Simple Randomized Algorithms for Assignment Problems

Pushkar Tripathi
Mentors : Prasad Tetali and Vijay Vazirani

June 28, 2011

Abstract

In this proposal I will present two problems that I would like to consider over the course of the next semester. Both problems fall in the general realm of allocation problems for which the known optimal algorithms have large polynomial running time. I wish to study the performance of simple randomized algorithms for these problems. I will also present a short literature survey of prior work on these problems.

1 Submodular Welfare Maximization Problem

In the submodular welfare maximization problem(SWMP) we are given a set X of m items and n agents(indexed by i). Each agent has a monotone submodular utility function over the subsets of X denoted by v_i, that is given as a value oracle. The objective is to allocate the items among the agents so that the sum of their utilities is maximized i.e. We wish to partition X into n disjoint sets S_i, $i \in [n]$ so as to maximize $\sum_i v_i(S_i)$.

The problem is known to be NP-hard and constant factor approximation algorithms are known for this problem. I propose to develop an online algorithm for the submodular welfare maximization problem(SWMP) by analyzing a randomized version of the algorithm proposed by [9] and show that it achieves a factor better than $1/2$.

1.1 Literature Survey

The SWMP was first proposed by Lehmann, Lehmann and Nisan in [9]. The problem is hard to approximate to a factor better than $1 - 1/e$ under information theoretic considerations [8, 11].The approximability of the SWMP for the value query model was settled by Vondrak in [3, 12] where he gave an optimal $1 - 1/e$ factor algorithm for this problem. However the algorithm is quite involved and has a large polynomial running time.

Consider the following algorithm to solve this problem. Iterate through the items according to an arbitrary order and for every item, assign it to the agent who receives the maximum marginal (incremental) benefit from receiving the item. It can be proved that this simple *greedy* algorithm achieves an approximation factor of $1/2$. The question that I seek to answer is that what happens if instead of an arbitrary order, we consider the items in a random order. I aim to show that this *randomized greedy* algorithm infact does better than $1/2$ on an average. The algorithm may be viewed as an online algorithm where the items arrive in online and need to be assigned greedily to the agents as they arrive. This generalizes two well known online allocation problems namely - Online Bipartite Matching, and Online Budgeted Allocation Problem.

Online Bipartite Matching. In this problem there is a bipartite graph $G(L \cup R, E)$ that is not known in advance. The vertices from the right bipartition R arrive online. Every time a vertex arrives it reveals its neighbors in L and the algorithm has to match the newly arrived vertex to an available(if any) neighbor in L. The objective is to maximize the number of vertices that get matched.

Online Budgeted Allocation Problem(OBAP). We are given a set of agents each of who have a daily budget B_i. Items arrive online and each agent places a (possibly different) bid on the item. Each arriving item can be allocated to at most one agent and the agent is charged his bid for the item from his budget. The objective is to maximize the amount of money spent by the agents. [10] considered the special case when the bids are small with respect to the daily budgets and gave a deterministic $1 - 1/e$ factor algorithm for the problem. Nothing better than the trivial 0.5 factor algorithm is known when bids can be arbitrarily large.

1.2 Possible Approaches

In this section I will present a high level view of three approaches towards solving this problem.

Randomized Primal-Dual. Recently Jain and Devanur [5, 4] developed a randomized primal dual interpretation of the Online Bipartite Matching Algorithm proposed in [7]. They use a linear programming relaxation for the bipartite matching problem and maintain a feasible primal and dual solution throughout the algorithm. They however violate the linear equations corresponding to the complimentary slackness conditions but only to a multiplicative factor of $1 - 1/e$ in expectation. Even though we cannot write a linear program for the general SWMP, an important subcase, i.e. the OBAP has a linear programming relaxation with a small integrality gap where the same approach might work.

Charging Approach. The main tool in the analyses of all online matching algorithms has been the development of a charging argument that charges 'losses' incurred by the algorithm to its 'gains'. Since in general an optimal solution for the SWMP may assign multiple items to the same agent it is difficult to define these gains and losses for submodular functions. The proof by Lehmann et al. in [9] does not use a charging scheme but that makes it harder to estimate the expected additional gains when the items are considered in random order. I have been able to develop an alternate proof that uses a charging scheme and establishes the same factor. My hope is that this proof would be more amenable to extension and would lead to a factor better than 1/2 in our setting.

Correlation Gap Trick. The third approach is based on a recent result by Agarwal et.al [1] relating to the correlation gap of a submodular function. Let f be a function over subsets of a given finite discrete domain X and let $\mathcal{D}$ be any distribution over subsets of X. Let $P_{\mathcal{D}}$ be another distribution such that for all $x \in X$, $Pr\left[x \in S | S \hookleftarrow \mathcal{D}\right] = Pr\left[x \in S | S \hookleftarrow P_{\mathcal{D}}\right]$. The correlation gap for the f is defined as $max_{P_{\mathcal{D}}} \left\{ \frac{E_{S \hookleftarrow \mathcal{D}}[S]}{E_{S \hookleftarrow P_{\mathcal{D}}}[S]} \right\}$.

