Economic Efficiency Requires Interaction

Elliot Marx and Nikhil Bhargava

May 27, 2014

Consider the matching scenario where there are n players and n goods. Each player i has unit demand, and values each good in its private subset S_i of goods equally. Our goal is to determine an allocation of the items so as to maximize social welfare, the number of players who receive a good from their desired set. If we know each S_i, then this problem reduces to simple bipartite matching. Each player i is a node $p_i \in P$ and each good j is a node $q_j \in Q$. We say that an edge (p_i, q_j) exists when $j \in S_i$. Then, this graph is clearly bipartite and by finding a maximum matching, we have maximized social welfare.

However, sending the preference of all n bidders when subsets could be up to size n requires n^2 communication. This paper investigates how close an approximation to maximum social welfare algorithms which send $l << n$ bits per player can achieve. Note that if $l = n$, then we can specify the complete set preferences of each player i as a bit vector, so we consider cases where $l \approx O(\log n)$.

1 Single Message Lower Bounds

We first consider a model where we allow the players to send a single l-bit message to a central authority, and then the central authority determines a matching. The paper provides lower-bounds on the possible approximation guarantees for both deterministic and randomized algorithms, proving a seperation between the two, with randomized algorithms outperforming deterministic algorithms.

1.1 Deterministic Lower Bound

Theorem 2. *The approximation ratio of any single-round simultaneous deterministic algorithm for the subset allocation algorithm that uses at most $l = n^\epsilon$-bits per player is $\Omega(n^{1-\epsilon})$.*

The full proof of this theorem is involved, but the special case where $l = \log n$ is easy to understand. Here, a player can send only a single arbitrary good that she is interested in obtaining. We can construct a graph which has a perfect matching by pairing player i with good i, and then connect every player to some good j. Because the algorithm is deterministic, there is some way of constructing the subsets such that each player reports his preference as the shared good j. With n players requesting the same good, the best possible matching is to give the good to one player, which is an approximation factor of n. The more involved statement of the above theorem says that given l bit messages, the approximation ratio is $\Omega(\frac{n}{8l+4\log n})$, which for $l = \log n$ is $\frac{n}{12\log n}$, making this counter example is nearly tight.

2.1 Deterministic Algorithm with $\max(2, \frac{n \log n}{l})$-approximation Factor

Let $l' = \frac{l}{\log n}$. Each player i report the indices of l' arbitrary verticies in his set S_i. Then, we decide an optimal matching given the reported indices. We provide a proof of the $\max(2, \frac{n \log n}{l})$-approximation factor below.

Proof. For the proof, we distinguish between two cases:

1. At least half of the players matched in the optimal solution have subsets of size less than l'
   In this case, at least half the players from the optimal solution can report their full set of preferences. Therefore, we will match at least half the players from the optimal solution as well as in the optimal solution, which yeilds an approximation ratio of 2.

2. Fewer than half of the players matched in the optimal solution have subsets of size less than l'
   Let S be the set of players with subsets of size less than l' who are matched in the optimal solution. If all of the players in S are matched, then we have an approximation ratio of no worse than 2 (more than half the players are matched as well as in the optimal solution). Otherwise, there is some player $i \in S$ not matched. Because i was not matched, at least l' goods have already been taken, because i's subset was at least size l' and all his goods have been taken, as otherwise he would have been matched. Therefore, the algorithm makes at least l' pairings. We know that the optimal algorithm cannot match more then n players to goods, so the approximation ratio in this case is $n/l' = \frac{n \log n}{l}$.

