Optimization II **Winter 2009/10**

Lecture 1 — October 12

Lecturer: Julián Mestre

1.1 Maximum weight matching via auctions

In this lecture we cover an auction-based algorithm for maximum weight matching in bipartite graphs.

MAXIMUM WEIGHT BIPARTITE MATCHING (MWBM)

Input: bipartite graph (U, V, E) and edge weights $w : E \to \mathcal{R}^+$

Output: matching $M \subseteq E$

Objective: maximize $w(M)$

Let n be the total number of vertices ($|U| + |V|$) and m be the total number of edges ($|E|$). We assume, without loss of generality, that the graph is connected so $m \geq n - 1$. Our algorithm will regard one side of the bipartition as a set of *bidders* (U), and the other side as a set of objects (V). For each bidder-object pair $(i, j) \in E$ the number w_{ij} captures how much i values j.

Algorithm 1 AUCTION-MECHANISM(U, V, E, w, δ)

1. For each object $j \in V$, set $p_j \leftarrow 0$ and $owner_j = \perp$
2. $Q \leftarrow$ a queue containing all the bidders
3. **while** $Q \neq \emptyset$ **do**
4. $\quad i \leftarrow$ pop a bidder from Q
5. $\quad j \leftarrow$ an object maximizing $w_{ij} - p_j$
6. $\quad$ **if** $w_{ij} - p_j > 0$ **then**
7. $\quad\quad$ enqueue $owner_j$, if any, into Q
8. $\quad\quad owner_j \leftarrow i$
9. $\quad\quad p_i \leftarrow p_i + \delta$
10. **return** $\{(owner_j, j) \mid j \in V \text{ and } owner_j \neq \perp\}$

Notice that the algorithm is parametrized by δ. We will do the analysis for a generic δ and then choose a suitable value to get the desired result.

Definition 2. *A bidder i is δ-satisfied if one of the following is true:*

- *Bidder i owns some object $j \in V$ (i.e., $owner_j = i$) and for all objects $j' \in V$ we have $\delta + w_{ij} - p_j \geq w_{ij'} - p_{j'}$, or*

- *Bidder i does not own anything and for all $j \in V$ we have $w_{ij} \leq p_j$.*

Lemma 2.1. *At any point during the execution of the algorithm, if bidder i does not belong to the queue Q then bidder i is δ-satisfied.*

Lemma 2.2. *If all bidders are δ-satisfied then for every matching M' we have that*

$$|V|\,\delta \;+\; \sum_{(i,j)\,:\,owner_j=i} w_{ij} \;\geq\; w(M')$$

Lemma 2.3. *The number of iterations is at most $\frac{|V|w_{max}}{\delta} + |U|$, where $w_{max} = \max_{(i,j)\in E} w_{ij}$.*

Theorem 2.4. *If all the weights are integral then* AUCTION-MATCHING *with $\delta = \frac{1}{|V|+1}$ solves* MWBM *in $O\left(n^3 w_{max}\right)$ time.*

Proof: From Lemma 2.3 and our choice of δ we know that the algorithm runs for at most $O(n^2 w_{max})$ iterations. Each iteration can be implemented in $O(n)$ time, so the claimed running time follows.

Since all weights are integral, any two matching either have the same weight or the differ by at least 1 unit. From Lemma 2.2 and our choice of δ we know that the weight of any mother matching is no larger than the one the algorithm outputs. □

In fact, we can refine the analysis of the algorithm if we are more careful about the implementation details behind the it. The end result is a faster algorithm for sparse instances; that is, instances where $m = o(n^2)$.

Theorem 2.5. *If edge weights are integral then* AUCTION-MATCHING *with $\delta = \frac{1}{|V|+1}$ solves* MWBM *in $O(nmw_{max})$ time and $O(m)$ space.*

Proof (sketch): The idea is to keep for each bidder i a list of the objects j such that $(i,j) \in E$ sorted in decreasing value of $w_{ij} - p_j$. Given this information, when bidder i is dequeued, finding the object j maximizing $w_{ij} - p_j$ becomes trivial. After updating the price of j we may need to update the position of j in the list of every other bidder i' such that $(i', j) \in E$. All these updates can be carried out in $O(\deg_E(j))$ time. The price of an object can be updated at most w_{max}/δ and the theorem follows. □