A Comparison of Performance Measures via Online Search[☆]

Joan Boyar[a], Kim S. Larsen[a,*], Abyayananda Maiti[a]

[a]*Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, DK-5230 Odense M, Denmark*

Abstract

Since the introduction of competitive analysis, a number of alternative measures for the quality of online algorithms have been proposed, but, with a few exceptions, these have generally been applied only to the online problem for which they were developed. Recently, a systematic study of performance measures for online algorithms was initiated [Boyar, Irani, Larsen: Eleventh International Algorithms and Data Structures Symposium 2009], first focusing on a simple server problem. We continue this work by studying a fundamentally different online problem, online search, and the Reservation Price Policies in particular. The purpose of this line of work is to learn more about the applicability of various performance measures in different situations and the properties that the different measures emphasize. We investigate the following analysis techniques: Competitive, Relative Worst Order, Bijective, Average, Relative Interval, Random Order, and Max/Max. In addition, we have established the first optimality proof for Relative Interval Analysis.

Keywords: Online algorithms, Online search, Performance measures

[☆]A preliminary version of this paper appeared in the *SIAM Joint FAW-AAIM Conference*, Lecture Notes in Computer Science, vol. 7285, Springer, 2012, pp. 303–314. This work was supported in part by the Danish Council for Independent Research.

[*]Corresponding author

Email addresses: `joan@imada.sdu.dk` (Joan Boyar), `kslarsen@imada.sdu.dk` (Kim S. Larsen), `abyaym@imada.sdu.dk` (Abyayananda Maiti)

1. Introduction

An optimization problem is *online* if input is revealed to an algorithm one piece at a time and the algorithm has to commit to the part of the solution involving the current piece before seeing the rest of the input [4]. The first and most well-known analysis technique for determining the quality of online algorithms is *competitive analysis* [18]. The competitive ratio expresses the asymptotic ratio of the performance of an online algorithm compared to an optimal offline algorithm with unlimited computational power. Though this works well in many contexts, researchers realized from the beginning [18] that this "unfair" comparison would sometimes make it impossible to distinguish between online algorithms of quite different quality in practice.

In recent years, researchers have considered alternative methods for comparisons of online algorithms, some of which compare algorithms directly, as opposed to computing independent ratios in a comparison to an offline algorithm. See references below and [11] for a fairly recent survey. Most of the new methods have been designed with one particular online problem in mind, trying to fix problems with competitive analysis for that particular problem. Not that much is known about the strengths and weaknesses of these alternatives in comparison with each other. In [7], a systematic study of performance measures was initiated by fixing a simple online server problem and applying a collection of performance measures. Partial conclusions were obtained in demonstrating which measures focus on greediness as an algorithmic quality. It was also observed that some measures could not distinguish between certain pairs of algorithms where the one performed at least as well as the other on every sequence.

We continue this systematic study here by investigating a fundamentally different problem which has not yet been studied as an online problem other than with competitive analysis, the *online search problem* [13, 14]. Online search is a very simple online (profit) maximization problem; the online algorithm tries to sell a specific item for the highest possible price. Prices, between the minimum price of m and the maximum price of M, arrive online one at a time, and each time a price is revealed, the algorithm can decide to accept that price and terminate or decide to wait. The length of the input sequence is not known to the algorithm in advance, but is revealed only when the last price is given, and the algorithm must accept that price, if it has not accepted one earlier.

This simple model of a searching problem has enormous importance due to its simplicity and its application in the much more complex problems of lowest or highest price searching in various real-world applications in the fields of Economics and Finance [17]. The online search problem is very similar to that of the one-way trading problem [13, 14, 9]. In fact, one-way trading can be seen as randomized searching. Note that the assumption of a known minimum and maximum price is often used for these types of problems because of the difficulties of defining and analyzing algorithms without them. Reasonable bounds can often be chosen by observing high and low values (of stock prices, currency exchange rates, or whatever is being traded) over an appropriate period of time.

The long-term goal of systematically comparing performance measures is to be able to determine, based on characteristics of an online problem, how online algorithms should be analyzed theoretically so as to accurately predict the relative quality of the algorithms in practice. Online search differs from the server problem studied earlier in many respects, particularly in its consisting of a "one-shot" choice, as opposed to incremental decisions, so the greediness studied in [7] is not relevant here. In addition, online search is a maximization problem, instead of minimization, and its last request has a different requirement than the others (if nothing was chosen before then, the last value must be chosen). Thus, the findings obtained here are complementary to the results obtained in [7]. The difference between online search and many other problems also forced us to extend earlier definitions for some of the measures so that they could be applicable here as well. In this paper, we discuss seven measures. There are also other important measures that we have not included here as they are less relevant to the online search problem. Resource Augmentation [15] and the Accommodating Function [8], for example, are two well studied modifications of Competitive Analysis, both of which depend on some resource used in the online problem being considered. However, the online search problem does not include any appropriate resource, so these two types of analysis are irrelevant here.

Our primary study is of the class of Reservation Price Policy (RPP) algorithms [13, 14]. This is a parameterized class, where the behavior of $\mathcal{R}_p$ is to accept the first price greater than or equal to the so-called reservation price p.

As a "sanity check" to confirm that the measures "work" at all on this problem, we also define $\mathcal{R}_p^2$, which accepts the *second* price greater than or equal to p, and investigate its relationship to $\mathcal{R}_p$. Whereas $\mathcal{R}_p$ "decides what it wants and takes it when it sees it", $\mathcal{R}_p^2$ "knows what it wants, but does not take it until the second time it sees it". One would expect $\mathcal{R}_p$ to be the better algorithm. With the exception of Max/Max Analysis [3], all the measures "pass this test" and favor $\mathcal{R}_p$, though some redefinition was necessary for Relative Interval Analysis.

Since the measures pass this test, we also consider the more interesting task of comparing the different quality measures on RPP algorithms with different parameters. We have considered having an integral interval of possible prices between m and M as well as a real-valued scenario; for the most part, the results are similar. The following discussion in this introduction is assuming a real-valued scenario, allowing us to state the results better typographically, without rounding.

We find that Competitive Analysis and Random Order Analysis favor $\mathcal{R}_{\sqrt{mM}}$, the reason being that they focus on limiting the worst case ratio compared to an optimal algorithm, independent of input length. Relative Interval Analysis favors $\mathcal{R}_{\frac{m+M}{2}}$, similarly limiting the worst case difference, as opposed to ratio. Average Analysis favors $\mathcal{R}_M$. This is basically due to focusing on the limit, i.e., when input sequences become long enough, any event will occur eventually. In Bijective Analysis, basically all algorithms are incomparable. Finally Relative Worst Order Analysis deems the algorithms incomparable, but gives indication that $\mathcal{R}_{\sqrt{mM}}$ is the best algorithm.

In addition to these findings, this paper contains the first optimality result for Relative Interval Analysis, where we prove that no $\mathcal{R}_p$ algorithm can be better than $\mathcal{R}_{\frac{m+M}{2}}$. For Relative Worst Order Analysis, we refine the discussion of which algorithm is best through the concept of "superiority", which seems to be interesting for classes of parameterized algorithms. A first use of this concept, without naming it, appeared when analyzing a parameterized variant of Lazy Double Coverage for the server problem in [7].

Finally, we have investigated the sensitivity of the different measures with regards to the choice of integral vs. real-valued domains, and most of the measures seem very stable in this regard. Not surprisingly, using real values, Bijective Analysis indicates that all RPP algorithms are equivalent. Average analysis is inapplicable for a real-valued interval, but a generalization, which we call Expected Analysis, can be applied, giving similar results to what Average Analysis gives for integral values. Expected Analysis may be useful for other problems as well.

