An Optimal Deterministic Algorithm for Online b-Matching

*Bala Kalyanasundaram** *Kirk R. Pruhs*†

Abstract

We study the online unweighted b-matching problem where at most $b \geq 1$ requests can be matched to any server site. We present a deterministic algorithm BALANCE whose competitive ratio is $1 - \frac{1}{(1+\frac{1}{b})^b}$. We show that the competitive ratio of every deterministic online algorithm is at least $1 - \frac{1}{(1+\frac{1}{b})^b}$. Hence, BALANCE is optimally competitive, including low order terms. For large b, the competitive ratio of BALANCE approaches $1 - \frac{1}{e} \approx .63$.

1 Introduction

We consider the natural online version of the well-known b-matching problem on an unweighted bipartite graph $G = (S, R, E)$, where S and R are the vertex partitions and E is the edge set. At the ith unit of time, $1 \leq i \leq n$, the vertex $r_i \in R$ and all the edges incident to r_i are revealed to the online algorithm $\mathcal{A}$. $\mathcal{A}$ then must either decline to ever service r_i, or irrevocably select a site s_k adjacent to r_i in G to service r_i. No server site may be used more than b times. Hence, it may well not be possible to service every request. The goal of the online algorithm is to maximize the number of requests that it services. We analyze this problem using the standard competitive ratio. For this problem, the competitive ratio of an online algorithm $\mathcal{A}$ is the supremum over all possible instances I, of the cardinality of the matching constructed by $\mathcal{A}$ on I divided by the maximum cardinality

**kalyan@cs.pitt.edu*, Computer Science Dept., University of Pittsburgh, Pittsburgh, PA 15260, Supported in part by NSF under grant CCR-9202158.

†*kirk@cs.pitt.edu*, Computer Science Dept., University of Pittsburgh, Pittsburgh, PA 15260, Supported in part by NSF under grant CCR-9209283.

matching in I. Note that the instance I specifies G as well as the order in which the r_i's appear.

As one example application, consider the problem of assigning client computers to support stations studied by Grove, Kao, Krishnan and Vitter [1]. In this problem each support station has a maximum range of service and a limit on the number of clients that it can support. Clients arrive over time and must each be assigned to a support station that is not too distant and that is not fully utilized. So the competitive ratio will be the fraction (relative to the maximum matching) of the clients that can be guaranteed support without reassignment.

1.1 Related Results

Karp, Vazirani, and Vazirani [5] give the following results for online bipartite matching, the special case of b-matching where $b = 1$. It is not hard to observe that any deterministic algorithm that never refuses to match a request, if it is possible to do so, is $\frac{1}{2}$-competitive, and that no deterministic algorithm can be better than $\frac{1}{2}$-competitive. [5] give a randomized algorithm RANKING whose competitive ratio is $1 - \frac{1}{e} + o(1)$ against an oblivious adversary that must specify the input a priori. RANKING initially selects uniformly at random a linear order of the server sites, and then matches each request with the the first available server. [5] show that the competitive ratio of every randomized algorithm is at least $1 - \frac{1}{e} + o(1)$. Hence, RANKING is optimally competitive, up to low order terms.

Kao and Tate [4] extended the results of [5] by considering the case where requests appear in batches. They showed that the results of [5] cannot be improved even if request appear in batches of size $o(n)$ each.

Grove, Kao, Krishnan and Vitter [1] consider the problem of maintaining a maximum cardinality matching with a minimal number of reassignments of servers in the special case that the maximum degree of each $r_i = 2$. [1] show that the greedy algorithm, that switches assignments along the shortest augmenting path, is $O(\log n)$-competitive, i.e. the greedy algorithm makes at most $O(\log n)$ times as many reassignments as the optimal number of reassignments required to maintain a maximum cardinality matching. [1] show the competitive ratio of every deterministic algorithm for this problem is $\Omega(\log n)$. [1] also give some results for case that requests may depart.