They showed that the correlation gap for submodular functions is bounded by $1 - 1/e$. This result can be useful in analyzing our algorithm as follows. Suppose we can write a linear program which gives the optimal solution to the SWMP but this program may have exponentially many variables and constraints. The solution to this linear program may be viewed as a distribution over the subsets of X. Since it is difficult to sample from this distribution(it would require us to solve a *huge* LP) we can try to approximate the distribution with another distribution that preserves the marginal probabilities for the elements. By the result in [1] this would only take us a factor of $1 - 1/e$ from the optimal solution. Using this approach we can show that for the special case when all agents have the same valuation functions then the proposed algorithm does lead to a $1 - 1/e$ factor algorithm.

2 An Oblivious Algorithm for Maximum Matching

Consider the following problem. We are given the vertex set V, of an undirected graph G(V,E) but the edge set E is not revealed to us. For every pair $(u, v) \in V \times V$ we are not told apriori whether there is an edge connecting these vertices, until we *probe/scan* this pair. If we scan a pair of vertices and find that there is an edge connecting them we are constrained to *pick* this edge and in this case both u and v are removed from the graph. However, if we find that u and v are not connected by an edge, they continue to be available to be matched in the future. The goal is to maximize the number of vertices that get matched.

We call this the oblivious matching problem since we are required to find a large matching without knowing the edge structure of the graph. The problem finds applications in barter exchanges and also models several previously studied versions of what is known as the kidney exchange problem. It is also important from the point of view of mechanism design where we may view the vertices as malicious agents who may hide their incident edges to maximize their chances of getting matched. The remedy in all these scenarios is a matching algorithm that acts independent of the edge structure.

It is easy to see that the greedy algorithm that considers one vertex at a time in arbitrary order and matches it to an arbitrary neighbor returns a maximal matching and is therefore a 1/2 approximation. To the best of our knowledge, there is just one oblivious algorithm that beats this barrier. In [2] Frieze et. al showed that if we consider the vertices in random order and match each vertex to a random unmatched neighbor then this algorithm attains a factor of 0.5000001. Our goal is to design an algorithm that beats this bound.

I propose to analyze the following randomized algorithm for this problem - As before, order the vertices according to a random permutation σ. Consider the vertices according σ and match each vertex to the first available(if any) neighbor in σ. The algorithm is inspired by the *Ranking* algorithm for online bipartite matching discussed in section 1.1. In fact for the special case of bipartite graphs this reduces to the result in [6].

2.1 Techniques and Results

Together with Gagan Goel and using tools developed in [6] we can show that the above algorithm infact does slightly better than Frieze's result. We can show that it attains a factor of at least 0.526. However despite an extensive(computer aided) search over various families of graphs, it seems the worst examples for the algorithm are bipartite graphs. This presents the intriguing possiblity that the actual factor for the algorithm may be much higher, possibly better than

$1 - 1/e$.Our results are based on a charging argument similar to the one described in section 1.2. However the presence of odd cycles makes the argument quite involved and I will omit the details due to space constraints.

References

[1] Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Correlation robust stochastic optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1087–1096, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[2] Jonathan Aronson, Martin Dyer, Alan Frieze, and Stephen Suen. Randomized greedy matching. ii. *Random Struct. Algorithms*, 6:55–73, January 1995.

[3] Gruia Calinescu, Chandra Chekuri, Martin Pal, and Jan Vondrak. Maximizing a submodular set function subject to a matroid constraint. In *Proc. of 12 th IPCO*, page 2007, 2007.

[4] Kamal Jain. Generalized online matching with concave utilities. Presented at the Approximation Algorithms Workshop: The Last Decade and the Next, Princeton, 2011.

[5] Kamal Jain. Understanding karp-vazirani-vaziranis online matching (1990) via randomized primal-dual. Presented at the Approximation Algorithms Workshop: The Last Decade and the Next, Princeton, 2011.

[6] Chinmay Karande, Aranyak Mehta, and Pushkar Tripathi. Online bipartite matching in the unknown distributional model. In *43rd ACM Symposium on Theory of Computing(STOC)*, 2011.

[7] R.M. Karp, U.V. Vazirani, and V.V. Vazirani. An optimal algorithm for online bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, 1990.

[8] Subhash Khot, Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Inapproximability results for combinatorial auctions with submodular utility functions. In *in Proceedings of WINE 2005*, pages 92–101, 2005.

[9] Benny Lehmann, Daniel Lehmann, and Noam Nisan. Combinatorial auctions with decreasing marginal utilities. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, EC '01, pages 18–28, New York, NY, USA, 2001. ACM.

[10] Aranyak Mehta, Amin Saberi, Umesh Vazirani, and Vijay Vazirani. Adwords and generalized online matching. In *FOCS*, 2005.

[11] Vahab Mirrokni, Michael Schapira, and Jan Vondrak. Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions. In *Proceedings of the 9th ACM conference on Electronic commerce*, EC '08, pages 70–77, New York, NY, USA, 2008. ACM.

[12] Jan Vondrak. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 67–74, New York, NY, USA, 2008. ACM.