CS 684 Algorithmic Game Theory	Scribe: Lars Backstrom
Instructor: Éva Tardos	Monday, November 28, 2005

Truthful algorithms

So far, we've mostly looked at Nash equilibria in games. Here, we will examine something slightly different called truthful algorithms. In these algorithms, there are players, as before, and players typically have some function (like utility) that they trying to optimize. However, unlike the Nash framework, these algorithms will involve a coordinator. The players will report their utility functions to the coordinator, and the coordinator will then assign the stategies to the players. The coordinator must do the assignment in such a way that the players have no incentive to lie. That is, the players shouldn't benefit by reporting their utility functions as something other than they are. In addition to having the players tell the truth, the coordinator often tries to optimize some function (like social welfare).

For example, we looked at the bandwidth sharing game of Johan & Tsitsiklis, where each user had some utility function, $u_i(x)$, which gives the utility to player i for getting x bandwidth. In the version of the game we looked at, the players announced how much they were willing to pay, and bandwidth was then assigned to them relative to how much they paid. Player i then got a net utility of $u_i(x_i) - w_i$, the utility from the bandwidth, minus the cost of the bandwidth, and players played to optimize this.

A truthful algorithm for this game works in a fundamentally different way. The players simply announce their utility functions, and the coordinator then assigns bandwidth to the players. The problem, of course, is that if the coordinator naively assigns the bandwidth to the players who claim the highest utility, then the players will lie about their utility functions in order to get more bandwidth. The trick to the truthful algorithm is to do the assignment in such a way that the players have no incentive to lie. For instance, in the bandwidth sharing game, the coordinator might assign the bandwidth, but also charge some fee for it so that a player who lied about his utility function would be worse off than one who told the truth. Thus, the output from the coordinator would be:

- $x_1, \ldots, x_k$ bandwidth shares, and
- $w_1, \ldots, w_k$ costs for bandwidth

User i gets utility $u_i(x_i)$ from the bandwidth for which he pays w_i.

Hence, the goal is to design an algorithm for a game that gives players no incentive to lie. Or, another way of thinking about it is to design the algorithm such that telling the truth is a Nash equilibrium. One good thing about such algorithms is that the optimal strategy for the players is simple, they just report their utility function, independent of what any other players might do. This is also something of a drawback though, as in the real world, most people would have a hard time putting a function to their utility for something.

Examples

1) *Vickery auction.* Here, there is a single item on sale, like a rhinoceros. All k players have a utility u_i of owning the item, so they will have net positive utility if they pay less than u_i for it, and they are not willing to pay more than u_i. The players will all announce their utilities to the coordinator who will then let one player buy they item at some price x.

In a real auction, the person selling the item typically wants to maximize the sale price of the item. However, here we really just want to come up with a truthful algorithm where no one has an incentive to lie about his utility. Of course, one way to do this would be to randomly give the item to one person, but this is sort of silly, and we can in fact devise a more useful and meaningful truthful algorithm.

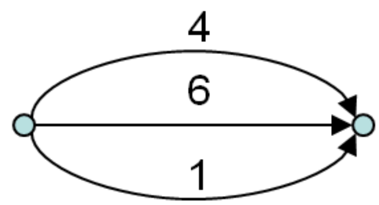
This algorithm is known as *second price auction.* User i, the user with the maximum utility is asked to pay $\max_{j \neq i} u_j$, i.e. the second highest price.

First off, this is in fact a truthful algorithm. The user with the maximum utility has no incentive to claim his utility is higher, since he would still pay the price of the second highest utility. He also has no incentive to claim a lower utility since doing so would either not effect the price he paid, or else cause him not to win. Since he has a positive net utility from winning, he clearly doesn't want to do this either. Finally, the non-winning players don't have any incentive to change their bids since lowering the bids won't change things for them, and raising their bids might cause them to pay more than the item is worth to them.

Furthermore, this algorithm has the nice property that it maximizes social welfare. User i benefits $u_i - x$ and the auctioneer benefits x, so the total welfare is simply u_i, and we picked the maximum u_i as the winner.

2) *Shortest path auction.* Consider the problem of trying to ship a car from Ithaca to China. There are many different companies, some of them regional, some of them global, who might ship the car part of the way. For instance, we might pay a train company to bring the car to a port, then train a different company to ship the car over seas, and then another train company to ship the car the rest of the way. Alternatively, we might imagine putting the car on the back of a truck for part of the journey, and paying someone to drive it somewhere for another part. In this problem, we are the coordinators of the game, and the shippers are the players who announce the price of shipping the car some distance. In a truthful algorithm, the shippers must tells us the true cost they incur from shipping the car, and then we must figure out a way to ship the car and pay the shippers in such a way that they have no incentive to tell us anything but the true price they incur.

More generally, we have some graph with a distinguished source, s, and target t. The edges in the graph correspond to the users, and user e announces the cost c_e of edge e. c_e is the true cost to the user to use the edge (so the user won't let the coordinator use the edge for less than c_e). The coordinator wants buy a path from the s to t (though not necessarily at the same costs that the players announce). To make this a truthful algorithm, the coordinator needs to buy the path in such a way that the users (edges) have no incentive to lie about the costs.