Results for online weighted matching problems, on graphs where the edge weights satisfy the triangle inequality, can be found in [2, 3, 6]. In particular, an optimally competitive deterministic algorithm for the case

$b = 1$ can be found in [2, 6]. In [3], the case of arbitrary b is studied under the assumption that the online algorithm has more servers per site than the adversary. Note that the triangle inequality is not generally satisfied by non-edges in unweighted matching.

1.2 Summary of Results

In this paper, we give the following results for online b-matching. In section 2, we show that the competitive ratio of any deterministic online algorithm for this problem is at least $1 - \frac{1}{(1+\frac{1}{b})^b}$. This lower bound holds even if each request has degree at least at least $\frac{m}{3b}$. In section 3, we give a simple deterministic algorithm BALANCE with competitive ratio $1 - \frac{1}{(1+\frac{1}{b})^b}$. Hence, BALANCE is optimally competitive, including low order terms. As b grows, $1 - \frac{1}{(1+\frac{1}{b})^b}$ approaches $1 - \frac{1}{e} \approx .63$. In response to a request r_i, BALANCE selects an arbitrary server site among all server sites adjacent to r_i in G that have used a minimum number of servers to date. The idea of trying to balance the number of servers used per site can also be found in an online matching algorithm given in [3].

As in [3, 7], we also compare the performance of the online algorithm against the performance of an offline algorithm with fewer servers. This will give us an idea on how well BALANCE peforms against a less malicious input given by the adversary. We show that BALANCE, with αb servers, is $1 - \frac{1}{(1-\frac{1}{b})^{\alpha b}}$ competitive against an offline adversary with b servers per site. We also show that this ratio is optimal for deterministic algorithms. Here α must be an integer.

2 The Lower Bound

In order to prove the desired lower bound, we first present an adversary. Throughout our arguments we think of the server site s_i as containing b different servers that handle the requests.

Adversary : Let $\mathcal{A}$ be the given deterministic online algorithm.

There are $(b+1)^b$ server sites with exactly b servers per site, and there will be $b \cdot (b+1)^b$ requests. The requests are partitioned into groups. The first group R_1 consists of the first $b(1+b)^{b-1}$ requests, the second group R_2 consists of the next $b^2(1+b)^{b-2}$ requests, and in general, R_i, i, $1 \le i \le b$, contains the $b^i(1+b)^{b-i}$ requests from request numbered $1+\sum_{j=1}^{i-1} b^j(1+b)^{b-j}$

to the request numbered $\sum_{j=1}^{i} b^j(1+b)^{b-j}$, inclusive. R_{b+1} contains the last b^{b+1} requests.

The adversary maintains $b+1$ sets $S_1, S_2, \ldots S_{b+1}$ of server sites such that $S_i \supset S_{i+i}$. Initially, S_1 is the set S of all server sites. The first b groups are handled in the following manner. The adversary makes a request $r_j \in R_i$ adjacent to those vertices in S_i that have not yet answered a request in R_i. If $\mathcal{A}$ uses s_k to service r_j then s_k is added to S_{i+1}. In R_{b+1}, each request can be matched to any server site in S_{b+1}. So the following is the adversary's algorithm for requests in R_i, $i \leq b$. Note that for every server site s_j, the set M_j is initialized to the empty set. Assuming that the first $i-1$ phase has been completed, consider the ith phase where $i \leq b$.

$S_{i+1} = \emptyset$
for each request $r_j \in R_i$ in chronological order
    reveal r_j and edges from r_j to sites in $S_i - S_{i+1}$.
    if $\mathcal{A}$ matches r_j to a server at site $s_k \in S_i - S_{i+1}$.
        Add s_k to S_{i+1}.
    else { $\mathcal{A}$ opts not to match r_j }
        choose some arbitrary site $s_k \in S_i - S_{i+1}$.
        Add s_k to S_{i+1} and r_j to M_k.
endfor

Lemma 1 *For each i $(1 \leq i \leq b)$,*
(a) $S_i \supset S_{i+1}$,
(b) $|S_{i+1}| = |R_i| = b^i(1+b)^{b-i}$, and
(c) $|S_i| - |S_{i+1}| = b^{i-1}(1+b)^{b-i}$,

For ease of notation, let us assume that the set S_{b+2} is empty.

