CSC2420: Algorithm Design, Analysis and Theory
Fall 2023
An introductory (i.e. foundational) level graduate course.

Allan Borodin

October 31, 2023

Week 8

Announcements
I was going to post an IP/LP rounding problem forthe 4th and final question for Assignment 2. Instead I will put that question in Assignment 3. So please submit Assignment 2 by the due date (this Friday, Nov 3, 11AM).

Todays agenda

- Quick review of bipartite matching
- Getting past the $1 - 1/e$ approximation ratio.
- AdWords and Display Ads
- A better than $\frac{3}{4}$ algorithm for Max-2-Sat by vector programming.
- Random walk algorithms for 2SAT and SAT

The randomized ranking algorithm

- The algorithm chooses a random permutation of the nodes in V and then when a node $u \in U$ appears, it matches u to the highest ranked unmatched $v \in V$ such that (u, v) is an edge (if such a v exists).
- Aside: making a random choice for each u is still only a $\frac{1}{2}$ approx.
- Equivalently, this algorithm can be viewed as a deterministic greedy (i.e. always matching when possible and breaking ties consistently) algorithm in the ROM model.
- That is, let $\{v_1, \ldots, v_n\}$ be any fixed ordering of the vertices and let the nodes in U enter randomly, then match each u to the first unmatched $v \in V$ according to the fixed order.
- To argue this, consider fixed orderings of U and V; the claim is that the matching will be the same whether U or V is entering online.

The KVV result and recent progress

KVV Theorem

Ranking provides a $(1 - 1/e)$ approximation.

- Original analysis is not rigorous. There is an alternative proof (and extension) by Goel and Mehta [2008], and then another proof in Birnbaum and Mathieu [2008]. Other alternative proofs have followed.
- Recall that this positive result can be stated either as the bound for a particular deterministic algorithm in the stochastic ROM model, or as the randomized Ranking algorithm in the (adversarial) online model.
- KVV show that the $(1 - 1/e)$ bound is essentially tight for any randomized online (i.e. adversarial input) algorithm. In the ROM model, Goel and Mehta state inapproximation bounds of $\frac{3}{4}$ (for deterministic) and $\frac{5}{6}$ (for randomized) algorithms.
- In the ROM model, Karande, Mehta, Tripathi [2011] show that Ranking achieves approximation at least .653 (beating $1 - 1/e$) and no better than .727. This ratio was improved to .696 by Mahdian and Yan [2011]].

And some more recent progress

- Karande et al show that any ROM approximation result implies the same result for the unknown i.i.d. model.
- Manshadi et al give a .823 inapproximation for biparitie matching in the known i.i.d. distribution model. This implies the same inapproximation in the unknown i.i.d. and ROM models improving the $\frac{5}{6}$ inapproximation of Goel and Mehta.
- There is a large landscape (and continuing research) of weighted versions of online bipartite matching such as the *adwords* problem and the *display ads* problem that are motivated by applications to online advertising.
- Although out of data, the survey by Mehta [2013] is a good starting reference. Note: The table in the survey seems to identify the ROM and unknown i.i.d. as equivalent models. model. Recently, Correa et al [Math of OR 2022] show that there is a provable gap between the known and unknown i.i.d. ratios when there is one offline node.

Getting past the $(1 - 1/e)$ bound

- The ROM model can be considered as an example of what is called stochastic optimization in the OR literature. As we have discussed early in the term, there are other stochastic optimization models that are perhaps more natural, namely i.i.d sampling from known and unknown distributions and Markov distributions.
- Feldman et al [2009] study the known distribution case. More specifically, we assume each onlin node is drawn independently from the same known distribution and refer to the online nodes in the supoort of this distribution as the types of the distribution. Furthermore, they assume the the expected number of draws from each typo is integral. The best results for online bipartite matching with i.i.d. arrivals assume integral arrival rates. In fact, only recently has the $1 - \frac{1}{e}$ "barrier" been broken for arbitrary arrival rates.
- They achieve a .67 ratio which has since been improved in a sequence of papers. (I will fill in references for best known unweighted and edge weighetd results).

