CS 684 Algorithmic Game Theory	Scribe: Selin Erdogan
Instructor: Eva Tardos	Friday, October 14, 2005

In today's lecture we will talk about finding some sort of equilibria instead of evaluating the quality of the equilibria as we have done in the previous lectures. Specifically, we will talk about potential games and finding the deterministic Nash equilibria.

Reminder: In potential games, we have a potential function Φ which keeps track of the change in objective function value when one of the players changes his strategy while others keep their strategies.

Example: The selfish routing game is an example of the potential games. In this game, we are given a network and k users. Each user i wants to route 1 unit of flow from source s_i to sink t_i using a path P_i between the source and the sink. In any solution of the game, if we have x_e users on edge e, the delay suffered by the x_e users is given by the delay function $l_e(x_e)$. The delay function is usually assumed to be monotonically increasing which is a realistic assumption but this maybe removed.

In this routing game each user i wants to minimize his delay, i.e. $min \sum_{e \in P_i} l_e(x_e)$, and the potential function is equivalent to $\Phi = \sum_{e \in E} \sum_{k=1}^{x_e} l_e(k)$.

Today we will provide a generalization of this game:

General Discrete Potential Game (Congestion Game):

In a General Discrete Potential Game, we are given k users (1,2,..,k), a ground set E, and delay functions $l_e(x)$ for each element e in E where x denotes the number of users using e. Each user i is associated with a set S_i, which denotes the set of allowed subsets of E for user i, and chooses an action P_i from set S_i. Like the routing game, x_e denotes the number of users i s.t. $e \in P_i$, each user i suffers delay $\sum_{e \in P_i} l_e(x_e)$, and the potential function is equal to $\Phi = \sum_{e \in E} \sum_{k=1}^{x_e} l_e(k)$.

Note that the General Discrete Potential Game in which we have actions from a discrete set (instead of paths) is in perfect analogy with the routing game. The network structure and the paths of the routing game are not relevant to the game structure and the general discrete potential game is also a potential game.

Since the delay of each user on an edge depends on the number of other users using the same edge, general discrete potential game is also referred as the Congestion Game.

Theorem 1 (Monderer and Shapley, 1997) *All discrete potential games are of the form given above, i.e. they coincide with the congestion games introduced by Rosenthal.*

Recall that in a potential game, a solution which minimizes Φ is a Nash equilibrium. A natural question arises at this point: Can we find a solution which minimizes the potential function Φ?

Special case: If all users have the same source-sink pair then we can find the solution which minimizes the potential function in polynomial time using the minimum cost flow algorithm. In order to use this algorithm, we replace each $e \in E$ with k copies with capacity 1 and cost $l_e(i)$ for i^{th} copy. So the first copy has the delay on the edge when there is one user using the edge e, second has the delay when there are two users, and so on...

Theorem 2 *Minimum cost flow of value k is equal to the set of paths that minimizes Φ (when the delay function l_e is monotone increasing).*

We needed that the delay function is increasing, as the minimum cost flow prefers to use the cheapest copy of each edge, and with the monotone delay function, the first copies are always cheaper.

Note that, this is a very special case. For most of the congestion games, minimizing Φ is NP-hard. (Since this is not the feasibility version we use the term NP-hard, instead of NP-complete.)

We must also be aware of the fact that previous paragraph states that finding the Nash that minimizes the potential function is NP-hard, but it does not give information about the difficulty of finding any Nash. Actually, finding any Nash solution in polynomial time for the congestion game is still an open problem. However, we have some information about the difficulty level:

Theorem 3 (Fabrikant, Papadimitriou, and Talwar, 1997) *Finding Nash in general congestion game is PLS-complete.*

***PLS*-Complete(Polynomial Local Search-Complete):**

A local search problem L is defined by an optimization problem, such as the traveling salesman problem, or max-3SAT where we want to satisfy the maximum number of clauses in a 3SAT formula. The goal of the problem is NOT to find the true optimum, but rather to find a local optimum. To make this meaningful, we need to also add to the problem definition a local move. For example, a natural local step for the 3SAT problem is to change the truth value of one variable. So if we change a variable x_i from true to false (or the other way around), this may make some clauses true that used to be false, and turn some other clauses false. The change is a local improvement, if the overall number of true clauses increased. In other words, the goal of the local search is to find a solution where a single switch of variable will not improve the number of satisfied clauses.

More generally, the class PLS (Polynomial Local Search) is defined by an optimization problem, and a local step. A local optimum is a solution with no improving local step, and the problem is to find a local optimum. We require that given a solution, we can decide in polynomial time if there is a local steps that improves the solution. In the case of the 3SAT problem defined above, this is easy to do: simply try swapping the truth value of each variable in turn, and see if it improves the total. In this local search problem, there are only a polynomial number of possible local steps (one for each variable), and hence we can try them all. Other local search problems can allow many more possible local steps, even exponentially many. The only requirement is that an improving local step can be found in polynomial time, if one exists. For example, in disjoint path routing problems, a local step would be to change one of the paths, and we may use a shortest path subroutine, to see if an improving path exists.

First note that the problem of finding a Nash equilibrium in a potential game is in PLS, assuming the problem the users face in finding their best move given the strategies of all other players in polynomial time solvable. For example, finding a Nash in the routing problem is in PLS (with the shortest path subroutine needed to find improving local steps). This is easy to see, as we know that local optima of the potential function Φ are exactly the Nash equilibria.

A problem $L \in PLS$ is PLS-complete, if every $\pi \in PLS$ is reducible to L. We have to be a bit careful to formally define what a reduction means here (how does it relate to what is a local step?) We will not go into a formal definition here.

Now note that a local optimum in the max-3SAT problem just defined above is easy to find in polynomial time: start with any solution and just takes repeated improving steps. Each step increases the number of satisfied variables. So in at most m steps (for 3SAT formulas with m clauses) it will terminate. So the procedure runs in polynomial time. The analog of this procedure won't work so well in our routing problem as the potential function Φ can take exponentially many different values. A similar problem arises in the weighted version of the 3-SAT problem. We also give each clause a weight (which can be a large number), and the problem is to maximize the sum of the weights of the satisfied clauses. In this problem, a sequence of local improvements can go on for more than polynomial time, as the total weight can increase very slowly.

The local search problem for the weighted version of the max 3-SAT problem, which aims to maximize the total weight of the clauses that are satisfied in an instance of 3-SAT problem, is PLS-Complete when the local search step is to change the truth value of one variable each time.

The result of Fabricant-Papadimitriou-Talwar shows that the problem of finding a Nash in a general congestion game (with increases l_e functions) is PLS complete via a reduction from the weighted max-3SAT problem. In the reduction, they use a congestion problem where players correspond to variables in the 3SAT formula, and each player has only two strategies that correspond to setting the variable true or false. They also show via a much harder reduction that the problem of finding a Nash equilibrium in the routing game is also PLS complete.