Dissertation zur Erlangung des Grades des
Doktors der Ingenieurwissenschaften (Dr.-Ing.) der
Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Analysis of Algorithms for Online Uni-directional Conversion Problems

von Iftikhar AHMAD
im September 2013

SAARLAND UNIVERSITY
COMPUTER SCIENCE

ORBI
Operations Research and Business Informatics

Tag des Kolloquiums:	30.Sep.2013
Dekan:	Prof. Dr. Mark Groves
Mitglieder des Prüfungsausschusses:	Prof. Dr. Dr. h.c. mult. Kurt Mehlhorn Univ.-Prof. Dr.-Ing. Günter Schmidt PD Dr. Rob van Stee Dr. Andreas Wiese

To:
Ammi , Abbu
&
LR Ahmad

Acknowledgement

Words of thanks, to Allah Almighty for whose beneficence and merciness their is nothing worthy in the diction of all languages. He, the Lord of lords provided opportunity that enabled me to exploit his gifted capabilities. Under His umbrella of constant benevolence I managed to accomplish my research work at a fine level.

I am grateful indeed and would proceed with cordial thanks to Günter Schmidt, who supervised my thesis and pointed directions through his sheer guidance throughout my research. His support and encouragement enabled me to carve out and fulfill my tasks. Without his wise counsel and able guidance, it would have been impossible to complete the thesis in this manner. I would also deem it obligatory on my part to forward my gratitude to Rob van Stee, my co-supervisor who helped me to his utmost for making my work praise able and worth able. I owe him an infinite debt for all I gained from his expertise and work. I express my gratitude to Kurt Mehlhorn and Andreas Wiese for their support and time.

The research environment at ORBI was particularly encouraging and motivating. I will like to thank Javeria Iqbal and Esther Mohr for their fruitful discussions on various topics, Mike Kersch for his co-operation in arranging data for the experiments and for the German version of the summary (Zusammenfassung), Hai Dung Dinh, Robert Dachow, Claude Gangolf, Christian Ebli and Marcus Pirron for their constructive criticism on the work. A special thanks to Hedi Staub for her tireless help and support in all organizational matters. My stay in Saarbrücken was made memorable by the vibrant Pakistani community, though limited in number, their cheerful and joyous company kept the homesickness at bay. I will cherish these moments for ever, especially the intensity of cricket matches, our ambition of Naya Pakistan and our love for BBQ. I am also thankful to all my friends from Freiburg to Berlin (and elsewhere in Germany) as well as in other parts of the world. I also forward thanks to my friends in Pakistan who always mustered up my courage and motivated/threatened me whenever the going got tougher. I will always cherish the lovely time spent with Javeria and Ahmed Shariq with whom I had lots of dinners and lunches together.

In one's life kith and kins are what one's live for and adore. In this regard, I would like to give infinite piles of reverence, regards and gratitude to my parents and siblings for pointing me ways to complete my research work efficiently. I am immensely grateful to my parents who provided me immeasurable moral, spiritual and financial support. To *Ammi*, who like all mothers of the world is exceptionally caring, loving and wonderful and *Abbu*, who sacrificed his needs for our wants. I am thankful to my siblings Gulzi and Saba, Huma, Romi, Faiq, Faiza and Gul Rukh. Your love and support is priceless. Love to Ghazali, Afifa and Shaheer, whome I am looking forward to meet soon. Last but equally (more) importantly, I am thankful to my wife *LR Ahmad* for her love, patience and support whilst I worked on my thesis and a lot of love to Areeba Ahmad and Aroofa Ahmad, *baba jani* misses you both.

Iftikhar Ahmad

Summary

In an online uni-directional conversion problem, an online player wants to convert an asset D to a desired asset Y. The objective of the player is to obtain the maximum amount of the desired asset. Competitive analysis is used as a tool for the design, and analysis of online algorithms for conversion problems. Although widely used, competitive analysis has its own set of drawbacks when the applicability of online algorithms in real world is considered. In this work, we investigate online uni-directional conversion problems with the objective to suggest measures for improving the applicability of online conversion algorithms in real world. The work is based on four different aspects, each examining online conversion problems with a different perspective.

