These lecture notes have not undergone rigorous peer-review. Please email quanquan.liu@yale.edu if you see any errors.

1 Introduction

Today we'll discuss approximate maximum bipartite matching (MCM) algorithm of Assadi, Liu, and Tarjan [ALT21]. The maximum cardinality matching problem for bipartite graphs asks for a matching of maximum size. This paper gives a $(1-\varepsilon)$-approximate MCM in the streaming model after $O(1/\varepsilon^2)$ passes. The main techniques used in this paper are *auction algorithms* which are typically used for welfare-maximumizing assignment of items to bidders. In our setting, we let the set of notes on the left hand size of the bipartite graph be the set of bidders and the set of nodes on the right be the items. We denote these sets of nodes by L and R, respectively.

2 Streaming Auction-Based MCM

Initially bidders are *unallocated* and all items start with price 0. Prices for items are capped at 1. The algorithm proceeds in iterations where in each iteration, bidders bid on their lowest price items. Then, these items are allocated to the bidder who currently does not have an item and bid on an item. The pseudocode for this algorithm is given in Algorithm 1. To define the notation we will use, let a bidder be $i \in L$ and the valuation for i for items in R as the function $v_i : R \to \{0, 1\}$ where $v_i(j) = 1$ if $j \in N(i)$ and $v_i(j) = 0$, otherwise.

Theorem 1. *We define the* **utility** *of bidder i as $u_i := v_i(a_i) - p_{a_i}$ where a_i is the item allocated to i and p_{a_i} is the price of the item a_i allocated to i. An unallocated bidder i has $u_i = 0$.*

The number of iterations is trivially $O(1/\varepsilon^2)$ since we only run our algorithm for that many iterations.

Now we prove the approximation factor. Let M^* be a maximum matching of G and OPT $\subseteq L$ be the set of bidders in L that are matched by M^*. For any $i \in$ OPT, let $o_i \in R$ be the item allocated to i in M^*. We now define a *happy* bidder.

Theorem 2. *Bidder i is ε-happy if and only if $u_i \geq v_i(j) - p_j - \varepsilon$ for all $j \in N(i)$.*

In other words, a happy bidder i is one where changing the allocation of i to any other item does not increase the utility of I by more than ε.

We now show the following lemma.

Lemma 2.1. *In each iteration, allocated bidders and unallocated bidders with empty demand sets are ε-happy.*

Proof. First, each allocated bidder picked a minimum price item in its neighborhoods and increased the price of this item by only ε. Thus, the prices are monotonically non-decreasing and each bidder with an allocated item cannot change to another item and get an increase in utility by more than ε.

Then, every item in the empty demand set of a bidder's neighborhood has price 1. Hence, the statement trivially holds since $v_i(j) - p_j - \varepsilon \leq 0$ for all such bidders. $\square$

Now, we show that if a "large" number of bidders in OPT become ε-happy at any point in the auction, then the final matching gives a $(1-\varepsilon)$-approximation.

Let $\mu(G)$ be the size of OPT.

Lemma 2.2. *If at the end of some iteration,* $(1-\varepsilon)\cdot\mu(G)$ *bidders in* OPT *are* ε*-happy, then the final matching* M *has size at least* $(1-2\varepsilon)\cdot\mu(G)$.

Proof. Let $a_1,\ldots,a_n$ be the set of allocations and $p_1,\ldots,p_n$ be the prices of the items. Let $HAPPY$ be the set of ε-happy bidders. For any bidder $i \in HAPPY \cap \text{OPT}$, it holds that $u_i \geq v_i(o_i) - p_{o_i} - \varepsilon$.

Now, we sum over all ε-happy bidders:

$$\begin{aligned}
\sum_{i\in HAPPY} u_i &\geq \sum_{i\in HAPPY\cap\text{OPT}} u_i \quad \text{since } u_i \geq 0 \\
&\geq \sum_{i\in HAPPY\cap\text{OPT}} (v_i(o_i) - p_{o_i} - \varepsilon) \\
&\geq \sum_{i\in HAPPY\cap\text{OPT}} (v_i(o_i) - \varepsilon) - \sum_{i\in HAPPY\cap\text{OPT}} p_{o_i} \quad \text{by assumption } |HAPPY\cap\text{OPT}| \geq (1-\varepsilon)\mu(G) \\
&\geq (1-\varepsilon)\mu(G)(1-\varepsilon) - \sum_{i\in HAPPY\cap\text{OPT}} p_{o_i} \quad \text{since } v(o_i) = 1 \\
&\geq (1-2\varepsilon)\mu(G) - \sum_{i\in HAPPY\cap\text{OPT}} p_{o_i}.
\end{aligned}$$

Furthermore, we can lower bound the sum of the utilities as follows:

