International Journal of Foundations of Computer Science

THE FREQUENT ITEMS PROBLEM IN ONLINE STREAMING UNDER VARIOUS PERFORMANCE MEASURES*

JOAN BOYAR† and KIM S. LARSEN‡ and ABYAYANANDA MAITI§

Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark
†*joan@imada.sdu.dk*
‡*kslarsen@imada.sdu.dk*
§*abyaym@gmail.com*

This is a contribution to the ongoing study of properties of performance measures for online algorithms. It has long been known that competitive analysis suffers from drawbacks in certain situations, and many alternative measures have been proposed. More systematic comparative studies of performance measures have been initiated recently, and we continue this work, considering competitive analysis, relative interval analysis, and relative worst order analysis on the frequent items problem, a fundamental online streaming problem.

Keywords: Online algorithms; streaming algorithms; performance measures.

1. Introduction

It has been known since its introduction that competitive analysis does not always give good results [13] and many alternative performance measures have been proposed for analyzing online algorithms. However, as a general rule, these alternatives have been fairly problem specific and most have only been compared to competitive analysis. A more comprehensive study of a larger number of performance measures applied to the same online problem was initiated in [6], where a simple k-server problem was investigated, and this line of work has been continued in [7], considering the problem of online searching. Continuing this line of work, we would like

*Supported in part by the Danish Council for Independent Research and the Villum Foundation. Part of this work was done while the authors were visiting the University of Waterloo, Ontario, Canada. A preliminary version of some of these results appeared in the proceedings of the 19th International Symposium on Fundamentals of Computation Theory, Liverpool, United Kingdom, 2013.
‡Corresponding author

to produce complete and tight results for the algorithms studied. For that reason, we focus on a fairly simple combinatorial problem and on simple algorithms for its solution.

The analysis of problems and algorithms for streaming applications, treating them as online problems, was started by Becchetti and Koutsoupias [2]. In online streaming, the items must be processed one at a time by the algorithm, making some irrevocable decision in the processing of each item, using a small fixed amount of memory. In the frequent items problem [9], an algorithm must store an item, or more generally a number of items, in a buffer, and the objective is to maintain the items appearing most frequently in the entire stream. This problem has been studied by Giannakopoulos and Koutsoupias [11]. In addition to analyzing deterministic algorithms using competitive analysis, they proved lower bounds which also hold for randomized algorithms against oblivious adversaries. They also analyzed algorithms using the distributional adversarial model, where the adversary, instead of producing an input stream, selects a probability distribution on the universe of elements. Here, since we are aiming for tight results comparing different performance measures, for simplicity, we consider only simple deterministic algorithms and a buffer of size one. Giannakopoulos and Koutsoupias [11] also concentrated on a buffer of size one, but have some lower bound results for general larger buffer sizes. We analyze the frequent items problem using relative interval analysis [10] and relative worst order analysis [4]. In addition, for the specific models considered here, we tighten the competitive analysis results from [11].

We develop results for a finite universe of items, but we also consider the unbounded case, where we could keep seeing new items no matter how long the stream is. For this case, we obtain results which are functions of the input length. In order to compare algorithms, we consider the constants in front of the leading term. In this paper, we name this competitive function and give a formal definition in order to be able to state our results clearly, without ambiguity. These ideas are also used to generalize relative worst order analysis.

Note that our contribution is towards understanding performance measures, rather than algorithmic improvements for the online streaming problem. A dedicated study of the online streaming problem for the sake of this problem alone would likely be directed towards more evolved algorithms than considered here and buffer sizes larger than one.

2. Preliminaries

The frequent items problem is a streaming problem, but as usual in online algorithms, we use the term sequence or input sequence to refer to a stream. We denote an *input sequence* by $I = a_1, a_2, \ldots, a_n$, where the items a_i are from some universe $\mathcal{U}$. We may refer to the index also as the *time step*. In this context, the size of the

universe $\mathcal{U}$ is very crucial. The universe can be finite or it can be unbounded as in [11]. In the bounded case, we denote the size of $\mathcal{U}$ by N. We study both cases and refer to them as the *frequent items problem with finite universe* (denoted FIF) and the *frequent items problem with unbounded universe* (denoted FIU).

We consider the simplest possible set up for the FIF and FIU problems: An algorithm has a *buffer* with space for one item. When processing an item, the algorithm can either discard the item or replace the item in the buffer by the item being processed. The objective is to keep the most frequently occurring items in the buffer, where frequency is measured over the entire input, i.e., when an algorithm must make a decision, the quality of the decision also depends on items not yet revealed to the algorithm. We define this objective function formally:

Given an online algorithm $\mathcal{A}$ for this problem, we let $s_t^{\mathcal{A}}$ denote *the item in the buffer at time step* t. We may omit the superscript when it is clear from the context which algorithm we discuss.

Given an input sequence I and an item $a \in \mathcal{U}$, the *frequency* of the item is defined as $f_I(a) = \frac{n_I(a)}{n}$, where $n_I(a) = |\{i \mid a_i = a\}|$ is the number of occurrences of a in I. The objective is to maximize the *aggregate frequency* [11], defined by $F_{\mathcal{A}}(I) = \sum_{t=1}^{n} f_I(s_t^{\mathcal{A}})$, i.e., the sum of the frequencies of the items stored in the buffer over the time.

When the same item occurs in two consecutive time steps, we refer to that as a *repeated item*. In the case of the FIF problem, since $N = 1$ is a trivial case, we always assume $N \geq 2$.

We compare the quality of the achieved aggregate frequencies of three different deterministic online algorithms from [11]: the naive algorithm (NAI), the eager algorithm (EAG), and the majority algorithm (MAJ). All three are practical streaming algorithms, being simple and using very little extra space. We study them for both the FIF and FIU problems. Recall that we consider a scenario where the buffer has room for only one item.

Definition 1. [NAI] NAI *buffers every item as it arrives, i.e.,* $s_t^{\text{NAI}} = a_t$ *for all* $t = 1, 2, \ldots, n$.

EAG switches mode upon the detection of the first repeated item in the sequence.

Definition 2. [EAG] *Initially,* EAG *buffers every item as it arrives. If it finds a repeated item, then it keeps that item until the end, i.e., let*

$$t^* = \min_{1 \leq t \leq n-1} \{t \mid a_t = a_{t+1}\},$$

if such a t *exists, and otherwise* $t^* = n$. *Then* EAG *is the algorithm with* $s_t^{\text{EAG}} = a_t$ *for all* $t \leq t^*$ *and* $s_t^{\text{EAG}} = a_{t^*}$ *for all* $t > t^*$.

Definition 3. [MAJ] MAJ *keeps a counter along with the buffer. Initially, the counter is set to zero. If the counter is zero, then* MAJ *evicts the item currently in the buffer and instead buffers the arriving item and sets the counter to one. Otherwise, the content of the buffer does not change, but if the arriving item is the same as the one currently buffered,* MAJ *increments the counter by one, and otherwise decrements it by one.*

As an example, consider the sequence in Fig. 1 and the item in the buffer after each item is processed for the three different algorithms:

Sequence	1	2	3	2	2	1	4	4	1	4	2	3	5
NAI	1	2	3	2	2	1	4	4	1	4	2	3	5
EAG	1	2	3	2	2	2	2	2	2	2	2	2	2
MAJ	1	1	3	3	2	2	4	4	4	4	4	4	5
MAJ counter	1	0	1	0	1	0	1	2	1	2	1	0	1

Fig. 1. Example buffer behavior for the three algorithms. The last line shows the value of MAJ's counter immediately after the item has been processed.

Finally, as usual in online algorithms, we let OPT denote an optimal offline algorithm. OPT is used in competitive analysis as a benchmark. If $\mathcal{A}$ is an algorithm, we let $\mathcal{A}(I)$ denote the result, the aggregate frequency, of the algorithm on the sequence I, i.e., $\mathcal{A}(I) = F_{\mathcal{A}}(I)$.

In comparing algorithms using a particular performance measure, it is useful to find families of sequences where one algorithm does well and the other does poorly. In comparing these three online algorithms, we repeatedly use the same three families of sequences; E_n, on which EAG performs particularly poorly for both the FIF and FIU problems, and W_n and $W_{n,r}$, on which MAJ performs particularly poorly. W_n is intended for FIU problems and $W_{n,r}$ for FIF problems, where r depends on the universe size N. Before considering the different performance measures, one at a time, we first define these families of sequences and consider the performance of each algorithm on each family.

Definition 4. *For an arbitrary universe $\mathcal{U}$, we define the sequences of length n:*

$$E_n = a, a, b, b, \ldots, b,$$

where $a, b \in \mathcal{U}$ and there are $n-2$ copies of b, and

$$W_n = \begin{cases} b_1, b_0, b_2, b_0, \ldots, b_{\frac{n}{2}}, b_0 & \text{for even } n \\ b_1, b_0, b_2, b_0, \ldots, b_{\lfloor \frac{n}{2} \rfloor}, b_0, b_{\lceil \frac{n}{2} \rceil} & \text{for odd } n, \end{cases}$$

where all $b_i \in \mathcal{U}$. For a finite universe $\mathcal{U}$ of size N, we define

$$W_{n,r} = (b_1, b_0)^r, (b_2, b_0)^r, \ldots, (b_{N-1}, b_0)^r,$$

where all $b_i \in \mathcal{U}$, r *is any positive integer indicating* r *repetitions, and* $n = 2r(N-1)$.

The four algorithms, including OPT, obtain the aggregate frequencies below on these three families of sequences. The arguments are simple, but fundamental, and also serve as an introduction to the heuristic behavior of these algorithms.

Proposition 5. *The algorithms' results on* E_n, W_n, *and* $W_{n,r}$ *are as in Fig. 2.*

	E_n	W_n	$W_{n,r}$
NAI	$n-4+\frac{8}{n}$	$\begin{cases} \frac{n}{4}+\frac{1}{2} & \text{for even } n \\ \frac{n}{4}+\frac{3}{4n} & \text{for odd } n \end{cases}$	$\frac{n}{4}+\frac{r}{2}$
EAG	2	as NAI	as NAI
MAJ	$n-6+\frac{16}{n}$	1	r
OPT	as NAI	$\begin{cases} \frac{n}{2}-\frac{1}{2}+\frac{1}{n} & \text{for even } n \\ \frac{n}{2}-1+\frac{3}{2n} & \text{for odd } n \end{cases}$	$\frac{n-1}{2}+\frac{r}{n}$

Fig. 2. The algorithms' aggregate frequencies on E_n, W_n, and $W_{n,r}$.

Proof. In E_n, the frequency of a is $\frac{2}{n}$ and the frequency of b is $\frac{n-2}{n}$. Thus, $\text{NAI}(E_n) = 2\frac{2}{n}+(n-2)\frac{n-2}{n} = n-4+\frac{8}{n}$. In W_n, the frequency of b_0 is $\lfloor\frac{n}{2}\rfloor/n$, and the frequencies of all the other b_i, $1 \le i \le \lceil\frac{n}{2}\rceil$, are $\frac{1}{n}$. Thus, $\text{NAI}(W_n) = \lceil\frac{n}{2}\rceil\frac{1}{n}+\lfloor\frac{n}{2}\rfloor\frac{\lfloor\frac{n}{2}\rfloor}{n}$. Considering both even and odd n gives the stated result. In $W_{n,r}$, the frequency of b_0 is $\frac{1}{2}$, and the frequencies of all the other b_i, $1 \le i \le N-1$, are $\frac{r}{n}$. Thus, $\text{NAI}(W_{n,r}) = \frac{n}{2}\frac{1}{2}+\frac{n}{2}\frac{r}{n} = \frac{n}{4}+\frac{r}{2}$.