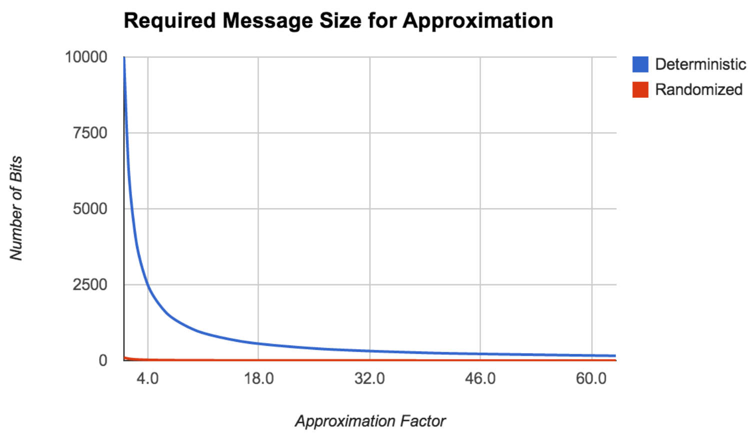
As in the first case, we had an approximation ratio of 2 and in the second we had an approximation ratio of $\max(2, \frac{n \log n}{l})$, the approximation ratio of the algorithm is $\max(2, \frac{n \log n}{l})$. □

We claim that the algorithm approaches optimal. To see this, note that with $l = n^\epsilon$-bit messages, we achieve a $\max(2, \frac{n \log n}{n^\epsilon}) = \max(2, n^{1-\epsilon} \log n)$ approximation ratio, which approaches the optimal of $n^{1-\epsilon}$ bound.

2.2 Randomized Algorithms

Theorem 3. *Fix $\epsilon > 0$. The approximation ratio of any single-round simultaneous randomized algorithm for the subset allocation algorithm that uses at most $l = n^{1/2-\alpha-\epsilon/2}$-bits per player is $\Omega(n^\alpha)$ for $\alpha \leq \frac{1}{2} - \epsilon$.*

We don't give a proof of this theorem, but an algorithm which achieves this bound is given in section 8 (the k round matching algorithm with k set to 1).



The graph above provides a comparison of the randomized and deterministic algorithms from the bounds provided above for $n = 10000$. Clearly, the randomized algorithms achieve a much better approximation factor for a fixed n, but neither provides a constant approximation factor as n grows.

4 Interactive Matching

Consider a model where each player is allowed to send a $\log(n)$-bit message in a series of rounds. Can k rounds of $\log(n)$-bit messages do better than one round of $k\log(n)$-bit messages?

4.1 An algorithm for $(1+\delta)$-approximation in rounds

The algorithm presented in the paper is based on a standard ascending auction, with two notable differences. First, bids are submitted simultaneously for items, rather than considered in a sequential and fixed ordering. This helps us decrease the number of rounds, though it leaves uncertainty about who wants a good the most. Second, bids are randomized, which helps us avoid players consistently bidding on the same item until it reaches a maximum price, thereby decreasing the number of price increments we have. The algoirthm is given below:

1. Set the price p_i of each item i to 0
2. For each round $r = 1, \ldots, 2\log(n)/\delta^2$:
   (a) Let D_i be the demand of a player i, the goods in its subset S_i which have the lowest price. We say that no player demands a good with price ≥ 1.
   (b) Each unallocated player i sends in a random good in D_i.
   (c) Loop over the bids received in the round in a fixed, arbitrary order. Accept each bid we encounter, so long as the good hasn't been given away in this round. Each player who has a good taken from him becomes unallocated, and the price of the item reallocated in this round rises by δ.
   (d) Report the new prices and allocations back to the players.

We show that this algorithm achieves a $(1 + \delta)$-approximation. With interaction, we are able to achieve a constant approximation factor. For comparison, if we set $\delta = 1$, we get a 2-approximation in $2\log(n)$ rounds. As we showed earlier, using a single round with $2\log^2(n)$ bit-messages, we can only report slightly more than a single index. Reporting only a single index, we can only achieve a $O(\sqrt{n})$-approxmation. So, by allowing interaction, we are able to achieve a constant approximation factor, whereas without interaction, we have an approximation factor that degrades with increasing number of bidders.

