Lecture 8

The Quality of Solutions

April 27, 2005
Lecturer: Anna Karlin
Notes: Chris Ré

8.1 Overview

So far we have been studying games that are motivated in Computer Science in areas like Ecommerce or Networks. We've seen 3 different types of equilbria, Dominant, Nash and Combined. We have seen existence and talked a little bit about their complexity. Another direction, and our topic in this lecture, is the quality of these solutions.

It seems that we are giving up some amount of efficiency for the sake of stability. By efficiency, we mean an abstract performance measure that could be instaniated with quantities like throughput or latency. One measure of this loss of efficiency is called the *cost of anarchy*, which was coined by Papadimitriou. In this definition, we compare the performance of the worst Nash equilibream with the performance of the optimal centralized solution. Another measure is called the *price of stability*, were we measure the ratio of the best Nash equilbream to the optimal centralized solution.

Remark. The existence of simple (reasonable) learning strategies that never converge to a Nash equilbream, makes these definitions a little bit unsatisfying.

We will begin by examining a resource allocation problem and generalize it to a network setting. As we go we will generalize the behavior of the participants so that we (hopefully) get closer to how real actors will behave.

8.2 The Setting

Here we consider the setting for the different games.

8.2.1 Resource Allocation

In this general situation there are consumers and producers who produce some single product. The i^{th} consumer is characterized by a value d_i, that represents the demand that consumer i has for the product. Additionally the consumer has a function $U_i : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ which is the utility to the i^{th} consumer of receiving x units of product.

The i^{th} producer has some cost to produce a unit of product $C_i : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$.

8.2.2 Single Link in a Network

There are n users interested in sharing a link with bandwidth B. Here the cost function is

$$c(s) \begin{cases} 0 & s \leq B \\ \infty & s > B \end{cases}$$

Our goal is efficient allocation, that is choosing a vector a such that $\max \sum_i^n U_i(a_i)$. This is subject to the constraints that $\sum a_i \leq B$ and $a_i \geq 0$. We will later refer to this program as **LP**.

8.3 Mechanisms

Here we consider first a centralized auction solution, the VCG (Vickrey-Clarke-Groves) method. We will see that this mechansim is *truthful*, that is participants do not have incentive to lie and that is social optimal. We will then look at distributed mechanisms, such as Kelly's mechanism and the Johari-Tsitsiklis' mechanism. These two mechanisms achieve varying bounds on the cost of anarchy under differing models of how players will play.

8.3.1 Run an auction

Run VCG mechanism, choose d^* optimally. First each user submits U_i - the entire function. Let $\widetilde{a_{ij}}$ be the j^{th} component of the solution without i playing. Then we set U_i's payment

$$p_i = U_i(a_i^*) - (\sum_i U_i(a_i^*) - \sum_{j \neq i} U_i(\widetilde{a_{ij}}))$$

The key property of this game is that it is truthful, that is it is a dominant strategy to be truthful. To see this, let's first look at an example. Let the utility functions $U_i(a_i) = \begin{cases} v_i & a_i = 1 \\ 0 & \text{otherwise} \end{cases}$. Without loss, we can reorder the players so that $v_1 \geq v_2 \cdots \geq v_n$. Our allocation is 1 to the B highest bidders. Notice the i_{th} consumer pays

$$p_i = v_i - (\sum_i U_i(a_i^*) - \sum_{j \neq i} U_i(\widetilde{a_{ij}})) = v_i - (v_i - v_{B+1}) = v_{B+1}$$

Let us see why this mechanism is truthful in this example. Suppose it would have been better for the conusmer to bet $v_i' < v_i$. There is a scenario where v_i would have been one of the highest bidders but not v_i', so the player would get 0 utility. However in every situation where we bet v_i and still get the item with v_i' we pay the same amount. As a result, it is never better to bid less than the truthful amount.

Suppose $v_i' > v_i$, you should have bid more. If you get the item with both bids, then your price is independent of your bid, so you could not have done strictly better. So it can only be in the case that you get the item at v_i' but not at v_i. This means you pay some price p. Notice $p \geq v_i$, since p = v_{B+1} and you did not get the item with v_i meaning v_i was not strictly greater than this price. This means your utility is ≤ 0. So you are no better off with a higher bid.

