CMSC 858F: Assignment 3
Due Date: Friday, April 18, 2014 before 4 pm

Instructor: MohammadTaghi Hajiaghayi

Thursday 24^{th} April, 2014

Please TYPE in your solutions after each problem and put your homework in my mailbox in the first floor of AV Williams. For definitions, please see slides, handwritten notes, and other course materials (or even Wikipedia).

Question 1 : Assume we want to sell k identical items to n bidders each wants one item with a value for the item. Design a polynomial-time mechanism which is truthful with maximum efficiency (social welfare) and charges each bidder a non-negative value (note that n can be smaller than k).

Proof : We use the VCG mechanism with Clarke Pivot Rule. This ensures the the mechanism is truthful and maximizes social welfare (see lecture notes). Let the values of the bidders be $b_1 \geq b_2 \geq \ldots \geq b_n$. If $k \geq n$, then everyone get an item for free. So, suppose $n > k$. Let A, B be the set of events including, not including bidder i respectively. Let a, b the optimal allocaions in A, B respectively. By the Clarke Pivor Rule, the payment of bidder i is given by

$$\begin{aligned} p_i &= \sum_{j \neq i} v_i(b) - \sum_{j \neq i} v_i(a) \\ &= (b_1 + b_2 + \ldots + b_{i-1} + b_{i+1} + \ldots + b_k + b_{k+1}) - (b_1 + b_2 \ldots + b_{i-1} + b_i + \ldots + b_k) \\ &= b_{k+1} \end{aligned}$$

So, the first k bidders are charged b_{k+1} each and all others pay 0. Since the values $\{b_1, b_2, \ldots, b_n\}$ can be sorted in polynomial time, the mechanism is polynomial time.

Question 2 : Assume we want to sell k items to n bidders each has different values for different items but only wants at most one item at the end. Design a polynomial-time mechanism which is truthful with maximum efficiency (social welfare) and charges each bidder a non-negative value (note that again n can be smaller than k).

Proof : This is similar to Question 1. Again we use the VCG mechanism with Clarke Pivot Rule. This ensures the the mechanism is truthful and maximizes social welfare (see lecture notes). Consider a bipartite graph G with the set of bidders B and the set of items I as the bipartitions. Since each bidder wants at most one item, we add an edge $\{b, i\}$ with weight equal to valuation of bidder b for item i. Now, an allocation corresponds to a weighted matching in G. For each $b \in B$, let G_b denote the graph $G \setminus \{b\}$ and M_b denote the weight of max weight matching in G_b. Furthermore, let M be the weight of max weight matching in G and ω_b denote the weight of edge incident on b in the max weight matching in G.

By the Clarke Pivot Rule, the payment of bidder b is given by

$$p_b = M_b - (M - \omega_b)$$

Since, we have to compute $(n + 1)$ weighted bipartite matchings, the mechanism runs in polynomial time.

Question 3 : (A) Assume an auctioneer has m items to sell where each has an unlimited supply. Also assume there are n single-minded bidders who want subsets of these items. Give a polynomial-time mechanism which only sets prices for items and has revenue at least $\Omega(\frac{1}{\log n+\log m})$-fraction of the optimum revenue.

Proof : This solution is from [5]. Bidder i values bundle S_i at v_i, and all other bundles at 0. For each $u \in [n]$ define $q_i = \frac{v_i}{|S_i|}$. We index the bidders so that $q_1 \geq q_2 \geq \ldots q_n$. Consider the algorithm which prices all items at q_i. Then a person j buys his bundle S_j if and only if $v_j \geq q_i \cdot |S_j|$, i.e., $q_j \geq q_i$. Hence, the profit of this algorithm is $R_i = \sum_{j=1}^{i} |S_j| \cdot \frac{v_i}{|S_i|}$. Run this algorithm for each i, and output the i with the maximum profit. Let us call this maximum profit as R. Then, we have