Since our problem is a profit maximization problem, for those analysis methods which have previously only been defined for cost minimization problems, we have presented profit maximization versions. In online search since profit is a constant (between m and M), independent of the sequence length, for measures of an asymptotic nature, we modify the definitions accordingly. In Competitive Analysis and Relative Worst Order Analysis we use the strict version since asymptotic results (allowing an additive constant) would deem all algorithms optimal (a ratio of one compared with an optimal algorithm—up to the additive constant). For the same reason, the measures which were originally defined using limits on the profit (or cost) achieved are modified. Both Relative Interval Analysis and the Max/Max ratio have previously been defined with limits and need alternative definitions which are more appropriate for this type of scenario. In each section of the paper, we give the precise definition of the measures used.

The rest of this paper is organized as follows. Section 2 defines the notation used and each subsequent section treats one of the measures described above.

2. Problem Preliminaries

Unless otherwise stated, we assume that the prices are integral and drawn from some integral interval $[m, M]$ with $0 < m \leq M$. In any time step, any value from this closed interval can be drawn as a price, and there will be $N = M - m + 1$ possible prices. This assumption is made for the sake of consistency; some methods of analysis are uninteresting for real-valued intervals; see Section 4, for example. Also, this assumption is compatible with the real-world problems of online search as the set of prices is generally finite (the market decides on an agreed-upon number of digits after the decimal point).

We denote the length of the price sequence by n. Denote by $\mathcal{I}_n$ the set of all input sequences of length n. Thus, the total number of possible input sequences of length n is N^n. For an online algorithm $\mathcal{A}$ and an input sequence I, let $\mathcal{A}(I)$ be the profit gained by $\mathcal{A}$ on I, i.e., the price chosen. In some analyses (for example in Relative Worst Order Analysis), we need to permute the input sequences. We always use σ as a permutation and denote the permuted sequence by $\sigma(I)$.

Some of the analysis methods compare the online algorithms with a hypothetical optimal offline algorithm which receives the input in its entirety in advance and has unlimited computational power in determining a solution. We denote this optimal algorithm by OPT and the profit gained by it from an input sequence I as $OPT(I)$, which is the maximum price in that sequence.

To denote the relative performance of two online algorithms $\mathcal{A}$ and $\mathcal{B}$ according to an analysis method, x, we use the following notation. If $\mathcal{B}$ is better than $\mathcal{A}$, then we write $\mathcal{A} \prec_x \mathcal{B}$, and if $\mathcal{B}$ is no worse than $\mathcal{A}$, this is denoted by $\mathcal{A} \preceq_x \mathcal{B}$. If the measure deems the algorithms equivalent, then this is denoted by $\mathcal{A} \equiv_x \mathcal{B}$. Usually, we merely define either $\prec_x$ or $\preceq_x$ and the other relations follow in the standard way from that.

If $n = 1$, any algorithm must take the only price in the sequence, so all online search algorithms are equivalent. To streamline the presentation of results, we always assume that $n \geq 2$.

The core of this paper is concerned with the comparison of $\mathcal{R}_p$ and $\mathcal{R}_q$ for $p \neq q$. To avoid stating this every time, we always assume that $m \leq p < q \leq M$.

3. Competitive Analysis

The online search problem was first studied from an online algorithms perspective using Competitive Analysis by El-Yaniv et al. [13]. Competitive Analysis evaluates an online algorithm in comparison to an optimal offline algorithm.

Definition 1. An online search algorithm $\mathcal{A}$ is *strictly c-competitive* if for all finite input sequences I, $OPT(I) \leq c \cdot \mathcal{A}(I)$. The *competitive ratio* of algorithm $\mathcal{A}$ is $\inf\{c \mid \mathcal{A} \text{ is } c\text{-competitive}\}$.

Denote the competitive ratio of an online algorithm $\mathcal{A}$ by $c_{\mathcal{A}}$. If $c_{\mathcal{A}} > c_{\mathcal{B}}$, $\mathcal{B}$ is better than $\mathcal{A}$ according to Competitive Analysis and we denote this by $\mathcal{A} \prec_c \mathcal{B}$.

In [13], El-Yaniv formulated the Reservation Price Policy algorithm and proved that for real-valued prices, the *reservation price* $p^* = \sqrt{Mm}$ is the optimal price according to Competitive Analysis, and using this price, the competitive ratio is $\sqrt{M/m}$. A very similar result and proof holds for integer-valued prices.

Theorem 1. *According to Competitive Analysis, $\mathcal{R}_p \prec_c \mathcal{R}_q$, $\mathcal{R}_p \equiv_c \mathcal{R}_q$, and $\mathcal{R}_q \prec_c \mathcal{R}_p$ if and only if $Mm > p(q-1)$, $Mm = p(q-1)$, and $Mm < p(q-1)$, respectively.*

Proof. In any price sequence for an RPP algorithm $\mathcal{R}_p$, we consider two cases: (i) all the prices are less than p, in which case the performance ratio, offline to online, will be at most $\frac{p-1}{m}$, with equality when there is a price $p-1$ and the last price is m; and (ii) at least one price is greater than or equal to p, in which case the offline to online performance ratio would be at most $\frac{M}{p}$, with equality when the first price greater than or equal to p is exactly p and there is another price M somewhere later. So, the competitive ratio of $\mathcal{R}_p$ will be $c_{\mathcal{R}_p} = max(\frac{p-1}{m}, \frac{M}{p})$. It is easy to verify that $c_{\mathcal{R}_p} > c_{\mathcal{R}_q}$ if and only if $\frac{M}{p} > \frac{q-1}{m}$, since $\frac{p-1}{m} < \frac{q-1}{m}$ and $\frac{M}{p} > \frac{M}{q}$. This argument proves that $\mathcal{R}_p \prec_c \mathcal{R}_q$ if and only if $Mm > p(q-1)$. The remaining two results in the theorem follow similarly. $\square$

Corollary 1. *Let $s = \left\lceil \sqrt{Mm} \right\rceil$. According to Competitive Analysis, the best RPP algorithm is $\mathcal{R}_s$.*

Proof. Assume that $p < s$. The comparison between $\mathcal{R}_p$ and $\mathcal{R}_s$ gives $p\left(\left\lceil \sqrt{Mm} \right\rceil - 1\right) < Mm$. Thus, $\forall p < s$, $\mathcal{R}_p \prec_c \mathcal{R}_s$.

Now, assume that $q > s$. Then the comparison between $\mathcal{R}_q$ and $\mathcal{R}_s$ gives $\left\lceil \sqrt{Mm} \right\rceil (q-1) \geq Mm$. Thus, $\forall q > s$, $\mathcal{R}_q \prec_c \mathcal{R}_s$. $\square$

Theorem 2. *According to Competitive Analysis, $\mathcal{R}_p^2 \prec_c \mathcal{R}_p$ and $\mathcal{R}_p^2 \equiv_c \mathcal{R}_p$ if and only if $p > m$ and $p = m$, respectively.*

Proof. From the proof of Theorem 1, we know that the competitive ratio of $\mathcal{R}_p$ is $c_{\mathcal{R}_p} = \max\left(\frac{p-1}{m}, \frac{M}{p}\right)$. For the competitive ratio of $\mathcal{R}_p^2$, we consider a price sequence with only one M followed by $n-1$ occurrences of m. Clearly the competitive ratio is M/m, and it is the maximum ratio that can be obtained by any algorithm. So, $c_{\mathcal{R}_p^2} \geq c_{\mathcal{R}_p}$, and equality holds if and only if $p = m$. $\square$

4. Bijective Analysis

In the Bijective Analysis model [2], we construct a bijection on the set of possible input sequences. In this bijection, we aim to pair input sequences for online algorithms $\mathcal{A}$ and $\mathcal{B}$ in such a way that the cost of $\mathcal{A}$ on every sequence I is no more than the cost of $\mathcal{B}$ on the image of I, or vice versa, to show that the algorithms are comparable. We present a version of the definition from [2] which is suitable for profit maximization problems such as online search.

