Homework 5 - Programming report

Haukur Páll Jónsson
Game Theory

March 15, 2017

1 Problem setup

I wrote a program which computes the outcome for a combinatorial auction with single-minded bidders under the VCG mechanism. The most computationally hard part of the computation is finding the goods allocation which gives the maximum social welfare. Finding this allocation is a NP-hard problem, in particular it is a permutation of the set packing problem. Since there is no known efficient algorithm solving this problem my plan was to get an working naive implementation to see how one might tackle this problem. This was successful.

For simplicity sake I assumed that *free disposal* would be acceptable, since the randomness of the bundles (see later) would otherwise often lead to no solutions. Furthermore, I assumed that *single minded bidders* would only give a valuation to exactly the bundle they wish for, i.e. the supersets of the bundles receive valuation of 0. This allows the allocation solution to be represented as a binary string s.t. if in index i of the string is 1 then player's i bundle is in the allocation, otherwise it is not included. Thus the idea was to simply start from the binary representation of 0 (with padded 0's to the left) up to $2^|N|$ and find the maximum social welfare of any given allocation.

The price in a VCG mechanism is defined as the price player i offers minus the marginal contribution made by i. For player i compute $\sum_{j \neq i} \hat{v}(f(\hat{\boldsymbol{v}}_{-i})) - \sum_{j \neq i} \hat{v}(f(\hat{\boldsymbol{v}}_{i}))$. Thus, a new allocation, $f(\hat{\boldsymbol{v}}_{-i})$, needs to be computed per player to compute the price.

2 Implementation

A combinatorial auction is a tuple $G = (N, G, \boldsymbol{v})$, as defined in lectures. In this program a random game is generated given $|N|$ and $|G|$ and then $\boldsymbol{v}$ is created by assigning random integers from 1 to $|G|$ to each player. Using this randomization empty bundles are also given a valuation. I did not see this as a problem since it does not affect the solution but it clearly makes little sense in an auction setting. This valuation is the valuation for that player for her bundle B_i. Bundles were were created randomly s.t. a random integer from 0 to $2^{|G|^{|N|}}$ was drawn. If 0 would be drawn, all players will have empty bundles, if 1 is drawn, the last player gets good number 1 in her bundle, if the highest number is drawn, all players will have have all goods in their bundle. In this case the player with the highest valuation will win the auction and get all the goods.

To find the best social welfare allocation the program iterates from 0 to $2^{|}N|$ as mentioned previously. In each loop it first checks if this allocation is legal, by checking if the disjunction of all the bundles in this allocation is empty. Next it computes the outcome of this allocation and compare it to the previous best allocation and keeps the better one.

After the best allocation has been found it computes the price for each player as described above. Here I implemented $\hat{\boldsymbol{v}}_{-i}$ by setting the valuation of player i to 0. Thus, making player's i participation in the auction irrelevant. After this, a new allocation was calculated and the price was computed as described above.

The program then prints out the game definition; N, G, $\boldsymbol{v}$ and bundles. Next it prints out the best social welfare outcome, the allocation of that outcome and prices the players paid for that outcome.

3 Results

The efficiency of this program is not very good since the run time grows exponentially with number of players, $O(2^N)$. This can be seen in the following graphs in which different number of players and goods are tested and compared to each other with run time. When testing different N, G is set to 10 and similarly when testing different G, N is set to 10. The exponential time can be clearly seen in the graph. Adjusting G did not seem to affect the running time but linearly. This is probably due to the distribution of bundles, since given more goods, each player is simple expected to have more goods (bundles are typical) thus the probability of an intersection should remain the same and thus there will be similarly as many legal/illegal allocations as before.