Function & Total Search Complexity Classes

Markos Epitropou

National Technical University of Athens

November 29, 2012

Function Problems

$L \in NP$

There is a polynomial-time decidable, polynomially balanced relation R_L such that for all strings x: there is a string y with $R_L(x, y)$ if and only if $x \in L$.

FL

Given x, find a string y such that $R_L(x, y)$ if such a string exists; if no such string exists, return "no".

Function Problems

Reductions

Functions Problems: $A \leq B$

- if x is an instance of A, then $R(x)$ is an instance of B.
- if there exists a solution for A with input x, then there exists a solution for B with input $R(x)$.
- if z is a solution for $R(x)$, then $S(z)$ is a solution for x.
- R, S are computable in logarithmic space.

$FP = FNP \iff P = NP$

- SAT can be solved in polynomial time if and only if FSAT can be solved in polynomial time
- FSAT is FNP-complete

Total Search Problems

Function Problem

1. Decision Problem: Decide if a solution exists ("yes", "no")
2. Search Problem: if "yes", find a solution

Total Search Problem

A "Total" FNP (TFNP) problem is an FNP problem where a solution is guaranteed to exist.

Total Search Problems

$$FP \subseteq TFNP \subseteq FNP$$

- $FP = TFNP \Rightarrow P = NP \cap coNP$
- $TFNP = FNP \Rightarrow NP = coNP$

Interesting to define classes of problems where solution is guaranteed to exist by a non-constructive proof.

Plan

1. Represent possible configurations with nodes.
2. Find a relation between nodes (Edges). The relation must be chosen so that the solutions are nodes with a special property (The non-consturctive proof helps!)

Total Search Problems

$$FP \subseteq TFNP \subseteq FNP$$

- $FP = TFNP \Rightarrow P = NP \cap coNP$
- $TFNP = FNP \Rightarrow NP = coNP$

Interesting to define classes of problems where solution is guaranteed to exist by a non-constructive proof.

Plan

1. Represent possible configurations with nodes.
2. Find a relation between nodes (Edges). The relation must be chosen so that the solutions are nodes with a special property (The non-consturctive proof helps!)

PLS

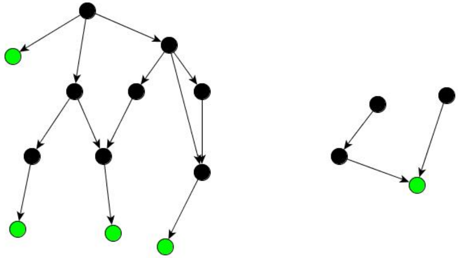


Figure: FIND SINK

Argument in Proof of Existence

Every finite directed acyclic graph has a sink.

PLS

Stable configuration for neural networks

Neural Network:

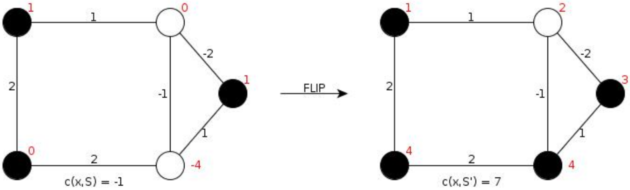
- $G = (V, E)$
- $S : V \to \{-1, 1\}$ (Nodes)
- Stable Configuration: $(\forall i \in V) S(i) \cdot \sum_{\{i,j\} \in E} S(j) w_{ij} \geq 0$

Define:

- Cost: $c(x, S) = \sum_{\{i,j\} \in E} S(i) S(j) w_{ij}$
- Neigborhood (Edges):

$$S' \in N(x, S) \iff (S' \in FLIP(S) \wedge (c(x, S') > c(x, S))$$

PLS



Stable Configuration for neural networks $\leq$ FIND SINK

Sinks $\subseteq$ *Solutions*: If node i is flipped and $S(i) \cdot \sum_{\{i,j\} \in E} S(j) w_{ij} = -\delta < 0$, then $c(x, S') = c(x, S) + 2\delta$.

PLS-complete Problems

- TSP, under the Kernighan-Lin neighborhood.
- MAX-CUT, under the flip neighborhood.
- Stable configuration for neural networks.