Definition 2. We say that an online search algorithm $\mathcal{A}$ is no better than an online search algorithm $\mathcal{B}$ according to *Bijective Analysis* if there exists an integer $n_0 \geq 1$ such that for each $n \geq n_0$, there is a bijection $b\colon \mathcal{I}_n \to \mathcal{I}_n$ satisfying $\mathcal{A}(I) \leq \mathcal{B}(b(I))$ for each $I \in \mathcal{I}_n$. We denote this by $\mathcal{A} \preceq_b \mathcal{B}$.

Theorem 3. *According to Bijective Analysis, $\mathcal{R}_p \prec_b \mathcal{R}_q$ if $p = m$ and $m < q \leq M$. Otherwise, $\mathcal{R}_p$ and $\mathcal{R}_q$ are incomparable.*

Proof. First assume that $m < p$. Considering $\mathcal{R}_p$, a price k in the range from m to $p-1$ will be chosen as output if and only if it is the last price of the sequence and all the preceding prices are smaller than p. As there are $p-m$ such prices smaller than p and, not counting the last price, there are $n-1$ prices in the sequence, the number of possible sequence with k as output is $(p-m)^{n-1}$.

For any prices in the range from p to M, algorithm $\mathcal{R}_p$ chooses this price as output at its first occurrence in the price sequence if no price greater than or equal to p has occurred before it. So all the preceding prices before this first occurrence must be smaller than p (in the range from m to $p-1$) and the following prices can have any value. The number of sequences which give output k if the reservation price is p is

$$N_{p,k} = \begin{cases} (p-m)^{n-1}, & \text{for } m \leq k < p \\ \sum_{i=1}^{n} (p-m)^{i-1} N^{n-i}, & \text{for } p \leq k \leq M \end{cases} \tag{1}$$

Recall the assumption throughout the paper that $q > p$. We consider two cases depending on p:

Case $\mathbf{p > m}$: From Eq. (1), we can derive the fact that when $p > m$, the number of sequences with output m for algorithm $\mathcal{R}_q$ is greater than that for algorithm $\mathcal{R}_p$, since $N_{q,m} > N_{p,m}$. Thus, we cannot have any bijective mapping $b\colon \mathcal{I}_n \to \mathcal{I}_n$ that shows $\mathcal{R}_p(I) \leq \mathcal{R}_q(b(I))$ for every $I \in \mathcal{I}_n$. On the other hand, it is also the case that the number of sequences with output M for algorithm $\mathcal{R}_q$ is greater than that of algorithm $\mathcal{R}_p$, since $N_{q,M} > N_{p,M}$. So there is no bijection b such that $\mathcal{R}_p(I) \geq \mathcal{R}_q(b(I))$ for every $I \in \mathcal{I}_n$. Thus for this case, $\mathcal{R}_p$ and $\mathcal{R}_q$ are incomparable according to the Bijective Analysis.

Case $\mathbf{p = m}$: For algorithm $\mathcal{R}_m$, since the first price will be accepted, each price will be the output for exactly N^{n-1} sequences. We can derive the number of sequences with specific output for algorithm $\mathcal{R}_q$ using Eq. (1). In this case, each price in the range from m to $q-1$ will emerge as output in $(q-m)^{n-1}$ sequences and the number of sequences with output in the range from q to M will be $N^{n-1}+(q-m)N^{n-2}+(q-m)^2N^{n-3}+\cdots+(q-m)^{n-1}$. Clearly, here we can construct a bijective mapping $b\colon \mathcal{I}_n \to \mathcal{I}_n$ where each sequence with output $k<q$ of algorithm $\mathcal{R}_m$ is mapped to sequences with the same output for algorithm $\mathcal{R}_q$. Let E_m denote the number of excess sequences with output $k<q$ of $\mathcal{R}_m$ which cannot be mapped in the above manner. We map each sequence with output $k\geq q$ of algorithm $\mathcal{R}_m$ to sequences with the same output in algorithm $\mathcal{R}_q$. Let E_q denote the number of excess sequences with output $k\geq q$ of $\mathcal{R}_q$ which can not be mapped in above manner. Clearly, $E_m=E_q$. Note that, for all of these E_m sequences, we can construct a mapping such that $\mathcal{R}_m(I)<\mathcal{R}_q(b(I))$. This mapping shows that $\mathcal{R}_m(I)\leq\mathcal{R}_q(b(I))$ for each $I\in\mathcal{I}_n$, but there is no bijection b' such that $\mathcal{R}_m(I)\geq\mathcal{R}_q(b'(I))$ for all $I\in\mathcal{I}_n$. Thus, if $p=m$, $\mathcal{R}_m$ and $\mathcal{R}_q$ are comparable according to Bijective Analysis and $\mathcal{R}_m \prec_b \mathcal{R}_q$. □

Theorem 4. *According to Bijective Analysis, $\mathcal{R}_p^2 \prec_b \mathcal{R}_p$ if $p>m$, and $\mathcal{R}_p^2 \equiv_b \mathcal{R}_p$ when $p=m$.*

Proof. Let $\hat{N}_{p,k}$ denote the number of sequences giving output k for algorithm $\mathcal{R}_p^2$. As in the case of $\mathcal{R}_p$, $\mathcal{R}_p^2$ will choose m as the output if it is the last price of the sequence and all the preceding prices are smaller than p. From the proof of Theorem 3, we know that there are exactly $(p-m)^{n-1}$ such sequences. In addition to those sequences, m will also be the output if the preceding $n-1$ prices have exactly one price greater than or equal to p. The above reasoning is valid for each price in the range from m to $p-1$. So,

$$\hat{N}_{p,k}=(p-m)^{n-1}+(p-m)^{n-2}(n-1)(M-p+1),\ \text{for } m\leq k<p \qquad (2)$$

For any price in the range from p to M, algorithm $\mathcal{R}_p^2$ chooses this price as output if in the price sequence there is exactly one price greater than or equal to p which occurs before it, or it is the last price and no other price p or larger occurred earlier. So, for $m<p\leq k\leq M$, the number of sequences with any output k in the range from p to M is

$$\hat{N}_{p,k}=\sum_{i=2}^{n}(p-m)^{i-2}(i-1)(M-p+1)N^{n-i}+(p-m)^{n-1} \qquad (3)$$

We first consider the case $p > m$. From Eq. (1) and (2), it is evident that if $n > 1$, then $\hat{N}_{p,k} > N_{p,k}$ for $m \leq k < p$. As the total number of sequences is fixed and neither Eq. (1) nor Eq. (3) depend on k, $\hat{N}_{p,k} < N_{p,k}$ for $p \leq k < M$. So, we can always build a bijective mapping $b: \mathcal{I}_n \rightarrow \mathcal{I}_n$ that has $\mathcal{R}_p^2(I) \leq \mathcal{R}_p(b(I))$ for every $I \in \mathcal{I}_n$, but the reverse mapping cannot be constructed. However, if $p = m$, then $\hat{N}_{p,k} = N_{p,k} = N^{n-1}$ for any k, since $\mathcal{R}_p$ always takes the first price and $\mathcal{R}_p^2$ the second. So, according to Bijective Analysis, $\mathcal{R}_p^2 \prec_b \mathcal{R}_p$ if $p > m$ and $\mathcal{R}_p^2 \equiv_b \mathcal{R}_p$ when $p = m$. □

The result of comparing the two algorithms using Bijective Analysis changes significantly when the values of the prices are real numbers: Bijective Analysis cannot differentiate between algorithms when the number of sequences is uncountable.