When processing E_n, EAG keeps a in its buffer. Hence, $\text{EAG}(E_n) = n\frac{2}{n} = 2$. Since W_n and $W_{n,r}$ have no repeated items, $\text{EAG}(W_n) = \text{NAI}(W_n)$ and $\text{EAG}(W_{n,r}) = \text{NAI}(W_{n,r})$.

For E_n, MAJ will have a in its buffer for the first four time steps, so $\text{MAJ}(E_n)$ is $4\frac{2}{n}+(n-4)\frac{n-2}{n} = n-6+\frac{16}{n}$. For W_n, MAJ brings each b_i, $1 \le i \le n$, into its buffer and never brings b_0 into its buffer. Thus, $\text{MAJ}(W_n) = n\frac{1}{n} = 1$. For $W_{n,r}$, MAJ brings each b_i, $1 \le i \le N-1$, into its buffer and never brings b_0 into its buffer. Thus, $\text{MAJ}(W_{n,r}) = n\frac{r}{n} = r$.

For E_n, OPT is forced to perform the same as NAI. For W_n, OPT must buffer b_1 in the first time step, but it buffers b_0 for the remainder of the sequence. Thus, $\text{OPT}(W_n) = \frac{1}{n}+(n-1)\frac{\lfloor\frac{n}{2}\rfloor}{n}$. Considering both even and odd n gives the stated result. For $W_{n,r}$, OPT must buffer b_1 in the first time step, but it buffers b_0 for the

remainder of the sequence. Thus,

$$\textsc{Opt}(W_{n,r}) = \frac{r}{n} + (n-1)\frac{r(N-1)}{n} = \frac{n-1}{2} + \frac{r}{n}. \qquad \square$$

Definition 6. *For any online algorithm $\mathcal{A}$, we denote the worst aggregate frequency of $\mathcal{A}$ over all the permutations σ of I by $\mathcal{A}_W(I) = \min_\sigma \mathcal{A}(\sigma(I))$.*

It is convenient to be able to consider items in order of their frequencies. Let $D(I) = b'_1, b'_2, \ldots, b'_n$ be a sorted list of the items in I in nondecreasing order of frequency. For example, if $I = a, b, c, a, b, a$, then $D(I) = c, b, b, a, a, a$. We will use the notation $D(I)$ throughout the paper.

Lemma 7. *For odd n, $\textsc{Maj}_W(I) = 2\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_I(b'_i) + f_I(b'_{\lceil \frac{n}{2} \rceil})$, and for even n, $\textsc{Maj}_W(I) = 2\sum_{i=1}^{\frac{n}{2}} f_I(b'_i)$, where the b'_i are the items of $D(I)$.*

Proof. The counter in Majority is always nonnegative and always changing, so at least half the items lead to increases. We order the items, so that exactly the $\lceil \frac{n}{2} \rceil$ requests to the least frequent items are buffered, as follows: Assuming n is even, then the worst permutation is $b'_1, b'_n, b'_2, b'_{n-1}, \ldots, b'_{\frac{n}{2}}, b'_{\frac{n}{2}+1}$. All (but the last request when n is odd) of the requests which lead to an item entering the buffer contribute twice, since they are also in the buffer for the next step. $\square$

The following observation, which we use repeatedly, is easy to verify and it also follows from the Cauchy-Schwarz inequality.

Proposition 8. *Let k non-negative numbers $x_1, x_2, \ldots, x_k$ be such that*

$$x_1 + x_2 + \ldots + x_k = n.$$

Then the sum of the squares of all x_i, $1 \leq i \leq k$, has a lower bound of $\frac{n^2}{k}$ and it achieves this bound when all x_i are equal to $\frac{n}{k}$.

3. Competitive Analysis

A streaming problem was first studied from an online algorithms perspective using competitive analysis by Becchetti and Koutsoupias [2]. Competitive analysis [13, 12] evaluates an online algorithm in comparison with an optimal offline algorithm. For maximization problems, an algorithm $\mathcal{A}$ is c-competitive for some constant c if there is a constant α such that for all finite input sequences I,

$$\textsc{Opt}(I) \leq c \cdot \mathcal{A}(I) + \alpha.$$

The infimum over the set of all values of c such that $\mathcal{A}$ is c-competitive is called the *competitive ratio* of $\mathcal{A}$. If there is no such constant c such that this inequality

holds for all I, then $\mathcal{A}$ is referred to as *not* being competitive. In particular, $\mathcal{A}$ is not competitive when c must be a function of the length of I.

For the FIU problem, it turns out that the relative performance of algorithms will depend on the length of I. This is different from the usual situation in competitive analysis, where we most often have results stating that an algorithm is c-competitive for some constant c. When using a function instead, we would prefer not to loose any further precision in our statements, so instead of talking about an algorithm being $O(C(n))$-competitive, we would like to be able to talk about the constant in front of the fastest growing term. To be precise in our statements, we define a modified and more general version of competitive analysis using ratios which are functions. Our definition can be adapted easily to minimization problems in the same way that such adaptations are handled for standard competitive analysis. In all these definitions, when n is not otherwise defined, we use it to denote $|I|$, the length of the sequence I. As usual, when using asymptotic notation in inequalities, notation such as $f(n) \leq g(n) + o(g(n))$ means that there exists a function $h(n) \in o(g(n))$ such that $f(n) \leq g(n) + h(n)$. Thus, we focus on the high order term of the ratio of the online algorithm to the optimal algorithm, including the multiplicative constant in front of it.

The following definition is for maximization problems.

Definition 9. *An algorithm $\mathcal{A}$ is* $f(n)$-competitive *if*

$$\forall I\colon\ \textsc{Opt}(I) \leq (f(n) + o(f(n))) \cdot \mathcal{A}(I).$$

$\mathcal{A}$ has competitive function $f(n)$ *if $\mathcal{A}$ is* $f(n)$-competitive *and for any $g(n)$ such that $\mathcal{A}$ is* $g(n)$-competitive, $\lim_{n\to\infty} \frac{f(n)}{g(n)} \leq 1$.

If algorithm $\mathcal{A}$ has competitive function $f_{\mathcal{A}}(n)$ *and algorithm $\mathcal{B}$ has* competitive function $f_{\mathcal{B}}(n)$*, then $\mathcal{A}$ is better than $\mathcal{B}$ according to competitive analysis if* $\lim_{n\to\infty} \frac{f_{\mathcal{A}}(n)}{f_{\mathcal{B}}(n)} < 1$.

Thus, the concept of a competitive function is an exact characterization up to the level of detail we focus on. It can be viewed as an equivalence relation, and if $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$ for two functions $f(n)$ and $g(n)$, then they belong to (and are representatives of) the same equivalence class. For example, $\frac{\sqrt{n}}{2}$ and $\frac{\sqrt{n}}{2-\frac{1}{\sqrt{n}}}$ are considered equivalent, whereas $\frac{\sqrt{n}}{2}$ and $\frac{\sqrt{n}}{4}$ are not.

For the FIF problem, we follow the original definition of competitive analysis. For the FIU problem, none of the three algorithms discussed here is competitive according to the original definition. However, information regarding the relative quality of these algorithms can be obtained from their respective competitive functions.

Giannakopoulos and Koutsoupias proved that no randomized algorithm for the FIU problem, where the buffer has room for one item, can have a competitive function

better than $\frac{1}{3}\sqrt{n}$ [11]. Using their technique, the result can be strengthened for the deterministic case.

Theorem 10. *No deterministic algorithm for the FIU problem can have a competitive function better than* $\frac{\sqrt{n}}{2}$.

Proof. Consider any deterministic algorithm $\mathcal{A}$, and input of the form

$$I_n = b_1, b_2, \ldots b_{n-\sqrt{n}}, x, x, \ldots, x$$

where the first $n-\sqrt{n}$ items are distinct and the last $\sqrt{n}$ items are identical. Since $\mathcal{A}$ is deterministic, an adversary will know whether b_1 or b_2 is in the buffer upon completion of time step 2. The value of x is based on this. If it is b_2, then the adversary sets $x = b_1$, and if it is b_1, then it sets $x = b_2$. As x does not occur among the next $n-\sqrt{n}-2$ items, $\mathcal{A}$ has no chance of bringing x into its buffer until the last $\sqrt{n}$ items arrive, so it stores x in its buffer at most $\sqrt{n}+1$ times. OPT stores x at least $n-1$ times. That gives the ratio of

$$\begin{aligned}\frac{\text{OPT}(I_n)}{\mathcal{A}(I_n)} &\geq \frac{\frac{1}{n} + (n-1)\frac{\sqrt{n}+1}{n}}{(n-\sqrt{n}-1)\frac{1}{n} + (\sqrt{n}+1)\frac{\sqrt{n}+1}{n}} = \frac{1+(n-1)(\sqrt{n}+1)}{n-\sqrt{n}-1+(\sqrt{n}+1)^2} \\ &= \frac{n+\sqrt{n}-1}{2\sqrt{n}+1} \geq \frac{\sqrt{n}}{2}, \text{ for } n \geq 4\end{aligned}$$

□

A similar technique gives a lower bound on the competitive ratio for the FIF problem as a function of the size of the universe.

Theorem 11. *No deterministic algorithm for the FIF problem can have a competitive ratio better than* $\frac{\sqrt{N}+1}{2}$.

Proof. For large enough n, let q be the smallest non-negative number (not necessarily an integer) such that $N-1$ divides $n-(\frac{n}{\sqrt{N}}+q)$. Then $r(N-1) = n-(\frac{n}{\sqrt{N}}+q)$ for some positive integer r. Let $s = \frac{n}{\sqrt{N}}+q$. Note that $q < N-1$, $s > q$, and both r and s are linear functions of n. Consider any deterministic algorithm, $\mathcal{A}$, and input of the form

$$I_n = b_1, b_2, y^{r-1}, b_3^r, b_4^r, \ldots, b_N^r, x^{(\frac{n}{\sqrt{N}}+q-1)}$$

where the b_i are N distinct items and x and y are determined based on the behavior of $\mathcal{A}$ in the first two time steps. Since $\mathcal{A}$ is deterministic, an adversary will know whether b_1 or b_2 is in the buffer upon completion of time step 2. If it is b_2, then the adversary sets $x = b_1$ and $y = b_2$, and if it is b_1, then it sets $x = b_2$ and $y = b_1$. Hence x is the most frequent item in I_n with frequency $\frac{s}{n}$ and all the other items have frequency $\frac{r}{n}$. As x does not occur among the next $n-s-1$ items, $\mathcal{A}$ has no

chance of bringing x into its buffer until the last $s-1$ items arrive, so it stores x in its buffer at most s times. OPT stores x at least $n-1$ times. That gives the ratio of

$$\begin{aligned}
\frac{\text{OPT}(I_n)}{\mathcal{A}(I_n)} &\geq \frac{\frac{r}{n}+(n-1)\frac{s}{n}}{(n-s)\frac{r}{n}+\frac{s^2}{n}} = \frac{ns-(s-r)}{(n-s)r+s^2} \\
&= \frac{(N-1)ns-Ns+n}{Ns^2-2ns+n^2}, \text{ since } r=\tfrac{n-s}{N-1} \\
&= \frac{\left(N-1-\frac{N}{n}\right)\left(\frac{1}{\sqrt{N}}+\frac{q}{n}\right)+\frac{1}{n}}{N\left(\frac{1}{\sqrt{N}}+\frac{q}{n}\right)^2-2\left(\frac{1}{\sqrt{N}}+\frac{q}{n}\right)+1}, \text{ since } s=\tfrac{n}{\sqrt{N}}+q \\
&\to \frac{(N-1)\frac{1}{\sqrt{N}}}{2-\frac{2}{\sqrt{N}}} \text{ for } n\to\infty \\
&= \frac{\sqrt{N}+1}{2}.
\end{aligned}$$

□

In [11], Giannakopoulos and Koutsoupias proved that for all sequences I of length n, $\text{OPT}(I) \leq \sqrt{n}\cdot\text{NAI}(I)$ for the FIU problem. Here we give tighter results for both the FIF and FIU problems.