$$
\begin{aligned}
R &\leq \text{OPT} \\
&\leq \sum_{i=1}^{n} v_i && \text{Since } \sum_{i=1}^{n} v_i \text{ is a trivial upper bound on OPT} \\
&= \sum_{i=1}^{n} \frac{R_i \cdot |S_i|}{\sum_{j=1}^{i} |S_j|} && \text{Since } R_i = \sum_{j=1}^{i} |S_j| \cdot \frac{v_i}{|S_i|} \\
&= R \cdot \sum_{i=1}^{n} \frac{|S_i|}{\sum_{j=1}^{i} |S_j|} && \text{Since } R_i \leq R \text{ for each } i \in [R] \\
&\leq R \cdot \Big(\sum_{i=1}^{n} \sum_{k=1}^{|S_i|} \frac{1}{k + \sum_{j=1}^{i-1} |S_j|} \Big) && \text{Since } |S_i| \geq k \text{ for each } 1 \leq k \leq |S_i| \\
&\leq R \cdot \sum_{i=1}^{\sum_{j=1}^{n} |S_j|} \frac{1}{i} && \text{Rearranging} \\
&\leq R \cdot O(\log(\sum_{j=1}^{n} |S_j|)) && \text{since } H_n \leq O(\log n) \\
&\leq R \cdot O(\log n + \log m) && \text{since } \sum_{j=1}^{n} |S_j| \leq n \cdot m
\end{aligned}
$$

Question 3 : (B) In case each single-minded bidder wants a set of size at most 2, give a mechanism with revenue at least a constant fraction of the optimum revenue.

Proof :This proof is from [2]. We build a graph G with the vertex set as the set of items. The edge set is added as follows:

- For each bundle $\{i, j\}$ which is valued at value $w_{i,j}$ by a customer, we add an edge $\{i, j\}$ of weight $w_{i,j}$.
- For each bundle $\{i\}$ which is valued at value w_i by a customer, we add a self-loop $\{i, i\}$ of weight w_i

First, we give a simple 2-approximation for the case when G is bipartite. Specically, consider the optimal price-vector p^*, and let OPT_L be the amount of money it makes from nodes on the left, and OPT_R be the amount it makes from nodes on the right.Thus, $\mathrm{OPT} = \mathrm{OPT}_L + \mathrm{OPT}_R$. Notice that if one takes p^* and zeroes out all prices for nodes on the right, then this has profit at least OPT_L since all previous buyers still buy (and some new ones may too). Therefore, we can algorithmically make profit at least OPT_L by setting all prices on the right to 0, and then separately fixing prices for each node on the left so as to make the most money possible on each node. This makes the optimal profit subject to all nodes on the right having price 0 because no edges have two distinct endpoints on the left and so the profit made from some node i on the left does not affect the optimal price for some other node j on the left. Similarly we can make at least OPT_R by setting prices on the left to 0 and optimizing prices of nodes on the right. So, taking the best of both options, we make $\max\{\mathrm{OPT}_L, \mathrm{OPT}_R\} \geq \dfrac{\mathrm{OPT}}{2}$.

Now we consider the general (non-bipartite) case. Define opt_e to be the amount of profit that OPT makes from edge e. We will think of opt_e as the weight of edge e, though it is unknown to our algorithm. Let E_2 be the subset of edges that go between two distinct vertices, and let E_1 be the set of self-loops. Let OPT_1 be the profit made by p^* on edges in E_1 and let OPT_2 be the profit made by p^* on edges in E_2, so $\sum_{e \in E_i} \mathrm{opt}_e = \mathrm{OPT}_i$ for $i = 1, 2$ and $\mathrm{OPT}_1 + \mathrm{OPT}_2 = \mathrm{OPT}$. Randomly partition the vertices into two sets L and R. Since each edge $e \in E_2$ has a $1/2$ chance of having its endpoints on different sides, in expectation $\frac{\mathrm{OPT}_2}{2}$ weight is on edges with one endpoint in L and one endpoint in R. Thus, if we simply ignore edges in E_2 whose endpoints are on the same side and run the algorithm for the bipartite case, the profit we make in expectation is at least $\frac{1}{2} \cdot \left(\mathrm{OPT}_1 + \frac{\mathrm{OPT}_2}{2}\right) \geq \frac{\mathrm{OPT}}{4}$. This algorithm can be derandomized using standard methods.

Question 4 : In the UNIQUE COVERAGE problem, given a universe U of n elements, a collection $\mathcal{S}$ of m subsets of U, we want to find a sub-collection $\mathcal{S}$ which maximizes the number of elements that are uniquely covered, i.e., appear in exactly one set of $\mathcal{S}$. Assuming there is no $\Omega(\log n)$-approximation algorithm for the unique coverage problem, prove that we cannot approximate the problem in Question 3(A) by a factor better than $\Omega(\log n)$ by a transformation of Unique Coverage to the problem in Question 3(A) where n is the number of buyers (and thus the algorithm that you designed for question 3(A) is essentially tight).