Intuitively, the price captures how much a player wants a good. Players bid for the goods least demanded by other players (on expectation), which helps spread bidding over all players. If the goods a player demands have a high price, then goods a player demands are popular among many players. It is important to note that the pricing mechanism does not influence the players; it is simply a mechanism for determining an efficient allocation of goods.

4.2 Proof of Approximation Factor

We claim that the above algorithm produces a $(1+\delta)$-approximation after $2\log(n)/\delta^2$ rounds.

Definition 1. *We call a player i satisfied in a given round if he either is allocated a good or if $D_i = \emptyset$.*

Let $N', |N'| = n'$, be the players satisfied in the optimal solution, and let END be a random variable that denotes the number of rounds until at least $(1 - \delta)n'$ of the players in N' are satisfied.

The plan for the proof is as follows. First we show that the $E[END] \leq 4\log n/\delta^2$. We will do this by using budgets, which act as potential functions to measure our progress. This will give us a bound on the number of rounds to satisfy some number of players. Then, we will show that when $(1 - \delta)n'$ players are satisfied, we have made meaningful progress. Namely, we will show that we achieve at least a $1 - 2\delta$ approximation factor in this case. We will do this using a variant on the first welfare theorem.

Theorem 5. $E[END] \leq 4\log n/\delta^2$

Proof. As we noted in our outline of the proof, the first step is to bound the number of rounds until $(1 - \delta)n'$ of the players in N' are satisfied (equivilantly, less than $\delta n'$ players in N' are unsatisfied). Note that this algorithm doesn't make clear progress in each round. There is no guarantee that each rounds we satisfy any more players (though we also don't satisfy any fewer), so we need to turn to craft a proxy for progress of the algorithm. We do so with two budgets: one for the number of price increments on items, and one for the number of demand halving actions for players in N'. We show that each round makes progress in depleting at least one of these budgets, and then show that these budgets expire in the desired number of rounds. Note that these budgets intuitively must run out. For price increments, we have a bound on the maximum price of a good, and for demand of players, we know that all items a player demands have the same price (as otherwise there would be some good of higher price), and that prices must be rising on the goods.

Let p_{D_i} be the price of the goods in D_i and let D_i^p be the goods in a player i's subset, S_i where the price is equal to p (formally, $D_i^p = S_i \cap \{j | p_j = p\}$). We use the following theorem to show the depletion of one of the two budgets in a single round:

Theorem 6. *In any round with t players unsatisfied at the beginning of the round, at least one of the following is true:*

1. $E[\sum_j \Delta p_j] > t\delta/4$
2. *For at least $t/2$ players, the demand of the player from one round to the next has shrunk by a factor of 2 (Formally, $D_i^{p_{D_i}}$ has shrunk by at least a factor of 2).*

Proof. We consider each player individually before considering their aggregate effect on the round. Consider some unsatisfied player i after submitting its bid. When i is considered for having his bid accepted, either half the goods it demanded at the beginning of the round were taken by players who were considered previously in the round, or they were not. If half of the goods it demanded have been taken, then we have made progress toward the second case of the theorem (its demand has fallen by half). Otherwise, at least half the items D_i selects are still available when he is considered in the allocation phase of the round. Because i randomizes over the goods he wants and more than half of those goods are available, he has at least probability $1/2$ of getting the item. If player i is allocated the item j, he causes p_j to rise by δ, so on expectation, because he gets the good with at least $1/2$ probability, he causes p_j to rise by at least $\delta/2$.

Now, we consider the players in aggregate. For at least half the unsatisfied players, either their demand falls by half before they are considered, or it does not. If it does, then we trivially fulfill the second case of the lemma. If it does not, then we know that at least half of the players cause a price increase of at least $\delta/2$ on some item. So, by linearity of expectation, the at least $t/2$ unsatisfied players cause $\sum_j \Delta p_j$ to increase by $t\delta/4$ on expectation. □

With this key lemma in hand, we return to considering the bound on the number of rounds. First, notice that the price of each item reaches a maximum at 1, and prices increase in increments of δ, so an item can increase in price at most $1/\delta$ times. At most n' goods can be allocated (this is the optimal bound). We can only be incrementing prices only on goods that the algorithm in the end allocates (We also know that once a good is allocated, it remains allocated. We see this by noticing that goods are only unallocated if they are to be given to another player immediately). Therefore, because we increment prices on no more than n' goods and each good can increase at most $1/\delta$ times, we can increase prices at most n'/δ total times.