Theorem 12. *For the FIF problem, the competitive ratio of* NAI *is* $\frac{\sqrt{N}+1}{2}$*. It is an optimal deterministic online algorithm for the FIF problem.*

Proof. Let f be the frequency of the most frequent item in the input sequence I, so there are nf occurrences of the most frequent item. Let the other $N-1$ items be $b_1, b_2, \ldots, b_{N-1}$. Then, $\sum_{i=1}^{N-1} n_I(b_i) = n-nf$. The total contribution of these $N-1$ items to NAI's result is $\sum_{i=1}^{N-1} n_I(b_i)\frac{n_I(b_i)}{n}$. By Proposition 8, the aggregate frequency of all these $N-1$ distinct items has a lower bound of $\frac{(n-nf)^2}{n(N-1)}$. Thus,

$$\text{NAI}(I) \geq nf^2 + \frac{(n-nf)^2}{n(N-1)} \quad \text{and} \quad \text{OPT}(I) \leq nf.$$

So,

$$\frac{\text{OPT}(I)}{\text{NAI}(I)} \leq \frac{nf}{nf^2+\frac{(n-nf)^2}{n(N-1)}} = \frac{(N-1)f}{Nf^2-2f+1}. \tag{1}$$

The right hand side of Ineq. 1 reaches its maximum when $f=\frac{1}{\sqrt{N}}$. Substituting this value into Ineq. 1, we get

$$\frac{\text{OPT}(I)}{\text{NAI}(I)} \leq \frac{\sqrt{N}+1}{2}.$$

Hence by Theorem 11, the competitive ratio of NAI is $\frac{\sqrt{N}+1}{2}$, and it is an optimal deterministic online algorithm for the FIF problem. □

Corollary 13. *For the FIU problem,* Nai *has competitive function* $\frac{\sqrt{n}}{2}$*. It is an optimal deterministic online algorithm for the FIU problem.*

Proof. Assume $N \leq n$. By Theorem 12, $\frac{\text{Opt}(I)}{\text{Nai}(I)} \leq \frac{\sqrt{n}+1}{2}$. Since there are only n items in the sequence, increasing N cannot make the ratio worse. Therefore, for any value of N, $\frac{\text{Opt}(I)}{\text{Nai}(I)} \leq \frac{\sqrt{n}+1}{2}$. Thus, by the lower bound from Theorem 10, the competitive function of Nai is $\frac{\sqrt{n}}{2}$ and it is optimal. □

For Maj, Giannakopoulos and Koutsoupias [11] proved a competitive ratio of $\Theta(n)$ for the FIU problem. We give asymptotically tight bounds for both the FIF and FIU problems..

Theorem 14. *For the FIF problem, the competitive ratio of* Maj *is* $N-1$ *for* $N \geq 4$ *and* $\frac{4\sqrt{2}+5}{7}$ *and* $1+\frac{2}{\sqrt{3}}$ *for the cases of* $N=2$ *and* $N=3$*, respectively.*

Proof. Let f be the largest frequency of any item in some input sequence I of length n. Opt cannot have an aggregate frequency larger than nf.

Maj buffers the first $\lceil \frac{n}{2} \rceil$ items of $D(I)$ while processing its worst permutation of I. Maj buffers all those items twice, except the $\lceil \frac{n}{2} \rceil$th item in case of odd n, which is buffered only once. If $f \leq \frac{1}{2}$, then the set of the first $\lceil \frac{n}{2} \rceil$ items of $D(I)$ does not contain the most frequent item. Maj's aggregate frequency is smallest when the first $\lceil \frac{n}{2} \rceil$ places of $D(I)$ are equally distributed over all the available $N-1$ remaining items. Therefore $\text{Maj}(I) \geq \frac{n}{2(N-1)}$ is a lower bound (which cannot be tight when $\frac{n}{2(N-1)}$ is not integer). This gives

$$\frac{\text{Opt}(I)}{\text{Maj}(I)} \leq \frac{nf}{\frac{n}{2(N-1)}} \leq N-1.$$

It remains to consider the range $\frac{1}{2} < f \leq 1$. Let b_0 denote the most frequent item in I. Note that b_0 must be in the buffer at some point since $f > \frac{1}{2}$. Since there are $n-fn$ items different from b_0, the total length of all subsequences where b_0 is not in the buffer is at most $2(n-fn)$. This means that b_0 *is* in the buffer at least $n-2(n-fn) = 2fn-n$ times, obtaining an aggregate frequency of at least $(2fn-n)f = 2nf^2 - nf$. By Proposition 8, the remaining items obtain at least $2(n-fn)\frac{n-fn}{n(N-1)}$. Therefore,

$$\text{Maj}(I) \geq 2nf^2 - nf + \frac{2n(1-f)^2}{N-1} \tag{2}$$

and

$$\frac{\text{Opt}(I)}{\text{Maj}(I)} \leq \frac{nf}{2nf^2 - nf + \frac{2n(1-f)^2}{N-1}} = \frac{f}{2f^2 - f + \frac{2(1-f)^2}{N-1}}. \tag{3}$$

The ratio in Ineq. 3 is maximized when $f = \frac{1}{\sqrt{N}}$. However, if $N > 4$, then with $f = \frac{1}{\sqrt{N}}$, the least number of times b_0 can be in the buffer, $(2fn - n)$, becomes negative, which is impossible. For $N = 2$ and $N = 3$, Ineq. 3 gives the upper bounds $\frac{4\sqrt{2}+5}{7}$ and $1 + \frac{2}{\sqrt{3}}$, respectively, on the ratios. Both of these two bounds are greater than the corresponding values of $N - 1$.

Now we consider the case when $N \geq 4$ and $\frac{1}{2} < f \leq 1$. If we can prove that the right hand side of Ineq. 2 is at least $\frac{nf}{N-1}$, then $\frac{\textsc{Opt}(I)}{\textsc{Maj}(I)} \leq \frac{nf}{nf/(N-1)} = N - 1$, and we will be done. The right hand side of Ineq. 2 is at least $\frac{nf}{N-1}$ if and only if

$$2Nf^2 - (N + 4)f + 2 \geq 0. \tag{4}$$

Taking the derivative of the left side of Ineq. 4 shows that it is an increasing function of f for $N \geq 4$ and $f > \frac{1}{2}$. The left side of Ineq. 4 is greater than zero for $N = 4$ and $f > \frac{1}{2}$, so the condition of Ineq. 4 is true. Thus, $\textsc{Opt}(I) \leq (N - 1)\textsc{Maj}(I)$ holds for all f and $N \geq 4$. So, Maj is $(N - 1)$-competitive for $N \geq 4$, $\frac{4\sqrt{2}+5}{7}$-competitive for $N = 2$, and $1 + \frac{2}{\sqrt{3}}$-competitive for $N = 3$.

For a lower bound in the cases $N = 2$ and $N = 3$, we construct a sequence I' with $\frac{n}{\sqrt{N}} + q$ occurrences of the most frequent item where q is the smallest non-negative number such that $N - 1$ divides $n - (\frac{n}{\sqrt{N}} + q)$. We argue that we can find such a q: If $N - 1$ should divide $n - (\frac{n}{\sqrt{N}} + q)$, we must find an integer z such that $n - (\frac{n}{\sqrt{N}} + q) = z(N - 1)$, implying that $q = z(N - 1) - n + \frac{n}{\sqrt{N}}$. For $z = -1$, the right-hand side is negative. It is also clear that we can keep increasing z until the right-hand side is positive. Thus, we can find a smallest integer z making the right-hand side positive, and thereby defining a q with the desired property. With this, $f = \frac{1}{\sqrt{N}} + \frac{q}{n}$. Letting $r(N - 1) = n - (\frac{n}{\sqrt{N}} + q)$, we decide on r occurrences of each of the other $N - 1$ items, and place all those $r(N - 1)$ items in the first $r(N - 1)$ odd places of I'. Then,

$$\textsc{Opt}(I') = (n - 1)f + \frac{r}{n}$$

and

$$\textsc{Maj}(I') = 2nf^2 - nf + \frac{2n(1 - f)^2}{N - 1}.$$

For large values of n, $\frac{\textsc{Opt}(I')}{\textsc{Maj}(I')}$ becomes very close to the rightmost expression of Ineq. 3 and f becomes very close to $\frac{1}{\sqrt{N}}$, which gives the expression for the upper bound. Hence, the competitive ratio of Maj is $\frac{4\sqrt{2}+5}{7}$ and $1 + \frac{2}{\sqrt{3}}$ for the cases of $N = 2$ and $N = 3$ respectively. Note that these values are greater than the corresponding competitive ratios of Nai. Therefore Nai is better than Maj for $N = 2$ and $N = 3$.

For a lower bound for $N \geq 4$, consider the family of sequences $W_{n,r}$ from Definition 4, where $r = \frac{n}{2(N-1)}$. By Proposition 5,

$$\frac{\text{OPT}(W_{n,r})}{\text{MAJ}(W_{n,r})} = \frac{n-1}{2r} + \frac{1}{n} = (N-1) - \frac{N}{n} + \frac{2}{n}.$$

Thus, the competitive ratio of MAJ is $N-1$ for $N \geq 4$. □

Theorem 15. *For the FIU problem,* MAJ *has competitive function* $\frac{n}{2}$.

Proof. For the lower bound, consider the family of sequences W_n from Definition 4. By Proposition 5, $\text{MAJ}(W_n) = 1$, and

$$\text{OPT}(W_n) = \begin{cases} \frac{n}{2} - \frac{1}{2} + \frac{1}{n} & \text{for even } n \\ \frac{n}{2} - 1 + \frac{3}{2n} & \text{for odd } n \end{cases}$$

Consequently, $\text{OPT}(W_n) \geq \frac{n}{2}\text{MAJ}(W_n) - 1$. Thus, the competitive function cannot be better than $\frac{n}{2}$.

For the upper bound, let f be the largest frequency of any item in some input sequence I of length n. OPT cannot have an aggregate frequency larger than nf. If $f \leq \frac{1}{2}$, then, since no algorithm can have an aggregate frequency less than one in total, $\frac{\text{OPT}(I)}{\text{MAJ}(I)} \leq nf \leq \frac{n}{2}$. It remains to consider the range $\frac{1}{2} < f \leq 1$. Since $f > \frac{1}{2}$, the number of distinct items in the input sequence is at most $\frac{n}{2}$, so $N \leq \frac{n}{2}$. By Theorem 14, $\frac{\text{OPT}(I)}{\text{MAJ}(I)} \leq N - 1 < \frac{n}{2}$. This implies that MAJ is $\frac{n}{2}$-competitive and, in total, that the competitive function of MAJ is $\frac{n}{2}$. □

Thus, according to competitive analysis, NAI is better than MAJ for both the FIU and FIF problems.