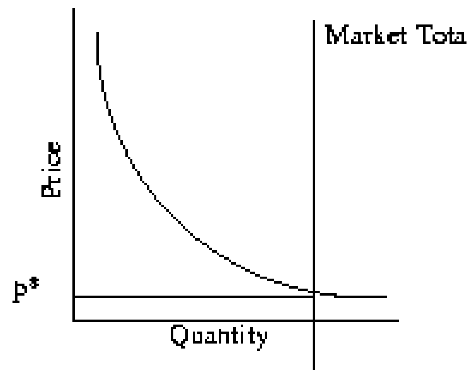

Figure 8.1: Market clearing Price

Claim 8.1. The mechanism is truthful

Proof. Suppose i should have submitted U_i' that leads to a new allocation y_i. Let the original allocation be called a and the allocation without player i be called z, $\widetilde{a_i j}$ above.

What is the price when i lied about U_i?

$$p_i' = \sum_{j \neq i} U_j(z_j) - \sum_{j \neq i} U_j(y_j)$$

What is the price when i is truthful?

$$p_i = \sum_{j \neq i} U_j(z_j) - \sum_{j \neq i} U_j(x_i)$$

Now suppose that this was better for player i. That is the real utility function of the new allocation minus the new price is better: $U_i(x_i) - p_i < U_i(y_i) - p_i'$.

Notice that the term depending only on z is the same so we can write:

$$U_i(x_i) + \sum_{j \neq i} U_j(x_j) < U_i(y_i) + \sum_{j \neq i} U_j(y_j) \Leftrightarrow \sum_j U_j(x_j) < \sum_j U_j(y_j)$$

This is a contradiction to the way the algorithm works, the x_j are supposed to attain a maximium value in their sum. $\square$

One problem with this algorithm is that it requires us to send our whole utility function to the centralized party. This setup may be undesriable, if the parties do not wish to share their entire function. This may also be infeasible, for example if the function is not known.

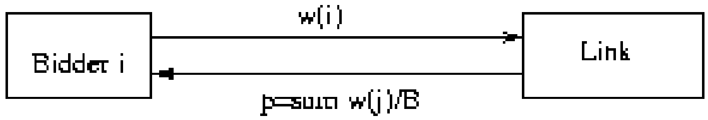

Figure 8.2: Kelly's Mechanism

8.3.2 Resource Allocation as a Market

We can view resource allocation as a market. Here let the u_i be continuous, increasing, differentiable, strictly convex - about as nice as can be. We are considering the situation when the prices are fixed for all participants.

This discussion relates to the fundemental theorem of welfare economics.

Definition 8.1. A *Market Clearing* means that all of the resource (bandwidth) are allocated.

Let $x_i(p)$ be the value of x_i that maximizes $U_i(x_i) - x_i * p$. This is roughly how much the player is willing to buy at price p. Let p^* be defined as the largest price such that the market clears. This is graphically represented in figure 7.1. Let $\boldsymbol{x}^* = x_i(p^*)$.

Theorem 8.1. *$\boldsymbol{x}^*$ optimizes **LP***

Proof. It is clear that the conditions of **LP** hold at $\boldsymbol{x}^*$. Let y be an optimal feasible solution. Now we notice term by term, $\sum_i(U_i(y_i) - y_i p^*) \leq \sum_i(U_i(x_i) - x_i p^*)$ because our mechanism maximizes this quantity. Since the market clears $\sum_i x_i p^* = Bp^*$. This means we can write, $\sum_i U_i(y_i) - Bp^* \leq \sum_i U_i(x_i) - Bp^*$, so $\sum_i U_i(y_i) \leq \sum_i U_i(x_i)$. Since y maximizes $\sum_i U_i(y_i)$, we conclude that $\sum_i U_i(y_i) = \sum_i U_i(x_i)$. □

8.3.3 Kelly's Mechanism

We now consider Kelly's mechanism depicted for our single link case in figure 7.2. Here, the bidder i offers a bid of w_i to the link. Each bidder is sent back $p = \frac{\sum w_j}{B}$. The player i gets $\frac{w_i}{\sum_j w_j} B$. Each bidder computes $\max v_i(\frac{w_i}{p} - w_i)$ at their next level.

Theorem 8.2. *Outcome of this mechanism is the same outcome.*

Proof. Next time by LaGrangian Duality of Convex Programs. □

Remark. Notice there is nothing on how quickly it converges to this solution.

8.3.4 Johari-Tsitsiklis

A strangeness in Kelly's process is that someone who wants more units does not take into consideration the effect of their bid on the price. Johari-Tsitsiklis considers when the user is anticipating the increase in cost of their own bid. Here they compute w_i to maximize $U_i(\frac{w_i B}{Bp - w_{old} + w_i}) - w_i$

Theorem 8.3.

$$\sum_i U_i(x_i) \geq \frac{3}{4} \max_{x_i} \sum_i U_i(x_i)$$