Online Matchings via Randomized Queueing

Thomas P. Hayes*

December 5, 2011

Abstract

We study a very simple and natural algorithm for finding a matching of "hats" with "men": line up the men in a random order, then, taking the hats in random order, assign each hat to the first man still in line who wants it, removing the man and hat.

The competitive ratio of this algorithm, defined as the expected size of the matching obtained, divided by the size of the maximum matching possible, is not known. We prove that, in the case of n men and n hats, where hat i is acceptable to man j whenever $i \leq j$, the competitive ratio is nearly optimal, specifically $1 - O(n^{-1/5})$. This input has been a "hard case" for a number of other online matching algorithms.

We conjecture that, in general, this algorithm has a competitive ratio of $1 - o(1)$ for any acceptability graph whose minimum degree tends to infinity. This may be the simplest known algorithm for "random arrival order online matching" for which such a conjecture has been made. We also conjecture that the worst-case competitive ratio of our algorithm is $1 - 1/e$, which is also optimal.

*Department of Computer Science, University of New Mexico, Albuquerque, NM 87108, U.S.A. Email: hayes@cs.unm.edu.

1 The Hats Problem

Suppose there are n men, and n hats, both labelled $1, \ldots, n$. We call the labels on the hats "ranks", and the labels on the men "selectivities", and say that the hat of rank i is acceptable to those men of selectivity $j \geq i$, and no others. Then there is exactly one perfect matching of men with hats, namely pair each man i with hat i. However, if we do not know the labels on the hats and men, and wish to pair the hats with men one at a time, in an online fashion, it is not obvious how to proceed.

[We call the above notion of acceptability the "1D ranked hats" setting, to distinguish it from more general settings, such as Online Matching in a bipartite graph, which we shall also discuss briefly.]

A very simple process for assigning hats to men proceeds as follows: place the men in a queue, and hand hats one by one to the first man in the queue. Whenever a man is handed a hat he finds acceptable, he takes it and goes home. Whenever a man is handed an unacceptable hat, he passes it to the next man in the queue, and remains in line. The algorithm terminates when every hat has either been taken home or been tossed into a "rejects" pile at the end of the queue. For convenience, we will refer to this as the "queueing" algorithm, Q.

Obviously the number of matched hats is a function of both the order of the men in the queue, and the order that the hats are passed out. For instance, if the men are in increasing order, or the hats are in decreasing order, then all n hats will be matched. On the other hand, if the men are decreasing order while the hats are in increasing order, then only half of the hats will be matched (the worst case).

In the present work, we analyze this process under the assumption that both the hats and the men are in random order. We denote this special case of Q by RQ, for "randomized queueing." Surprisingly, the expected size of the matching produced is nearly optimal: $(1 - o(1))n$. Indeed, our main result is as follows.

Theorem 1.1. *In the 1D ranked hats setting, with probability $1 - 1/n^2$, the matching obtained by RQ is of size $n - O(n^{4/5} \log(n))$.*

We also prove a somewhat weaker lower bound, that shows that the regret (defined as the fraction of wasted hats) is in fact inverse-polynomial in n.

Theorem 1.2. *In the 1D ranked hats setting, with probability $\Omega(1)$, the matching obtained by RQ is of size $n - \Omega(\sqrt{n})$.*

1.1 Motivation

Our interest in this problem is motivated by the problem of *online resource allocation,* where we want to allocate a set of goods to particular bidders subject to certain constraints, in order to maximize some objective function, such as revenue, or total utility.

For instance, in the *online matching* problem, there is an unknown bipartite n by n bipartite graph, G, and we receive vertices from the right-hand side one by one, discovering their neighbors on the left-hand side when they arrive. The goal is to build up as pair up as many of these arriving vertices as we can, in a one-to-one matching, where we must commit to the pairing of each arriving vertex, before discovering anything about the subsequent vertices.

For this problem, the best possible competitive ratio is known to be $1 - 1/e$. Indeed, a hard case for this problem is given by the bipartite graph of the Hats problem. Specifically, if the hats arrive in decreasing order, but the labels of the men are not known, then, as the eligible men are indistinguishable based on the hats seen so far, nothing better than random assignment is possible,

and it is an easy exercise to check that this leads to a matching of expected size asymptotically $(1 - 1/e)n$. A proof sketch is included in Section 6.