Theorem 16. *For the FIU problem, the competitive function of* EAG *is* $\frac{n}{2}$.

Proof. For the lower bound, consider the family of sequences E_n from Definition 4. By Proposition 5, $\text{EAG}(E_n) = 2$, and $\text{OPT}(E_n) = n - 4 + \frac{8}{n}$. Thus, $\text{OPT}(E_n) = \frac{n}{2}\text{EAG}(E_n) - 4 + \frac{8}{n}$, and EAG's competitive function cannot be better than $\frac{n}{2}$.

If there are no repeated items in I, then EAG behaves like NAI and that will give $\text{OPT}(I) \leq (\frac{\sqrt{n}}{2} + o(\sqrt{n}))\text{EAG}(I)$ by Corollary 13. It is evident from the lower bound result that the competitive function for EAG is worse than $\frac{\sqrt{n}}{2}$, so we assume that there is at least one repeated item in I. Let time steps $p+1$ and $p+2$ be the first occurrence of a repeated item in I. Let b be the most frequent item in I. Note that b is not necessarily the item which arrived at time steps $p+1$ and $p+2$. After p, all the items could conceivably be b, but among the first p items, at most $\frac{p}{2}$ items can be b, because $p+1$ and $p+2$ are the indices of the first repeated item. So,

an upper bound on the maximum frequency $f_I(b)$ is $\frac{n-p+\frac{p}{2}}{n} = \frac{n-\frac{p}{2}}{n}$. This gives an upper bound of $\textsc{Opt}(I) \leq n\frac{n-\frac{p}{2}}{n} = n - \frac{p}{2}$.

Now we consider a lower bound on $\textsc{Eag}(I)$. In the worst case for Eag, all the items before $p+1$ are distinct, so their contribution to $\textsc{Eag}(I)$ is at least $\frac{p}{n}$. In the worst case for Eag, the item that occurs at time steps $p+1$ and $p+2$ has frequency $\frac{2}{n}$, so the contribution to $\textsc{Eag}(I)$ from the items after p is at least $(n-p)\frac{2}{n}$. Thus, $\textsc{Eag}(I) \geq \frac{p}{n} + (n-p)\frac{2}{n} = 2 - \frac{p}{n}$, and

$$\frac{\textsc{Opt}(I)}{\textsc{Eag}(I)} \leq \frac{n-\frac{p}{2}}{2-\frac{p}{n}} = \frac{n}{2}.$$

Hence, Eag has competitive function $\frac{n}{2}$. □

Corollary 17. *For the FIF problem,* Eag *is not competitive.*

Proof. From the lower bound proof of Theorem 16, we can see that irrespective of the value of $N \geq 2$, Eag has competitive function $\frac{n}{2}$, i.e., it is a function of n. Thus, Eag is not competitive according to the original definition of the competitive ratio. □

Although Eag has the same competitive ratio as Maj for the FIU problem, according to competitive analysis, Maj is better than Eag for the FIF problem. Clearly, competitive analysis indicates that Nai is better than Eag for both problems.

4. Relative Interval Analysis

Dorrigiv et al. [10] proposed another analysis method, relative interval analysis, in the context of paging. Relative interval analysis compares two online algorithms directly, i.e., it does not use the optimal offline algorithm as the baseline of the comparison. This direct comparison, which is also a property of the measure in the next section on relative worst order analysis, gives an advantage over competitive analysis in being able to differentiate between pairs of algorithms, where one does at least as well as the other on every sequence. For example, for the paging problem, both relative interval analysis [10] and relative worst order analysis [5] indicate that Least-Recently-Used (LRU) is better than Flush-When-Full (FWF), whereas these two very different algorithms have the same competitive ratio. In general, it compares two algorithms on the basis of their minimal and maximal differences in profit relative to the length of the input sequence, where the profit in our context is the aggregate frequency. In this way, both best and worst case performances matter. Furthermore, results are made comparable by dividing by the input length. This seems reasonable if one assumes that the maximal profit possible depends linearly

on the input length. Here we define this analysis for maximization problems for two algorithms $\mathcal{A}$ and $\mathcal{B}$, following [10].

Definition 18. *Define*

$$\mathit{Min}_{\mathcal{A},\mathcal{B}}(n) = \min_{|I|=n} \{\mathcal{A}(I) - \mathcal{B}(I)\} \text{ and } \mathit{Max}_{\mathcal{A},\mathcal{B}}(n) = \max_{|I|=n} \{\mathcal{A}(I) - \mathcal{B}(I)\},$$

and

$$\mathit{Min}(\mathcal{A},\mathcal{B}) = \liminf_{n\to\infty} \frac{\mathit{Min}_{\mathcal{A},\mathcal{B}}(n)}{n} \text{ and } \mathit{Max}(\mathcal{A},\mathcal{B}) = \limsup_{n\to\infty} \frac{\mathit{Max}_{\mathcal{A},\mathcal{B}}(n)}{n}.$$

The relative interval *of $\mathcal{A}$ and $\mathcal{B}$ is defined as*

$$l(\mathcal{A},\mathcal{B}) = [\mathit{Min}(\mathcal{A},\mathcal{B}), \mathit{Max}(\mathcal{A},\mathcal{B})].$$

If $\mathit{Max}(\mathcal{A},\mathcal{B}) > |\mathit{Min}(\mathcal{A},\mathcal{B})|$, then $\mathcal{A}$ is said to have better performance *than $\mathcal{B}$ in this model.*

Note that $\mathrm{Min}(\mathcal{A},\mathcal{B}) = -\mathrm{Max}(\mathcal{B},\mathcal{A})$ and $\mathrm{Max}(\mathcal{A},\mathcal{B}) = -\mathrm{Min}(\mathcal{B},\mathcal{A})$.

For any pair of algorithms, $\mathcal{A}$ and $\mathcal{B}$, for the frequent items problem, there is a trivial upper bound on $\mathrm{Max}(\mathcal{A},\mathcal{B})$ and lower bound on $\mathrm{Min}(\mathcal{A},\mathcal{B})$.

Proposition 19. *For any pair of algorithms $\mathcal{A}$ and $\mathcal{B}$, $\mathit{Max}(\mathcal{A},\mathcal{B}) \leq 1$ and $\mathit{Min}(\mathcal{A},\mathcal{B}) \geq -1$.*

Proof. The maximum aggregate frequency any algorithm could have is for a sequence where all items are identical, giving the value n. The minimum is for a sequence where all items are different, giving the value 1. The stated bounds follow since $\limsup_{n\to\infty} \frac{n-1}{n} = 1$. □

4.1. *Naive vs. Eager*

According to relative interval analysis, NAI has better performance than EAG.

Theorem 20. *For the FIF problem, according to relative interval analysis,* $l(\mathrm{NAI}, \mathrm{EAG}) = [-\frac{1}{4} + \frac{1}{4N}, 1]$.

Proof. By Proposition 19, $\mathrm{Max}(\mathrm{NAI}, \mathrm{EAG}) \leq 1$.

We now consider a lower bound on $\mathrm{Max}(\mathrm{NAI}, \mathrm{EAG})$. By Proposition 5, $\mathrm{NAI}(E_n) - \mathrm{EAG}(E_n) = (n - 4 + \frac{8}{n}) - 2$, so

$$\limsup_{n\to\infty} \frac{\mathrm{NAI}(E_n) - \mathrm{EAG}(E_n)}{n} = \limsup_{n\to\infty} \frac{n - 6 + \frac{8}{n}}{n} = 1.$$

Thus, $\mathrm{Max}(\mathrm{NAI}, \mathrm{EAG}) = 1$.

We now consider $\text{Min}(\text{NAI}, \text{EAG})$. For a lower bound on $\text{Min}(\text{NAI}, \text{EAG})$, assume that among the sequences of length n, I gives the smallest possible value of $\text{NAI}(I) - \text{EAG}(I)$. From the definitions of NAI and EAG, it is evident that there must be at least one repeated item if $\text{NAI}(I) - \text{EAG}(I) < 0$. Suppose that the first repeated item is a, which is repeated at time steps t and $t+1$. The item a must be the most frequent item in I; otherwise two occurrences of the most frequent item can be swapped with the item at t and $t+1$ to get a smaller value of $\text{NAI}(I) - \text{EAG}(I)$. Since at and after time step t, EAG buffers only the most frequent item, $\text{NAI}(I) - \text{EAG}(I)$ is minimized when $t = 1$, i.e., a must occur at the first two time steps of I. That gives

$$\text{NAI}(I) \geq \frac{n_I(a)^2}{n} + (n - n_I(a))\frac{n - n_I(a)}{n(N-1)} \text{ and } \text{EAG}(I) = n\frac{n_I(a)}{n} \text{ so,}$$

$$\begin{aligned} \text{NAI}(I) - \text{EAG}(I) &\geq \frac{(n - n_I(a))^2}{n(N-1)} - \frac{n_I(a)(n - n_I(a))}{n} \\ &= \left(1 - \frac{n_I(a)}{n}\right)\frac{n - Nn_I(a)}{N-1} \end{aligned} \tag{5}$$

The right hand side of Ineq. 5 is minimized when $n_I(a) = \frac{n(N+1)}{2N}$. Substituting this value of $n_I(a)$ into Ineq. 5, we get

$$\text{NAI}(I) - \text{EAG}(I) \geq \left(1 - \frac{N+1}{2N}\right)\frac{n - \frac{n(N+1)}{2}}{N-1} = -\frac{n(N-1)}{4N}$$

So,

$$\text{Min}(\text{NAI}, \text{EAG}) = \liminf_{n\to\infty} \frac{\text{Min}_{\text{NAI},\text{EAG}}(n)}{n} \geq -\frac{1}{4} + \frac{1}{4N}. \tag{6}$$

For an upper bound on $\text{Min}(\text{NAI}, \text{EAG})$, we construct an input sequence I' with $n = 2Nr$ for any positive integer r. Suppose that the most frequent item in I' is a, $n_{I'}(a) = r(N+1)$, and the other $N-1$ items have r occurrences each. Let the first two items of I' be a. The input sequence I' is identical to the lower bound sequence I, so $\text{NAI}(I') - \text{EAG}(I') = -\frac{n(N-1)}{4N}$ and $\text{Min}(\text{NAI}, \text{EAG}) \leq -\frac{1}{4} + \frac{1}{4N}$. Thus, $l(\text{NAI}, \text{EAG}) = [-\frac{1}{4} + \frac{1}{4N}, 1]$. □

Corollary 21. *For the FIU problem, according to relative interval analysis,* $l(\text{NAI}, \text{EAG}) = [-\frac{1}{4}, 1]$.

Proof. The proof of Theorem 20 holds here also and gives this result asymptotically when the size of the universe is unbounded. Note that in the upper bound for $-\frac{1}{4} + \frac{1}{4N}$, one can let $r = 1$ and $N = n/2$ to give this asymptotic result. □

4.2. *Naive vs. Majority*

NAI and MAJ are equally good according to relative interval analysis.