Lemma 2 *There exists an offline perfect matching that, for each i satisfying $1 \leq i \leq b+1$, matches every request in R_i to a server site in $S_i - S_{i+1}$.*

Proof Sketch: Notice that every request in R_i can be matched to any server in sites from $S_i - S_{i+1}$. It suffices to show that $b \cdot |S_i - S_{i+1}| \geq |R_i|$ since each site has b servers. For $1 \leq i \leq b$ this follows from lemma 1. For $i = b+1$ it follows since $|S_{b+1}| = b^b$, $|S_{b+2}| = 0$, and $|R_{b+1}| = b^{b+1}$ ■

Lemma 3 *The number of requests matched by $\mathcal{A}$ does not exceed $\sum_{i=1}^{b} |R_i|$.*

Proof Sketch: Consider the sites in S_{b+1}. Since each site in S_{b+1} went through b phases starting from the set S_1, the maximum number of requests from R_{b+1} that the servers from sites in S_{b+1} can match is $\sum_{s_k \in S_{b+1}} |M_k|$. The number of requests in $\cup_{i=1}^{b} R_i$ matched by $\mathcal{A}$ to servers in $S - S_{b+1}$ is $\sum_{i=1}^{b} |R_i| - \sum_{s_k \in S} |M_k|$. The result follows since $S \supseteq S_{b+1}$. ■

Theorem 4 *The competitive ratio of any deterministic online algorithm for the b-matching problem is at most* $1 - \frac{1}{(1+\frac{1}{b})^b}$.

Proof Sketch: Let $\mathcal{A}$ be the given online algorithm, and apply the adversary described in this section. Combining lemma 3 and lemma 2 we get that the competitive ratio is

$$\frac{\sum_{i=1}^{b} |R_i|}{\sum_{i=1}^{b+1} |R_i|} = \frac{\sum_{i=1}^{b} |R_i|}{|R_{b+1}| + \sum_{i=1}^{b} |R_i|}$$

Substituting $\sum_{i=1}^{b+1} |R_i| = b(b+1)^b$, $|R_{b+1}| = b^{b+1}$, and $\sum_{i=1}^{b} |R_i| = b(b+1)^b - b^{b+1}$ yields the claimed bound. ■

Note that in this lower bound each request has degree at least $\frac{|S|}{3b}$.

Theorem 5 *Assume that the adversary has b servers per site while the online algorithm has αb servers per site, where α is some positive integer. In this model, the competitive ratio of any deterministic online algorithm for the b-matching problem is* $1 - \frac{1}{(1+\frac{1}{b})^{\alpha b}}$.

Proof Sketch: We modify the adversary in the following way. Let the number of server sites be $(b+1)^{\alpha b}$ and for all $1 \leq i \leq \alpha b$, R_i consists of $b^i(1+b)^{\alpha b - i}$ requests. $R_{\alpha b+1}$ consists of $b^{\alpha b+1}$ requests. The rest of the argument go through as before. ■

3 The Algorithm Balance

In this section we present the algorithm BALANCE, and show that the competitive ratio of BALANCE **exactly** matches the deterministic lower bound from the previous section.

Algorithm BALANCE: Each request r_j is served by an arbitrary adjacent server site that has a maximum number of servers remaining.

Definition 6
(a) Let OPT be an arbitrary maximum cardinality matching.
(b) Let B be the set of request vertices matched in OPT.
(c) Let X be the set of requests in B not matched by BALANCE.
(d) If the request r_j is matched by BALANCE *to a server site that has already used $i-1$ servers, then we say that the* rank *of r_j is i.*
(e) For $1 \leq i \leq b$, let R_i be the set of all requests with rank i.
(f) For $1 \leq i \leq b$, let $M_i \subseteq R_i$ be the set of all requests with rank i that are not in B.
(g) For $1 \leq i \leq b+1$, let S_i be the set of server sites in the maximum matching that service requests in $X \cup (\cup_{j=i}^{b}(R_j - M_j))$.

