CSE 5319 Notes 5: Complexity of Equilibria

(Last updated 4/19/23 9:06 AM)

Roughgarden 19/20

Nisan 2/3/19.3.4

5.A. Tractability Issues (R 19.1)

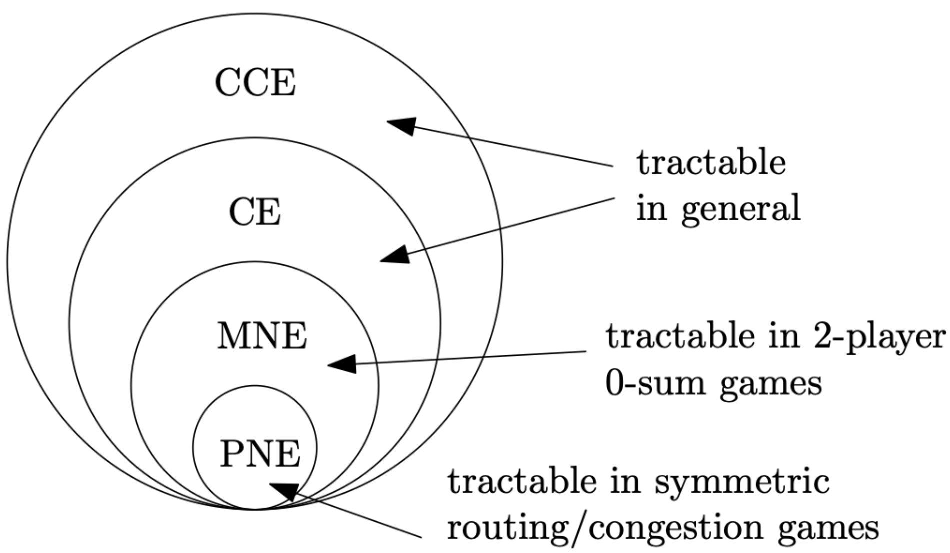
Four Positive Results (R 19.1.1)

No-regret dynamics for approximate coarse correlated equilibria

No-swap-regret dynamics for approximate correlated equilibria

No-regret dynamics for approximate mixed Nash equilibria for two-person, zero-sum

$\in$-best-response dynamics for approximate pure Nash equilibria for restricted atomic routing



Replacing dynamics (approximate) with algorithms (exact)

Linear programming?

Number of constraints

Behavior of solution methods

Lack of "guidance" as occurs with dynamics

Analogies to NP-Completeness (Garey & Johnson `https://www.amazon.com/dp/0716710455/`)

PLS-COMPLETENESS (R 19.2)

Aside: Parallel Computation (Greenlaw et.al.
`https://www.drraymondgreenlaw.com/research/PARALLEL/limits.pdf`)

(Highly) Parallelizable: NC - polylog time using polynomial processors

Cleverness 1: `https://en.wikipedia.org/wiki/Prefix_sum`

Cleverness 2: `https://en.wikipedia.org/wiki/List_ranking`

(`https://ranger.uta.edu/~weems/NOTES4351/09notes.pdf`)

Inherently Sequential: P-complete (maximum flow, linear programming)

"Maximum" Cut - Partition weighted undirected graph to maximize sum of edge weights across cut

NP-hard to achieve global maximum

"Maximal" cut - cannot be improved by moving one vertex, i.e. no local improvement exists

Components of any Abstract Local Search Problem - if each is in P, then problem is in PLS

1. Initial feasible solution.

2. Compute objective function for a feasible solution.

3. Detect termination or produce improvement.

The number of applications of (2.) is not required to be polynomial.

For a (new) problem in PLS to be PLS-complete (R Definition 19.1), there must be a polynomial-time reduction from every other (old) PLS problem.

1. Polynomial time algorithm to translate each instance of old problem to instance of new problem.

2. Polynomial time algorithm to translate each local optimum for instance of old problem to local optimum for instance of new problem.

Aside: The "first" PLS-complete problem (FLIP) is a general boolean circuit simulation that can capture the properties of *any* PLS problem. FLIP is more directly reducible to maximal cut. (D.S. Johnson et. al. "How Easy is Local Search", *JCSS* 37, 79-100, 1988)

PURE NASH EQUILIBRIA OF CONGESTION GAMES (R 19.3; N 19.3.4)

It is very straightforward to reduce maximum cut to a congestion game.

Potential function is the sum of the weights that do not cross the cut. (Dual minimization problem)

Aside: Other local search topics

https://en.wikipedia.org/wiki/3-opt

https://en.wikipedia.org/wiki/Gradient_descent

https://en.wikipedia.org/wiki/Simulated_annealing

https://en.wikipedia.org/wiki/Genetic_algorithm

5.B. MIXED NASH EQUILIBRIA OF BIMATRIX GAMES (R 20.1; N 2/3; `daskalakis.pdf`)

Bimatrix game

Write the two values of each matrix entry for general two-person game in separate matrices.

A = row player matrix

B = column player matrix

(B = -A for zero-sum game)

Wish to compute one mixed Nash equilibrium (MNE) strategy for each player:

$\hat{\mathbf{x}}^T A\hat{\mathbf{y}} \geq \mathbf{x}^T A\hat{\mathbf{y}}$ for all row mixed strategies $\mathbf{x}$ and

$\hat{\mathbf{x}}^T B\hat{\mathbf{y}} \geq \hat{\mathbf{x}}^T B\mathbf{y}$ for all column mixed strategies $\mathbf{y}$

What is a suitable complexity class for this problem (MNEBG)?

