Chapter 4

Zachary Campbell

March 28, 2018

We now look at a cool application of weighted bipartite matching. Consider an auction scenario, where there is a set B of bidders, a set G of goods, and costs c_{bg} that quantifies the amount that bidder $b \in B$ is willing to pay for good $g \in G$. This can easily be modeled with a weighted bipartite graph. Our goal is to maximize the total amount earned in the auction – i.e. to maximize $\sum_{(b,g)} c_{bg}$, under the constraint that no bidder gets more than one good, and no good is purchased by more than one bidder.

The algorithm presented here is an alternative to the alternating path algorithms we have discussed.

ALG 1

For each $g \in G$, set $p_g := 0$ and $owner_g := null$.
Queue $Q := B$.
Set $\delta = 1/(|G| + 1)$.
while $Q \neq \emptyset$
    $b = Q.deque()$
    Find $g \in G$ that maximizes $c_{bg} - p_g$
    if $c_{bg} - p_g \geq 0$
        $Q.enqueue(owner_g)$
        $owner_g = b$
        $p_g = p_g + \delta$
return $(owner_g, g)$ for all g.

Definition . We say that a bidder b is δ-happy if one of the following is true:

1. b is the owner of some good g, and for all other goods $g^{'}$, we have that $\delta + c_{bg} - p_g \geq c_{bg'} - p_{g'}$, i.e. our b has the good g that maximizes the value of their contribution.

2. for no good g does it hold that $owner_g = b$ and for all goods g $w_{bg} \leq p_g$.

What we will show is that the algorithm reaches an equilibrium in which all bidders are "happy," as defined above. The loop invariant is that all bidders not in Q are δ-happy. This is clearly true at the beginning, as all bidders start in our queue. When a bidder b is dequeued, line 6 in our loop chooses good g that maximizes $w_{bg} - p_g$, which means it chooses a good that makes b δ-happy, if such a good exists. We need to confirm that this step does not hurt the invariant for any other bidder $b^{'}$. Well, an increase in p_g by δ for any g not own by $b^{'}$ does not violate the inequality $\delta + w_{b^{'}g^{'}} - p_{g^{'}} \geq w_{b^{'}g} - p_g$. On the other hand, if $b^{'}$ did own g, this means that $b^{'}$ has been thrown back into Q, so $b^{'}$ no longer owns anything.

Lemma . If all bidders are δ-happy then for every matching $M^{'}$ we have that

$$n\delta + \sum_{b=owner_g} w_{bg} \geq \sum_{(b,g)\in M'} w_{bg},$$

where n is the number of bidders.

First, note that this lemma implies the correctness of our algorithm. Since $n\delta < 1$ and all of our weights are integral, this proving this inequality will show that the matching we output is a maximum matching.

Proof. Fix a bidder b, and let good g be such that $b = owner_g$. Let $g^{'}$ be the good assigned to b in $M^{'}$. (Note: these could be *null*, in which case we adopt the convention of assigning their weights and prices to be zero.) Since b is δ-happy, we have that $\delta + w_{bg} - p_g \geq w_{bg} - p_g$. Now let's sum over all b to get

$$\sum_{b=owner_g} (\delta + w_{bg} - p_g) \geq \sum_{(b,g^{'})\in M'} (w_{bg^{'}} - p_g).$$

We know that our algorithm gives us a matching, as does $M^{'}$, which means that each good g can only appear at most once on each side of this inequality. So if we substract $\sum_g p_g$ from each side, and rearrange a bit, we get

$$\sum_{b=owner_g} \delta + \sum_{b=owner_g} w_{bg} \geq \sum_{(b,g^{'})\in M'} (w_{bg^{'}}).$$

But $\sum_{b=owner_g} \delta \leq n\delta$, which gives us what we want. □

Let's think about how this algorithm relates to the corresponding primal and dual linear programs. First, note that this algorithm maintains primal feasibility at all times. However, at no point are we maintaining dual feasibility. We can define the "price" on bidders b as $p_b = 0$ for all b. Furthermore, the prices on goods p_g never exceeds $\max_b\{w_{bg}\}$. These p_b and p_g are the corresponding dual variables in the linear program, so we never have that $p_b + p_g \geq w_{bg}$. It is still useful to think about the linear program since, although infeasible as a dual solution, we our final price vector $\mathbf{p}$ is the pointwise minimum such that all bidders are δ- happy. So the algorithm is still performing a minimization over $\sum_b p_b + \sum_g p_g = \sum_g p_g$, but relaxing the constraint that $p_b + p_g = p_g \geq w_{bg}$.

Note that this algorithm functions as an accurate model of real-worl multi-item auctions. When potential buyers try to outbid each other, a rational agent outbids another bidder by as small an amount they think will still put the other bidder out of the running. Correspondingly, we choose δ to mimick this process. Larger values of δ will speed up the algorithm, but will only offer approximate solutions, which may be good enough when the auction size is in the thousands, or tens of thousands of items.