Lemma 7 *The competitive ratio of* BALANCE *is*

$$\frac{\sum_{i=1}^{b}|R_i|}{|X| + \sum_{i=1}^{b}|R_i - M_i|} = \frac{\sum_{i=1}^{b}|R_i|}{(|X| - \sum_{i=1}^{b}|M_i|) + \sum_{i=1}^{b}|R_i|}$$

Lemma 8 *For any i satisfying $2 \leq i \leq b+1$, each $s_k \in S_i$ is matched by* BALANCE *to at least $i-1$ requests. Hence, $|R_{i-1}| \geq |S_i|$.*

Proof Sketch: First consider the case that s_k is matched in OPT with an $r_j \in X$. Then since BALANCE didn't match r_j, it must be the case that BALANCE has used all the servers from s_k. Hence, s_k is adjacent to a rank $i-1$ request.

Now suppose that in OPT the site s_k matches a request $r_a \in R_j$, $j \geq i$. Notice that r_a is also matched by BALANCE to some server at site s_b. Note that it may be the case $s_k = s_b$. Since r_a can be matched to either s_k or s_b, it must be the case that BALANCE has already matched a $j-1$st rank request to s_k. The result then follows since $j \geq i$. ∎

Lemma 9 *For any $1 \leq i \leq b+1$,*

$$|S_i| \geq \frac{1}{b}(|X| - \sum_{j=i}^{b}|M_j| + \sum_{j=i}^{b}|R_j|)$$

Proof Sketch: This follows from the definition of S_i, the fact that each site has at most b servers, the fact that $M_i \subseteq R_i$, and the fact that the R_i's are disjoint. ∎

Lemma 10 *For $0 \leq i \leq b$,*

$$b \cdot |S_{b-i+1}| \geq (1 + \frac{1}{b})^i \cdot (|X| - \sum_{j=b-i+1}^{b} |M_j|)$$

Proof Sketch: We prove this by induction on i. First consider the case $i = 0$. Since at most b requests can be matched to servers at any site, we have $b \cdot |S_{b+1}| \geq |X|$.

Assume that the induction hypothesis holds for $i \leq k$. We now want to show that it also holds for $i = k + 1$. Applying lemma 9 we have

$$b \cdot |S_{b-(k+1)+1}| = |S_{b-k}| = |X| - \sum_{j=b-k}^{b} |M_j| + \sum_{j=b-k}^{b} |R_j|$$

Applying lemma 8, that is $|R_j| \geq |S_{j+1}|$, we get,

$$\sum_{j=b-k}^{b} |R_j| \geq \sum_{j=b-k}^{b} |S_{j+1}| = \sum_{i=0}^{k} |S_{b-i+1}|$$

Applying the induction hypothesis we get,

$$\begin{aligned} b \cdot \textstyle\sum_{i=0}^{k} |S_{b-i+1}| &\geq \textstyle\sum_{i=0}^{k}(1 + \frac{1}{b})^i \cdot (|X| - \sum_{j=b-i+1}^{b} |M_j|) \\ &\geq \textstyle\sum_{i=0}^{k}(1 + \frac{1}{b})^i \cdot (|X| - \sum_{j=b-k+1}^{b} |M_j|) \\ &= (|X| - \textstyle\sum_{j=b-k+1}^{b} |M_j|) \cdot \sum_{i=0}^{k}(1 + \frac{1}{b})^i \\ &\geq (|X| - \textstyle\sum_{j=b-k}^{b} |M_j|) \cdot \sum_{i=0}^{k}(1 + \frac{1}{b})^i \\ &= (|X| - \textstyle\sum_{j=b-k}^{b} |M_j|) \cdot b \cdot ((1 + \frac{1}{b})^{k+1} - 1) \end{aligned}$$

