The Online Matching Problem on a Line[⋆]

Elias Koutsoupias[1,2] and Akash Nanavati[2]

[1] Department of Informatics, University of Athens,
Panepistimiopolis, Athens 15784, Greece.
elias@di.uoa.gr
[2] Computer Science Department, University of California Los Angeles,
Los Angeles, CA 90095, USA.
akash@cs.ucla.edu

Abstract. We study the *online matching* problem when the metric space is a single straight line. For this case, the offline matching problem is trivial but the online problem has been open and the best known competitive ratio was the trivial $\Theta(n)$ where n is the number of requests. It was conjectured that the generalized *Work Function Algorithm* has constant competitive ratio for this problem. We show that it is in fact $\Omega(\log n)$ and $O(n)$, and make some progress towards proving a better upper bound by establishing some structural properties of the solutions. Our technique for the upper bound doesn't use a potential function but it reallocates the online cost in a way that the comparison with the offline cost becomes more direct.

1 Introduction

We study the online version of the weighted bipartite matching problem in which the nodes of one partition, called the *servers*, are known in advance and the nodes of the other partition, the *requests*, arrive online and must be matched immediately. The objective is to minimize the cost of the matching. The more general problem in which the nodes of both partitions arrive online and must be matched as soon as possible is not essentially different; by appropriate partitioning the sequence of points, it is simply a repeated version of the simple version and all our results apply to the more general problem. Of special interest is the metric matching problem in which the points, both servers and requests, lie in some metric space and especially in a Euclidean space.

The offline matching problem is one of the most fundamental algorithmic problems and it has played a central role in the development of the theory of algorithms [2, 3, 10]. The online version of the problem has also been studied before.

Karp, U. Vazirani, and V. Vazirani [8] were the first to study some online version of the matching problem. They studied randomized algorithms for online matching in an unweighted bipartite graph when the optimum matching was a perfect matching. Khuller, Mitchell, and Vazirani [9] studied the online version

[⋆] Research supported in part by NSF.

of minimum bipartite matching. They showed that for non-metric spaces the competitive ratio is unbounded. They also proposed a natural algorithm, *Permutation*, which has optimal competitive ratio for arbitrary metric spaces; its competitive ratio is $2n - 1$, where n is the number of requests. The same algorithm was studied independently by Kalyanasundaram and Pruhs [5, 7]. They conjectured that the Work Function Algorithm has constant competitive ratio (see their review article [6]).

Unlike the minimization version of the online matching the maximization version has been completely settled. For non-metric spaces the competitive ratio is unbounded [9] and for metric spaces the greedy algorithm is optimal and has competitive ratio 3 [5].

Here we are concerned with the special case of the metric online minimum matching where the servers and requests lie on a line (1-dimensional Euclidean space). This is perhaps the most interesting case of online matching for many reasons. First, the offline matching problem for this case is trivial and therefore competitive analysis appropriately captures the deterioration of performance (that is, the approximation ratio or competitive ratio) due to lack of complete information. Second, it is a natural generalization of other fundamental online problems such as the cow-path problem (also known as the bridge problem) [1]. Third, versions of online matching problem play a central role in electronic markets where "buyers" and "sellers" arrive online, name their prices, and must be "matched". One can view the prices as points on a line that must be matched online. The version of the matching problem that we study in this work together with the analogous maximization version, are the simplest, pure abstractions of these online matching problems.

It is easy to show that no online algorithm can have competitive ratio less than 9 since the online matching problem is a generalization of the cow-path problem (see the beginning of the Section 3). Very recently Fuchs, Hochstättler, and Kern [4] gave a lower bound of 9.001 which suggests that the problem is not simply a disguised generalization of the cow-path problem. It has been conjectured [6] that the matching problem on the line and in particular the Work Function Algorithm (WFA) have constant competitive ratio. Here we disprove the conjecture and establish a lower bound of $\Omega(\log n)$ where n is the number of requests. We also show an upper bound $O(n)$ for the competitive ratio of WFA. The bound by itself is weak but the proof technique is more important. It exhibits structural properties of the solutions that can lead to improved bounds. Using these properties we show that the competitive ratio is no more that the height of the online matching (i.e., the maximum number of lines through a point). We can also establish a tight upper bound $(O(\log n))$ for some special cases (restricted adversaries), but due to lack of space we omit these results from this abstract.

Outline

In Section 2 we define the WFA and prove some properties that are useful in general and in particular they simplify the argument for the lower bound. We then give the relatively simple lower bound (Section 3).