ROM arrivals implies i.i.d, with integral arrival rates

Before discussing i.i.d. bipartite matching, let's consisder the Karande et "obsrvation". The result has not been stated in what I think is its generality but was observed implicitly by Kenyon [SODA 1986] for online bin packing, and Karande et al [STOC 2011] for online bipartite matching.

Theorem (informal) Consider any problem $\mathcal{P}$ for which the ROM model and i.i.d. models are applicable. Suppose algorithm $\mathcal{A}$ obtains expected competitive ratio c in the ROM model. Then $\mathcal{A}$ achieves expected competitive ratio at least c in the unknown (and hence known) i.i.d. model with general arrival rates.

The proof is remarkably simple. Consider the algorithm on instances (i.e. multi sets) consisting of n input items. Partition the input instances into classes, such that each class is made up of the $n!$ ways to permute the input items in that class. Each input sequence in a class occurs with the same probability. Thus each class becomes an instance of the random order model and hence algorithm $\mathcal{A}$ has competitive ratio at least c on each class. We then take the expectation over the different classes.

Results for i.i.d. bipartite matching with integral arrival rates

With respect to the known i.i.d. model, most of the work concerns integral arrival rates. Huang et al [STOC 2022] point discuss the status of general arrival rates.

The Feldman et al *Best of Two Worlds Algorithm* achieved a .67 ratio for unweighted online biparitite known distribution model (i.i.d.) with integral arrival rates.

This was improved in successive papers by

- Haeupler et al [WINE 2011] obtaining .661 for edge weights
- Mansadi et al [Math of OR 2012] .705 unweighted.
- Jailet and Lu [Math of OR 2013] .725 for offline vertex weights
- Brubach et al [ESA 2016] .705 edge weighted and .7299 offline vertex weight.

Bipartite matching with general arrival rates

Huang et al [STOC 2022] recently improve the results for unweighted and vertex weighted bipartite matching by Huang and Shu [STOC 2021] as well as proving that for edge weighted bipartite matchning, no online algorithm can achieve a .703 competitive ratio which shows that general arrivals cannot be as good as integral arrivals for which Brubach et al obtained a .705 ratio.

More generally they achieve the following results for known i.i.d arrivals with general arrivals given in Table 1 in their paper.

Table 1: Summary of Results. We round down algorithmic results and round up hardness results, to three decimal places. The results in this paper are on the right of the arrows in bold.

	Algorithms		Hardness	
Unweighted	0.711 [16]	$\rightarrow$ **0.716 (§4)**	0.823 [25]	
Vertex-weighted	0.700 [16]	$\rightarrow$ **0.716 (§4)**	0.823 [25]	
Edge-weighted (Free Disposal)	0.632 [13] †	$\rightarrow$ **0.706 (§3)**	0.823 [25]	
Edge-weighted	0.632 [13] †		0.823 [25]	$\rightarrow$ **0.703 (§5.1)**

† Although Feldman et al. [13] only analyzed unweighted matching, they effectively showed that every edge is matched with probability $1 - 1/e$ times the LP variable, which is sufficient for edge-weighted matching as well.

Figure: Huang et al results for general arrivals

Sketch of ideas for the *Best of Two* algorithm

Let U be the offline vertices and V the online vertices in the type graph. With integral arrival rates we can assume (by making copies of each type) that the the expected number of arrivals of any type is 1. When $n = |V|$ is sufficiently large the actual realized graph is "close enough" to the type graph. There will be an offline stage using the type graph and an online stage using the realized graph.

Offline stage. We set up a flow graph by appending a source node s to the offline vertices in U with capacity 2, a target node t connected to all vertices in V with capacity 2 and all edges (u, v) in the type graph having capacity 1. Since all capacities are integral, there is an optimal integral flow from s to t. Let U^* be the offline nodes in the optimal flow. There is a procedure to color the edges, blue and red. Edges to a node in U in the optimal flow will have one or two colors and when there is only one edge, call that the blue edge.

Online stage. When an online vertex arrives, match it to a free node in U^* by a blue edge. If no such node exists, match it to a free node (if one exists) by a red edge.

