CS6208 Lecture 2 Supplementary Material and Updates

August 20, 2025

1 Short clarifications

1. Q: In Lemke Howson, we said that there was a special action (the n-th one) that we used to define "almost-Nash" strategies. Does this mean that using a different action id could make us end up in a different Nash?

   Ans: Yes. In fact, the Lemke Howson algorithm can be seen as a special type of *homotopy* method. In such methods, we create a new game (say, G) with one special action as dominant action and then mixes this with the original game A giving

   $$\alpha \cdot A + (1 - \alpha) \cdot G$$

   When $\alpha = 0$ we get a dominant strategy corresponding to that distinguished action id. The idea is to see how the set of Nash evolve as we we shift gradually from $\alpha = 0$ to $\alpha = 1$, we get back Nash of the original game. (In practice, we cannot decrease α monotonically, but we can trace such a path; it can be shown that this is a connected set). The Lemke paths correspond to such a homotopy. The Lemeke Howson Algorithm is not the only homotopy type method, another is the one based on quantal response equilibria — this is Nash but using softmax instead of max for the best response; we can gradually reduce temperature to 0 which gives us Nash. See Herings and Peeters [2010] for an example.

2. Q: What happens when there are more than 2 players?

   Ans: generally, things get a bit messy. Yes, Nash's theorem holds so we will still get existence of equilibrium. But, it can be hard to find. Lemke's doesn't work. For instance, even in "nice games" like team games, there can be games where the only equilibria have irrational probabilities of being played. This sounds like a mere nuisance, but it means that it cannot be the direct solution of a linear or even integer program. I believe these are still algebraic numbers, i.e., solutions to polynomials, but still, things are tricky. Many multiplayer games of interest have much more special structure that make them easy to analyze, especially for certain types of *succinct games* e.g., routing games. See the AGT textbook Roughgarden [2010] for examples.

3. If you run our version of Lemke's algorithm and get a Nash, what happens if you "continue" to run it?

   Ans: What happens is that your wind up with the Nash you started with (if we started from the "fake" all-zero Nash, then we end up there again).

4. Q: Why are the graphs directed? Are the directions arbitrary?

   Ans: No, they are not arbitrary. The directions are what is known as an *index*, either +1 or -1, which dictates the direction. This is a topological concept that is defined in terms of signs of certain determinants. The index, as it turns out is closely related to the stability of the corresponding Nash (stability not in terms of player deviating, but whether these equilibria are "robust" to perturbations). It turns out that NE belong to the endpoints of Lemke paths, and one of them has to be +1 and the other one -1, which makes sense since the Lemke paths are directed. It turns out that $z = 0$ always has index -1, which means the edges point *away* from it, as we expect. The index is a rather deep topic and unfortunately nearing the boundary of my expertise. See von Schemde [2005] for a more complete account.

5. Q: Finding NE is PPAD complete. Are there approximation algorithms? What about parameterized complexity?

   Ans: I am not very familiar with parameterized complexity, but my general understanding is that there are special classes of games (e.g., graphical games) where find the NE can be bounded in terms of the treewidth of a certain graph. For approximations, we do not have a polynomial time approximation scheme PTAS or FPTAS, last I checked. But there are constant factor approximations, see Tsaknakis and Spirakis [2008] for an example.

6. Q: Solving integer programs are also NP hard, but Nash is only PPAD hard. Why do I recommend using Mixed Integer Linear Programming then?

   Ans: You are right, MILPs are hard to solve. But in practice, they are easy at least to implement, and allow for an objective. That makes it more convenient to prototype with.

2 The Lemke Howson Algorithm

In the lecture we gave a "reduced" version of NE for symmetric games under the assumption of nondegeneracy. We also glossed over two parts of the proof

2.1 Assumption about symmetric games.

We said that we didn't lose that much by restricting ourselves to symmetric games, assuming that our goal was to find *one* NE. Assume that A, B are strictly positive. Then if (A, B) are payoff matrices to the original game (not necessarily square), then consider the payoff matrix

$$A' = \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix}, B' = A'^T$$

such that (A', B') is now a symmetric game. I claim that

- There exists a symmetric equilibrium to (A', B') is of the form $[x', y'] \in \mathbb{R}^{n+m}$, where $x' \in \mathbb{R}^n$ and $y' \in \mathbb{R}^m$ and are both nonzero (may not sum to 1 though).
- The renormalized versions of x', y' given by $x^* = x'/||x'||_1$ and $y^* = y'/||y'||_1$ constitute a NE of the original game (A, B).

The existence of some symmetric equilibrium follows from Nash's or Kakutani's theorem (you can try proving it by hand). Now we want to show that both x', y' are nonzero. Suppose WLOG that $x' = 0$. That would mean that both players are only playing the "bottom part" of their actions, which gives them utility of zero.

$$[0, y']A'[0, y']^T = [0, y'] \begin{bmatrix} 0 & A \\ B^T & 0 \end{bmatrix} [0, y']^T = 0$$

and likewise for P2, $[0, y']B'[0, y']^T = 0$. But since A, B are strictly positive, either player can deviate to play the "top part" of their actions and do better. So having $x' = 0$ cannot constitute symmetric Nash. By symmetry it also holds that $y' \neq 0$. (Note that individual actions may be played with 0 probability, it just cannot be x' or y' are 0 for all entries. This in turns makes the renormalization in part 2 possible.

The second half of the theorem holds since if $(x'/||x'||_1, y'/||y'||_1)$ was not Nash in the original game, then WLOG player 1 can deviate from $x'/||x'||_1 \to x''$ and do strictly better. Thus it must be the case that one of the player s can go from $[x', y'] \to [x'' \cdot ||x'||_1, y']$ and do strictly better in the augmented symmetric game (A', B'). Note that $[x'' \cdot ||x'||_1, y']$ is still a valid probability distribution. This however violates the assumption that $[x', y']$ was a Nash to (A', B').

2.2 A non-zero z that is fully represented iff it corresponds to a Nash after renormalization

Let A be the matrix in the symmetric game. Since it is nondegenerate and fully represented, for every action i either $z_i = 0$ or $(Az)_i = 1$ but not both. Let $x = z'/||z'||_1$, the renormalized vector and $L = x^T A x = z^T A z/||z||_1^2 = (\sum_i z_i \cdot (Az)_i)/||z||_1^2 = ||z||_1/||z||_1^2 = 1/||z||_1$. Then for every action $x_i = 0$ or $(Ax)_i = 1/||z||_1$. This satisfies the best response condition and hence is a Nash in the original game.

Conversely, suppose (x^*, x^*) is a symmetric Nash in the original game. Then, dividing x^* by $L = {x^*}^T A x^*$ and using the best response condition gives the z we desire that is fully represented (and nonzero).

References

P Jean-Jacques Herings and Ronald Peeters. Homotopy methods to compute equilibria in game theory. *Economic Theory*, 42(1):119–156, 2010.

Tim Roughgarden. Algorithmic game theory. *Communications of the ACM*, 53(7): 78–86, 2010.

Haralampos Tsaknakis and Paul G Spirakis. An optimization approach for approximate nash equilibria. *Internet Mathematics*, 5(4):365–382, 2008.

Arndt von Schemde. *Index and Stability in Bimatrix Games: A Geometric-Combinatorial Approach*. Springer, 2005.