Proof : This solution is from [3]. The UNIQUE COVERAGE problem is defined as follows:

> **Unique Coverage**
> **Input**: An universe $U = \{e_1, e_2, \ldots, e_n\}$ of elements, and a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of subsets of U
> **Objective:** Find a sub-collection $\mathcal{S} \subseteq \mathcal{S}$ which maximizes the number of elements that are uniquely covered, i.e., appear in exactly one set of $\mathcal{S}$.

Consider an instance of UNIQUE COVERAGE given by $(U, \mathcal{S})$. We build an instance of the problem from Quesion 3(a) as follows:

- Each element e_i maps to a buyer b_i
- Each set S_j maps to an item I_j
- For each $i \in [n]$, the buyer b_i has valuation 1 for the bundle given by $B_i = \{I_j : j \in [m], e_i \in S_j\}$

Because all valuations are either 0 or 1, we can assume the prices are from the range $(0, 1)$. The next lemma uses *randomized rounding* to show that we can assume that the prices are from the set $\{0, 1\}$ by paying a constant factor loss in the total profit.

> **Integral Prices at a Cost of Constant Factor in Profit**
> **Lemma**: There is a price assignment that uses prices from $\{0, 1\}$, and whose prot is within a constant factor of optimal.
> **Proof:** Consider the optimal assignment of prices p_i to items I_i. If any price p_i is larger than 1, we set it to 1 at no cost. Now we round by setting the new price p_i' of item I_i to 1 with probability $\frac{p_i}{2}$, and to 0 otherwise. We claim that if $u_i = \sum_{I_j \in B_i} p_j < 1$ (i.e., the optimal solution gets a prot u_i from buyer b_i), then the probability that the seller prots 1 from buyer b_i is at least $\frac{u_i}{2e}$.
> The probability that the seller prots 1 from buyer b_i who desires bundle B_i, is $\sum_{I_j \in B_i} \frac{p_j}{2} \pi_{I_j \neq I_k \in B_i} (1 - \frac{p_k}{2}) = \pi_{I_k \in B_i} (1 - \frac{p_k}{2}) \sum_{I_j \in B_i} \frac{\frac{p_j}{2}}{1 - \frac{p_j}{2}}$. Because $\sum_{I_j \in B_i} p_j \leq \frac{1}{2}$, it is easy to see that the quantity is minimized when all of the p_j's are equal for $I_j \in B_i$. Thus the probability of profit from b_i is at least $(1 \frac{u_i}{2|Bi|})^{|Bi|} \cdot \frac{\frac{u_i}{2}}{1 \frac{u_i |B_i|}{2}}$. Because $1 - x \geq e^{2x}$ for $0 \leq x \leq \frac{1}{2}$, this probability is at least $e^{u_i} \cdot \frac{u_i}{2} \geq \frac{u_i}{2e}$, as claimed.
> Thus the expected total prot in the modied solution is at least $\sum_i \frac{u_i}{2e}$, which is $\frac{1}{2e}$ times the prot of the optimal solution. This algorithm can be derandomized using the method of conditional expectation. □

Now, each buyer b_i will buy its bundle precisely if at most one item is priced at 1, and the rest of the items are priced at 0. If all items in a bundle are priced at 0, then the seller makes no prot. I exactly one item is priced at 1 and the rest are priced at 0, then the seller prots by 1. Thus the effective goal is to assign prices of 0 or 1 in order to maximize the number of bundles for which exactly one item is priced at 1, which is identical to the original UNIQUE COVERAGE problem.

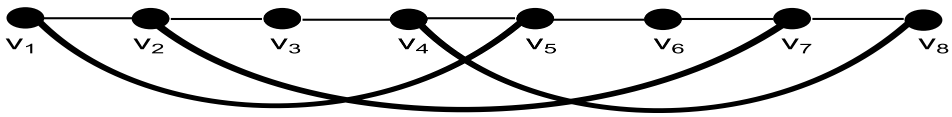

Figure 1: A diameter-3 swap equilibrium graph G

Question 5 : (A) In a network creation game, a swap-equilibria is a connected unweighted graph in which each vertex v is stable, i.e., if vertex v swaps one of its neighboring edges $\{v, u\}$ with another non-neighboring edge $\{v, w\}$, its sum of distances to all nodes does NOT decrease. Prove there is a graph of at most 9 vertices with diameter 3 (there was a conjecture that all swap-equilibria have diameter at most 2, i.e., a star with some extra edges; here you disprove this conjecture). **Hint**: you may use a computer program.

Proof : Note that Figure 3 from [1] is **not** a swap-equilibrium graph: If d_1 replaces the edge $\{d_1, c_{1,1}\}$ with the edge $\{d_1, c_{2,1}\}$, then the total distance for d_1 decreases from 27 to 26.