PPA

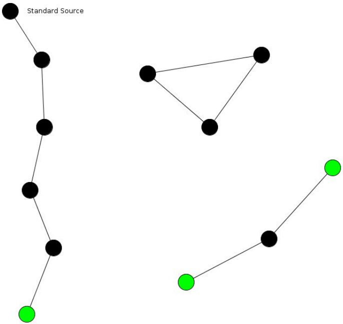
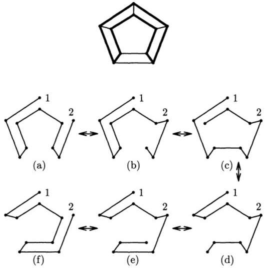


Figure: ODD DEGREE NODE

Argument in Proof of Existence

Any finite graph has an even number of odd-degree nodes *OR* All graphs of degree two or less have an even number of leaves.

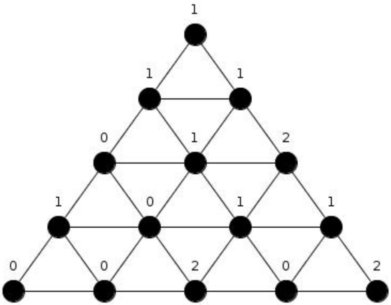


SMITH

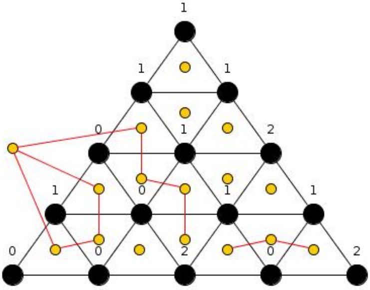
Given a graph G with odd degrees, and a Hamilton cycle, find another one.

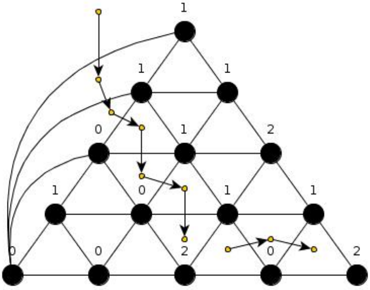
- Nodes: Hamilton Paths without edge $\{1, 2\}$.
- Edges: Add an edge to the endpoint ($\neq 1$) and break cycle in a unique way.

Sperner's Lemma



Sperner's Lemma





PPAD

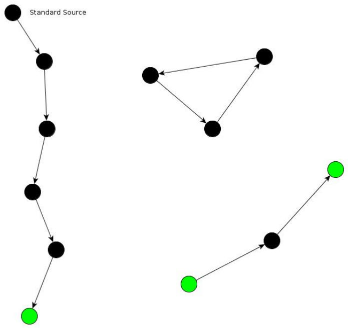


Figure: END OF THE LINE

Argument in Proof of Existence

If a finite directed graph has an unbalanced node (a vertex with different in-degree and out-degree), then it has another one.

PPAD-complete Problems

- 3D SPERNER
- BROUWER
- NASH

Hierarchy

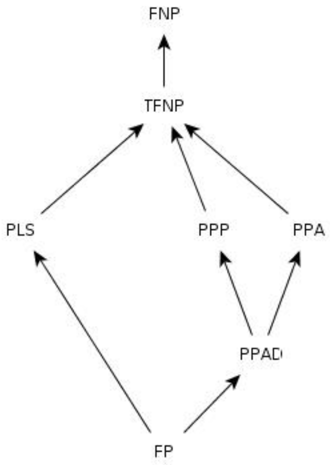


Figure: Search Classes

References

D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis.
How easy is local search?
Journal of Computer and System Sciences, pages 79–100, 1988.

C. H. Papadimitriou.
Computational Complexity.
Addison-Wesley, 1994.

C. H. Papadimitriou.
On the complexity of the parity argument and other inefficient proofs of existence.
Journal of Computer and System Sciences, pages 498–532, 1994.

C. H. Papadimitriou.
Algorithmic Game Theory, Chapter 2.
Cambridge University Press, 2007.

M. Yannakakis.
Survey: Equilibria, fixed points, and complexity classes.