CS6840 - Algorithmic Game Theory (4 pages) Spring 2014

# March 28 - Greedy Algorithm as a Mechanism

*Instructor:Eva Tardos* Thodoris Lykouris (tl586)

**Main result**

This lecture is based on a result by Brendan Lucier and Alan Borodin [1]. The main result is the following:

**Theorem 1.** If a greedy algorithm is a $c$-approximation in the optimization version of the problem then, in the game-theoretic version of the problem, it derives a Price of Anarchy of at most $c$ with first price and $(c+1)$ with second price.

Before proving the result, we need to first understand what we really mean with this theorem.

**Framework of the optimization version**

- Set $S$ of items on sale.
- Each bidder $i \in [n]$ has value $v_i(A)$ for subset $A \subseteq S$.

The goal of the greedy algorithm is to maximize the social welfare:

$$\max_{\text{disjoint } A_1,\ldots,A_k \subseteq S} \sum v_i(A_i)$$

**Mechanism for the game-theoretic version**

- All users $i \in [n]$ declare a bid $b_i(A)$ for every subset $A \subseteq S$.
- We then run the previous algorithm to determine the allocation.
- For the pricing, we could have:
  1. If $i$ gets $A_i$, charge her $b_i(A_i)$ (first price)
  2. If $i$ gets $A_i$, charge her $\Theta_i(A_i)$ (second price), where $\Theta_i(A)$ will be defined later. Note that, in this case, we need an extra no overbidding assumption: $\forall i, A : b_i(A) \leq v_i(A)$.

**Greedy algorithm** We will consider the case that the greedy algorithm uses some function $f(i, A, v) \to \mathbb{R}$ to determine its next step in the allocation. This function $f$ should be monotone non-decreasing in the value $v$ for fixed $i, A$ and satisfy the property $\forall i, v, A \subseteq A' : f(i, A, v) \geq f(i, A', v)$.

The algorithm is the following: In decreasing order of $f(i, A, v_i(A))$ give $A$ to $i$ and remove $i$ from the game.

The latter (removing part) gives a unit-demand feature in the players and captures the fact that the valuation that some player has on the union of two sets is not the sum of their valuations. Hence, we are not allowed to assign him another set, once something is assigned to him as then the valuations are no more valid.

**Possible catches**

1. There is no assumption on the valuation function (monotonicity/submodularity) in the theorem. The reason why this is not a problem is hidden in the "if" statement. These assumptions guarantee the existence of a greedy algorithm in most settings. However, the theorem just takes care in transforming an approximation algorithm for the optimization version of the problem to a mechanism with decent Price of Anarchy to the game-theoretic version of the problem.
2. There is exponential amount of information. This is, as well, related to the greedy algorithm and not with the theorem. In fact, there exist greedy algorithms that behave well and fit in our framework. We will give some examples of this form.

**Examples**

1. The problem of finding a matching of maximum value has a very simple 2-approximation greedy algorithm (sorting edges by value and iteratively adding the edge with the maximum value among the edges that have unassigned adjacent vertices). This case behaves well as the number of items is small.
2. A case more close to our problem is when every player $i$ is interested in just one set $A_i$. By sorting them by $v_i$ or $\frac{v_i}{|A_i|}$, we get a $n$-approximation, which gets better if we sort by $\frac{v_i}{\sqrt{|A_i|}}$. This case behaves well as just few items have non-zero value.
3. The routing problem where there is a graph $G$ and some $\{s_i, t_i\}$ and we have value $v_i$ for any $(s_i - t_i)$ path. Although we might have an exponential numbers of possible paths/items, their values are given implicitly.

**$c$-approximation algorithm**

An algorithm is called a $c$-approximation for a maximization problem if the value of its solution is at least $\frac{1}{c}$ the value of the optimal solution.

**Second price**

Last but not least, we need to define what $\Theta_i(A)$ (used in second price auction) is. This corresponds to the critical price related to player $i$ and set $A$, i.e. the smallest price which would allow him to still win the set.

More formally, $\Theta_i(A)$ equals to the minimum bid that gets set $A$ to player $i$ when the algorithm favors $i$ in all ties. The latter is to avoid the need of bidding slightly above to strictly win the auction. The number depends on $b_{-i}$ but not in $b_i$.

**Proof of Theorem**

Suppose that $b$ is the bids' trajectory in Nash, which results in solution $A_1, \dots, A_n$ and *Opt* is the solution of disjoint sets $O_1, \dots O_n$ that maximizes $\sum_i v_i(O_i)$.

Suppose that $X_1, \dots, X_n$ is the allocation that maximizes $\sum_i b_i(X_i)$ (different from *Opt* as we are not maximizing on the real valuations but on the bids). It holds that $\sum_i b_i(O_i) \leq \sum_i b_i(X_i)$ (as *Opt* was among the possible allocations).

In addition, as the algorithm is $c$-approximation, it holds that $\sum_i b_i(X_i) \leq c \sum_i b_i(A_i)$.

Hence, we have the following inequality to which we will refer as (*):

$$\sum_i b_i(O_i) \leq c \sum_i b_i(A_i)$$

**Claim 2.**

$$\sum_i \Theta_i(O_i) \leq c \sum_i b_i(A_i)$$

*Proof.* Let the following bids:

$$b'_i(A) = \begin{cases} b_i(A) & \text{if } A \neq O_i \\ \Theta_i(A) - \epsilon & \text{else} \end{cases}$$

We define $b^*_i(A) = \max(b_i, b'_i)$. As a result, the outcome is not affected as, either:

- $A$ is in the winning set in which case it doesn't alter
- it keeps its value without being in the winning set
- it increases to slightly less than its critical value thus not getting in the winning set.

Applying (*) on $b^*$, using that $b'_i(A) \leq b^*_i(A)$ and taking $\epsilon \to 0$, the claim follows. □

We will continue the proof for the case of the second price (the case of the first price is similar).

$$b_i^*(A) = \begin{cases} v_i(A) & \text{if } A = O_i \\ 0 & \text{else} \end{cases}$$

As $b$ is Nash, we have $\forall i : u_i(b) \geq u_i(b_i^*, b_{-i})$. Furthermore, $u_i(b_i^*, b_{-i}) \geq v_i(O_i) - \Theta_i(O_i)$ as the right hand is negative in the case that $i$ has 0 utility and the inequality holds with equality from the definition of utility otherwise.

Combining the two inequalities and summing over all $i$, we have:

$$\sum_i u_i(b) \geq \sum_i (b_i^*, b_{-i}) \geq \sum_i v_i(O_i) - \sum_i \Theta_i(O_i) = OPT - \sum_i \Theta_i(O_i)$$

By the Claim, we have $\sum_i \Theta_i(O_i) \leq c \sum_i b_i(A_i)$ and, by the no overbidding assumption, $b_i(A_i) \leq v_i(A_i)$. Hence, it holds

$$\sum_i u_i(b) \geq OPT - \sum_i \Theta_i(O_i) \geq OPT - c \sum_i b_i(A_i) \geq OPT - c \sum_i v_i(A_i)$$

This inequality $\sum_i u_i(b) \geq OPT - c \sum_i v_i(A_i)$ is smoothness-like. Adding the prices on the left hand, we have:

$$\sum_i v_i(A_i) \geq OPT - c \sum_i v_i(A_i)$$

which results in a Price of Anarchy of at most $(c + 1)$.

**Open Questions** An interesting open question is to what extent the above technique can be extended to other (non-greedy) approximations. That is, when turned into games, can they generate good Price of Anarchy results?

## References

[1] B. Lucier and A. Borodin. Price of anarchy for greedy auctions. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 537–553, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.