$|S_i| \leq n$ for every player i, so his demand at a price, D_i^p can be shrunk by a factor of 2 at most $\log n$ time. So, combining the fact that the number of price increments we can make is at most n'/δ and demand at a price can shrink by a factor of 2 at most $\log n$ time, the total number of shrinkage steps for the players in N' is at most $n' \log n/\delta$.

In each round, we either see a drop in the prices, or a drop in the demand for each player. So, if the demand is to drop in each round, our budget would run out in:

$$\frac{\text{total shrinkage steps}}{\text{shrinkage per round}} = \frac{n' \log n/\delta}{t/2} = \frac{2n' \log n}{\delta t}$$

rounds. Prior to END, the number of unsatisfied players t is at least $\delta n'$, so it would take at most $2 \log n/(\delta^2)$ rounds to run out this budget if the demand drop occured in each round.

If the price increase occured in each round, then we would see a total price increase of $t\delta/4$ in each round. As the price can increase at most n'/δ times, it would take $n'/\delta/(t\delta/4) = 4n'/(t\delta^2)$ rounds to run out this budget. Again, the number of unsatisfied players t is at least $\delta n'$, so this budget would run out in no more than $4/\delta^3$ rounds. Therefore, both budgets will run out in $4 \log n/\delta^2$ rounds. □

Finally, we show that satisfying $(1-\delta)n'$ of the players in N' implies a meaningful approximation factor.

Theorem 7. *If at least $(1-\delta)n'$ of the players in N' are satised, then the approximation of the algorithm is $1-2\delta$.*

Proof. Let the optimal solution be $(o_1, \ldots, o_n)$, where o_i specifies the good that the i-th player receives. Consider a player who has received an item j_i. The player's demand is maximized up to δ by the first welfare theorem, so:

$$v_i(j_i) - p_{j_i} \geq v_i(o_i) - p_{o_i} - \delta$$

Where $v_i(\cdot)$ is the value of receiving the good $\cdot$. If a player is satisfied and did not receive an item, the the good was too expensive, so $0 \geq v_i(o_i) - p_{o_i}$. Now, summing over all satisfied players N_s:

$$\begin{aligned} \sum_{i \in N_s} v_i(j_i) - p_{j_i} &\geq \sum_{i \in N_s : iallocated} (v_i(o_i) - p_{o_i} - \delta) + \sum_{i \in N_s : iunallocated} (v_i(o_i) - p_{o_i}) \\ &= \sum_{i \in N_s} V_i(o_i) - p_{o_i} - \delta n' \end{aligned}$$