To prove the upper bound, we first establish some useful properties and invariants of the WFA (beginning of Section 4). We now want to compare the cost of the WFA to the optimal matching (*opt*). However, the optimal matching can use different set of servers than WFA. We show (Subsection 4.1) that this is not a serious problem: the WFA uses an almost optimal set of servers in the sense that the optimal offline matching involving the servers of WFA (*popt*) is within a constant factor from the optimal (*opt*). We need therefore only to compare the online cost with *popt*. We don't know how to do it directly though. As it is frequently the case, it is not easy to apply the potential function method to online algorithms with non-constant competitive ratio. Indeed, we don't use a potential function but we find a way to reallocate the online cost in such a way that it becomes easier to compare it to *popt*. A major obstacle is that *popt* is not monotone: more requests can decrease the matching. Furthermore, new requests may decrease the number of *popt* lines that cross a particular point (see β_x in the Section 2). We get around these problems by introducing a monotone quantity, the alive parts of the offline matchings (*alive*), which counts each interval with multiplicity roughly equal to the maximum number of *popt* lines that crossed it in the past. We then show that *alive* is also within a constant factor from *popt* (Subsection 4.2). Finally, it is easy to bound the online cost by *alive*, and thus indirectly with the optimum (Subsection 4.3).

We conclude and mention some open problem in Section 5.

2 Properties of the Work Function Algorithm

The metric online matching problem is defined on a metric space $\mathcal{M}$ endowed with distance d. A (possibly infinite) subset of $S \subseteq \mathcal{M}$ is the set of servers. Some of these servers are going to be matched with a sequence of requests $\{r_1, r_2, \ldots\} \subseteq \mathcal{M}$[3]. Let $R_n = \{r_1, \ldots, r_n\}$ denote the set of the first n requests and let $S_n = \{s_{r_1}, \ldots, s_{r_n}\}$ denote the set of servers used by the online algorithm to match these requests.

For the problem we study here the metric space is the 1-dimensional Euclidean space. Let $M(A, R)$ denote the optimal (minimum) matching between sets A and R with $|A| = |R|$. Clearly $M(A, R)$ can be obtained by matching the leftmost request to the leftmost server and recursing in the same fashion. There may be other optimal matchings but we will use the notation $M(A, R)$ for this particular matching. For simplicity, we also denote the weight of the optimal matching by $M(A, R)$.

Given the definition of the matching $M(A, R)$ we define

$$\beta_x(A, R) = |\{a : a \in A \text{ and } a < x\}| - |\{r : r \in R \text{ and } r < x\}|,$$

[3] Here we assume that all servers and requests are distinct points. It is obvious that the general case of allowing multiple servers and/or requests on the same point is the limit case when some distances tend to zero. For simplicity we sometimes allow multiple servers/requests on points though. Also, all sets are assumed to be multisets.

which is the number of lines in $M(A, R)$ that cross point x with appropriate sign.

We denote singletons $\{a\}$ by their unique element a, and the multiset consisting of k copies of a by a^k. In particular, $B + a^k$ will denote the multiset that results if we add k copies of a to multiset B.

Definition 1 (Work Function Algorithm). *The* Generalized Work Function Algorithm *(*γWFA*) matches request r_n to an unmatched server $s_{r_n} \in S - S_{n-1}$ that minimizes the expression: $\gamma M(S_{n-1} + s_{r_n}, R_{n-1} + r_n) + d(s_{r_n}, r_n)$.*

We state the following easy result without proof.

Proposition 1. *For $\gamma = 0$ this is the greedy algorithm with competitive ratio at least $2^n - 1$, and in general, for $0 \leq \gamma < 1$, competitive ratio is exponential, at least $(\zeta^n - 1)/(\zeta - 1)$, where $\zeta = 2/(\gamma + 1)$. The case $\gamma = 1$ is known simply as the Work Function Algorithm, with competitive ratio at least n. The case $\gamma = \infty$ (i.e. ignore $d(x, r)$ and choose s_{r_n} that minimizes $M(S_{n-1} + s_{r_n}, R_{n-1} + r_n)$) is the retrospective algorithm also known as the* Permutation *algorithm [5] with competitive ratio at least n (and at most $2n - 1$).*

Our analysis applies to any $\gamma > 1$. There are indications that the minimum competitive ratio is achieved for approximately $\gamma = 3$; in particular for the special case of the cow-path problem the value $\gamma = 3$ is optimal. We decided to state our result for general γ instead of $\gamma = 3$ because this slight generalization is sometimes helpful to the reader (by reducing the number of "magic constants").