On the other hand, if the hats arrive in random order, our randomized queueing process shows that a competitive ratio approaching one is possible.

More generally, we boldly conjecture that, in the random arrival model, any bipartite graph having an $o(1)$ fraction of vertex degrees below any particular constant threshold C, the randomized queueing algorithm achieves a competitive ratio $1 - o(1)$.

2 Preliminaries

2.1 Efficiency and Regret

An alternative measure of the success of a hat-matching algorithm would be to ask, *"How close is the matching to the optimal matching?"* More concretely, we define the inefficiency of a matching S to be the sum, over all pairs $(h, m) \in S$, of the gap $m - h$. Note that this is a non-negative quantity, because hat h is acceptable to man m if and only if $m \geq h$.

Interestingly, this notion turns out to be more or less equivalent to our chosen measure of success, namely the competitive ratio, $|S|/n$, or, alternatively, the *regret*, $n - |S|$.

Proposition 2.1. *Consider the 1D ranked hats setting. Let S be any maximal matching, having inefficiency x and regret $R = n - |S|$. Then*

$$R^2 \leq x \leq R(n - R).$$

Proof. Pair up the unused hats and men arbitrarily, to complete S to a perfect matching of the complete graph $K_{n,n}$. Note that for this matching, the inefficiency is 0, since the sum of all hats equals the sum of all men. However, this also equals x plus R additional terms, all of which are negative by our maximality assumption— in any positive term, the hat is acceptable to the man, so the pair cannot be omitted from a maximal matching. Indeed, each of the discarded hats must be greater than each of the bareheaded men. Now, since the average of these negative terms is easily checked to fall in the range $[-(n - R), -R]$, the result follows immediately. □

As an immediate corollary, for any *maximal* matching, such as those produced by the queueing algorithm, the competitive ratio is $1 - o(1)$ if and only if the inefficiency is $o(n^2)$; that is, if on average, the errors $(m - h)/m$ must be $o(1)$.

One reason to be interested in this connection, apart from its intrinsic appeal, is that it seems to indicate an obstacle which must be overcome in order to prove Theorem 1.1. Namely, Theorem 1.1 is equivalent to the statement that, despite assigning the first few hats almost at random, the RQ algorithm very quickly begins assigning almost all the hats to owners whose selectivity is very nearly identical to the hat's rank. But, how can such a simple algorithm achieve this goal?

3 Monotonicity

In this section, we will prove three key monotonicity properties of the queueing algorithm. The first, Lemma 3.1, which applies even in the most general setting of an arbitary acceptability graph, says that adding extra men to the line, without rearranging the others, can never decrease the size of the matching obtained. The next two, in Lemma 3.2, are specific to the "ranked hats" setting, say that if we decrease the selectiveness of any man, or exchange adjacent men in line, to move the more selective man forward in line, we cannot decrease the size of the matching obtained.

For the purposes of these lemmas, we will relax our assumption that the sequence $\mathbf{m}$ of men, and the sequence $\mathbf{h}$ of hats, contain the n distinct ranks $1, \ldots, n$. Instead, we will allow arbitrary sequences of men and hats. We will take advantage of this relaxation in the proofs of Theorems 1.1 and 1.2 in the succeeding sections, in order to "round" the selectivities of men in line, and thereby reduce the number of distinct types of men and hats under consideration, while keeping the problem size the same.

We will use the notation $\mathbf{m} = (m_1, \ldots, m_k)$ to denote an ordered sequence of men, and $\mathbf{h} = (h_1, \ldots, h_\ell)$ to denote an ordered sequence of hats. When we are in the "ranked hats" setting, we will identify men with their selectivities, and hats with their ranks. Let $Q(\mathbf{m}, \mathbf{h})$ be the number of hats matched by the queueing process given the ordered sequences $\mathbf{m}$ and $\mathbf{h}$.