The Feldman et al Best of Two Algorithm

Algorithm 52 Main algorithm

procedure TwoSuggestedMatching($G = (U, V, E)$)

▷ Preprocessing stage

Create flow network $G_f = (V_f, E_f)$.
$V_f = V \cup U \cup \{s, t\}$.
$E_f = \{(u, v) : \forall u \in U \text{and} (u, v) \in E\} \cup \{(s, u) : \forall u \in U\} \cup \{(v, t) : \forall v \in V\}$.
for all $(u, v) \in E_f$ **do**
  if $u = s$ or $v = t$ **then**
    $cap_{u,v} = 2$
  else
    $cap_{u,v} = 1$
Find the maximum integral flow f on G_f.
$(E_b, E_r) = BlueRedColoring(G_f)$

▷ Online stage

Let *count* a list of zeros with size $|V|$.
while arriving a new vertex v' **do**
  Let v be the type of v'.
  if $count[v] == 0$ and there exist a $u : (u, v) \in E_b$ **then**
    Match u and v.
  if $count[v] == 1$ and there exist a $u : (u, v) \in E_r$ **then**
    Match u and v.
  $count[v] + +$

Figure: The Feldman et al algorithm

Weighted extensions of online bipartite matching

As mentioned, there are various edge weighted versions of online bipartite matching motivated by online auction advertising.

- Adwords with small and large (compared to the budget) bids.
- The Display Ads problem with free disposal.
- The adwords problem with small bids is equivalent to the display ads problem with large capacities. Both of these problems are generalized by the submodular welfare maximization problem.
- The Display Ads Problem with free disposal is an example of online bipartire bipartite matching with recourse. More generally, one can consider reassigning an online node.

The adwords problem: an extension of weighted bipartite matching

- In the (single slot) adwords problem, the nodes in U are queries and the nodes in V are advertisers. For each query q and advertiser i, there is a bid $b_{q,i}$ representing the value of this query to the advertiser.
- Each offline advertiser i also has a hard budget B_i. An adviser i receives benefit min $\{B_i$, sum of matched bids$\}$. The objective is to maximize the sum of the offline benefits. The goal is to match the nodes in U to V so as to maximize the sum of the accepted bids without exceeding any budgets.
- In the online case, when a query arrives, all the relevant bids are revealed.

Some results for the adwords problem

- Here we are just considering the combinatorial problem and ignoring game theoretic aspects of the problem.
- The problem has been studied for the special (but well motivated case) that all bids are small relative to the budgets. As such this problem is incomparable to the matching problem where all bids are in $\{0,1\}$ and all budgets are 1.
- For this small bid case, Mehta et al [2005) provide a deterministic online algorithm achieving the $1 - 1/e$ bound and show that this is optimal for all randomized online algorithms (i.e. adversarial input).
- Recently, a new algorithmic approach called *online correlated selection (OCS)* was introduced by Zadimoghaddam [arXiv 2017] and first publish in Fahrbach et al [FOCS 2020 and JACM 2022] breaking the $\frac{1}{2}$ "barrier" for the display ads (capacity 1). OGS was also utilized by Huang et al [FOCS 2020] for the adwords (arbitrary bids) problem. I will try to sketch the OCS approach later.

Greedy for a class of adwords problems

- Goel and Mehta [2008] define a class of adwords problems which include the case of small budgets, bipartite matching and b-matching (i.e. when all budgets are equal to some b and all bids are equal to 1).
- For this class of problems, they show that a deterministic greedy algorithm achieves the familiar $1 - 1/e$ bound in the ROM model. Namely, the algorithm assigns each query (.e. node in U) to the advertiser who values it most (truncating bids to keep them within budget and consistently breaking ties). Recall that Ranking can be viewed as greedy (with consistent tie breaking) in the ROM model.

Vertex weighted bipartite matching

- Aggarwal et al [2011] consider a vertex weighted version of the online bipartite matching problem. Namely, the vertices $v \in V$ all have a known weight w_v and the goal is now to maximize the weighted sum of matched vertices in V when again vertices in U arrive online.
- This problem can be shown to subsume the adwords problem when all bids $b_{q,i} = b_i$ from an advertiser are the same.
- It is easy to see that Ranking can be arbitrarily bad when there are arbitrary differences in the weight. Greedy (taking the maximum weight match) can be good in such cases. Can two such algorithms be somehow combined? Surprisingly, Aggarwal et al are able to achieve the same 1-1/e bound for vertex weighted bipartite matching.