Theorem 5. *$\mathcal{R}_p$ and $\mathcal{R}_q$ are equivalent according to Bijective Analysis if the prices are drawn from real space in $[m, M]$.*

Proof. As any closed or open interval in real space has the cardinality of the continuum, the cardinality of $[m, p]$ and $[m, q]$ will be same. Taking the Cartesian product of such sets preserves their cardinality. So the cardinality of the set of sequences with any output, k, will be same for both algorithms. Hence, we can find a bijective mapping between the sequences where each sequence is mapped to another with same output. This shows that in this situation, all reservation price algorithms are equivalent according to Bijective Analysis. □

The same problem clearly arises for $\mathcal{R}_p$ versus $\mathcal{R}_p^2$ and for other online problems with real-valued inputs.

5. Average Analysis

In general, using Bijective Analysis, algorithms could be incomparable because it is impossible to find a bijection showing that one algorithm dominates the other. In some of these cases, if we take the average performance of the algorithms, then we can still get an indication of which algorithm is better. In [2], Average Analysis is defined with that aim and is formulated here in terms of online search.

Definition 3. We say that an online search algorithm $\mathcal{A}$ is no better than an online search algorithm $\mathcal{B}$ according to *Average Analysis* if there exists an integer $n_0 \geq 1$ such that for each $n \geq n_0$, $\sum_{I \in \mathcal{I}_n} \mathcal{A}(I) \leq \sum_{I \in \mathcal{I}_n} \mathcal{B}(I)$. We denote this by $\mathcal{A} \preceq_a \mathcal{B}$.

Theorem 6. *For all* $n \geq \left\lfloor \frac{\log(N/(q-p))}{\log(N/(N-1))} \right\rfloor + 1$, $\sum_{I \in \mathcal{I}_n} \mathcal{R}_p(I) < \sum_{I \in \mathcal{I}_n} \mathcal{R}_q(I)$. *Thus, according to Average Analysis,* $\mathcal{R}_p \prec_a \mathcal{R}_q$.

Proof. Let $S_{p,n}$ denote the summation $\sum_{I \in \mathcal{I}_n} \mathcal{R}_p(I)$. We can derive the value of $S_{p,n}$ using Eq. (1) and that $N = M - m + 1$. $S_{p,n}$ equals

$$\begin{aligned}
& \sum_{i=m}^{p-1} iN_{p,i} + \sum_{i=p}^{M} iN_{p,i} \\
= \; & (p-m)^{n-1} \sum_{i=m}^{p-1} i + \left(\sum_{i=1}^{n} (p-m)^{i-1} N^{n-i} \right) \sum_{i=p}^{M} i \\
= \; & \frac{(p-m)^n (p+m-1)}{2} + \left(\sum_{i=1}^{n} (p-m)^{i-1} N^{n-i} \right) \frac{(M+p)(M-p+1)}{2} \qquad (4) \\
= \; & \frac{(p-m)^n (p+m-1)}{2} + \left(\frac{N^n - (p-m)^n}{N-(p-m)} \right) \frac{(N+m+p-1)(N+m-p)}{2} \\
= \; & \frac{N^{n+1} + pN^n + mN^n - N^n - N(p-m)^n}{2} \qquad (5)
\end{aligned}$$

To compare $\mathcal{R}_p$ and $\mathcal{R}_q$, we show that the difference between the two corresponding sums $(S_{q,n} - S_{p,n})$ is greater than zero for some $n_0 \geq 1$ and for each $n \geq n_0$. Using Derivation (5), we have

$$S_{q,n} - S_{p,n} > 0 \;\; \Leftrightarrow \;\; N^{n-1} > \frac{(q-m)^n - (p-m)^n}{q-p} \qquad (6)$$

Since $q - m \leq M - 1$ and $q > p$, Ineq. (6) holds for any n_0. Thus, it holds for any n_0 satisfying $N^{n_0-1} > \frac{(N-1)^{n_0}}{q-p}$. Solving for n_0 gives

$$(n_0 - 1)\log N > n_0 \log(N-1) - \log(q-p) \;\; \Leftrightarrow \;\; n_0 > \frac{\log(N/(q-p))}{\log(N/(N-1))} \qquad (7)$$

Therefore, for all $n_0 \geq \left\lfloor \frac{\log(N/(q-p))}{\log(N/(N-1))} \right\rfloor + 1$, $\sum_{I \in \mathcal{I}_n} \mathcal{R}_p(I) < \sum_{I \in \mathcal{I}_n} \mathcal{R}_q(I)$. $\square$

Corollary 2. *According to Average Analysis, the best RPP algorithm is* $\mathcal{R}_M$.

PROOF. According to Theorem 6, for any two RPP algorithm $\mathcal{R}_p$ and $\mathcal{R}_q$ with $q > p$, $\mathcal{R}_p \prec_a \mathcal{R}_q$. Thus, the maximal possible reservation price is best. □

Theorem 7. *According to Average Analysis,* $\mathcal{R}_p^2 \prec_a \mathcal{R}_p$ *if* $p > m$, *and* $\mathcal{R}_p^2 \equiv_a \mathcal{R}_p$ *when* $p = m$.

PROOF. Follows immediately from the Theorem 4. □

In Average Analysis, algorithms are compared by comparing the sums of their outputs on all possible sequences. For integral valued problems, this is equivalent to comparing the sum of the outputs and the expected outputs over a uniform distribution on all input sequences. In contrast, in the case of real-valued problems, calculating the sum of the outputs of the infinitely many sequences is impossible. However, if we know the distribution of the input prices in the sequences, then we can derive the expected output of a sequence. We generalize Average Analysis to *Expected Analysis*. This generalization may prove useful for other online problems as well.

Definition 4. We say that an online search algorithm $\mathcal{A}$ is no better than an online search algorithm $\mathcal{B}$ according to *Expected Analysis* if there exists an integer $n_0 \geq 1$ such that for each $n \geq n_0$, $E_{I \in \mathcal{I}_n}[\mathcal{A}(I)] \leq E_{I \in \mathcal{I}_n}[\mathcal{B}(I)]$. We denote this by $\mathcal{A} \preceq_e \mathcal{B}$.