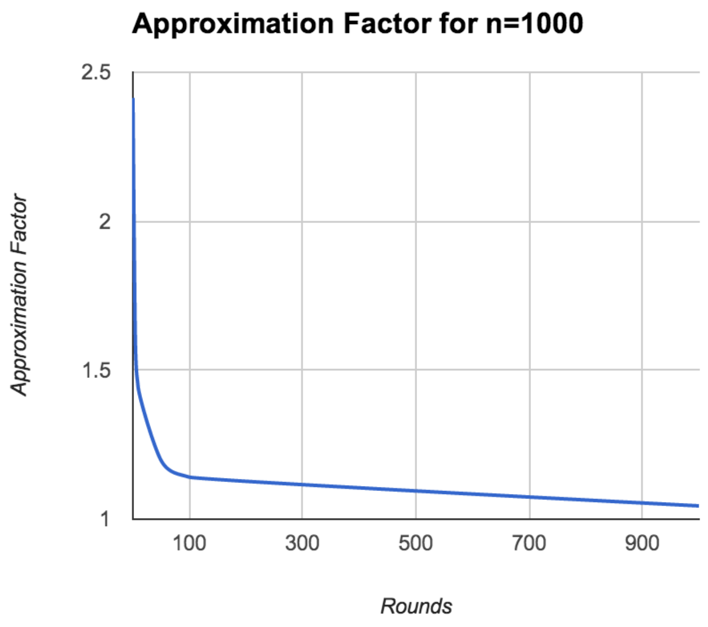
We know that

$$\sum_{i \in N_s} v_i(j_i) - p_{j_i} = ALG - \sum_{j \in N} p_j$$

because goods that are unallocated by the algorithm have a price of 0 (once a good is bid on, it will be allocated). Now, because $N' \cap N_s \geq (1-\delta)N'$:

$$\begin{aligned} ALG - \sum_{j \in N} p_j &\geq OPT - \delta n' - \sum_{i \in N_s} p_{o_i} - n'\delta \\ ALG &\geq OPT - 2n'\delta = (1-2\delta)n' \end{aligned}$$

□



The figure below gives shows the number of rounds required to achieve a given approximation ratio with 1000 players. After 5 rounds, we have an approximation factor of 1.63, which then gradually falls as the number of rounds increases.

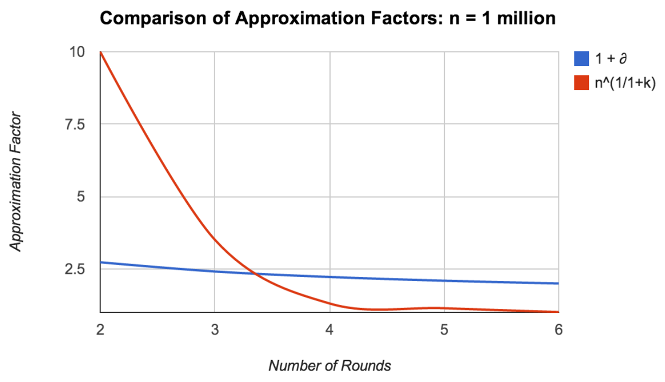
8 A k-Round Algorithm for Matching

The authors also propose an algorithm which gives approximation bounds given the number of rounds. In k rounds, for $k < \log n$, the algorithm achieves a $O(n^{\frac{1}{k+1}})$-approximation ratio.

8.1 The algorithm

1. Let N_1 be the set of all players and U_1 be the set of all goods
2. For each round $r = 1, \ldots, k$
   (a) Each player i chooses a good uniformly at random from U_i that he demands
   (b) Loop over the bids received in the round in a fixed, arbitrary order. Accept each bid we encounter, so long as the good hasn't been given away already.
   (c) Let $N_{r+1} \subseteq N_r$ be the players who have yet to receive a good
   (d) Let $U_{r+1} \subseteq U_r$ be the goods which have not yet been allocated.

In this algorithm, goods are only allocated once, rather than changing the allocation at each round. We compare the performance of the k-round matching algorithm with the $(1 + \delta)$-approximation presented earlier below for $n = 1$ million:



With a limited number of rounds the re-allocation algorithm performs better. However, after this point, the algorithm above begins to outperform the $(1+\delta)$-approximation algorithm. Note that the k-round matching algorithm is practical only when the number of players is very large, as the guarantee holds only when $k \leq \log n$. Further, in a single round, it is outperformed by by the $(1+\delta)$-approximation algorithm.

9 Conclusion

This paper surveyed the hardness of finding matchings for the subset allocation problem when communication is restricted. Using single round algorithms, we found that randomized algorithms performed significantly better than their deterministic counterparts, but that we could not achieve a constant approximation ratio. When we changed our model to allow a series of rounds, we were able to achieve a constant approximation factor. We studied two algorithms, one which gave better approximations when we had a large number of players (k rounds), and one which was better with a greater number of rounds a fewer players.