Theorem 22. *For the FIF problem, according to relative interval analysis,* $l(\text{NAI}, \text{MAJ}) = [-\frac{1}{4} + \frac{1}{4N}, \frac{1}{4} - \frac{1}{4N}]$.

Proof. For the maximum value of $\text{NAI}(I) - \text{MAJ}(I)$, it is sufficient to consider the worst permutation of I for MAJ since NAI has the same output for all permutations of I. For the worst permutation, $\text{MAJ}_W(I)$ will buffer only the first $\lceil \frac{n}{2} \rceil$ items of the distribution $D(I)$. The first $\lfloor \frac{n}{2} \rfloor$ items will be buffered twice and in case of odd n, the $\lceil \frac{n}{2} \rceil$th item will be stored once at the last time step. Let $D(I) = b'_1, b'_2, b'_3, \ldots, b'_n$.

$$\begin{aligned}\text{NAI}(I) - \text{MAJ}_W(I) &= \sum_{i=1}^{n} f_I(a'_i) - 2\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_I(a'_i) - \left(\left\lceil \frac{n}{2} \right\rceil - \left\lfloor \frac{n}{2} \right\rfloor\right) f_I(a'_{\lceil \frac{n}{2} \rceil}) \\ &= \sum_{i=\lceil \frac{n+2}{2} \rceil}^{n} f_I(a'_i) - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_I(a'_i). \end{aligned} \tag{7}$$

The right hand side of Eq. 7 is the difference between the aggregate frequencies of the last $\lfloor \frac{n}{2} \rfloor$ items and the first $\lfloor \frac{n}{2} \rfloor$ items of $D(I)$. Denote these two values of aggregate frequencies by B_l and B_f, respectively. Let p be the number of occurrences of the most frequent item a in I. An upper bound on B_l is $\lfloor \frac{n}{2} \rfloor \frac{p}{n}$. For a lower bound on B_f, the aggregate frequency of $n - p$ items other than a must be minimum. If $p \geq \lceil \frac{n}{2} \rceil$, then a contributes $(p - \lceil \frac{n}{2} \rceil)\frac{p}{n}$ to B_f, so by Proposition 8, a lower bound on B_f is $(p - \lceil \frac{n}{2} \rceil)\frac{p}{n} + \frac{(n-p)^2}{n(N-1)}$. Therefore, if $p \geq \lceil \frac{n}{2} \rceil$, then

$$\begin{aligned}\text{NAI}(I) - \text{MAJ}(I) &\leq \left\lfloor \frac{n}{2} \right\rfloor \frac{p}{n} - (p - \left\lceil \frac{n}{2} \right\rceil)\frac{p}{n} - \frac{(n-p)^2}{n(N-1)} \\ &= p - \frac{p^2}{n} - \frac{(n-p)^2}{n(N-1)}. \end{aligned} \tag{8}$$

From Ineq. 8, the upper bound on the difference is maximized when $p = \frac{n(N+1)}{2N}$. This gives an upper bound on $\text{NAI}(I) - \text{MAJ}(I)$ of $\frac{n}{4} - \frac{n}{4N}$.

If $p < \lceil \frac{n}{2} \rceil$, then an upper bound on B_l is $\frac{n}{2}\frac{1}{2} = \frac{n}{4}$ and by Proposition 8, a lower bound on B_f is $\frac{n-1}{2}\frac{n-1}{2(N-1)n}$. Therefore, if $p < \lceil \frac{n}{2} \rceil$, then

$$\text{NAI}(I) - \text{MAJ}(I) \leq \frac{n}{4} - \frac{n^2 - 2n + 1}{4n(N-1)}. \tag{9}$$

Thus, for any value of p, $\text{Max}(\text{NAI}, \text{MAJ}) \leq \frac{1}{4} - \frac{1}{4N}$.

For a lower bound on the maximum value of $\text{NAI}(I) - \text{MAJ}(I)$, we construct a family of sequences $I_{n,r}$ of length $n = 2rN$ for any positive integer r. $I_{n,r}$ is defined as

$$I_{n,r} = (b_1, b_0)^r, (b_2, b_0)^r, \ldots, (b_{N-1}, b_0)^r, (b_0)^{2r},$$

where there are $(N-1)r + 2r = (N+1)r$ copies of b_0, which is the most frequent item in $I_{n,r}$. All the other items in $I_{n,r}$ have $r = \frac{n}{2N}$ occurrences each. Thus,

$$\textsc{Nai}(I_{n,r}) = \frac{(N+1)^2 r^2}{n} + (N-1)\frac{r^2}{n} \text{ and}$$
$$\textsc{Maj}(I_{n,r}) = 2(N-1)r\frac{r}{n} + 2r\frac{(N+1)r}{n}.$$

Therefore, by substituting the value of r,

$$\textsc{Nai}(I_{n,r}) - \textsc{Maj}(I_{n,r}) = \frac{n}{4} - \frac{n}{4N}.$$

Thus, $\text{Max}(\textsc{Nai}, \textsc{Maj}) \geq \limsup_{n\to\infty} \frac{\textsc{Nai}(I_{n,r}) - \textsc{Maj}(I_{n,r})}{n} = \frac{1}{4} - \frac{1}{4N}$, matching the upper bound.

To derive the minimum value of $\textsc{Nai}(I) - \textsc{Maj}(I)$, we calculate the maximum value of $\textsc{Maj}(I) - \textsc{Nai}(I)$. For an upper bound, we consider the best permutation I_B for Maj of an arbitrary sequence I. For I_B, Maj buffers the half of the requests in the sequence with the highest frequencies. The difference, $\textsc{Maj}(I_B) - \textsc{Nai}(I_B)$, is

$$\begin{aligned} & 2\sum_{i=\lceil \frac{n+2}{2} \rceil}^{n} f_I(a'_i) + \left(\left\lceil \frac{n}{2} \right\rceil - \left\lfloor \frac{n}{2} \right\rfloor \right) f_I(a'_{\lceil \frac{n}{2} \rceil}) - \sum_{i=1}^{n} f_I(a'_i) \\ = & \sum_{i=\lceil \frac{n+2}{2} \rceil}^{n} f_I(a'_i) - \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} f_I(a'_i). \end{aligned} \tag{10}$$

This expression is exactly the same as the expression for $\textsc{Nai}(I) - \textsc{Maj}_W(I)$ from Eq. 7, so we get the same upper bound, $\frac{n}{4} - \frac{n}{4N}$. For a lower bound on $\text{Max}(\textsc{Maj}, \textsc{Nai})$, we use a specific permutation $I'_{n,r}$ of $I_{n,r}$.

$$I'_{n,r} = (b_0)^{r(N+1)}, (b_1)^r, (b_2)^r, \ldots, (b_{N-1})^r.$$

Consequently,

$$\textsc{Nai}(I'_{n,r}) = \textsc{Nai}(I_{n,r}) = \frac{(N+1)^2 r^2}{n} + (N-1)\frac{r^2}{n} \text{ and}$$
$$\textsc{Maj}(I'_{n,r}) = n\frac{(N+1)r}{n} = (N+1)r.$$

Therefore, by substituting the value of r,

$$\textsc{Maj}(I'_{n,r}) - \textsc{Nai}(I'_{n,r}) = \frac{n}{4} - \frac{n}{4N}.$$

Thus, $\text{Max}(\textsc{Maj}, \textsc{Nai}) \geq \limsup_{n\to\infty} \frac{\textsc{Maj}(I'_{n,r}) - \textsc{Nai}(I'_{n,r})}{n} = \frac{1}{4} - \frac{1}{4N}$, matching the upper bound. Hence, $\text{Min}(\textsc{Nai}, \textsc{Maj}) = -\text{Max}(\textsc{Maj}, \textsc{Nai}) = -\frac{1}{4} + \frac{1}{4N}$, and $l(\textsc{Nai}, \textsc{Maj}) = [-\frac{1}{4} + \frac{1}{4N}, \frac{1}{4} - \frac{1}{4N}]$. □

Corollary 23. *For the FIU problem, according to relative interval analysis,* $l(\textsc{Nai}, \textsc{Maj}) = [-\frac{1}{4}, \frac{1}{4}]$.

Proof. Substituting the values $r = 1$ and $N = n/2$ into the proof of Theorem 22 gives these asymptotic results. □

4.3. *Majority vs. Eager*

According to relative interval analysis, MAJ has better performance than EAG.

Theorem 24. *For the FIF problem, according to relative interval analysis,* $l(\text{MAJ}, \text{EAG}) = [-\frac{1}{2} + \frac{1}{2N}, 1]$.

Proof. By Proposition 19, $\text{Max}(\text{MAJ}, \text{EAG}) \leq 1$. For a lower bound on $\text{Max}(\text{MAJ}, \text{EAG})$, we consider the family of sequences E_n from Definition 4. By Proposition 5, $\text{MAJ}(E_n) - \text{EAG}(E_n) = (n - 6 + \frac{16}{n}) - 2 = n - 8 + \frac{16}{n}$, and $\text{Max}(\text{MAJ}, \text{EAG}) \geq \limsup_{n\to\infty} \frac{n-8+\frac{16}{n}}{n} = 1$. Thus, $\text{Max}(\text{MAJ}, \text{EAG}) = 1$.

For $\text{Min}(\text{MAJ}, \text{EAG})$, we consider $\text{Max}(\text{EAG}, \text{MAJ})$. First we calculate an upper bound on $\text{EAG}(I) - \text{MAJ}(I)$. Suppose that the input sequence I of length n gives the maximum value of $\text{EAG}(I) - \text{MAJ}(I)$ over all sequences of length n. Suppose I has k distinct items $b_1, b_2, b_3, \ldots, b_k$, and let $f_i = f_I(b_i)$ and $n_i = n_I(b_i)$ for all i. Assume that $f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_k$, so b_k is the most frequent item.