Lemma 3.1. *Consider the general bipartite matching setting, with an arbitrary acceptability graph. Let* $\mathbf{m}$ *be a sequence of* k *men, and let* $\mathbf{s}$ *be a subsequence (not necessarily consecutive) of* $\mathbf{m}$*, of length* $k - j$*. Let* $\mathbf{h}$ *be any sequence of hats. Then*

$$Q(\mathbf{m}, \mathbf{h}) - j \leq Q(\mathbf{s}, \mathbf{h}) \leq Q(\mathbf{m}, \mathbf{h}).$$

Proof. The proof is by induction on $\ell := \text{len}(\mathbf{h})$.

The base case, $\ell = 0$, is trivial. Now suppose $\ell > 0$. Look at what happens to the first hat h_1. If it is taken by the same man in $\mathbf{s}$ and $\mathbf{m}$, or is discarded in both cases, we immediately reduce to the inductive hypothesis, and the result follows.

Otherwise, hat h_1 is taken by a man in $\mathbf{m}$ who is not in $\mathbf{s}$, and either discarded or taken by a different man in $\mathbf{s}$. In either case, we maintain the subsequence relation between the two resulting shorter lines of men. In the former case, we have shrunk $\mathbf{m}$ by one and not $\mathbf{s}$, but also paired an extra hat in $\mathbf{m}$. In the latter case, we have shrunk both lines of men by one, and paired one hat each. Either way, the inductive hypothesis now supplies the desired inequality. □

The second monotonicity property, which is specific to the half-complete graph, says that increasing the selectiveness of some of the men can also only shrink the matching. By symmetry, the same applies to decreasing the quality of some of the hats. Put another way, decreasing the selectiveness of a man, or increasing the quality of a hat, can only increase the size of the matching.

To prove this, we will actually need to prove a slightly stronger statement: the size of our matching will not decrease if we decrease the selectiveness of a man, *and move him backwards in lne*, as long as we do not move him past any less selective man.

Lemma 3.2. *Consider the 1D ranked hats setting. Let* $\mathbf{m}$ *and* $\mathbf{m}'$ *be two sequences of men, and* $\mathbf{h}$ *any sequence of hats. Suppose that either*

(a) $m_j = m'_j$ *for all* $j \neq i$*, and* $m_i > m'_i$*, or*

(b) $m_j = m'_j$ *for all* $j \notin \{i, i+1\}$*, and* $m_i = m'_{i+1} < m_{i+1} = m'_i$*.*

Then, in either case,

$$Q(\mathbf{m}, \mathbf{h}) - Q(\mathbf{m}', \mathbf{h}) \in \{0, 1\}.$$

Proof. We will prove the result for conditions (a) and (b) simultaneously by induction on the length, ℓ, of $\mathbf{h}$. The base case, $\ell = 0$, is trivial. Suppose $\ell > 0$, and the result holds for all strings of length $< \ell$.

In case (a), if hat h_1 is either rejected by both $\mathbf{m}$ and $\mathbf{m}'$, or is accepted by the same man $m_j = m'_j$ in both, then either $j = i$, and we are done because the new $\mathbf{m}$ and $\mathbf{m}'$ are identical, or, otherwise, we are done by induction, because the new $\mathbf{m}$ and $\mathbf{m}'$ still satisfy condition (a).

Two possibilities remain in case (a), namely:

(i) hat h_1 is accepted by man m_i, but rejected by $\mathbf{m}'$. In this case, the new $\mathbf{m}$ is a subsequence of $\mathbf{m}'$, and so Lemma 3.1 implies that $\mathbf{m}$ will be behind by 0 or 1 for the rest of the game, which, together with having successfully paired hat h_1, puts $\mathbf{m}$ ahead by 0 or 1 overall.

(ii) hat h_1 is accepted by man m_i, and by some later man m'_j, where $j > i$. In this case, the new $\mathbf{m}$ can be obtained from the new $\mathbf{m}'$ by a sequence of improvements: first increase m'_i to equal m_j—an improvement of type (a)—then swap position i with $i+1$, $i+1$ with $i+2$, $\ldots$, $j-1$ with j, each an improvement of type (b), because otherwise hat h_1 would have been matched differently in $\mathbf{m}'$. This proves that $Q(\mathbf{m}, \mathbf{h}) \geq Q(\mathbf{m}', \mathbf{h})$. To see that these values can differ by at most one, note that we can make $\mathbf{m}$ and $\mathbf{m}'$ equal by adding m'_i to $\mathbf{m}$ in between positions i and $i+1$, and by adding m_i to $\mathbf{m}'$ in between positions $i-1$ and i. But, Lemma 3.1 says these changes will affect $Q(\mathbf{m}, \mathbf{h}) - Q(\mathbf{m}', \mathbf{h})$ by adding an element of $\{0, 1\} + \{0, -1\} = \{-1, 0, 1\}$. This completes the analysis of case (a).