We denote the probability of the first price being from the range $[m, p)$ by $P_{m,p}$ and the probability of the first price being from the range $[p, M]$ by $P_{p,M}$. Additionally we denote the expected value of prices smaller than p by $E_{m,p}$ and the expected value of the prices greater than or equal to p by $E_{p,M}$. We assume that the prices in an input sequence are independent and uniformly distributed over the range $[m, M]$. If $n = 1$, then the expected value of the output is $P_{p,M}E_{p,M} + P_{m,p}E_{m,p}$. Assume now that we are dealing with sequences of length two. Hence, with probability $P_{p,M}$, the algorithm $\mathcal{R}_p$ accepts the first price and with probability $P_{m,p}$ it does not. So, for $n = 2$, the expected value of the output will be $P_{p,M}E_{p,M} + P_{m,p}P_{p,M}E_{p,M} + P_{m,p}P_{m,p}E_{m,p}$. Inductively, the expected value of the output for a sequence of length n can be calculated from that of sequences of length $n - 1$.

$$E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] = P_{p,M}E_{p,M}\sum_{i=1}^{n} P_{m,p}^{i-1} + E_{m,p}P_{m,p}^{n}$$

For the real-valued case, $P_{m,p} = \frac{p-m}{M-m}$, $P_{p,M} = \frac{M-p}{M-m}$, $E_{m,p} = \frac{p+m}{2}$, and $E_{p,M} = \frac{M+p}{2}$. For the integral case, $P_{m,p} = \frac{p-m}{M-m+1}$, $P_{p,M} = \frac{M-p+1}{M-m+1}$, $E_{m,p} = \frac{p+m-1}{2}$, and $E_{p,M} = \frac{M+p}{2}$. For the integral case of the algorithm $\mathcal{R}_p$, the above values give

$$E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] = \tfrac{(M-p+1)(M+p)}{2(M-m+1)}\sum_{i=1}^{n}\left(\tfrac{p-m}{M-m+1}\right)^{i-1} + \tfrac{p+m-1}{2}\left(\tfrac{p-m}{M-m+1}\right)^{n}$$

It is easily verifiable from the Eq. (4) that $E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] = S_{p,n}/N^n$. Thus, Definition 3 and 4 produce the same result as stated in Theorem 6 for the integral case.

Neither Definition 3 nor Theorem 6 can be used in the real-valued case. Here, $\mathcal{R}_m$ always chooses the first price of any sequence, whereas $\mathcal{R}_M$ does not choose any of the first $n-1$ prices as $P_{M,M} = 0$ and it has to take the last price. As all the prices are identically distributed, the expected value of the first price and the last price are the same. That makes $\mathcal{R}_m$ and $\mathcal{R}_M$ equivalent.

Proposition 1. *In case of real-valued online search*

$$E_{I\in\mathcal{I}_n}[\mathcal{R}_m(I)] = E_{I\in\mathcal{I}_n}[\mathcal{R}_M(I)] = \frac{m+M}{2}$$

Thus, according to Expected Analysis, $\mathcal{R}_m \equiv_e \mathcal{R}_M$.

For the rest of the cases, we denote the distance between m and M by U, i.e., $U = M - m$.

Theorem 8. *In case of real-valued online search, for all* $n \geq \left\lfloor \frac{\log(U/(q-p))}{\log(U/(q-m))} \right\rfloor + 1$, *if either* $p > m$ *or* $q < M$, $E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] < E_{I\in\mathcal{I}_n}[\mathcal{R}_q(I)]$. *Thus, in this case, according to Expected Analysis,* $\mathcal{R}_p \preceq_e \mathcal{R}_q$.

Proof. For the real-valued case, using

$$E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] = P_{p,M}E_{p,M}\sum_{i=1}^{n} P_{m,p}^{i-1} + E_{m,p}P_{m,p}^{n},$$

the expression $E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)]$ becomes

$$\begin{aligned}
E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] &= \frac{(M-p)(M+p)}{2U}\sum_{i=1}^{n}\left(\frac{p-m}{U}\right)^{i-1} + \frac{p+m}{2}\left(\frac{p-m}{U}\right)^{n} \\
&= \frac{(p-m)^n(p+m)}{2U^n} + \frac{(M+p)(M-p)}{2U^n}\sum_{i=1}^{n}(p-m)^{i-1}U^{n-i} \\
&= \frac{1}{2U^n}\left[(p-m)^n(p+m) + (M+p)(M-p)\frac{U^n-(p-m)^n}{U-(p-m)}\right] \\
&= \frac{1}{2U^n}[MU^n + pU^n - U(p-m)^n] \qquad (8)
\end{aligned}$$

To prove that $\mathcal{R}_p \prec_e \mathcal{R}_q$, it is sufficient to show that the difference between the corresponding two expectations is greater than zero for some $n_0 \geq 1$ and for each $n \geq n_0$. We use Eq. (8).

$$\begin{aligned}
& E_{I\in\mathcal{I}_n}[\mathcal{R}_q(I)] - E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] > 0 \\
& \Updownarrow \\
& \frac{qU^n - pU^n - (q-m)^nU + (p-m)^nU}{2U^n} > 0 \\
& \Updownarrow \\
& U^{n-1} > \frac{(q-m)^n - (p-m)^n}{q-p}
\end{aligned}$$

Since $q < M$, the inequality above becomes similar to Ineq. (6), with the only difference being that here $U = M - m$ in place of $N = M - m + 1$.

To get the value of n_0, we can follow the derivation of Ineq. (7), concluding that in the case of real-valued problems, for all $n \geq \left\lfloor \frac{\log(U/(q-p))}{\log(U/(q-m))} \right\rfloor + 1$, $E_{I\in\mathcal{I}_n}[\mathcal{R}_p(I)] < E_{I\in\mathcal{I}_n}[\mathcal{R}_q(I)]$, if either $p > m$ or $q < M$. From the previous statement and Proposition 1, this proves that according to Expected Analysis, $\mathcal{R}_p \preceq_e \mathcal{R}_q$. $\square$

6. Random Order Analysis

Kenyon [16] proposed another method for comparing the average behaviors of online algorithms by considering the expected result of a random ordering of an input sequence and comparing that to *OPT*'s result on the same sequence. Kenyon defines the random order ratio in the context of the bin packing problem which is a cost minimization problem.

Definition 5. The *random order ratio* of an online bin packing algorithm $\mathcal{A}$ is

$$\limsup_{OPT(I)\to\infty} \frac{E_\sigma[\mathcal{A}(\sigma(I))]}{OPT(I)}$$

where the expectation is taken over all permutations of I.

An online algorithm $\mathcal{B}$ is better than an online algorithm $\mathcal{A}$ according to Random Order Analysis if the random order ratio of $\mathcal{A}$ is larger than the random order ratio of $\mathcal{B}$. We denote this by $\mathcal{A} \prec_r \mathcal{B}$. Since the value of $OPT(I)$ is bounded above by the constant M, the following definition, a maximization version of the definition of random order ratio in [10], is used here in place of the original definition, to specify the quality of $\mathcal{R}_p$:

$$\limsup_{|I|\to\infty} \frac{OPT(I)}{E_\sigma[\mathcal{R}_p(\sigma(I))]} \tag{9}$$

Theorem 9. *The random order ratio of the RPP algorithm $\mathcal{R}_p$ is* $\max(\frac{M}{p}, \frac{p-1}{m})$ *when $p > 1$ and $p > m$. Consequently, $\mathcal{R}_p \prec_r \mathcal{R}_q$ if and only if $Mm > p(q-1)$.*

Proof. Considering the random order ratio of $\mathcal{R}_p$, OPT always chooses the highest price in the sequence as its output and $\mathcal{R}_p$ chooses the first price that is greater than or equal to p. There are two cases where the random order ratio could achieve the maximal value. First, suppose the sequence has one price with value M and all other prices are p. Then

$$E_\sigma[\mathcal{R}_p(\sigma(I))] = \frac{M + p(n-1)}{n} \tag{10}$$

Now substituting the expected value of the output of the algorithm $\mathcal{R}_p$ in Exp. (9), the ratio becomes

$$\limsup_{n\to\infty} \frac{nM}{M + p(n-1)} = \frac{M}{p} \tag{11}$$

The other case is when the sequence has one price with value $p-1$ and all other prices are equal to m. In this case, we can get a limit similar to Exp. (11) of $\frac{p-1}{m}$ when $p > 1$ and $p > m$. As we are seeking the maximum of these ratios, the random order ratio of $\mathcal{R}_p$ is the maximum of the two values, $\frac{M}{p}$ and $\frac{p-1}{m}$, when $p > 1$. This gives the same result as in the case of Competitive Analysis in Section 3. The relative performance of $\mathcal{R}_p$ and $\mathcal{R}_q$ follows from the arguments in the proof of Theorem 1. □

Corollary 3. *Let* $s = \left\lceil \sqrt{Mm} \right\rceil$. *According to Random Order Analysis, the best RPP algorithm is* $\mathcal{R}_s$.

