  - It is the implicit search graph of the problem.
- Let $M(x, 0 \dots 0) = 1 \dots 1$, and $0 \dots 0 \in M(x, 1 \dots 1)$, hence $0 \dots 0$ is the standart leaf.
- PPA: “Given x, find a leaf of $G(x)$ other than $0 \dots 0$.”

ANOTHER HAMILTON CYCLE is in PPA

- Let x be the input: (G, h) the graph and a hamilton cycle.
- C_x, the configuration space of x contains encodings of valid Hamilton Paths and other irrelevant strings.
- On input $x \in C_x$ machine M outputs in time $\mathcal{O}(p(n)))$ a set $M(x, c$ of, at most, two neigbors of a valid Hamilton Path
- c and c' are neigbors ($[c, c'] \in G(x)$) iff $c \in M(c', x)$ and $c' \in M(c, x)$, in the neigboring graph of "valid" Hamilton Paths
- the standart leaf is the Hamilton Path that occured from the removal or edge xy.

The class PPAD (I)

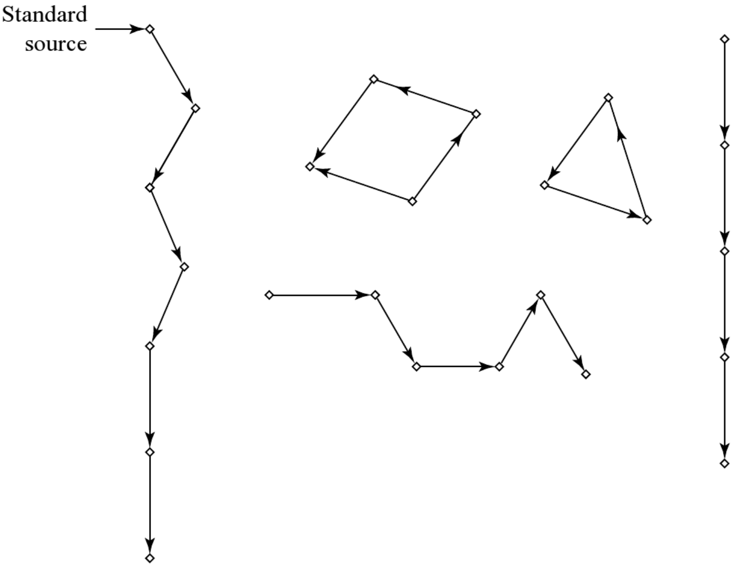
Same as PPA with the following differences:

- $M(x, c) = \mathrm{In}(x, c) \cup \mathrm{Out}(x, c)$ is an ordered pair of configs whith $|\mathrm{In}(x, c)| \leq 1$ and $|\mathrm{Out}(x, c)| \leq 1$.
- $G(x)$ is now directed $(c, c') \in G(x)$ iff $c \in \mathrm{In}(x, c')$ and $c' \in \mathrm{Out}(x, c)$.
- We search for a node (not the standart one) with $|\mathrm{In}(x, c)| + |\mathrm{Out}(x, c)| = 1$, i.e. any other source or sink.

2DSPERNER is in PPAD

- Let x be the input: A triangle, a triangulation and a valid color assignment (in terms of a cirquit/algorithm)
  - we need some sort of guarantee of the validity of the input.
- C_x is the configuration space of x, encodings of the active sub-triangles of the triangulation
- $M(x, c) = In(x, c) \cup Out(x, c)$ where:
  - $In(x, c)$ is the sub-triangle we came from.
  - $Out(x, c)$ is the sub-triangle we are heading into.
- c and c' are neighbors iff $[c, c'] \in G(x))$ iff $c \in M(c', x)$ and $c' \in M(c, x)$
- The standart node is the outer face of the triangulated graph.

The class PPAD (II)



End Of The Line

- In order to fix the setting for the next talk consider the following computational problem:
    - End Of The Line: Given two cirquits S and P, each with n input and output bits, such that $S(P(0^n)) \neq 0^n = P(S(0^n))$, find an input $x \in \{0, 1\}^n$ s.t. $P(S(x)) \neq x$ or $S(P(x)) \neq x \neq 0^n$
- The above problem is PPAD-complete (recall the proof that CirquitSAT is NP complete)
- From now on we can say that PPAD contains the problems that reduce to End Of The Line.

THANK YOU!!