The vertex weighted online algorithm

The perturbed greedy algorithm

For each $v \in V$, pick x_v randomly in $[0, 1]$
Let $f(x) = 1 - e^{1-x}$
When $u \in U$ arrives, match u to the unmatched v (if any) having the highest value of $w_v * f(x_v)$. Break ties consistently.

In the unweighted case when all w_v are identical this is the Ranking algorithm.

A result by Huang et al [TALG 2018, arXiv 2019] provdes a randomized algorithm that achieves competitive ratio .6534 in the random order arrival model. .

Some concluding remarks on max-sat and bipartite matching

- The ROM model subsumes the stochastic model where inputs are chosen i.i.d. from an unknown distribution (which in turn subsumes i.i.d. inputs from a known distribution). Why? Hence a positive result in the ROM model implies a positive result in the i.i.d. unknown distribution model.
- A research problem of interest is to see to what extent some form of an extended online or priority framework can yield a deterministic online bipartite matching algorithm with approximation ratio better than $1/2$.
- As mentioned before, Pena can show that a $3/4$ max-sat approximation can be obtained by a deterministic "poly width" online algorithm.
- One can formulate the Buchbinder and Feldman method in the framework of the priority BT model of Alekhnovich et al. Can we show that a bounded width online (or priority) BT algorithm cannot obtain a $3/4$ ratio?

Online and priority width bounds for max-sat and bipartite matching

We have the following width inapproximation results.

- To improve upon the $\frac{3}{4}$ approximation (using online width $2n$) result, we need exponential width. More precisely,
  For any $\epsilon > 0$ there exists $\delta > 0$ such that, for $k < e^{\delta n}$, no online width k algorithm can achieve an asymptotic approximation ratio of $3/4 + \epsilon$ for unweighted exact max-2-sat with input model 2.
- For any $\epsilon > 0$ there exists $\delta > 0$ such that, for $k < e^{\delta n}$, no pBT width k algorithm can achieve an asymptotic approximation ratio of $21/22 + \epsilon$ for unweighted max-2-sat with input model 3.
- $\Omega(\log \log n)$ advice bits (and therefor random bits) are needed to asymptoticall beat the $\frac{1}{2}$ online ratio. Recently Buchbinder et al prove that $O(\log \log n)$ random bits are sufficient to obtain $\frac{1}{2} + \epsilon$ for some $\epsilon > 0$.
- For any $\epsilon > 0$, no priority algorithm can achieve a $\frac{1}{2} + \epsilon$ approximation for bipartite matching.
- The first algorithm to use vectior programming for Max-2-Sat (and other problems such as Max-Cut) is due to Goemans and Williamson

The quadratic program for Max-2-Sat

The following discussion is taken from the Vazirani *Approximation Algorithms* textbook.

- We introduce {-1,1} variables y_i corresponding to the propositional variables. We also introduce a homogenizing variable y_0 which will correspond to a constant truth value. That is, when $y_i = y_0$, the intended meaning is that x_i is set *true* and *false* otherwise.
- We want to express the $\{-1, 1\}$ truth value $val(C)$ of each clause C in terms of these $\{-1, 1\}$ variables.
   1. $val(x_i) = (1 + y_i y_0)/2$
      $val(\bar{x}_i) = (1 - y_i y_0)/2$
   2. If $C = (x_i \vee x_j)$, then $val(C) = 1 - val(\bar{x}_i \wedge \bar{x}_j) = 1 - (\frac{1-y_i y_0}{2})(\frac{1-y_j y_0}{2}) = (3 + y_i y_0 + y_j y_0 - y_i y_j)/4 = \frac{1+y_0 y_i}{4} + \frac{1+y_0 y_j}{4} + \frac{1-y_i y_j}{4}$
   3. If $C = (\bar{x}_i \vee x_j)$ then $val(C) = (3 - y_i y_0 + y_j y_0 + y_i y_j)/4$
   4. If $C = (\bar{x}_i \vee \bar{x}_j)$ then $val(C) = (3 - y_i y_0 - y_j y_0 - y_i y_j)/4$

The quadratic program for Max-2-Sat continued

- The Max-2-Sat problem is then to maximize $\sum w_k val(C_k)$ subject to $(y_i)^2 = 1$ for all i
- By collecting terms of the form $(1 + y_i y_j)$ and $(1 - y_i y_j)$ the max-2-sat objective can be represented as the strict quadratic objective: max $\sum_{0 \leq i < j \leq n} a_{ij}(1 + y_i y_j) + \sum b_{ij}(1 - y_i y_j)$ for some appropriate a_{ij}, b_{ij}.
- Like an IP this integer quadratic program cannot be solved efficiently.