Let r be a request and let s_1 be the rightmost unmatched server in the interval $(-\infty, r)$. Similarly let s_2 be the leftmost unmatched server in (r, ∞). We call the two servers s_1 and s_2 the *surrounding* servers of r. It is easy to see that any online algorithm can be converted into one that services each request with one of its two surrounding servers with the same (or better) competitive ratio. Our first aim is to show that the γWFA has this nice property:

Property 1 (Locality). The γWFA services each request with one of its two surrounding servers.

To show the Locality Property we need to take into account the history of the matching created by the γWFA. To see this consider the following example: There are three servers s_1, s_2, s_3 (Figure 1).

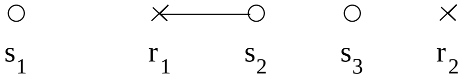

Fig. 1. Surrounding servers

The first request r_1 is in the interval (s_1, s_2) and is matched to s_2. The second request is in (s_3, ∞). If γWFA has Property 1, then it must match r_2 to s_3. This can be easily verified, but only if we take into account the history of γWFA and in particular the fact that $d(s_1, r_1) \geq d(r_1, s_2)$. This example shows that we need some kind of induction to show that γWFA has the Locality Property. It turns out that the Locality Property follows from the following invariant (in fact, it is equivalent to it).

Theorem 1. *Let s_1, s_2 be two servers that, before time t, have not been matched by the γWFA algorithm but every other server in (s_1, s_2) has been matched. Let A, R be the sets of servers and requests before time t in (s_1, s_2). Then the γWFA has matched the set of requests R to the same-cardinality set of servers A; furthermore*

$$\gamma M(A, R) \leq \gamma M(A + s_1, R + s_2) + d(s_1, s_2). \tag{1}$$

Before we proceed to prove the theorem, we point out that the Locality Property before time t follows immediately from it and in particular from the conclusion that A and R have the same cardinality and are matched by γWFA. Conversely, the theorem follows if we assume that γWFA has the Locality Property at time t: let the request r at time t be to the right of s_2 within very small distance; then the γWFA will prefer to match r to s_2 than to s_1 and this gives (1).

We will also need the following lemma about matchings:

Lemma 1. *Let s_1, s_2 be two servers and let A and R be two sets of points (servers and requests) in (s_1, s_2) of equal cardinality. Let also $r_1 \leq r_2$ be two points (requests) in $[s_1, s_2]$. Then*

$$M(A+s_1, R+r_1)+M(A+s_1, R+r_2) \leq M(A+s_1+s_1, R+r_1+r_2)+M(A, R). \tag{2}$$

Proof. Fix a point $x \in [s_1, r_1)$. The total number of lines that include x of the left-hand side is $|\beta_x(A+s_1, R+r_1)| + |\beta_x(A+s_1, R+r_2)| = 2|\beta_x(A, R)+1|$ and of the right-hand side is $|\beta_x(A+s_1+s_1, R+r_1+r_2)| + |\beta_x(A, R)| = |\beta_x(A, r)+2| + |\beta_x(A, r)|$. Since $2|a+1| \leq |a+2| + |a|$ (the absolute function is concave), it follows that the contribution of x to the left hand side of the inequality is no more than the right hand side. A similar situation holds when $x \in [r_1, r_2)$ and $x \in [r_2, s_2)$ and the lemma follows. □

Proof (of Theorem 1). To prove Theorem 1 we use induction on the number of requests in R and the properties of the γWFA. Let r be the most recent request in R and let s be the server matched to r by the γWFA. By induction, the Locality Property holds and therefore s is in the interval (s_1, s_2), that is $s \in A$. We consider two cases.

Case 1: $r < s$. Let A_1 and A_2 be the set of servers in (s_1, s) and (s, s_2), respectively. Let also R_1 and R_2 be the associated requests in these intervals (requests before r). By induction, A_1 (resp. A_2) has the same cardinality with R_1 (resp. R_2).