Special case: Here the source and target are connected by many parallel edges. This is analogous to the Vickery auction algorithm, except instead of selling, we are buying. Hence, instead of selling to the highest bidder at the second highest price, we will buy from the cheapest edge at the second lowest price. For all the same reasons, this is a truthful algorithm and it maximizes social welfare. In the above example, we would pay 4 for the edge of cost 1.

Extending the Vickery auction

Now, we would like to do to things:

- Extend the Vickery auction to the action of buying a general path.
- Buy the path that maximizes the social welfare – the path that has the lowest social cost, which in turn is the minimum cost path.

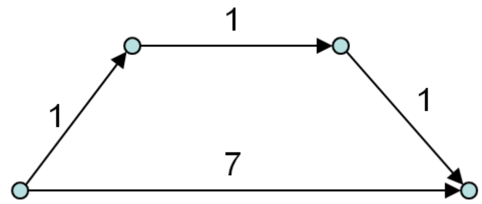
The General Approach:

- All players announce c_e (edge cost).
- We computer the shortest path
- We compute payments.

Note the differences between a truthful algorithm to buy a path and a Nash game. A Nash game might be to have players quote prices on the edges, and then buy the cheapest path, paying the quoted prices along the used edges, and paying 0 along unused edges. However, this version of the game is not truthful since players would clearly charge as high a price as they could and still sell their edges.

In a truthful algorithm, we will have to do something different to ensure that everyone tells us the true price. Given the prices that the users quote, we will find the shortest path using those quotes as edge weights, and buy those edges. Clearly, we will in fact find the path that maximizes the social welfare, since we'll be selected the shortest path. But, how can we do it to prevent users from lying about their costs.

Consider the graph below, where the edge weights are the true costs. The shortest path clearly has length 3, and so we will buy that one. But, we can't just pay each the true cost, or else the users would report higher costs to get paid more. In order to have a truthful algorithm, it turns out that we will have to pay each of the users 5. If we paid a user less than 5, he would have an incentive to report a higher cost, and we wouldn't have a truthful algorithm. With this in mind, we can now define a general payment rule that will give us a truthful algorithm for payments along a path P.



Define:

$$\begin{aligned} c_{\min} &= \text{cost of the shortest path in } G. \\ c^e_{\min} &= \text{cost of the shortest path in } G - e. \end{aligned}$$

Payment rule: Consider an edge e. If $e \notin P$ we pay 0. If $e \in P$, we pay $w_e = c_e + (c^e_{\min} - c_{\min})$

Notice that there was a cost to getting this truthful algorithm. We ended up paying 15 to buy the path where we certainly shouldn't have paid more than 7. In a first price auction, (the Nash game) the coordinator would not have had to pay more than 7.

Theorem 1 *This method is truthful and selects the shortest path.*

If an edge reports a lower cost than the true cost, it will either get paid the same, or else get paid less than the true cost, for a net loss. If an edge reports a higher cost, it will either get paid the same (c_e goes up by the same amount as $c_{\min}$), or else no longer be on the path, neither of which are good for that edge.

VCG mechanism

What we have just described is called the VCG (Vickery, Clarke, Groves) mechanism. In general, the VCG mechanism applies to any problem where we have we have users $1, \ldots, k$ with utility u_i (possibly a function). If S is set of possible outcomes of the problem then $u_i(s)$ for $s \in S$ is the utility for user i and outcome s.

We want to find the maximum social welfare:

$$\max_{s \in S} \sum_i u_i(s)$$

To do this, we will design a truthful algorithm where user i pays w_i, and naturally the user's goal is to maximize $u_i(s) - w_i$.

The VCG mechanism is a truthful algorithm that selects $s^* \in S$ such that $\sum_i u_i(s^*)$ is maximized and sets

$$w_i = u_i(s^*) - \left(\underbrace{\max_s \sum_j u_j(s) - \max_s \sum_{j \neq i} u_j(s)}_{\text{contribution to welfare by user } i} \right)$$

Note that no player has an incentive to give anything but his true utility function for all the same reasons we say in the examples above. Essentially, a user has to pay an amount equal to his utility, minus his contribution to the social welfare. If he reports a utility that is δ higher than the true utility, $u_i(s*)$ would go up by δ, and the first max in the expression would go up by at most δ, while the second max would stay the same. Hence, w_i would stay the same in the best case, and go up by δ in the worst case. Therefore, increasing your reported utility will never help you, and may hurt you by forcing you to pay a larger w_i.

A similar argument shows that reporting a lower $u_i(x)$ would not be advantageous since if you had postive contribution to the social welfare, reporting a lower utility would lower your contribution to the social welfare as much as it lowered $u_i(x_i)$, and hence w_i would remain unchanged. If your contribution to the social welfare was already 0, then when you lower your reported utility, your contribution will still be 0, and the coordinator essentially won't let you play in both cases.

Problems

- Computation: often can't find s that maximizes social welfare (may be NP complete, for instance).
- Socially unfair: it would be more fair to do a 1^{st} price auction.
- Not optimal for the coordinator. Had to pay 15 for the path, for example.