Proof. The values are the same as for Competitive Analysis in Corollary 1. □

The conditions of the following theorem are exactly the same as in the case of comparing $\mathcal{R}_p^2$ and $\mathcal{R}_p$ using Competitive Analysis in Section 3 and the proof resembles the proof of Theorem 2.

Theorem 10. *According to Random Order Analysis,* $\mathcal{R}_p^2 \prec_r \mathcal{R}_p$ *and* $\mathcal{R}_p^2 \equiv_r \mathcal{R}_p$ *if and only if* $p > m$ *and* $p = m$*, respectively.*

Proof. From the proof of Theorem 9, we know that the random order ratio of $\mathcal{R}_p$ is $\max(\frac{p-1}{m}, \frac{M}{p})$. For the random order ratio of $\mathcal{R}_p^2$, we consider a price sequence with only one M and $n-1$ occurrences of m. Clearly, OPT always takes M, whereas $\mathcal{R}_p^2$ never accepts the first occurrence of M unless it is the last price in the sequence. So,

$$E_\sigma[\mathcal{R}_p^2(\sigma(I))] = \frac{M + m(n-1)}{n} \tag{12}$$

Now, substituting the expected value of the output of the algorithm $\mathcal{R}_p^2$ in Exp. (9), we get the ratio

$$\limsup_{n\to\infty} \frac{nM}{M + m(n-1)} = \frac{M}{m} \tag{13}$$

This ratio is the maximum and worst ratio that can be obtained by any algorithm. So, $\mathcal{R}_p^2 \leq_r \mathcal{R}_p$, and equality holds if and only if $p = m$. □

7. Relative Interval Analysis

Dorrigiv et. al. [12] 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. It compares two algorithms on the basis of the rate of the outcomes over the length of the input sequence rather than their worst case behavior. Here we define this analysis for profit maximization problems for two algorithms $\mathcal{A}$ and $\mathcal{B}$, following [12].

Definition 6. Let

$$Min_{\mathcal{A},\mathcal{B}}(n) = \min_{|I|=n} \{\mathcal{A}(I) - \mathcal{B}(I)\} \text{ and } Max_{\mathcal{A},\mathcal{B}}(n) = \max_{|I|=n} \{\mathcal{A}(I) - \mathcal{B}(I)\}.$$

These functions are used to define the following two measures:

$$Min(\mathcal{A},\mathcal{B}) = \liminf_{n\to\infty} \frac{Min_{\mathcal{A},\mathcal{B}}(n)}{n} \text{ and } Max(\mathcal{A},\mathcal{B}) = \limsup_{n\to\infty} \frac{Max_{\mathcal{A},\mathcal{B}}(n)}{n}. \quad (14)$$

Note that $Min(\mathcal{A},\mathcal{B}) = -Max(\mathcal{B},\mathcal{A})$ and $Max(\mathcal{A},\mathcal{B}) = -Min(\mathcal{B},\mathcal{A})$. The *relative interval* of $\mathcal{A}$ and $\mathcal{B}$ is defined as $l(\mathcal{A},\mathcal{B}) = [Min(\mathcal{A},\mathcal{B}), Max(\mathcal{A},\mathcal{B})]$. If $Max(\mathcal{A},\mathcal{B}) > |Min(\mathcal{A},\mathcal{B})|$, then $\mathcal{A}$ is said to have *better performance* than $\mathcal{B}$ in this model. In particular, if $l(\mathcal{A},\mathcal{B}) = [0,\beta]$ for $\beta > 0$, then it is said that $\mathcal{A}$ *dominates* $\mathcal{B}$, since $Min(\mathcal{A},\mathcal{B}) = 0$ indicates that $\mathcal{A}$ is never worse than $\mathcal{B}$ and $Max(\mathcal{A},\mathcal{B}) > 0$ says that $\mathcal{A}$ is better at least for some case(s).

Given the finite nature of the online search problem, the above limits are always zero. So we propose a modification of Relative Interval Analysis to make it suitable for finite profit.

Definition 7. $Min_{\mathcal{A},\mathcal{B}}(n)$ and $Max_{\mathcal{A},\mathcal{B}}(n)$ are as in Definition 6. These functions are used to define the following two measures:

$$Min(\mathcal{A},\mathcal{B}) = \inf_{n\geq 2}\{Min_{\mathcal{A},\mathcal{B}}(n)\} \text{ and } Max(\mathcal{A},\mathcal{B}) = \sup_{n\geq 2}\{Max_{\mathcal{A},\mathcal{B}}(n)\}. \quad (15)$$

The *Finite Relative Interval* of $\mathcal{A}$ and $\mathcal{B}$ is defined as

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

Relative performance and dominance with regards to $fl(\mathcal{A},\mathcal{B})$ are defined as for $l(\mathcal{A},\mathcal{B})$ from Definition 6.

Theorem 11. *According to Finite Relative Interval Analysis,* $fl(\mathcal{R}_q,\mathcal{R}_p) = [m-q+1, M-p]$.

Proof. The minimum value of $\mathcal{R}_q(I) - \mathcal{R}_p(I)$ is obtained by any sequence of prices with all prices smaller than q, where the first price is $q-1$ and the last price is m. In this case, $Min(\mathcal{R}_q,\mathcal{R}_p) = m-q+1$. The maximum value of $\mathcal{R}_q(I) - \mathcal{R}_p(I)$ is $M-p$, which is obtained when the first price is p and the second price is M. This proves that $fl(\mathcal{R}_q,\mathcal{R}_p) = [m-q+1, M-p]$. $\square$

Corollary 4. *Let* $s = \left\lceil \frac{M+m}{2} \right\rceil$*. According to Finite Relative Interval Analysis, the best RPP algorithm is* $\mathcal{R}_s$*.*

PROOF. Let $p < s$. To compare $\mathcal{R}_s$ and $\mathcal{R}_p$, we have $Min(\mathcal{R}_s, \mathcal{R}_p) = m - \left\lceil \frac{M+m}{2} \right\rceil + 1$ and $Max(\mathcal{R}_s, \mathcal{R}_p) = M - p > M - \left\lceil \frac{M+m}{2} \right\rceil$. This shows that $\forall p < s$, $Max(\mathcal{R}_s, \mathcal{R}_p) > |\,Min(\mathcal{R}_s, \mathcal{R}_p)|$ and, consequently, $\mathcal{R}_s$ performs better than $\mathcal{R}_p$. Now assume $q > s$. Then a comparison between $\mathcal{R}_q$ and $\mathcal{R}_s$ gives $Min(\mathcal{R}_q, \mathcal{R}_s) = m - q + 1 < m - \left\lceil \frac{M+m}{2} \right\rceil + 1$, and $Max(\mathcal{R}_q, \mathcal{R}_s) = M - s = M - \left\lceil \frac{M+m}{2} \right\rceil$. This inequality shows that for all $q > s$, $Max(\mathcal{R}_q, \mathcal{R}_s) \leq |\,Min(\mathcal{R}_q, \mathcal{R}_s)|$ and $\mathcal{R}_s$ performs at least as well as $\mathcal{R}_q$. These two cases prove that $\mathcal{R}_s$ is a best RPP algorithm according to Finite Relative Interval Analysis. □