By the definition of γWFA we have $\gamma M(A_1+s, R_1+r)+d(s,r) \leq \gamma M(A_1+s_1, R_1+r)+d(s_1,r)$. By the induction hypothesis we also have $\gamma M(A_2, R_2) \leq \gamma M(A_2+s, R_2+s_2)+d(s,s_2)$. Therefore:

$$\begin{aligned}
\gamma M(A,R) &= \gamma M(A_1+s, R_1+r)+\gamma M(A_2,R_2) \\
&\leq [\gamma M(A_1+s_1, R_1+r)+d(s_1,r)-d(s,r)] \\
&\quad +[\gamma M(A_2+s, R_2+s_2)+d(s,s_2)] \\
&= \gamma M(A_1+A_2+s_1+s, R_1+R_2+r+s_2) \\
&\quad +[d(s_1,r)-d(s,r)+d(s,s_2)] \\
&\leq \gamma M(A_1+A_2+s_1+s, R_1+R_2+r+s_2)+d(s_1,s_2) \\
&= \gamma M(A+s_1, R+s_2)+d(s_1,s_2)
\end{aligned}$$

Case 2: $r \geq s$. Let again A_1, R_1 and A_2, R_2 be as in the first case. Then $M(A,R) = M(A_1+A_2+s, R_1+R_2+r) = M(A_1,R_1)+M(A_2+s, R_2+r)$.

By induction on the left interval, we have $\gamma M(A_1,R_1) \leq \gamma M(A_1+s_1, R_1+s)+d(s_1,s)$. Also by induction on the right interval, we have $\gamma M(A_2,R_2) \leq \gamma M(A_2+s, R_2+s_2)+d(s,s_2)$ or equivalently

$$\gamma M(A_2,R_2)-\gamma M(A_2+s, R_2+s_2) \leq d(s,s_2).$$

But by Lemma 1 we have

$$M(A_2+s, R_2+r)-M(A_2+s+s, R_2+r+s_2) \leq M(A_2,R_2)-M(A_2+s, R_2+s_2).$$

Combining the two inequalities we get $\gamma M(A_2+s, R_2+r)-\gamma M(A_2+s+s, R_2+r+s_2) \leq d(s,s_2)$. Therefore:

$$\begin{aligned}
\gamma M(A,R) &= \gamma M(A_1,R_1)+\gamma M(A_2+s, R_2+r) \\
&\leq [\gamma M(A_1+s_1, R_1+s)+d(s_1,s)] \\
&\quad +[\gamma M(A_2+s+s, R_2+r+s_2)+d(s,s_2)] \\
&= \gamma M(A_1+A_2+s+s_1, R_1+R_2+r+s_2)+d(s_1,s_2) \\
&= \gamma M(A+s_1, R+s_2)+d(s_1,s_2)
\end{aligned}$$

Notice the crucial use of Lemma 1. We remark that it we could get away without it when $A_1 \neq \emptyset$, but it is absolutely necessary otherwise. This is because in that case $|A_2+s| = |A|$ and so we cannot apply induction.

We also remark that in the second case we used only induction, no property of the γWFA at all. □

The following invariant is a generalization of Theorem 1 and we state it without proof:

Theorem 2. *Let s_1, s_2 be as in Theorem 1. Then for any points $v_1, \ldots, v_k$ with $s_1 \leq v_1 \leq v_2 \leq \cdots \leq v_k \leq s_2$ we have*

$$\gamma M(A+s_1^k, R+v_1+\cdots+v_k) \leq \gamma M(A+s_1^{k+1}, R+v_1+\cdots+v_k+s_2)+d(s_1,s_2), \quad (3)$$

where s_1^m denotes m copies of s_1.

Definition 2 (Balanced Interval). *Let s_1, s_2 be two unmatched servers such that inside (s_1, s_2) all servers are already matched. We will call such an interval* balanced.

Because of the Locality Property, we can focus our analysis to a balanced interval.

3 Lower bound

We remark that the cow-path problem is a special case of the matching problem and this implies a lower bound of 9 for any online algorithm. In the reduction to the cow-path problem, there is a server on each integral point of R except 0. The first request is at $r_1 = 0$, and request r_t, $t > 1$, is at $s_{r_{t-1}}$, the server used to service the previous request.

We will now show a lower bound of $\Omega(\log n)$ on the competitive ratio of γWFA. This disproves a conjecture of Kalyanasundaram and Pruhs. We take advantage of the Locality Property of γWFA to simplify the argument.

