Theoretical Computer Science 532 (2014) 2–13

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

A comparison of performance measures via online search [☆]

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

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

ARTICLE INFO

Keywords:
Online algorithms
Online search
Performance measures

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 [J. Boyar, S. Irani, K.S. Larsen, A comparison of performance measures for online algorithms, in: Eleventh International Algorithms and Data Structures Symposium, in: Lecture Notes in Computer Science, vol. 5664, Springer, 2009, pp. 119–130], 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.

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.

☆ A preliminary version of this paper appeared in the *SIAM Joint FAW–AAIM Conference*, in: 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.
E-mail addresses: joan@imada.sdu.dk (J. Boyar), kslarsen@imada.sdu.dk (K.S. Larsen), abyaym@imada.sdu.dk (A. Maiti).

0304-3975/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.tcs.2013.07.022

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 the 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