Theorem 12. *According to Finite Relative Interval Analysis,*

$$fl(\mathcal{R}_p, \mathcal{R}_p^2) = [p - M, M - m].$$

PROOF. For the minimum value of $\mathcal{R}_p(I) - \mathcal{R}_p^2(I)$, we use any sequence of prices starting with two prices in the order p followed by M. In this case, $Min_{\mathcal{R}_p, \mathcal{R}_p^2}(N) = p - M$. The maximum value of $\mathcal{R}_p(I) - \mathcal{R}_p^2(I)$ is $M - m$, which occurs when the first price is M and the other prices are m. That gives $fl(\mathcal{R}_p, \mathcal{R}_p^2) = [p - M, M - m]$. □

The above theorem shows that, according to Finite Relative Interval Analysis, $\mathcal{R}_p$ has better performance than $\mathcal{R}_p^2$ if $p > m$.

8. Relative Worst Order Analysis

Relative Worst Order Analysis [5] compares two online algorithms directly. It compares two algorithms on their worst orderings of sequences which have the same content, but possibly in different order. The definition of this measure is somewhat more involved; see [6] for more intuition on the various elements. Here we use the definitions for the strict Relative Worst Order Analysis for profit maximization problems.

Definition 8. Let I be any input sequence, and let n be the length of I. Let $\mathcal{A}$ be any online search algorithm. Then $\mathcal{A}_W(I) = \min_\sigma \mathcal{A}(\sigma(I))$.

Definition 9. For any pair of algorithms $\mathcal{A}$ and $\mathcal{B}$, we define

$$\begin{aligned} c_l(\mathcal{A},\mathcal{B}) &= \sup\{c \mid \forall I : \mathcal{A}_W(I) \geq c\mathcal{B}_W(I)\} \text{ and} \\ c_u(\mathcal{A},\mathcal{B}) &= \inf\{c \mid \forall I : \mathcal{A}_W(I) \leq c\mathcal{B}_W(I)\}. \end{aligned}$$

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 *strict 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.

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

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*.

When two algorithms happen to be incomparable, Relative Worst Order Analysis can still be used to express their relative performance.

Definition 10. If at least one of the ratios $c_u(\mathcal{A},\mathcal{B})$ and $c_u(\mathcal{B},\mathcal{A})$ is finite, the algorithms $\mathcal{A}$ and $\mathcal{B}$ are $(c_u(\mathcal{A},\mathcal{B}), c_u(\mathcal{B},\mathcal{A}))$-*related*.

Theorem 13. *According to Relative Worst Order Analysis, $\mathcal{R}_q$ and $\mathcal{R}_p$ are $(\frac{M}{p}, \frac{q-1}{m})$-related. They are comparable in $\mathcal{R}_q$'s favor if $p = m$ and $q = m+1$.*

Proof. For the maximum value of the ratio of $\mathcal{R}_{qW}(I)$ and $\mathcal{R}_{pW}(I)$, we can construct a sequence I with only one p and one M and all the other prices smaller than q. Among all the permutations of I, the worst output for $\mathcal{R}_q$ is M and that of $\mathcal{R}_p$ is p. This gives the value of the upper bound $c_u(\mathcal{R}_q, \mathcal{R}_p)$ as $\frac{M}{p}$. For the lower bound, assume I has only one $q-1$ and one m and all the other prices are smaller than p. Then, $\mathcal{R}_p$ takes $q-1$ as its output on every permutation of I, but the worst output of $\mathcal{R}_q$ gives m. On this sequence, $\mathcal{R}_q$ performs worse than $\mathcal{R}_p$, and the ratio of the outputs of the two algorithms can never be lower than that. So,

$$\begin{aligned} c_l(\mathcal{R}_q,\mathcal{R}_p) &= \frac{m}{q-1}\begin{cases} = 1, & \text{for } q = m+1 \text{ and } p = m \\ < 1, & \text{otherwise} \end{cases} \\ c_u(\mathcal{R}_q,\mathcal{R}_p) &= \frac{M}{p} > 1 \end{aligned}$$

From the above expressions and the definitions of strict Relative Worst Order Analysis, we can see that $\mathcal{R}_q$ and $\mathcal{R}_p$ are comparable when $p = m$ and $q = m+1$. For all the other cases, they are incomparable. For this single feasible condition of $c_l(\mathcal{R}_q, \mathcal{R}_p) = 1$, we have $WR_{\mathcal{R}_q,\mathcal{R}_p} = \frac{M}{p} > 1$, and we can say that algorithms $\mathcal{R}_q$ and $\mathcal{R}_p$ are comparable in $\mathcal{R}_q$'s favor. Using Definition 10, since all the ratios are finite, $c_u(\mathcal{R}_p, \mathcal{R}_q)$ is $\frac{q-1}{m}$ and the algorithms $\mathcal{R}_q$ and $\mathcal{R}_p$ are $(\frac{M}{p}, \frac{q-1}{m})$-related. □

Note that this relatedness result gives the same conditions indicating which algorithm is better as Competitive and Random Order Analysis. Although with the original definition of relatedness in Relative Worst Order Analysis the values are not interpreted further, one could use the concept of *better performance* (see [12] or Section 7) from Relative Interval Analysis, and state the following:

Corollary 5. *Let $s = \left\lceil \sqrt{Mm} \right\rceil$. If $q > s$ and $\mathcal{R}_q$ and $\mathcal{R}_s$ are (c, c')-related, then $c \leq c'$, and if $p < s$ and $\mathcal{R}_s$ and $\mathcal{R}_p$ are (c, c')-related, then $c > c'$.*

Proof. By Theorem 13, $c_u(\mathcal{R}_q, \mathcal{R}_s) = \frac{M}{\lceil \sqrt{Mm} \rceil}$ and $c_u(\mathcal{R}_s, \mathcal{R}_q) = c_l(\mathcal{R}_q, \mathcal{R}_s) = \frac{q-1}{m}$. As $q > s = \left\lceil \sqrt{Mm} \right\rceil$, $c_u(\mathcal{R}_q, \mathcal{R}_s) \leq c_u(\mathcal{R}_s, \mathcal{R}_q)$. Similarly, $c_u(\mathcal{R}_s, \mathcal{R}_p) = \frac{M}{p}$ and $c_u(\mathcal{R}_p, \mathcal{R}_s) = \frac{\lceil \sqrt{Mm} \rceil - 1}{m}$. As $p < s = \left\lceil \sqrt{Mm} \right\rceil$, $c_u(\mathcal{R}_p, \mathcal{R}_s) < c_u(\mathcal{R}_s, \mathcal{R}_p)$. □

A similar result on a parameterized family of algorithms can be found in [7]. This could be defined as a weak form of optimality within a class of algorithms, and we will say that $\mathcal{R}_s$ is *superior* to any other RPP algorithm.

Theorem 14. *According to Relative Worst Order Analysis, $\mathcal{R}_p$ and $\mathcal{R}_p^2$ are comparable in $\mathcal{R}_p$'s favor and $WR_{\mathcal{R}_p,\mathcal{R}_p^2} = \frac{M}{m}$.*

Proof. From the proofs of Theorems 2 and 10, we have already seen that the worst case performance ratio of $\mathcal{R}_p$ and $\mathcal{R}_p^2$ on the same sequence is M/m which is the largest possible value. So we can conclude that $c_u(\mathcal{R}_p, \mathcal{R}_p^2) = \frac{M}{m}$.