Theorem 3. *The competitive ratio of* γWFA *is at least* $2 + \lfloor \log n \rfloor$*, where* n *is the number of requests.*

Proof. Without loss of generality n is a power of 2. The set of servers is defined as follows: There is one server at 0, one server at n and 2 servers at each of the even integers in $(0, n)$. The requests come in stages. In the first stage all odd points in $(0, n)$ are requested. Consider the request at $2j + 1$. Using the Locality Property, we see that the γWFA can service it by either a server on the left, at $2j$, or a server at the right, at $2j + 2$, because they are at equal distance from the request. We want to allow the adversary to break the tie. This can be enforced by perturbing the request slightly. In particular, we service all requests with the servers at positions $2 + 4k$, as shown in Figure 2. Notice now that the remaining servers are in a configuration similar to the initial one, only now they

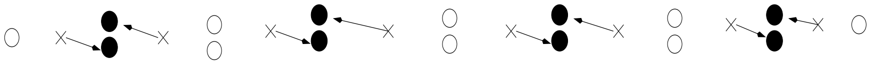

Fig. 2. Stage 1 (x denotes request and o denotes server)

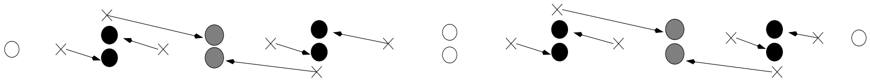

Fig. 3. Stage 2

are placed apart at distance 4 instead of 2. So we can repeat the construction. In particular, in the second stage all points $2+4k$ are requested and are serviced by the servers at points $4+8k$. We repeat it until only the server at 0 remains (in the last stage, a request is placed at $n/2$ which is serviced by the server at n). At the end, just to finish the construction we place a request at n and it is serviced by the only remaining server at 0.

The online cost at each stage is $n/2$ and the cost of the last request is n. There are $\log n$ stages, so the total online cost is $n + \frac{n}{2}\log n$. The total offline cost is easily seen to be $n/2$. The competitive ratio is $2 + \log n$. □

4 Upper bound

We define the set of points crossed exactly j times by the optimal matching $M(A,R)$ as

$$B^j(A,R) = \{x \,:\, \beta_x(A,R) = j\}.$$

We also use the notation $B^-(A,R) = \cup_{j<0}B^j(A,R)$ and $B^+(A,R) = \cup_{j>0}B^j(A,R)$. To keep the expressions simple, we use the same symbols for the measure of these sets. Given these definitions, observe that

$$M(A,R) = \sum_j |j| \cdot B^j(A,R). \tag{4}$$

Thus for any balanced interval (s_1,s_2) which includes the set of servers A and set of requests R, we have $M(A+s_1,R+s_2) = M(A,R)+B^+(A,R)+B^0(A,R)-B^-(A,R)$ and $d(s_1,s_2) = B^+(A,R) + B^0(A,R) + B^-(A,R)$. Substituting these values in Theorem 1, we have:

$$2\gamma\, B^-(A,R) \leq (\gamma+1)\, d(s1,s2), \text{ and}$$
$$2\gamma\, B^+(A,R) \leq (\gamma+1)\, d(s1,s2).$$

Now we are ready to prove a generalization that will be very useful later.

Lemma 2. *Let (s_1,s_2) be a γWFA balanced interval which includes the set of servers A and set of requests R. For any $x \in (s_1,s_2)$*

$$2\gamma\,\|B^-(A,R)\cap[s_1,x]\| \leq (\gamma+1)\, d(s_1,x), \text{ and}$$
$$2\gamma\,\|B^+(A,R)\cap[x,s_2]\| \leq (\gamma+1)\, d(x,s_2).$$

Proof. The intuition behind the lemma is that since s_1 is an unmatched server, it was rejected in favor of other servers in the interval (s_1,s_2). This in turn indicates that $\|B^-(A,R)\cap[s_1,x]\|$ cannot be very large.

We prove only the first part, the second is similar. Suppose that the statement was true before the last request, $r_t \in (s_1,s_2)$, was matched to a server $s_{r_t} \in (s_1,s_2)$. Let $A_{t-1} = A - s_{r_t}$ and $R_{t-1} = R - r_t$. We consider two cases depending on whether s_{r_t} is to the left of r_t or not. The first case when $s_{r_t} < r_t$ is trivial because such a request increases the values $\beta_x(A_{t-1},R_{t-1})$.

The other case, $r_t < s_{r_t}$, is more involved. We first notice that when x is in (s_1, r_t) then the lemma holds directly by induction. Also if the lemma holds for $x = s_{r_t}$ then it immediately holds for any $x \in [s_{r_t}, s_2)$ from the induction hypothesis for the balanced interval (s_{r_t}, s_2).