In fact, we will show that the graph G shown in Figure 1 is a swap-equilibrium graph. It is easy to see that G has diameter 3. We first define *local diameter*: consider a vertex $v \in G = (V, E)$. Let $d_G(x, y)$ denote the distance between x and y in G, i.e., the length for shortest $x - y$ path in G. Then we define local diameter of v to be equal to $\max_{w \neq v} d(v, w)$.

The following lemma shows that we do not need to worry about swaps by vertices which local diameter 2:

Vertices of local diameter cannot decrease by swapping
Lemma: Let G be a graph of diameter 3. Any vertex v which has local diameter 2 cannot decrease its sum by swapping any edge.
Proof: Let v be a vertex of local diameter 2 in G. Note that any swap keeps constant the number of vertices which are at a distance 1 from v. Thus, the number of vertices at distance ≥ 2 from v remains also the same. Therefore, it is optimal to keep the distance 2 for all these vertices, i.e., v cannot decrease its distance with any swap. $\square$

In Figure 1, the vertices v_2, v_4, v_5, v_7 have local diameter 2. By above lemma, we do not need to worry about them. Check manually for each swap for each of the vertices v_1, v_3, v_6, v_8 (by symmetry it is enough to check only for v_1 and v_3).

Question 5 : (B)**(Bonus problem)**. Can you give a swap-equilibria with diameter 4 or higher?

Proof : OPEN !!

Question 6 : **(Adword Auction with Free Disposal)**. In an Adword auction, adwords are coming one by one in an online manner and we should assign each coming adword to one bidder each with different bids for different adwords and collect his bid as our revenue (first-price auction). We assume that bidder i, wants at most C_i adwords, i.e., its capacity is C_i.

- First, if $C_i = \infty$ for all i, then present the optimum algorithm for the problem.
- Second, prove that in the worst case our revenue can be arbitrary lower that the revenue in the offline case that we know all adwords and all bids in advance.
- Third, to resolve the problem in the previous case, a free disposal assumption has been suggested in the literature. Here, if we already assigned an adword to a bidder and we used its capacity, we can still assign him to a new adword and do not charge the bidder for the previous adword (and thus we show an ad of the bidder for a previous adword for free.). In this case, propose a natural greedy algorithm for the problem and prove that it obtains at least half revenue of the best offline algorithm that knows all adwords and all bids for them in advance.

Proof :

- The algorithm is to assign each coming adword to its highest bidder. Since the capacities are infinite in this case, we can always assign the current adword to the highest bidder. Hence we make the most profit out of each adword, i.e., the algorithm is optimal.
- Consider the following example: There is a bidder u with capacity 1. For the first arriving adword, u has a bid equal to ϵ. Any algorithm would either assign the adword to u, or it would leave the adword. If the adword is assigned, the next adword would be such that the bidder has a bid equal to 1 for it. The offline algorithm would get a profit of 1 while we make a profit of ϵ. If the adword is unassigned, there would be no future adwords. This means we made 0 profit while the best offline algorithm would get a profit of ϵ. By changing ϵ, we can get arbitrarily low competitive ratio.
- This solution is from [4]. With free disposal, the primal is as follows:

$$\max_{i\in I, a\in A} w_{i,a}x_{i,a} \quad \text{such that} \sum_{a\in A} x_{i,a} \le 1 \;\forall\, i \in I \quad \text{and} \sum_{i\in I} x_{i,1} \le C_a \;\forall\, a\in A \tag{1}$$

The dual is the following

$$\min\Big(\sum_{a\in A} C_a\beta_a + \sum_{i\in I} z_i\Big) \quad \text{such that } \beta_a + z_i \ge w_{i,a} \;\forall\, i\in I, a\in A \quad \text{and } \beta_a, z_i \ge 0 \;\forall\, a\in A, i\in I \tag{2}$$

Let E be the edge set we have so far. For each $a \in A$, define

$$v(a) = \begin{cases} 0 & \text{if } |\{i : ia \in E\}| < C_a \\ \min_{i:ai\in E} w_{i,a} & \text{otherwise} \end{cases}$$