In case (b), if hat h_1 is rejected by $\mathbf{m}$ and $\mathbf{m}'$, or accepted by the same man $m_j = m'_j$, where $j \notin \{i, i+1\}$, then we are done by induction, since the new $\mathbf{m}$ and $\mathbf{m}'$ still satisfy condition (b). Otherwise, one of two remaining possibilities must occur:

(i) h_1 is accepted by m_i and by m'_i. In this case, we are done by induction, because the new $\mathbf{m}$ and $\mathbf{m}'$ satisfy condition (a). Or,

(ii) h_1 is accepted by m'_i and by m_{i+1}. In this case, we are done because the new $\mathbf{m}$ and $\mathbf{m}'$ are equal.

□

4 The Upper Bound: Proof of Theorem 1.1

In this section, we will prove that the regret for RQ is, with high probability, $O^*(n^{4/5})$.

Let $m_1, \ldots, m_n$ denote the full sequence of men $1, \ldots, n$. First, we modify this list by rounding each man m_i down to the nearest integer multiple of $n^{4/5}$. By Lemma 3.2, this can only make the matching smaller, regardless of the sequence of hats.

Second, we modify this list further, again pessimistically, in order to load the front of the list with "buffers", sorted in decreasing order of selectivity. Done properly, these buffers will persist throughout the running of the algorithm, and will ensure that we actually achieve the optimal pairing, given our new set of men and hats.

We now show how to define such buffers in a non-adaptive way, without reference to the ordering of the later part of the list. Despite this restriction, we can still achieve the desired optimality with high probability.

For notational convenience, define a "height" $h = n^{4/5}$, and "width" $w = Cn^{3/5}\log(n)$, where C is a suitably chosen constant. For convenience, we will assume these values are integers. Now, replace every m_j with

$$m'_j = \min\left\{m_j, \left\lceil \frac{j}{w} \right\rceil h\right\}.$$

This rounds down about half of the first $wn/h = Cn^{4/5}\log(n)$ men to particular target multiples of h. At a slight additional cost, we could remove the other half of the first wn/h men, thereby achieving a complete sorting of the initial men. However, this is not necessary.

Note that, again, by our monotonicity lemmas, the new sequence $\mathbf{m}'$ of men can only do worse on any sequence of hats than the original $\mathbf{m}$.

Now, let $1 \leq i < n/h - 1$, and for a moment, ignore all men in $\mathbf{m}'$ except those of selectivity ih and $(i+1)h$. First, note that, with high probability, the number of buffered men of selectivity ih exceeds the number of buffered men of selectivity $(i+1)h$ by about $wh/n = Cn^{2/5}\log(n)$. Since the ordering of men beyond the buffered zone is uniformly random, and there are about $2h = 2n^{4/5}$ of these men in all, it is very likely (probability can be made $> 1 - 1/n^2$, for suitable choice of C) that every initial segment of the line contains at least $wh/(2n)$ more men of selectivity ih than $(i+1)h$. Taking a union bound over all i, the failure probability remains less than $1/n$, and so such failures can affect the expected size of our matching by at most 1.

Finally, the same argument shows that, with probability $\geq 1 - 1/n$, every prefix of the sequence of hats has within $wh/(2n)$ of the same number of hats in all the ranges $[ih, (i+1)h)$, and so none of the buffers will ever be exhausted while men of that type remain. Therefore, the matching achieved will actually be the optimal matching between the men in $\mathbf{m}'$ with the hats in $\mathbf{h}$. Since the initial rounding down of the men's selectivities to the next multiple of h changed the size of the optimal matching by exactly $h = O(n^{4/5})$, and the second rounding touched only $wn/h = O(n^{4/5}\log(n))$ mens preferences, we have achieved, with high probability, a matching of size $n - O(n^{4/5}\log(n))$.