Therefore we concentrate on the case $x \in [r_t, s_{r_t}]$. To show that the lemma holds for $x = s_{r_t}$ we don't need the induction hypothesis, only the fact that γWFA services r_t with s_{r_t} instead of s_1:

$$\gamma M(A_{t-1} + s_1, R_{t-1} + r_t) + d(s_1, r_t) \geq \gamma M(A_{t-1} + s_{r_t}, R_{t-1} + r_t) + d(r_t, s_{r_t}).$$

With some work we can rewrite it as

$$2\gamma \left\| B^-(A, R) \cap [s_1, s_{r_t}] \right\| \leq (\gamma + 1)\, d(s_1, s_{r_t}) - 2\, d(r_t, s_{r_t}), \tag{5}$$

which is slightly stronger than the lemma when $x = s_{r_t}$.

Finally, for the case $x \in [r_t, s_{r_t})$, we use (5) and the second inequality of the inductive hypothesis for the interval (x, s_{r_t}). Specifically, by induction on this interval, we get $2\gamma \left\| B^+(A_{t-1}, R_{t-1}) \cap [x, s_{r_t}] \right\| \leq (\gamma + 1)\, d(x, s_{r_t})$, which can be rewritten as

$$-2\gamma \left\| B^-(A, R) \cap [x, s_{r_t}] \right\| \leq -(\gamma - 1) d(x, s_{r_t}) \tag{6}$$

Summing (5) and (6) we get

$$2\gamma \left\| B^-(A, R) \cap [s_1, x] \right\| \leq (\gamma + 1) d(s_1, x) - 2 d(r_t, x)$$

and the lemma follows. □

4.1 Optimal vs Pseudo-Optimal matching

Suppose that after n requests, γWFA has matched R_n to servers S_n. The (offline) optimal matching though may match R_n to some other set S'_n. We call the optimal matching $M(S_n, R_n)$ the pseudo-optimal matching and denote its weight by $popt_n$ to distinguish it from the optimal matching $M(S'_n, R_n)$ whose weight is opt_n. Although γWFA cannot use the optimal set of servers (in general $S_n \neq S'_n$), it has the nice property that it uses an almost optimal set of servers.

Theorem 4. *For any set of servers and requests,*

$$opt_n \leq popt_n \leq \frac{\gamma + 1}{\gamma - 1} opt_n.$$

We now proceed to prove this theorem with the help of Lemma 2. To prove Theorem 4, we assume without loss of generality that all requests are in a balanced interval (s_1, s_2), otherwise we sum up the parts for each balanced interval. We need to compare $popt_n = M(S_n, R_n)$ and $opt = M(S'_n, R_n)$, for any S'_n such that $|S'_n| = |S_n| = |R|$. We can obtain the matching $M(S'_n, R_n)$ by putting requests at server points in $S_n - S'_n$ and matching them to servers in $S'_n \setminus S_n$: $M(S'_n, R_n) = M(S'_n \cup S_n, R_n + S_n - S'_n)$. Observe now that in the worst case the servers in S'_n that are outside the interval (s_1, s_2) are at s_1 or s_2. Thus, instead of proving Theorem 4, we shall prove the following stronger lemma.

Lemma 3. *Let* (s_1, s_2) *be a* γWFA *balanced interval that contains the set of servers* A *which is matched to requests* R*. For any points* $v_1, \ldots, v_{k+m}$ *in* (s_1, s_2)

$$M(A + s_1^k + s_2^m, R + v_1 + v_2 + \cdots + v_{k+m}) \geq \frac{\gamma - 1}{\gamma + 1} M(A, R)$$

Proof. Assume $v_1 \leq \cdots \leq v_{k+m}$ and denote $v_0 = s_1$. We concentrate in showing the special case with $m = 0$ and then observe that the same proof applies to the general case. Define $\zeta_i^j = \|\{x : \beta_x(A, R) = j\} \cap [v_{i-1}, v_i]\|$ to be the measure of points between v_{i-1} and v_i which the optimal matching crosses j times (with appropriate sign). Applying Lemma 2 to interval (s_1, v_t) we obtain $\frac{\gamma-1}{\gamma+1} \sum_{i=1}^t \sum_{j<0} \zeta_i^j \leq \sum_{i=1}^t \sum_{j \geq 0} \zeta_i^j$ and summing for $t = 1, \ldots, k$, we get:

$$\frac{\gamma - 1}{\gamma + 1} \sum_{t=1}^k \sum_{i=1}^t \sum_{j<0} \zeta_i^j \leq \sum_{t=1}^k \sum_{i=1}^t \sum_{j \geq 0} \zeta_i^j,$$

which can be rewritten as

$$\frac{\gamma - 1}{\gamma + 1} \sum_{i=1}^k \sum_{j<0} (k + 1 - i) \zeta_i^j \leq \sum_{i=1}^k \sum_{j \geq 0} (k + 1 - i) \zeta_i^j.$$