The online algorithm runs as follows: when an adword $i \in I$ arrives, we match it to a bidder a who maximizes the marginal value given by $\mu(a) = w_{i,a} - v(a)$ (leave i unassigned if the marginal value is negative for all $a \in A$). Initialize all the dual variables $_a$ and z_i to 0 for each bidder. Set $x_{i,a} = 1$. If a previously had C_a adwords assigned to it then we drop the weight of the least valuable of these, say i'. That is, we set $x_{i',a} = 0$. Increase β_a by the marginal value $\mu(a)$, i.e., $\beta_a^{\text{NEW}} = \beta_a^{\text{OLD}} + \mu(a) \ge v(a) + \mu(a)$. Set $z_i = w_{i,a} - \beta_a^{\text{NEW}}$. Clearly, the choice of z_i ensures that the dual remains feasible: since we always increase the β variables, the only constraint that needs to be checked is $\beta_a + z_i \ge w_{i,a}$. Now, we claim that the increase in dual objective is at most $2\mu(a)$, which is the increase in the value of the assignment. Combining this with weak duality (Dual is greater equal Primal) gives us that our algorithm is (1/2)-competitive.

Note that it is easy to see that β_a is greater equal to v_a at the end of each step (a step is when an adword arrives and is matched). Initially $\beta_a = 0$, and at each step it is increased by the marginal value. Hence, we have $\beta_a^{\text{NEW}} = \beta_a^{\text{OLD}} + \mu(a) \geq v(a) + \mu(a) = v(a) + (w_{i,a} - v(a)) = w_{i,a} \geq v_a^{\text{NEW}}$ by definition of $v(a)$. Therefore, we have increase in dual objective is $\mu(a) + (w_{i,a} - \beta_a^{\text{NEW}})$. To show that this is at most $2\mu(a)$, it suffices to show that $w_{i,a} - \beta_a^{\text{NEW}} \leq \mu(a) = w_{i,a} - v(a)$ which holds since $\beta_a^{\text{NEW}} \geq v_a^{NEW} \geq v_a$ (since the $v(a)$ values can only go up after each step).

Question 7 : A "good thief who has stolen 130-page thesis of an MIT student (the thesis was inside a knapsack which was stolen from an unattended students office with an open door) wants to return the thesis and instead surely gets 100$ bonus advertised by the student in several places all over the campus. Can you help the good thief to complete the transaction without any possibility that police can catch him? Assume police is reasonably strong (say your assumption about police explicitly) and if you use any cryptographic or complexity assumption please mention them explicitly as well (the story is a real story and the police apprehended the poor good thief ☹).

Proof :

Solution by Melika: The thief should call from a public phone to the student and tell him his bitcoin account plus the place he has hidden half of the thesis. When the student finds half of the thesis he knows the thief has the thesis and is not lying. Therefore, he would send him the money. Then the thief can make another call from the public phone to tell the student where he has hidden the rest of the thesis. Bitcoin is a peer-to-peer payment system introduced as open source software in 2009 by developer Satoshi Nakamoto. The digital currency created and used in the system is alternatively referred to as a virtual currency, electronic money, or a cryptocurrency although it does not meet the generally recognized definition of money. The bitcoin system is not controlled by a single entity, like a central bank, which has led the US Treasury to call bitcoin a decentralized currency. (wikipedia)

Solution by Reza: Here we use the fact that the item he has is splittable and the owner would like to receive all of his thesis. Therefore, the thief would put half of the thesis somewhere he only knows. Then, inform the owner its place by a fake email. Now, the owner knows that he is the true holder of the thesis without his real name to be revealed. Now the thief can ask half of the money to be transferred to an online unrecognizable account. The owner strategy is to give him half of the money and wait for the rest. Now the thief should put one forth of the thesis somewhere else and ask for one forth of the money and continue doing so until he can get arbitrarily near the value he desires.

References

[1] Alon, N., Demaine, E.D., Hajiaghayi, M.T., Leighton, T.: Basic network creation games. SIAM J. Discrete Math. 27(2), 656–668 (2013)

[2] Balcan, M.F., Blum, A.: Approximation algorithms and online mechanisms for item pricing. In: EC. pp. 29–35 (2006)

[3] Demaine, E.D., Feige, U., Hajiaghayi, M., Salavatipour, M.R.: Combination can be hard: Approximability of the unique coverage problem. SIAM J. Comput. 38(4), 1464–1483 (2008)

[4] Feldman, J., Korula, N., Mirrokni, V.S., Muthukrishnan, S., Pál, M.: Online ad assignment with free disposal. In: WINE. pp. 374–385 (2009)

[5] Guruswami, V., Hartline, J.D., Karlin, A.R., Kempe, D., Kenyon, C., McSherry, F.: On profit-maximizing envy-free pricing. In: SODA. pp. 1164–1173 (2005)