Therefore, we get

$$\begin{aligned} b \cdot |S_{b-(k+1)+1}| &\geq (|X| - \textstyle\sum_{j=b-k}^{b} |M_j|) + (|X| - \sum_{j=b-k}^{b} |M_j|) \cdot [(1 + \frac{1}{b})^{k+1} - 1] \\ &= (1 + \frac{1}{b})^{k+1} \cdot (|X| - \textstyle\sum_{j=b-k}^{b} |M_j|) \end{aligned}$$

■

Theorem 11 *The competitive ratio of* Balance *is* $1 - \frac{1}{(1+\frac{1}{b})^b}$.

Proof Sketch: Applying lemma 8 and lemma 10, we get

$$\begin{aligned}
\sum_{i=1}^{b} |R_i| &\geq \sum_{i=2}^{b+1} |S_i| \\
&= \sum_{i=0}^{b-1} |S_{b-i+1}| \\
&\geq \frac{1}{b} \sum_{i=0}^{b-1} (1+\frac{1}{b})^i (|X| - \sum_{j=b-i+1}^{b} |M_j|) \\
&\geq \frac{1}{b} \cdot (|X| - \sum_{j=1}^{b} |M_j|) \cdot \sum_{i=0}^{b-1} (1+\frac{1}{b})^i \\
&\geq (|X| - \sum_{j=1}^{b} |M_j|) \cdot ((1+\frac{1}{b})^b - 1)
\end{aligned}$$

Applying the bound to the competitive ratio computed in lemma 7, yields the desired bound. ■

We now claim that BALANCE is optimally competitive against an adversary with fewer servers.

Theorem 12 *The competitive ratio of* BALANCE, *with αb servers per site, against an adversary, with only b servers per site, is* $1 - \frac{1}{(1+\frac{1}{b})^{\alpha b}}$.

Proof Sketch: The above arguments need to be modified by allowing the rank of a request to range from 1 to αb. By appropriately extending the definitions, the same argument will go through. ■

4 Conclusion

We show that the algorithm BALANCE is optimal optimally competitive among deterministic algorithms for the online b-matching problem. The obvious open question is to find an optimally competitive randomize algorithm. We are currently analyzing the following algorithm that is a mix of RANKING and BALANCE. Initially, uniformly at random linearly order the server sites. Then run BALANCE. If there is more than one site with a minimum number of servers that can handle a request, break the tie by selecting the highest ranked site.

In the case where the number servers per site vary from site to site, the competitive factor of BALANCE does not exactly match with that of the lower bound. Is there a deterministic algorithm with competitive factor $1 - \frac{1}{(1+\frac{1}{b_a})^{b_a}}$ where b_a is the average number of servers per site used by OPT?

References

[1] E. Grove, M. Kao, P. Krishnan, and J. Vitter, "Online Perfect Matching and Mobile Computing", *Proceedings of the Workshop on Algorithms and Data Structures*, 1995.

[2] B. Kalyanasundaram, and K. Pruhs, "Online weighted matching", *Journal of Algorithms*, **14**, 478–488, 1993.

[3] B. Kalyanasundaram, and K. Pruhs, "The Online Transportation Problem", *Proc. of European Symposium on Algorithms,* Vol. 979 (*LNCS*), 484–493, 1995.

[4] M. Kao, and S. Tate, "Online Matching with Blocked Input", *Information Processing Letters,* 38, 113–116, 1991.

[5] R. Karp, U. Vazirani, and V. Vazirani, "An Optimal Algorithm for Online Bipartite Matching", *STOC*, 352–358, 1990.

[6] S. Khuller, S. Mitchell, and V. Vazirani, "On-line algorithms for weighted matchings and stable marriages", *Theoretical Computer Science*, **127**(2), 255–267, 1994.

[7] D. Sleator and R. Tarjan, "Amortized efficiency of list update and paging rules", *Communications of the ACM*, **28**, 202–208, 1985.