Using this and the assumption $\gamma > 1$, we can bound

$$\begin{aligned} M(A, R) - M(A + s_1^k, R + v_1 + \cdots, v_k) &= \sum_{i=1}^k \sum_j |j| \zeta_i^j - \sum_{i=1}^k \sum_j |j + k + 1 - i| \zeta_i^j \\ &\leq \frac{2}{\gamma + 1} \sum_{i=1}^k \sum_{j<0} |j| \zeta_i^j \end{aligned}$$

The last quantity is at most equal to $\frac{2}{\gamma+1} M(A, R)$ and the lemma follows (for $m = 0$). When $m > 0$ the proof is very similar but now the right-hand side includes also the terms with $j > 0$; this is still bounded by $\frac{2}{\gamma+1} M(A, R)$ and the lemma holds. □

This theorem implies almost immediately that γWFA is $O(n)$-competitive as follows: It is not hard using the β_x values to verify that $d(r_n, s_{r_n}) \leq M(S_{n-1}, R_{n-1}) + M(S_n, R_n)$ which in turn implies that the cost to service request r_n is bounded above by $popt_{n-1} + popt_n \leq \frac{\gamma+1}{\gamma-1}(opt_{n-1} + opt_n)$. Since opt_n is non-decreasing function (which by the way is not true for $popt_n$) the claim follows. We however show a stronger result and in the process we will discover some important properties that may be useful to improve the competitive ratio. We will show that the competitive ratio is bounded (up to a constant factor) by the maximum number of crossing lines in the γWFA matching.

4.2 Alive vs Pseudo-Optimal

In this section, we take a closer look at the structure of the pseudo-optimal matching, and use this analysis to derive an $O(n)$ bound on the competitive ratio of γWFA.

We say that a point x contributes to $popt_n$ at time t if

$$0 \leq \beta_x(S_{t-1}, R_{t-1}) < \beta_x(S_t, R_t) \leq \beta_x(S_{t+1}, R_{t+1}), \ldots, \beta_x(S_n, R_n)$$

or

$$0 \geq \beta_x(S_{t-1}, R_{t-1}) > \beta_x(S_t, R_t) \geq \beta_x(S_{t+1}, R_{t+1}), \ldots, \beta_x(S_n, R_n)$$

We say that a point x contributes to $alive_n$ at time t if the same inequalities hold but not necessarily the leftmost ones (involving 0). For example, in Figure 4, x contributes to $alive_8$ at times 3,6,7, and 8 (which correspond to the bold lines; intuitively these are the lines that are visible from the point $(\infty, 0)$. In the same figure, x contributes to $popt_8$ at times 7 and 8 (which correspond to the bold lines after the last visit to the horizontal axis).

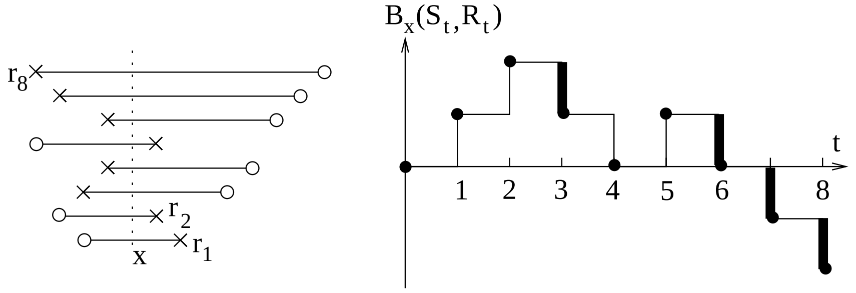

Fig. 4. $alive_n$ and $popt_n$

We say that a point contributes to $alive_n$ (or opt_n) k times when it contributes to $alive_n$ (or opt_n) at exactly k distinct times. For example, the point x above contributes 4 times to $alive_8$ and 2 times to $popt_8$. Clearly $popt_n$ is the measure of all points, each taken with multiplicity equal to the number of times that it contributes to $popt_n$. Similarly, $alive_n$ is the measure of all points, each taken with multiplicity equal to the number of times that it contributes to $alive_n$.

