CS 6840 Algorithmic Game Theory March 4, 2020

Lecture 17: Complexity of Finding Equilibria

Instructor: Eva Tardos *Scribes: Eugene Kim, Lucas Van Bramer*

Recap

Reminder: Nonatomic Routing

When we did nonatomic routing, where $c_e(x)$ is monotonically increasing, we saw that we could achieve Nash by minimizing the potential function $\Phi(f) = \sum_e \int_0^{f(e)} c_e(x)dx$. We also know that this minimum is unique since this function is convex due to the assumed monotonic-increasing cost function. We can use non-linear optimization to calculate this minimum in polynomial time.

What if we have a discrete, or atomic, routing game? Then the potential function is $\Phi(f) = \sum_e \sum_{k=1}^{f(e)} c_e(k)$. Nash are local minima of this function, but discrete convex functions on integers are not easily optimized.

Atomic Routing and the 3-SAT Game

Refresher on 3-SAT

In 3-SAT, we have $x_1, x_2, ..., x_n$ as boolean variables. We can set each x_i to either true or false. Our goal is to make formula ϕ true where ϕ is the conjunction of several clauses, each consisting of the disjunction of 3 boolean variables: for example, $(x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_2 \vee \overline{x_1} \vee x_3) \wedge ....$ Each variable can appear negated or not negated in any clause.

Setup

Player i controls variable x_i and can pick two strategies: $x_i = T$ or $x_i = F$. Whichever strategy i chooses, the congestion occurs on any clause where this assignment causes the variable (or its negation, depending on what literal is in the clause) to evaluate false. So if i picks x_i to be true, then if $\overline{x_i}$ appears in a clause, i congests that clause since $\overline{x_i}$ evaluates to false there. Given this view, we can think of i's strategy as choosing between two sets:

$$\begin{cases} A_i = \text{the set of clauses containing } \overline{x_i} \\ B_i = \text{the set of clauses containing } x_i \end{cases}$$

given that i will end up congesting all clauses in either A_i or B_i, depending on their choice. Now we need to determine a monotonically increasing cost function. We define cost in this game as

$$c_j(k) = \begin{cases} 0 & k = 0 \\ 0 & k = 1 \\ 0 & k = 2 \\ 1 & k = 3 \end{cases}$$

for clause j with k units of congestion (the number of variables evaluating to false in that clause). This cost function ensures that each player in a clause only feels pain once that clause evaluates to false.

Guiding Questions

Minimum Social Cost

If player i suffers a cost equal to the number of unsatisfied clauses in ϕ that contain either x_i or $\bar{x_i}$, then the Social Cost is equal to $3 \times$ (the number of unsatisfied clauses) given that each unsatisfied clause in ϕ makes 3 players suffer that cost. Thus, minimizing SC is NP-hard because it contains 3-SAT given that, if you could minimize SC and achieve $SC = 0$, you would solve 3-SAT. Note that minimizing SC is NP-hard, but not NP-complete because it is not a decision problem.

Minimum Social Cost Nash Equilibrium

This is the same problem as above; it is Nash Equilibrium for all players to be suffering cost 0, which is the same as the Social Optimum. Thus, finding the minimum Social Cost Nash also contains 3-SAT.

Minimizing Potential Function Φ

Potential function Φ is just going to be equal to the number of unsatisfied clauses in ϕ:

$$\Phi = \sum_{e} \sum_{k=1}^{f(e)} c_e(k) = \sum_{e\ unsat} c_e(1) + c_e(2) + c_e(3) = \sum_{e\ unsat} 0 + 0 + 1$$

Thus, minimizing Φ yet again contains 3-SAT.

The max 3-SAT game

The max 3-SAT game is as follows: given a formula ϕ, satisfy as many clauses in it as you can. Note that this also contains 3-SAT.

A local optimum of this game occurs when changing the value of one variable x_i from T to F or F to T does not increase the number of satisfied clauses. We then ask: can we find a local optimum (Nash equilibrium)?

If ϕ is explicitly given, then yes, this is doable by flipping variable's "strategies" (values) iteratively and checking the number of satisfied clauses after each flip until we reach a local optimum. Note that PLS is the class of local optimization problems, and routing with linear costs is complete for PLS. The proof for this is complex and not included in lecture.

NP-Completeness

We play the same 3-SAT game, but there is an additional player x_{n+1} who has strategies a and b available, and ask: is there a Nash in which player x_{n+1} plays a? (yes or no decision)

Claim: This game is NP-complete.

Proof:

Let $c_i(s)$ for $i \in [1...n]$ be the same as in the original 3-SAT game.

Let $c_{n+1}(s)$ be defined as follows:

$$c_{n+1}(s) = \begin{cases} 0 & \text{if } s_1...s_n \text{ satisfies } \phi \\ 0 & \text{if } s_{n+1} \neq a \\ 1 & \text{else (when } s_{n+1} = a \text{ and } s_1...s_n \text{ does not satisfy } \phi) \end{cases}$$

If the 3-SAT instance ϕ is solvable, the first n players solve ϕ and x_{n+1} can play a with 0 cost, so there exists a Nash in which x_{n+1} plays a. If it is not solvable, then it is advantageous for x_{n+1} to unilaterally switch away from a, so a Nash in which x_{n+1} plays a does not exist. Thus, the question of whether there exists a Nash equilibrium in which player x_{n+1} plays a is equal to 3-SAT. 3-SAT is, of course, NP-complete. Thus, this game is NP-complete.