First, assume $n_k \leq \frac{n}{2}$ (for odd n, $n_k \leq \frac{n-1}{2}$). MAJ buffers the first $\frac{n}{2}$ items of $D(I)$ if n is even and it buffers the first $\frac{n+1}{2}$ items of $D(I)$ if n is odd. We get a lower bound on $\text{MAJ}(I)$ by distributing the remaining $N - 1$ distinct items over these $\frac{n}{2}$ (for odd n, $\frac{n+1}{2}$) places. So, $\text{MAJ}(I) \geq n\frac{n}{2n(N-1)} = \frac{n}{2(N-1)}$, and

$$\text{EAG}(I) - \text{MAJ}(I) \leq nf_k - \frac{n}{2(N-1)} \leq n\frac{\frac{n}{2}}{n} - \frac{n}{2(N-1)} = \frac{n}{2}\left(1 - \frac{1}{N-1}\right) \quad (11)$$

It remains to consider the range $\frac{n}{2} < n_k \leq n$. Assume that for some positive integer q', such that $q = q' - \frac{\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor}{2}$, we have $n_k = \frac{n}{2} + q$. From Proposition 7, we know that MAJ's result has the lower bound $\text{MAJ}_W(I) \geq 2(\sum_{i=1}^{k-1} n_i f_i + q f_k)$. The summation is minimized when the $k - 1$ least frequent items have the same frequencies. Since those $k - 1$ distinct items are distributed over the remaining $\frac{n}{2} - q$ places, in this case, $\text{MAJ}(I) \geq 2\left((\frac{n}{2} - q)\frac{\frac{n}{2}-q}{n(N-1)} + q\frac{\frac{n}{2}+q}{n}\right)$. Hence,

$$\begin{aligned}\text{EAG}(I) - \text{MAJ}(I) &\leq \frac{n}{2} + q - 2\left(\left(\frac{n}{2} - q\right)\frac{\frac{n}{2} - q}{n(N-1)} + q\frac{\frac{n}{2} + q}{n}\right) \\ &= \frac{n}{2}\left(1 - \frac{1}{N-1}\right) - 2q\left(\frac{qN - n}{n(N-1)}\right) \quad (12)\end{aligned}$$

The right hand side of Ineq. 12 is maximized when $q = \frac{n}{2N}$. That gives

$$\text{EAG}(I) - \text{MAJ}(I) \leq \frac{n}{2}\left(1 - \frac{1}{N}\right). \quad (13)$$

For a lower bound on the maximum value of $\textsc{Eag}(I) - \textsc{Maj}(I)$, we construct a family of sequences $I_{n,r}$ of length $n = 2rN$ for any integer $r \geq 2$. $I_{n,r}$ is defined as

$$I_{n,r} = b_0, b_0, b_1, b_1, (b_1, b_0)^{r-2}, (b_2, b_0)^r, \ldots, (b_{N-1}, b_0)^r, (b_0)^{2r}$$

where there are $(N-1)r + 2r = (N+1)r$ occurrences of b_0, which is the most frequent item in $I_{n,r}$. All the other items in $I_{n,r}$ have $r = \frac{n}{2N}$ occurrences each.

$$\textsc{Eag}(I_{n,r}) = n\frac{(N+1)r}{n} \text{ and}$$
$$\textsc{Maj}(I_{n,r}) = 4\frac{(N+1)r}{n} + 2((N-1)r - 2)\frac{r}{n} + 2r\frac{(N+1)r}{n}.$$

Therefore, by substituting the value of r,

$$\textsc{Eag}(I_{n,r}) - \textsc{Maj}(I_{n,r}) = \frac{n}{2}(1 - \frac{1}{N}) - 2.$$

From this lower bound and the upper bound from Eq. 13, we have

$$\text{Min}(\textsc{Maj}, \textsc{Eag}) = -\text{Max}(\textsc{Eag}, \textsc{Maj}) = -\limsup_{n\to\infty} \frac{\textsc{Eag}(I) - \textsc{Maj}(I)}{n} = -\frac{1}{2} + \frac{1}{2N}.$$

Therefore $l(\textsc{Maj}, \textsc{Eag}) = [-\frac{1}{2} + \frac{1}{2N}, 1]$. □

Corollary 25. *For the FIU problem, according to relative interval analysis,* $l(\textsc{Maj}, \textsc{Eag}) = [-\frac{1}{2}, 1]$.

Proof. Substituting the values $k = N = n - n_k + 1$ in the proof of Theorem 24 gives these asymptotic results. □

5. Relative Worst Order Analysis

Relative worst order analysis [4] compares two online algorithms directly. It compares two algorithms on their worst orderings of sequences which have the same content, but possibly different order. Thus, rather than comparing algorithms directly to each other on the same sequences, it compares algorithms on the set of sequences having the same content, but in a different order. The result is that, compared with relative worst order analysis, fewer pairs of algorithms are simply incomparable. For example, under relative worst order analysis, conservative algorithms for paging can be proven to be equivalent, whereas this cannot be established for relative interval analysis due to incomparability issues [10, 5]. The definition of this measure is somewhat more involved; see [5] for more intuition on the various elements.

Definition 26. *Using the definition of worst aggregate frequency over permutations of sequences from Definition 6, for any pair of algorithms* $\mathcal{A}$ *and* $\mathcal{B}$*, we define*

$$c_l(\mathcal{A}, \mathcal{B}) = \sup\{c \mid \forall I : \mathcal{A}_W(I) \geq c\mathcal{B}_W(I) - b\} \text{ and}$$
$$c_u(\mathcal{A}, \mathcal{B}) = \inf\{c \mid \forall I : \mathcal{A}_W(I) \leq c\mathcal{B}_W(I) + b\}.$$

If $c_l(\mathcal{A},\mathcal{B}) \geq 1$ *or* $c_u(\mathcal{A},\mathcal{B}) \leq 1$*, the algorithms are said to be* comparable *and the* relative worst order ratio $WR_{\mathcal{A},\mathcal{B}}$ *of algorithm* $\mathcal{A}$ *to algorithm* $\mathcal{B}$ *is defined. Otherwise,* $WR_{\mathcal{A},\mathcal{B}}$ *is undefined.*

$$\begin{aligned} &\textit{If } c_l(\mathcal{A},\mathcal{B}) \geq 1 \textit{ then } \ WR_{\mathcal{A},\mathcal{B}} = c_u(\mathcal{A},\mathcal{B}), \textit{ and} \\ &\textit{if } c_u(\mathcal{A},\mathcal{B}) \leq 1 \textit{ then } \ WR_{\mathcal{A},\mathcal{B}} = c_l(\mathcal{A},\mathcal{B}). \end{aligned}$$

If $WR_{\mathcal{A},\mathcal{B}} > 1$*, the algorithms* $\mathcal{A}$ *and* $\mathcal{B}$ *are said to be* comparable in $\mathcal{A}$'s favor*. Similarly, if* $WR_{\mathcal{A},\mathcal{B}} < 1$*, algorithms are said to be* comparable in $\mathcal{B}$'s favor*.*

For the FIF problem, we use the above definition of relative worst order analysis. However, for the FIU problem, as in the case of competitive analysis, the relative performance of the algorithms depends on the length of the input sequence I. As in Section 3, we define a generalized version of relative worst order analysis for use with the FIU problem. The definition is given for a maximization problem, but is trivially adaptable to be used for minimization problems as well; only the decision as to when which algorithm is better would change.

The following definition is parameterized by a total ordering, $\sqsubseteq$, since we will later use it for both $\leq$ and $\geq$.

Definition 27. *f is a* $(\mathcal{A},\mathcal{B},\sqsubseteq)$-function *if*

$$\forall I\colon\ \mathcal{A}_W(I) \sqsubseteq (f(n) + o(f(n))) \cdot \mathcal{B}_W(I),$$

where $\mathcal{A}$ *and* $\mathcal{B}$ *are algorithms and* $\sqsubseteq$ *is a total ordering. Recall from Definition 6 that the notation* $\textsc{Alg}_W(I)$*, where* Alg *is some algorithm, denotes the result of* Alg *on its worst permutation of* I*.*

f is a bounding function with respect to $(\mathcal{A},\mathcal{B},\sqsubseteq)$ *if f is a* $(\mathcal{A},\mathcal{B},\sqsubseteq)$*-function and for any* $(\mathcal{A},\mathcal{B},\sqsubseteq)$*-function* g*,* $\lim_{n\to\infty} \frac{f(n)}{g(n)} \sqsubseteq 1$*.*

If f is a bounding function with respect to $(\mathcal{A},\mathcal{B},\leq)$ *and g is a bounding function with respect to* $(\mathcal{A},\mathcal{B},\geq)$*, then* $\mathcal{A}$ *and* $\mathcal{B}$ *are said to be* comparable *if* $\lim_{n\to\infty} f(n) \leq 1$ *or* $\lim_{n\to\infty} g(n) \geq 1$*.*

If $\lim_{n\to\infty} f(n) \leq 1$*, then* $\mathcal{B}$ *is better than* $\mathcal{A}$ *and* $g(n)$ *is a* relative worst order function of $\mathcal{A}$ and $\mathcal{B}$*, and if* $\lim_{n\to\infty} g(n) \geq 1$*, then* $\mathcal{A}$ *is better than* $\mathcal{B}$ *and* $f(n)$ *is a* relative worst order function of $\mathcal{A}$ and $\mathcal{B}$*.*

We use $\mathrm{WR}_{\mathcal{A},\mathcal{B}} = f(n)$ to indicate that $f(n)$ belongs to the equivalence class of *relative worst order functions of* $\mathcal{A}$ *and* $\mathcal{B}$*.*

The competitive function, as given in Definition 9, could also have been defined using this framework, but was defined separately as a gentle introduction to the idea.

5.1. *Naive vs. Optimal*

Relative worst order analysis can show the strength of the simple, but adaptive, NAI algorithm by comparing it with the powerful OPT. NAI is an optimal algorithm according to relative worst order analysis, in the sense that it is equivalent to OPT. We first prove that NAI and OPT are equivalent.

Theorem 28. *For both the FIU and FIF problems, according to relative worst order analysis,* $WR_{\text{OPT,NAI}} = 1$.

Proof. In the aggregate frequency problem, even though OPT knows the entire sequence in advance, it cannot store an item before it first appears in the sequence. Thus, for any input sequence I, the worst permutation for OPT is the sorting of I according to the increasing order of the frequencies of the items, as expressed by $D(I)$. On this ordering, OPT is forced to behave like NAI. Therefore, $c_u(\text{NAI}, \text{OPT}) = c_l(\text{NAI}, \text{OPT}) = 1$, proving the theorem. □

5.2. *Naive vs. Eager*

From Theorem 28, NAI must be at least as good as EAG according to relative worst order analysis. The following result shows that it is strictly better.

Theorem 29. *For the FIU problem, according to relative worst order analysis,* $WR_{\text{NAI,EAG}} = \frac{n}{2}$.

Proof. From Theorem 28, we know that for OPT's worst permutation I_W of any sequence I, $\text{OPT}(I_W) = \text{NAI}(I_W)$. Any arbitrary online algorithm $\mathcal{A}$ cannot be better than OPT on any sequence. Therefore $c_l(\text{NAI}, \text{EAG}) \geq 1$, so NAI and EAG are comparable.

For any arbitrary online algorithm $\mathcal{A}$ and a worst order I_W for $\mathcal{A}$ of any sequence I, $\frac{\text{NAI}(I_W)}{\mathcal{A}(I_W)} = \frac{\text{OPT}(I_W)}{\mathcal{A}(I_W)}$, so a competitive function of $\mathcal{A}$ is an upper bound on $c_u(\text{NAI}, \mathcal{A})$. By Theorem 16, a competitive function of EAG is $\frac{n}{2}$, for any value of $N \geq 2$. Thus, $c_u(\text{NAI}, \text{EAG}) \leq \frac{n}{2}$.

For a lower bound, consider the family of sequences E_n from Definition 4. These sequences are in the worst ordering for both EAG and OPT. By Proposition 5, $\text{NAI}(E_n) = n - 4 + \frac{8}{n}$ and $\text{EAG}(E_n) = 2$. Thus,

$$\text{NAI}_W(E_n) = \frac{n}{2}\text{EAG}_W(E_n) - 4 + \frac{8}{n}$$

and $\text{WR}_{\text{NAI,EAG}} = \frac{n}{2}$. □

Since we use the original definition of relative worst order for the FIF problem rather than relative worst order functions, according to this definition, NAI is unboundedly better than EAG.

Corollary 30. *For the FIF problem, according to relative worst order analysis,* $WR_{\text{NAI,EAG}} = \infty$.

Proof. Since the lower bound in the proof of Theorem 29 uses the sequence E_n, where $N = 2$, the proof also holds for the problem where the size of the universe is bounded. The relative worst order function from Theorem 29 is unbounded, so using the original definitions, $\text{WR}_{\text{NAI,EAG}} = \infty$. □

5.3. *Naive vs. Majority*

According to relative worst order analysis, NAI is better than MAJ, though not quite as much better as compared to EAG.