Theorem 5.
$$\frac{\gamma - 1}{2\gamma} alive_n \leq popt_n \leq alive_n.$$

Proof. The second inequality is obvious from the definitions. The first inequality is based on Lemma 2. Consider a request r_t which is serviced by request s_{r_t} and assume without loss of generality that $s_{r_t} < r_t$. The points that contribute to

$alive_n$ at time t form an interval (s_{r_t}, q) for some point q. By Lemma 2 at least a fraction $\frac{\gamma-1}{2\gamma}$ of this interval has $\beta_x(A_{t-1}, R_{t-1}) \geq 0$. This fraction contributes also to $popt_n$ at time t. □

4.3 Alive vs Cost

We can now put everything together to obtain the desired upper bound. We have considered the four quantities: opt_n, $popt_n$, $alive_n$ and $cost_n$ (the last one is the weight of the online matching). Theorems 4 and 5 establish that the first three quantities are almost equal within some constant factors:

$$opt_n \leq popt_n \leq alive_n \leq \frac{2\gamma}{\gamma-1}\frac{\gamma+1}{\gamma-1} opt_n$$

It is remarkable though that opt_n and $alive_n$ are non-decreasing while $popt_n$ is not. Furthermore, the cost to service request r_t is at most equal to $alive_t$ (each point in the interval (r_t, s_{r_t}) contributes to $alive_t$). Therefore we can bound the total $cost_n$ by

$$cost_n \leq \sum_{t=1}^{n} alive_t \leq n \cdot alive_n \leq \frac{2\gamma}{\gamma-1}\frac{\gamma+1}{\gamma-1} \cdot n \cdot opt_n.$$

We showed

Theorem 6. *The* γWFA *has competitive ratio at most* $\frac{2\gamma}{\gamma-1}\frac{\gamma+1}{\gamma-1} \cdot n = O(n)$, *where* n *is the number of requests. In particular, when each point* x *is crossed by at most* k *lines of the matching of* γWFA, *the competitive ratio is* $O(k)$.

5 Conclusions and Open Problems

For the online matching problem on a line, we showed that the WFA has competitive ratio between $\Omega(\log n)$ and $O(n)$. We believe that the lower bound is tight and that the tools we developed here can be very useful. We don't know whether there exists some other algorithm which has constant competitive ratio.

One important property is the relation between *popt* and *opt* which states that although WFA may not use the optimal subset of servers —the one that minimizes the matching with the requests— it uses an almost optimal one (up to a constant factor). This may be true for arbitrary metric spaces.

For the Euclidean space R^d, the lower bound for the cow-path problem, $\Omega(n^{1-1/d})$, can be extended to the matching problem. Is this tight and what is the competitive ratio of WFA? Finally, randomized algorithms may have much better competitive ratio; the best randomized lower bound for arbitrary metric spaces is only $\Omega(\log n)$ [6].

References

1. Ricardo A. Baeza-Yates, Joseph C. Culberson, and Gregory J. E. Raw lins. Searching in the plane. *Information and Computation*, 106(2):234–252, October 1993.
2. J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards B*, 69B:125–130, 1965.
3. J. Edmonds. Paths, trees and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.
4. Bernhard Fuchs, Winfried Hochstättler, and Walter Kern. Online matching on a line. In Hajo Broersma, Ulrich Faigle, Johann Hurink, Stefan Pickl, and Gerhard Woeginger, editors, *Electronic Notes in Discrete Mathematics*, volume 13. Elsevier, 2003.
5. Bala Kalyanasundaram and Kirk Pruhs. Online weighted matching. *Journal of Algorithms*, 14(3):478–488, May 1993. A Preliminary version appeared in the proceedings of the 2nd ACM-SIAM Symposium on Discrete Algorithms, 1991, pages 234–240.
6. Bala Kalyanasundaram and Kirk Pruhs. Online network optimization problems. In A. Fiat and G. Woeginger, editors, *Online Algorithms: The State of the Art*, volume 1442 of *Lecture Notes in Computer Science*, pages 268–280. Springer-Verlag, 1998.
7. Bala Kalyanasundaram and Kirk Pruhs. Online transportation problem. *SIAM Journal of Discrete Mathematics*, 13(3):370–383, 2000. A Preliminary version appeared in Proceedings of the European Symposium on Algorithms, pages 484–493, 1995.
8. Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd ACM Annual Symposium on Theory of computing*, pages 352–358, 1990.
9. Samir Khuller, Stephen G. Mitchell, and Vijay V. Vazirani. On-line algorithms for weighted bipartite matching and stable marriages. *Theoretical Computer Science*, 127:255–267, 1994. A preliminary version appeared as Technical Report 90–1143, Department of Computer Science, Cornell University, 1990.
10. L. Lovász and M. D. Plummer. *Matching Theory.* Elsevier Science Pub. Co., 1986.