5 The Lower Bound: Proof of Theorem 1.2

In this section, we will prove that the expected regret for RQ on randomized inputs is $\Omega(\sqrt{n})$. Specifically, we show that the expected regret is at least one-fourth of the expected maximum excess of heads in a uniformly random sequence of n coin flips. By "excess", I mean $\max\{0, \#\text{ heads} - \#\text{ tails}\}$. Since this expectation is $\Theta(\sqrt{n})$, this is a lower bound on the expected regret.

To see this note that, by our monotonicity lemmas, the size of our matching only increases if we increase the worst hat acceptable to each man to the next multiple of $n/2$. After doing this, half of the men have a selectivity of n, and will take any hat, while the other half have a selectivity of $n/2$, and will take only half of the hats. Let k be such that the excess of unselective men in the first k men is maximized. With probability at least half the first k hats include at least half high-quality hats. In this case, since all the unselective men in the first k will certainly be paired with hats among the first k, at least $(excess/2)$ unselective men will be paired with high-quality hats. But this is a lower bound on the regret, since the number of unpaired selective men equals the number of unselective men paired with high-quality hats.

6 Worst-case Behavior

It follows from our monotonicity lemmas that, for any ordering of men, the worst-case ordering of hats is the ascending sequence $1, 2, \ldots, n$. In this case, under a random ordering of the men, the expected competitive ratio approaches $1 - 1/e$.

To see this, observe that, with the hats in ascending order, each man who rejects a hat can be removed from line without changing the size of the matching. This means that our algorithm simplifies to removing men from the front of the line, up to and including the first one willing to take the current hat.

For now, let us ignore the fact that the n men are distinct, and instead allow each man to have a selectivity uniformly and independently selected from $\{1, \ldots, n\}$. In this case, the expected number of men removed from line to match hat i is $\frac{n}{n+1-i}$. Summing this over the first k hats, so that the sum is about n, we have

$$\sum_{i=1}^{k} \frac{n}{n+1-i} = n(H(n) - H(n+1-k)) \sim n(\log(n/(n-k)))$$

which equals n when $k = (1 - 1/e)n$. Here $H(i) = \sum_{j \leq i} 1/j$ denotes the harmonic series, and $\sim$ denotes asymptotic equality.

7 Further Remarks and Discussion

We have proven that RQ performs nearly optimally when the acceptibility graph is the half-complete graph. This is a promising first step, since this graph seems to be a frequent "hard case" for many different online resource allocation settings. As a next step, I would like to see that RQ performs well for some broader classes of acceptability graphs. The main obstacle to generalizing the current approach seems to be that Lemma 3.2 is very specific to the total orderings on the men and hats, that we enjoy in the 1D ranked hats setting.

As far as we know, RQ may achieve $1 - o(1)$ competitive ratio for any acceptability graph with almost all nodes having degree $\omega(1)$.

An intermediate class of acceptability graphs which may be of interest to study, is a higher-dimensional generalization of the 1D ranked hats setting. Although there are various possibilities, we will focus on one, corresponding to the usual elementwise definition of vector comparison, $\leq$. Suppose every hat and man has a secret label in $[0, 1]^d$, for some $d \geq 2$. Hat h is acceptable to man m if and only if $h_i \leq m_i$ for $1 \leq i \leq d$. We might assume the graph G is generated by sampling labels for the hats and for the men uniformly at random. Preliminary computer simulations suggest that RQ continues to perform well in this setting.

It is worth noting that our Lemma 3.2 does not generalize to this context. In particular, even for $d = 2$ and $n = 2$, it is possible to increase the selectivity of one man in line, and thereby improve the resulting matching.

One appealing feature of the RQ algorithm is that it is a *truthful* mechanism. That is, if the goal of each player is to maximize his chance of being assigned an acceptable hat, it is a strictly dominant strategy for each player to honestly report his set of acceptable hats, regardless of what the other players say. In settings where the players might try to "game the system", this could be an important consideration.