Algorithm 1: Auction-Based Algorithm for MCM

Input: A bipartite graph $G = (L, R, E)$, where L is the set of bidders, R is the set of items, and E is the set of edges
Output: A matching M of G

for *each bidder* $i \in L$ **do**
  set $a_i \leftarrow \perp$
end
for *each item* $j \in R$ **do**
  set $p_j \leftarrow 0$
end
for $r \leftarrow 1$ **to** $\lceil 2/\varepsilon^2 \rceil$ **do**
  for *each unallocated bidder* $i \in L$ **do**
    define $D_i \leftarrow \arg\min_{j\in N(i), p_j<1}(p_j)$ be the demand set of i
  end
  let G_r be the induced subgraph between unallocated bidders and demand set
  find a maximal matching M_r in G_r
  for *every bidder-item pair* $(i, j) \in M_r$ **do**
    reallocate j to i, set $a_i \leftarrow j$, and $a_{i'} \leftarrow \perp$ for previous owner i' of j
    increase price of j, $p_j \leftarrow p_j + \varepsilon$
  end
end
Return matching M as (i, a_i) for all $i \in L$ where $a_i \neq \perp$

$$\begin{aligned}
\sum_{i \in HAPPY} u_i &= \sum_{i \in HAPPY \cap a_i \neq \perp} (v_i(a_i) - p_{a_i}) \quad \text{unallocated bidder have } u_i = 0 \\
&\leq |M| - \sum_{i \in HAPPY \cap a_i \neq \perp} p_{a_i} \text{number of allocated bidders nondecreasing} \\
&= |M| - \sum_{j \in R} p_j \quad \text{only allocated items have } > 0 \text{ price.}
\end{aligned}$$

Combining both inequalities above gives us

$$\begin{aligned}
|M| - \sum_{j \in R} p_j \geq \sum_{i \in HAPPY} u_i &\geq (1 - 2\varepsilon)\mu(G) - \sum_{i \in HAPPY \cap \text{OPT}} p_{o_i} \\
|M| &\geq (1 - 2\varepsilon)\mu(G) + \left(\sum_{j \in R} p_j - \sum_{i \in HAPPY \cap \text{OPT}} p_{o_i} \right) \\
&\geq (1 - 2\varepsilon)\mu(G).
\end{aligned}$$

□

Now we prove that the conditions of the lemma are satisfied for some iteration.

Lemma 2.3. *There exists some iteration* $r \leq \lceil 2/\varepsilon \rceil$ *where at least* $(1-\varepsilon)\mu(G)$ *bidders in* OPT *are* ε*-happy.*

Proof. We define a set of potential functions $\Phi_{bidders}$ and Φ_{items} to represent the sum of the minimum prices of neighboring items to each bidder and the sum of the prices of all items. Formally, we define:

$$\begin{aligned}
\Phi_{bidders} &= \sum_{i \in \text{OPT}} \min_{j \in N(i)} (p_j) \\
\Phi_{items} &= \sum_{j \in R} p_j.
\end{aligned}$$

Now we first note that both $0 \leq \Phi_{bidders}, \Phi_{items} \leq \mu(G)$ since prices start at 0 and are capped at 1. The number of allocated items cannot be more than $\mu(G)$ as otherwise, we get a better matching. Thus, both potential functions are monotone.

Consider an iteration where $\geq \varepsilon \cdot \mu(G)$ bidders are not ε-happy. They are the unallocated bidders.

At the end of the iteration, either an unallocated bidder becomes matched or all of the items in its demand set are matched to other bidders. In the first case, the price of its matched item increases by ε. In the second case, $\min_{j \in N(i)}(p_j)$ increases since all items in its demand set becomes matched (i.e. all of its lowest price items increase in price). Hence, $\Phi_{items} + \Phi_{bidders}$ increase by at least $\varepsilon^2 \cdot \mu(G)$ in total. The maximum possible value of this sum is $2\mu(G)$ so by the pigeonhole principle, we have that in $\frac{2\mu(G)}{\varepsilon^2 \mu(G)} = \frac{2}{\varepsilon}$ iterations, in at least one iteration we have enough happy bidders. □

Finally, we show how to implement this algorithm in the streaming model and show its space bounds.

Lemma 2.4. *Algorithm 1 can be implemented in the streaming model in* $O\left(\frac{1}{\varepsilon}\right)$ *passes and* $O(n \log(1/\varepsilon))$ *space.*

Proof. To prove the number of passes, we simply need to prove that each iteration requires $O(1)$ passes. In each iteration, the demand set can be determined in 2 passes. In the first pass, the bidder determines their lowest price neighbor. Then, we do not store D_i for each bidder explicitly but implicitly calculate it when determining the maximal matching. For each unallocated neighbor, we greedily find a maximal matching in the second pass by taking an edge if it is in the demand set of a bidder and the bidder is not yet matched. Then, reallocation and price increases do not require additional passes. Hence, the algorithm can be implemented in $O(1/\varepsilon^2)$ passes.

The amount of space usage is equal to the number of bidders and the space required to store the prices for each bidder. Hence, a total of $O(n \log(1/\varepsilon))$ space is needed. $\square$

References

[ALT21] Sepehr Assadi, S Cliff Liu, and Robert E Tarjan. An auction algorithm for bipartite matching in streaming and massively parallel computation models. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 165–171. SIAM, 2021.