TOTAL NP SEARCH PROBLEMS (R 20.2)

NP = Decision problems for which a small "witness/certificate" may be *verified* in polynomial time (C.ACM Turing Award lectures: June 1983, Stephen Cook; February 1986, R. Karp; October 1994, J.Hartmanis; November 1994, R. Stearns)

coNP = Problems in NP have small certificates for "yes" instances and (presumably) large (exponential) certificates for "no" instances. Problems in coNP have small certificates for "no" instances and (presumably) large (exponential) certificates for "yes" instances. Satisfiability is in NP, Tautology is in coNP.

FNP (Functional NP) = outputs a certificate if one exists, otherwise indicates impossibility

PLS $\subset$ FNP (aside)

Theorem 20.1: MNEBG is not FNP-Complete unless NP = coNP.

Proof: Put together two "unlikely"algorithms:

1. Polynomial-time algorithm to map SAT formula to a bimatrix game.
2. Polynomial-time algorithm to map MNE to a satisfying assignment (if one exists) and "no" otherwise.

Existence of these would imply NP = coNP.

For an unsatisfiable input SAT formula, the intermediate MNE is a short (polynomial-length) certificate. The second algorithm would use this to obtain "no" in polynomial time.

The bigger picture: Nash showed that there is always an MNE.

TFNP (Total Functional NP) - a witness will always be output.

Theorem 20.2: No PLS problem is FNP-Complete unless NP = coNP.

(R 20.2.3 - Syntactic vs. Semantic Classes . . .)

Theorem 20.3: MNEBG is PPAD-Complete.

PPAD CONCEPTS (R 20.3)

Abstraction:

Directed graph

Generic path-following procedure

In- and out- degrees bounded by 1

Canonical source vertex

Sink vertex reachable from canonical source vertex

(Can have cycles. Can't have path crossing itself N p.36 Figure 2.2 . . . "rho" ρ `https://en.wikipedia.org/wiki/Cycle_detection` Chapter 2 of `http://elementsofprogramming.com` and its prequel `https://www.fm2gp.com`)

The Three Algorithms:

1. Returns canonical source vertex and its successor.
2. Given a vertex *x* other than those from (1.), returns the predecessor of *x* or "no predecessor".
3. Given a vertex *x* other than the canonical source vertex, returns the successor of *x* or "no successor".

CANONICAL PPAD PROBLEM: SPERNER'S LEMMA (R 20.4)

Theorem 20.4: (Sperner's Lemma) For every legal coloring of a subdivided triangle, there is an odd number of trichromatic triangles.

Proof: The trap-door walk of Notes 3.1 p. 9 finds one of these triangles. Roughgarden covers this by invoking the "Handshaking Lemma". (`https://en.wikipedia.org/wiki/Handshaking_lemma#Other_applications`)

MIXED NASH EQUILIBRIA AND THE CLASS PPAD (R 20.5)

Theorem 20.5: (Nash's Theorem) Every finite game has at least one Nash equilibrium.

Proof: `daskalakis.pdf` and KP 5.1

Folklore 1: Rational values in the bimatrix representation leads to rational values in the MNE strategy vectors.

Folklore 2: What fixpoints mean to many computer scientists: Lassez, J.-L. "Fixed Point Theorems and Semantics: A Folk Tale", *Information Processing Letters* 14 (2), May 16, 1982.

Lemke-Howson Algorithm (N 2.3)

N 2.3.1 Reduction to Symmetric Games

Theorem 2.4 *There is a polynomial reduction from* NASH *to* SYMMETRIC NASH.

$$C = \begin{pmatrix} 0 & A \\ B^T & 0 \end{pmatrix}$$

N 2.3.2 Pivoting on Supports

Convex polytope

Nondegeneracy w.r.t. constraints

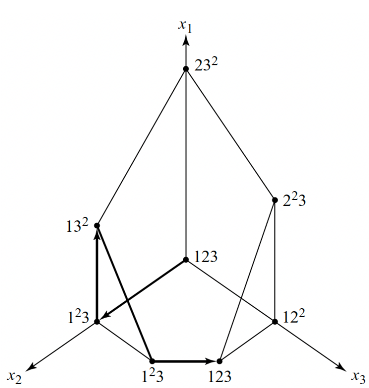
Strategy being *represented* at a vertex

Definition of vector

A vertex (besides the all-zero starting vertex) with all strategies represented is a Nash equilibrium

Best response condition N p. 55

Figure 2.1



N p. 30

$$A = \begin{pmatrix} 0 & 3 & 0 \\ 0 & 0 & 3 \\ 2 & 2 & 2 \end{pmatrix} \qquad B = A^{\mathrm{T}}$$

(0, 2/3, 1/3; 1/3, 0, 2/3) gives 2 as expected value

(1/3, 0, 2/3; 0, 2/3, 1/3) gives 2 as expected value

(0, 1/3, 2/3; 0, 1/3, 2/3) gives 2 as expected value

N p. 62-63

$$A = B^{\top} = \begin{bmatrix} 3 & 3 & 0 \\ 4 & 0 & 1 \\ 0 & 4 & 5 \end{bmatrix}$$

(1/2, 1/4, 1/4) gives 9/4 as expected value

(3/4, 1/4, 0) gives 3 as expected value

(0, 0, 1) gives 5 as expected value

`vonStengel.pdf` p. 8

$$A = \begin{bmatrix} 0 & 6 \\ 2 & 5 \\ 3 & 3 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 2 \\ 4 & 4 \end{bmatrix}$$

(2/3, 1/3, 0; 1/3, 2/3) gives 4 and 2/3 as expected values

(0, 0, 1; 1, 0) gives 3 and 4 as expected values

(0, 0, 1; 2/3, 1/3) gives 3 and 4 as expected values