Theorem 31. *For the FIF problem, according to relative worst order analysis,* $WR_{\text{NAI,MAJ}} = \frac{N}{2}$ *for* $N \geq 4$.

Proof. Since NAI and OPT perform the same on their worst orderings of any sequence, NAI and MAJ are comparable and $c_l(\text{NAI}, \text{MAJ}) \geq 1$.

Next we derive the value of $c_u(\text{NAI}, \text{MAJ})$. Since NAI's aggregate frequency is the same on any ordering of that sequence, we can compare NAI and MAJ on the same sequence, namely MAJ's worst ordering of it; that is also a worst ordering for NAI. Suppose the input sequence I gives the largest ratio of $\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)}$ for sequences of length n. Suppose I has k distinct items $b_1, b_2, \ldots, b_k$, and let $f_i = f_I(b_i)$ and $n_i = n_I(b_i)$ for all i. Assume that $f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_k$, so b_k is the most frequent item.

If $n_k \leq \lfloor \frac{n}{2} \rfloor$, then

$$\begin{aligned} \frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} &= \frac{\sum_{i=1}^{k} n_i f_i}{2(\sum_{i=1}^{j-1} n_i f_i + p f_j) + \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) f_j} \\ &= \frac{\sum_{i=1}^{k} n_i^2}{2(\sum_{i=1}^{j-1} n_i^2 + p n_j) + \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) n_j} \end{aligned} \tag{14}$$

where $j \leq k$ is the largest index such that $\sum_{i=1}^{j-1} n_i + p = \lceil \frac{n}{2} \rceil$ for some non-negative integer p. We create another sequence I' from I by replacing all the b_i's where $j < i < k$ with b_k, and by replacing $n_j - p - \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right)$ b_j's with b_k. I' will have $j+1$ distinct items and the most frequent item will have $\lfloor \frac{n}{2} \rfloor$ occurrences. Since all these changes will increase the numerator and not change the denominator

in Eq. 14, I' will give at least as large a ratio as I, so we consider the sequence I' instead of I. Suppose the distinct items of I', in nondecreasing order of frequency, are $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{j+1}$ and the corresponding counts are $\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_{j+1}$ and $\hat{n}_{j+1} = \lfloor \frac{n}{2} \rfloor$. Then,

$$\frac{\text{NAI}_W(I')}{\text{MAJ}_W(I')} \leq \frac{\lfloor \frac{n}{2} \rfloor^2 + \sum_{i=1}^{j} \hat{n}_i^2}{2\sum_{i=1}^{j} \hat{n}_i^2 - \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) \hat{n}_j} \tag{15}$$

The above ratio is maximized when the term $\sum_{i=1}^{j} \hat{n}_i^2$ obtains its minimum. Since $\sum_{i=1}^{j} \hat{n}_i = \lceil \frac{n}{2} \rceil$, by Proposition 8 this occurs when all $\hat{n}_i$ are equal and $j = N - 1$. Thus,

$$\frac{\text{NAI}_W(I')}{\text{MAJ}_W(I')} \leq \frac{\lfloor \frac{n}{2} \rfloor^2 + (N-1)\left(\lceil \frac{n}{2} \rceil \frac{1}{N-1}\right)^2}{2(N-1)\left(\lceil \frac{n}{2} \rceil \frac{1}{N-1}\right)^2} \leq \frac{N}{2} \tag{16}$$

is an upper bound (which cannot be tight when $\frac{n}{2(N-1)}$ is not integer).

It remains to consider the range $\frac{n}{2} < n_k \leq n$. In this case,

$$\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} = \frac{\sum_{i=1}^{k} n_i f_i}{2(\sum_{i=1}^{k-1} n_i f_i + q f_k)} = \frac{n_k^2 + \sum_{i=1}^{k-1} n_i^2}{2qn_k + 2\sum_{i=1}^{k-1} n_i^2} \tag{17}$$

where $\sum_{i=1}^{k-1} n_i + q = \frac{n}{2}$ for some positive integer q' such that $q = q' - \frac{\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor}{2}$. Note that for odd n, $2q = 2q' - 1$ and MAJ buffers b_k for $2q$ time steps. As in the case of $n_k \leq \lfloor \frac{n}{2} \rfloor$, the ratio is maximized when $\sum_{i=1}^{k-1} n_i^2$ is minimized. So,

$$\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} \leq \frac{(\frac{n}{2} + q)^2 + \frac{(\frac{n}{2} - q)^2}{N-1}}{2q(\frac{n}{2} + q) + 2\frac{(\frac{n}{2} - q)^2}{N-1}}. \tag{18}$$

The right hand side of Ineq. 18 is maximized when q is equal to either $\frac{n}{\sqrt{N}} - \frac{n}{2}$ or $-(\frac{n}{\sqrt{N}} + \frac{n}{2})$. Since q is positive, the only option is $q = \frac{n}{\sqrt{N}} - \frac{n}{2}$ and $N = 2$ or $N = 3$. For all the other values of N, the maximum ratio is given by the case of $n_k \leq \lfloor \frac{n}{2} \rfloor$. However, substituting the above value of q and $N = 2$ and $N = 3$ in Ineq. 18 gives

$$\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} \leq \begin{cases} \frac{6+2\sqrt{2}}{7} < 1.262 & \text{for } N = 2 \\ \frac{3+\sqrt{3}}{3} < 1.578 & \text{for } N = 3 \\ \frac{N}{2} & \text{for } N \geq 4 \end{cases} \tag{19}$$

In the case of $N = 2$ and $N = 3$, for a lower bound on the maximum value of the ratio, we take a sequence with sufficiently large n, such that we can get an almost integer value of q. A worst permutation for MAJ can easily be built by placing all the items that are not most frequent in consecutive odd places starting from the first place. This sequence approximately gives the same values as in Ineq. 19. In the case of $N \geq 4$, we use $W_{n,r}$, which gives the ratio of $\frac{n}{4r} + \frac{1}{2} = \frac{N}{2}$, since $r = \frac{n}{2(N-1)}$.

Hence, the upper bound and the lower bound on $c_u(\text{NAI}, \text{MAJ})$ are identical. Thus, $c_u(\text{NAI}, \text{MAJ}) = \frac{N}{2}$, and $\text{WR}_{\text{NAI,MAJ}} = \frac{N}{2}$ for $N \geq 4$. □

Theorem 32. *For the FIU problem, according to relative worst order analysis,* $WR_{\text{NAI,MAJ}} = \frac{n}{4}$.

Proof. As in the proof of the previous theorem, since NAI and OPT perform the same on their worst orderings of any sequence, NAI and MAJ are comparable. Next we derive a bounding function with respect to $(\text{NAI}, \text{MAJ}, \leq)$. According to Definition 27, we must exhibit a function f such that $\forall I: \ \text{NAI}_W(I) \leq (f(n) + o(f(n))) \cdot \text{MAJ}_W(I)$. To do this, we consider worst order sequences for the two algorithms and work with the ratio $\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)}$, from which we will be able to find f.

Since NAI's aggregate frequency is the same on any ordering of that sequence, we can compare NAI and MAJ on the same sequence, namely MAJ's worst ordering of it; that is also a worst ordering for NAI. Suppose the input sequence I gives the largest ratio of $\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)}$ for sequences of length n. Suppose I has k distinct items, $b_1, b_2, \ldots, b_k$, and let $f_i = f_I(b_i)$ and $n_i = n_I(b_i)$ for all i. Assume that $f_1 \leq f_2 \leq f_3 \leq \ldots \leq f_k$, so b_k is the most frequent item.

If $n_k \leq \lfloor \frac{n}{2} \rfloor$, then

$$\begin{aligned} \frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} &= \frac{\sum_{i=1}^{k} n_i f_i}{2(\sum_{i=1}^{j-1} n_i f_i + p f_j) + \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) f_j} \\ &= \frac{\sum_{i=1}^{k} n_i^2}{2(\sum_{i=1}^{j-1} n_i^2 + p n_j) + \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) n_j} \end{aligned} \tag{20}$$

where $j \leq k$ is the largest index such that $\sum_{i=1}^{j-1} n_i + p = \lfloor \frac{n}{2} \rfloor$ for some non-negative integer p. We create another sequence I' from I by replacing all the b_i's where $j < i < k$ with b_k, and by replacing $n_j - p - \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right)$ b_j's with b_k. I' will have $j+1$ distinct items and the most frequent item will have $\lfloor \frac{n}{2} \rfloor$ occurrences. Since all these changes will increase the numerator and not change the denominator in Eq. 20, I' will give at least as large a ratio as I, so we consider the sequence I' instead of I. Suppose the distinct items of I', in nondecreasing order of frequency, are $\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_{j+1}$ and the corresponding counts are $\hat{n}_1, \hat{n}_2, \ldots, \hat{n}_{j+1}$ and $\hat{n}_{j+1} \leq \lfloor \frac{n}{2} \rfloor$. Then,

$$\frac{\text{NAI}_W(I')}{\text{MAJ}_W(I')} \leq \frac{\lfloor \frac{n}{2} \rfloor^2 + \sum_{i=1}^{j} \hat{n}_i^2}{2\sum_{i=1}^{j} \hat{n}_i^2 - \left(\lceil \frac{n}{2} \rceil - \lfloor \frac{n}{2} \rfloor\right) \hat{n}_j} \tag{21}$$

Note that since we are first discussing the ratio in terms of a generic input sequence, this is the same formula as the corresponding formula (15) in the previous proof. However, when we now consider concrete instantiations of the sequence with items from the universe, here, as opposed to the previous proof, we are not constrained by the finite universe and thereby forced to use the same items repeatedly. We can choose distinct items when this will help us maximize the ratio. Consider any item $\hat{a}_i$ where $i \leq j$. Suppose its count is $\hat{n}_i > 1$. Replace the $\hat{n}_i$ copies of $\hat{a}_i$ by $\hat{n}_i$ distinct items which are different from all the other items in I'. In most cases, this

replacement will decrease the numerator in Eq. 21 by $\hat{n}_i^2 - \hat{n}_i$ and will decrease the denominator by $2(\hat{n}_i^2 - \hat{n}_i)$. The only exception is when $i = j$ and n is odd, in which case the denominator will decrease by $2\hat{n}_i^2 - 3\hat{n}_i + 1$. However, in either case, the decrease in the denominator is as large as that in the numerator. Since the lower bound on the ratio is 1, this replacement will increase the ratio. Hence, the maximum ratio will be achieved if all the items, except the most frequent item, have frequency $\frac{1}{n}$, so I' has the same form as W_n. Using Proposition 5,

$$\frac{\text{NAI}_W(I')}{\text{MAJ}_W(I')} = \begin{cases} \frac{n}{4} + \frac{1}{2} & \text{for even } n \\ \frac{n}{4} + \frac{3}{4n} & \text{for odd } n \end{cases} \tag{22}$$

It remains to consider the range $\lfloor \frac{n}{2} \rfloor < n_k \leq n$. Since $n_k > \lfloor \frac{n}{2} \rfloor$, the number of distinct items is at most $\lfloor \frac{n}{2} \rfloor + 1$. By Theorem 31, $\frac{\text{NAI}_W(I)}{\text{MAJ}_W(I)} \leq \frac{N}{2} \leq \frac{n}{4} + \frac{1}{2}$ for any I and for large enough n. Thus, by Eq. 22, $\frac{n}{4}$ is a $(\text{NAI}, \text{MAJ}, \leq)$-function.