The vector program relaxation for Max-2-Sat

- We now relax the quadratic program to a vector program where each y_i is now a unit length vector $\mathbf{v}_i$ in $\Re^{n+1}$ and scalar multiplication is replaced by vector dot product. This vector program can be (approximately) efficiently solved (i.e. in polynomial time).
- The randomized rounding (from $\mathbf{v}_i^*$ to y_i) proceeds by choosing a random hyperplane in $\Re^{n+1}$ and then setting $y_i = 1$ iff $\mathbf{v}_i^*$ is on the same side of the hyperplane as $\mathbf{v}_0^*$. That is, if $\mathbf{r}$ is a uniformly random vector in $\Re^{n+1}$, then set $y_i = 1$ iff $\mathbf{r} \cdot \mathbf{v}_i^* \geq 0$.
- The rounded solution then has expected value $2 \sum a_{ij} Prob[y_i = y_j] + \sum b_{ij} Prob[y_i \neq y_j]$; $Prob[y_i \neq y_j] = \frac{\theta_{ij}}{\pi}$ where θ_{ij} is the angle between $\mathbf{v}_i^*$ and $\mathbf{v}_j^*$.

The approximation ratio (in expectation) of the rounded solution

Let $\alpha = \frac{2}{\pi} \min_{\{0 \leq \theta \leq \pi\}} \frac{\theta}{(1-cos(\theta))} \approx .87856$ and let OPT_{VP} be the value obtained by an optimal vector program solution.
Then $\mathbf{E}[\text{rounded solution}] \geq \alpha \cdot (OPT_{VP})$.

2SAT and kSAT using random walks

- First, here is the idea of the deterministic polynomial time algorithm for 2-Sat: We can first eliminate all unit clauses. We then reduce the problem to the directed $s - t$ path problem. We view each clause $(x \vee y)$ in F as two directed edges $(\bar{x}, y)$ and $(\bar{y}, x)$ in a graph G_F whose nodes are all possible literals x and $\bar{x}$. Then the formula is satisfiable iff there does not exist a variable x such that there are paths from x to $\bar{x}$ and from $\bar{x}$ to x in G_F.
- There is also a randomized algorithm for 2-SAT (due to Papadimitriou [1991]) based on a random walk on the line graph with nodes $\{0, 1, \ldots, n\}$. We view being on node i as having a truth assignment τ that is Hamming distance i from some fixed satisfying assignment τ^* if such an assignment exists (i.e. F is satisfiable).
- Start with an arbitrary truth assignment τ and if $F(\tau)$ is true then we are done; else find an arbitrary unsatisfied clause C and randomly choose one of the two variables x_i occurring in C and now change τ to τ' by setting $\tau'(x_i) = 1 - \tau(x_i)$.

The expected time to reach a satisfying assignment

- When we randomly select one the the two literals in C and complement it, we are getting close to τ^* (i.e. moving one edge closer to node 0 on the line) with probability at least $\frac{1}{2}$. (If it turns out that both literal values disagree with τ^*, then we are getting closer to τ^* with probability $= 1$.)
- As we are proceeding in this random walk we might encounter another satisfying assignment which is all the better.
- It remains to bound the expected time to reach node 0 in a random walk on the line where on each random step, the distance to node 0 is reduced by 1 with probability at least $\frac{1}{2}$ and otherwise increased by 1 (but never exceeding distance n). This perhaps biased random walk is at least as good as the case where we randomly increase or decrease the distance by 1 with probability equal to $\frac{1}{2}$.

Claim:

The expected time to hit node 0 is at most $2n^2$.

- To prove the claim one needs some basic facts about Markov chains.