For deriving the lower bound on the ratio $c_l(\mathcal{R}_p, \mathcal{R}_p^2)$, we show that for any sequence, on its worst permutation of that sequence, $\mathcal{R}_p$'s output will be at least as large as $\mathcal{R}_p^2$'s on its worst ordering of that sequence. We can prove this fact by taking the worst output of $\mathcal{R}_p$ and $\mathcal{R}_p^2$ over all the permutations of a sequence I. Let x and y denote these outputs, respectively. If $y < p$, then there is no price in I smaller than y. So, the worst output of $\mathcal{R}_p$ must be greater than or equal to y, i.e., $x \geq y$. If $y \geq p$, it is the smallest price in I greater than p, so again $x \geq y$. Thus, $c_l(\mathcal{R}_p, \mathcal{R}_p^2) = 1$. From Definition 9, we conclude that according to Relative Worst Order Analysis, $\mathcal{R}_p$ and $\mathcal{R}_p^2$ are comparable in $\mathcal{R}_p$'s favor and $WR_{\mathcal{R}_p,\mathcal{R}_p^2} = \frac{M}{m}$. $\square$

9. Max/Max Analysis

In [3], Ben-David et. al. defined the Max/Max ratio for cost minimization problems. The Max/Max ratio compares an algorithm's worst cost for any sequence of length n to OPT's worst cost for any sequence of length n. If we want to preserve this worst output ratio behavior for profit maximization problems, the analysis must consider the minimum profit for each sequence length and could be named Min/Min Analysis. Here we define the Min/Min ratio by modifying the definition of the Max/Max ratio.

Definition 11. The *Min/Min ratio* of an online algorithm $\mathcal{A}$, $w_M(\mathcal{A})$, is $M(OPT)/M(\mathcal{A})$, where

$$M(\mathcal{A}) = \liminf_{n\to\infty} \min_{|I|=n} \mathcal{A}(I)/n. \tag{16}$$

In the online search problem, for any RPP algorithm, the minimum output is m for some sequence of length n. For example, the sequence of n consecutive prices of value m always has the output m. As m is a finite value, the limit of Eq. (16) is zero for any algorithm. Thus, Min/Min Analysis is not applicable in comparing these online search algorithms.

However, we can modify the previous definition to make it suitable for finite problems, as with Finite Relative Interval Analysis.

Definition 12. The *Finite Min/Min ratio* of an online algorithm $\mathcal{A}$, $w_M(\mathcal{A})$, is $M(OPT)/M(\mathcal{A})$, where

$$M(\mathcal{A}) = \inf_{n\geq 2}\{\min_{|I|=n} \mathcal{A}(I)\}. \tag{17}$$

For the sequences where every price is m, every algorithm, even OPT, outputs m. This makes all Min/Min ratios equal to one, making every algorithm equivalent according to Min/Min Analysis.

10. Concluding Remarks

With regards to the concrete results, for Competitive and Random Order Analysis, $\mathcal{R}_{\sqrt{mM}}$ is the best online algorithm. Relative Worst Order and Relative Interval Analysis have more nuanced answers, but point to $\mathcal{R}_{\sqrt{mM}}$ and $\mathcal{R}_{\frac{m+M}{2}}$, respectively. Bijective and Average Analysis seem to provide the least interesting information in this context; Average Analysis indicates $\mathcal{R}_M$ as the best algorithm, and Bijective Analysis deems most algorithms incomparable.

This points to three choices for the online player with regards to the optimal reservation prices, namely $\sqrt{mM}$, $\frac{m+M}{2}$, and M, depending on the different analysis methods, i.e., the geometric mean, the arithmetic mean, and the maximum M of all possible values. This clearly shows that the objectives of the different performance measures vary greatly, trying to limit poor performance in a proportional or additive sense, or focusing equally on all scenarios, including the possibly non-occurring upper bound of M. Thus, the different measures are tailored towards different degrees of risk aversion—cautiousness vs. aggressiveness. The observations above complement the findings regarding greediness and laziness from [7].

Studying performance measures and disclosing their properties and differences from each other is work in progress. With this study, we have added Online Search to the collection of problems that have been investigated with a spectrum of measures. More online problem scenarios must be analyzed this broadly before strong conclusions concerning the different performance measures can be drawn.

Another interesting direction for future work would be to incorporate other aspects of financial problems into the analysis in the context of other performance measures, as has been done for competitive analysis of financial games in the risk-reward framework of al Binali [1].

Acknowledgments

The authors would like to thank Leah Epstein, Asaf Levin, and Alejandro López-Ortiz for discussions regarding bijective analysis and real-valued inputs.

References

[1] S. al Binali. A risk-reward framework for the competitive analysis of financial games. *Algorithmica*, 25:99–115, 1999.

[2] Spyros Angelopoulos, Reza Dorrigiv, and Alejandro López-Ortiz. On the separation and equivalence of paging strategies. In *18th Symposium on Discrete Algorithms*, pages 229–237, 2007.

[3] Shai Ben-David and Allan Borodin. A new measure for the study of on-line algorithms. *Algorithmica*, 11(1):73–91, 1994.

[4] Allan Borodin and Ran El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[5] Joan Boyar and Lene M. Favrholdt. The relative worst order ratio for online algorithms. *ACM Transactions on Algorithms*, 3, 2007.

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

[7] Joan Boyar, Sandy Irani, and Kim S. Larsen. A comparison of performance measures for online algorithms. In *Eleventh International Algorithms and Data Structures Symposium*, volume 5664 of *Lecture Notes in Computer Science*, pages 119–130. Springer, 2009. Full version available at arXiv: 0806.0983v1 [cs.DS].

[8] Joan Boyar, Kim S. Larsen, and Morten N. Nielsen. The accommodating function: A generalization of the competitive ratio. *SIAM Journal on Computing*, 31:233–258, 2002.

[9] Gen-Huey Chen, Ming-Yang Kao, Yuh-Dauh Lyuu, and Hsing-Kuo Wong. Optimal buy-and-hold strategies for financial markets with bounded daily returns. *SIAM Journal on Computing*, 31(2):447–459, 2001.

[10] Edward G. Coffman Jr., János Csirik, Lajos Rónyai, and Ambrus Zsbán. Random-order bin packing. *Discrete Applied Mathematics*, 156(14):2810–2816, 2008.

[11] Reza Dorrigiv and Alejandro López-Ortiz. A survey of performance measures for on-line algorithms. *SIGACT News*, 36(3):67–81, 2005.

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

[13] Ran El-Yaniv. Competitive solutions for online financial problems. *ACM Computing Surveys*, 30(1):28–69, 1998.

[14] Ran El-Yaniv, Amos Fiat, Richard M. Karp, and G. Turpin. Optimal search and one-way trading online algorithms. *Algorithmica*, 30(1):101–139, 2001.

[15] Bala Kalyanasundaram and Kirk Pruhs. Speed is as powerful as clairvoyance. In *36th Annual IEEE Foundations of Computer Science*, pages 214–221, 1995.

[16] C. Kenyon. Best-fit bin-packing with random order. In *7th Symposium on Discrete Algorithms*, pages 359–364, 1996.

[17] Michael Rothschild. Searching for the lowest price when the distribution of prices is unknown. *Journal of Political Economy*, 82(4):689–711, 1974.

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