Since Eq. 22 shows that W_n gives the largest ratio among sequences of length n, we can use the same sequence for the lower bound, showing that $\text{WR}_{\text{NAI,MAJ}} = \frac{n}{4}$. □

5.4. *Majority vs. Eager*

Theorem 33. *For the FIU problem, according to relative worst order analysis,* MAJ *and* EAG *are incomparable.*

Proof. First, we show that MAJ can be much better than EAG. Consider the family of sequences E_n from Definition 4. These sequences are in their worst orderings for both MAJ and EAG. By Proposition 5, $\text{EAG}(E_n) = 2$, so

$$\text{MAJ}_W(E_n) = n - 6 + \frac{16}{n} \geq \left(\frac{n}{2} - 3 + \frac{8}{n}\right) \text{EAG}_W(E_n).$$

Now, we show that EAG can be much better than MAJ. Consider the family of sequences W_n from Definition 4. These sequences are in their worst orderings for MAJ, so by Proposition 5, $\text{MAJ}_W(W_n) = 1$. A worst ordering for EAG is

$$W_n' = a_1, a_2, \ldots, a_{\lceil \frac{n}{2} \rceil}, a_0, a_0, \ldots, a_0,$$

where there are $\lfloor \frac{n}{2} \rfloor$ copies of a_0. $\text{EAG}(W_n') = \text{NAI}(W_n)$, which by Proposition 5 is $\frac{n}{4} + \frac{1}{2}$ when n is even and $\frac{n}{4} + \frac{3}{4n}$ when n is odd. Thus,

$$\text{EAG}_W(W_n) \geq \frac{n}{4} \text{MAJ}_W(W_n).$$

These two families of sequences show that MAJ and EAG are incomparable under relative worst order analysis. □

Although MAJ and EAG are incomparable for the FIU problem under relative worst order analysis, if the number of possible items is bounded by some constant N, the situation changes. Now MAJ is deemed the better algorithm.

Theorem 34. *For the FIF problem, according to relative worst order analysis,* $WR_{\text{MAJ,EAG}} = \infty$.

Proof. The separation from Theorem 33 also holds here, so

$$\text{MAJ}_W(E_n) \geq \left(\frac{n}{2} - 3 + \frac{8}{n}\right) \text{EAG}_W(E_n).$$

To show that MAJ and EAG are comparable, we show that for any input sequence I, $\text{MAJ}_W(I) \geq \text{EAG}_W(I) - N$.

For some input sequence I, let $k \leq N$ denote the number of items occurring exactly once in I. In a worst ordering of I with respect to EAG, all these k items would come first, followed by two occurrences of an item with lowest frequency $\frac{x}{n}$ in I (other than the items occurring only once). Thus, EAG's aggregate frequency is $k\frac{1}{n} + (n-k)\frac{x}{n}$.

For MAJ, in a worst ordering of I, it would buffer all of the k items occurring only once in I. Clearly, such an item stays in the buffer exactly twice, since the next item is different and decrements the counter to zero. For the remaining time, the buffer must contain an item of frequency at least $\frac{x}{n}$. Thus, MAJ's aggregate frequency is at least $2k\frac{1}{n} + (n-2k)\frac{x}{n}$.

Using these two bounds for the respective algorithms' aggregate frequencies on a worst ordering of I, $\text{MAJ}_W(I) \geq \text{EAG}_W(I) + k\frac{1}{n} - k\frac{x}{n}$. Since $k\frac{x}{n} \leq N\frac{n}{n} = N$, $\text{MAJ}_W(I) \geq \text{EAG}_W(I) - N$. □

6. Conclusion and Future Work

We have considered the frequent items problem for streaming as an online problem. Two versions of the problem have been studied, namely the FIF problem, where we assume that the universe is finite, and the FIU problem, where the universe size is unbounded. Three deterministic algorithms, NAI, MAJ, and EAG have been compared using three different quality measures: competitive analysis, relative interval analysis, and relative worst order analysis. Since, for the FIU problem, performance ratios of algorithms depend on the length of the input sequence, we have used the competitive function and the relative worst order function in comparing the algorithms. On the other hand, for the FIF problem, performance ratios of algorithms are mostly independent of n and more often dominated by a function of the universe size, so we have used competitive ratio and relative worst order ratio. Tables 1 and 2

summarize the comparison of the three algorithms for the FIF and FIU problems, respectively.

Measure	NAI vs. MAJ	NAI vs. EAG	MAJ vs. EAG
Competitive ratio	$\frac{\sqrt{N}+1}{2}$, $N-1$ (NAI is better)	$\frac{\sqrt{N}+1}{2}$, ∞ (NAI is better)	$N-1$, ∞ (MAJ is better)
Relative interval	$[-\frac{1}{4}+\frac{1}{4N}, \frac{1}{4}-\frac{1}{4N}]$ (equivalent)	$[-\frac{1}{4}+\frac{1}{4N}, 1]$ (NAI is better)	$[-\frac{1}{2}+\frac{1}{2N}, 1]$ (MAJ is better)
Relative worst order ratio	$\frac{N}{2}$ (NAI is better)	∞ (NAI is better)	∞ (MAJ is better)

Table 1. The comparison of the three algorithms for the FIF problem as a function of the universe size.

Measure	NAI vs. MAJ	NAI vs. EAG	MAJ vs. EAG
Competitive function	$\frac{\sqrt{n}}{2}$ and $\frac{n}{2}$ (NAI is better)	$\frac{\sqrt{n}}{2}$ and $\frac{n}{2}$ (NAI is better)	$\frac{n}{2}$ and $\frac{n}{2}$ (equivalent)
Relative interval	$[-\frac{1}{4}, \frac{1}{4}]$ (equivalent)	$[-\frac{1}{4}, 1]$ (NAI is better)	$[-\frac{1}{2}, 1]$ (MAJ is better)
Relative worst order function	$\frac{n}{4}$ (NAI is better)	$\frac{n}{2}$ (NAI is better)	incomparable

Table 2. The comparison of the three algorithms for the FIU problem as a function of the length of the input sequence.

Comparing across the two tables, results are similar. With regards to relative interval analysis, the results for FIU follow from results for FIF for N going towards infinity. For the other entries, there are small variations that do not alter the conclusion as to which algorithms is better, with two exceptions, both involving MAJ and EAG. When comparing MAJ and EAG using competitive analysis, they are equivalent in the context of the FIU problem, but MAJ is better when considering the FIF problem. Similarly, when comparing MAJ and EAG using relative worst order analysis, they are incomparable with regards to the FIU problem, but MAJ is better for the FIF problem. Technically, this is because EAG's advantage over MAJ in the worst case sequences for MAJ can be controlled within an additive constant depending on N in the case of the FIF scenario. Intuitively, the FIF problem must be the more reasonable model in many cases. If one is interested in keeping frequent items in the buffer, then this must be for scenarios where many items appear with

some non-negligible frequency, so there is a limit to how many items there could be. Thus, it seems positive that MAJ is deemed the best algorithm in this scenario in comparison with the completely non-adaptable EAG.

All three analysis techniques studied here are worst case measures. According to both competitive analysis and relative worst order analysis, NAI is the best possible online algorithm, and according to relative interval analysis, it is as good as MAJ and better than EAG. This is a consequence of NAI being very adaptive and, as a result, good at avoiding the extreme poor performance cases. Both MAJ and EAG attempt to keep the most frequent items in the buffer for longer than their observed frequencies would warrant. The heuristic approaches hurt these algorithms in the worst case.

Relative interval analysis compares the algorithms on the same sequence in a manner which, in addition to the worst case scenarios, also takes the algorithms' best performance into account to some extent. This makes MAJ's sometimes superior performance visible, whereas EAG, not being adaptive at all, does not benefit in the same way from its best performance. In some sense, MAJ's behavior can be seen as oscillating around the behavior of NAI, with worse behavior on some sequences counter-acted by correspondingly better behavior on other sequences.

It seems that the problem would benefit from supplementary analyses based on performance measures that are not worst case measures. To that end, natural performance measures to consider would be bijective and average analysis [1] or other techniques based on analyses of expected results. However, due to the notion of the aggregate frequency calculated over the whole sequence, it is somewhat difficult to apply any average case measures, though one could explore empirical options. In this respect, an extension of this work would be to consider other frequency models. Becchetti et al. [2] studied the aggregate maximum of online streaming, where the maximum at each time step is calculated from a window of fixed size; also called the sliding window streaming model. Another option is to consider the ephemeral frequency [11], where the frequency of an item a at time t is calculated over the first t items of the input sequence, i.e., $f_I^t(a) = \frac{|\{i \leq t | b_i = a\}|}{t}$. In both the sliding window streaming model and the ephemeral frequency model, the frequency of an item does not depend on the future part of the input. Another natural extension of this work is to consider multiple buffers, which also allows for a richer collection of algorithms [3], or more complicated, not necessarily discrete, objective functions [8].

Acknowledgments

We would like to thank the anonymous reviewers for improving the presentation of our results.

References

[1] S. Angelopoulos, R. Dorrigiv, and A. López-Ortiz. On the separation and equivalence of paging strategies. In *18th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 229–237, 2007.

[2] L. Becchetti and E. Koutsoupias. Competitive analysis of aggregate max in windowed streaming. In *36th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 5555 of *Lecture Notes in Computer Science*, pages 156–170. Springer, Heidelberg, 2009.

[3] R. Berinde, G. Cormode, P. Indyk, and M. J. Strauss. Space-optimal heavy hitters with strong error bounds. In *28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 157–166, 2009.

[4] J. Boyar and L. M. Favrholdt. The relative worst order ratio for online algorithms. *ACM Transactions on Algorithms*, 3, 2007.

[5] J. Boyar, L. M. Favrholdt, and K. S. Larsen. The relative worst order ratio applied to paging. *Journal of Computer and System Sciences*, 73(5):818–843, 2007.

[6] J. Boyar, S. Irani, and K. S. Larsen. A comparison of performance measures for online algorithms. In *11th International Symposium on Algorithms and Data Structures (WADS)*, volume 5664 of *Lecture Notes in Computer Science*, pages 119–130. Springer, Heidelberg, 2009. To appear in *Algorithmica*.

[7] Joan Boyar, Kim S. Larsen, and Abyayananda Maiti. A Comparison of Performance Measures via Online Search. *Theoretical Computer Science*, 532:2–13, 2014.

[8] E. Cohen and M. J. Strauss. Maintaining time-decaying stream aggregates. *Journal of Algorithms*, 59(1):19–36, 2006.

[9] G. Cormode and M. Hadjieleftheriou. Finding frequent items in data streams. *Proceedings of the VLDB Endowment*, 1(2):1530–1541, 2008.

[10] R. Dorrigiv, A. López-Ortiz, and J.I. Munro. On the relative dominance of paging algorithms. *Theoretical Computer Science*, 410(38–40):3694–3701, 2009.

[11] Y. Giannakopoulos and E. Koutsoupias. Competitive analysis of maintaining frequent items of a stream. In *13th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, Lecture Notes in Computer Science, pages 340–351. Springer, Heidelberg, 2012.

[12] A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. *Algorithmica*, 3:79–119, 1988.

[13] D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.