First we investigate the use of competitive ratio as a coherent risk measure. In the last decade coherent measures of risk meeting a set of four desirable properties gained considerable importance. It is deemed important for a risk measure to be coherent in order to be in line with the basic risk paradigms and thus used in practice. We discuss competitive ratio as a risk measure and show that it can be used in practice as it satisfies all the required properties of a coherent measure of risk.

Secondly, we evaluate a selected set of online algorithms on the real world data to highlight the gap between theoretically guaranteed and experimentally achieved competitive ratio. In order to avoid the data snooping bias, we also use artificial data generated using the bootstrap procedure. From the experimental study, we observe a wide gap between theoretically guaranteed and experimentally achieved performance. The algorithms perform much better on the real world data as the worst case inputs are not (frequently) observed in real world.

The third aspect of the study deals with generating synthetic data that are true representation of all possible scenarios. It was found in the experimental study that although bootstrap can be used to generate artificial data sets, the procedure fails to replicate all possible scenarios such as market crashes. The bootstrap procedure omits the extreme values as outliers and the worst cases are not represented in the resultant synthetic data. In order to generate the test instances that have the ability to replicate all possible scenarios, we suggest the use of *Extreme Value Theory* (EVT) approach. Using *EVT* approach, we generate synthetic data and

execute a selected set of non-preemptive uni-directional online algorithms on it.

In the fourth and final part, we propose risk aware non-preemptive algorithms for uni-directional and bi-directional conversion problems. The proposed algorithms are flexible to accommodate the risk level of the online player whereas guaranteeing a bounded worst case competitive ratio as well. We evaluate our algorithms using the competitive analysis approach as well as testing the algorithms on the real world data. We observe that risk aware algorithms perform better than risk mitigating algorithms on the real world data.

The results are presented in the form of research papers and will help to improve the applicability of online conversion algorithms in real world. We conclude the work by discussing a number of open questions that will provide new directions for future research.

Zusammenfassung

In einem Online-uni-directional-conversion-Problem, möchte ein Online-Spieler ein Asset D in ein gewünschtes Asset Y konvertieren. Das Ziel des Spielers ist es, den maximalen Wert des gewünschten Assets zu erhalten. Die competitive analysis wird als Hilfsmittel verwendet, um Online-Algorithmen für Conversion-Probleme zu entwerfen und zu analysieren. Obwohl die competitve analysis weit verbreitet ist, besitzt sie mehrere Nachteile wenn ihre Anwendbarkeit auf Online-Algorithmen in der realen Welt betrachtet wird. In dieser Arbeit werden wir online Uni-directional-conversion-Probleme betrachten, mit dem Ziel, Kennzahlen zu erarbeiten, um die Anwendbarkeit von Online-Conversion-Algorithmen in der realen Welt zu verbessern. In einem ersten Schritt untersuchen wir die competitive ratio als kohärentes Risikomaß und schließen, wenn es alle notwendigen Kohärenzaxiome erfüllt wurden, dass die competitive ratio in der Praxis eingesetzt werden kann. In einem zweiten Schritt evaluieren wir eine ausgewählte Menge an Online-Algorithmen in der realen Welt. Außerdem werden die Daten gebootstrapped, um den Unterschied zwischen der theoretisch garantierten und empirisch erreichten competitive ratio. Der dritte Aspekt dieser Arbeit betrachtet das Generieren von synthetischen Daten, welche alle möglichen Szenarien, wie beispielsweise einen Marktcrash, repräsentieren. Wir empfehlen dafür den Einsatz der Extreme-Value-Theorie (EVT). Mit der EVT generieren wir synthetische Daten und führen eine ausgewählte Menge an nicht-präemptiven Uni-directionalen-Online-Algorithmen über diesen aus. Der vierte Beitrag dieser Arbeit beinhaltet das Design und die Analyse von risk-aware Reservationspreis-Algorithmen für Conversion-Probleme. Die vorgeschlagenen Algorithmen können das Risikoniveau des Online-Spielers aufnehmen und garantieren eine begrenzte Worst-Case-Competitive-Ratio. Wir evaluieren unsere Algorithmen mit Hilfe des competitive-analysis-Ansatzes sowie dem Testen der Algorithmen auf realen Daten. Die Resultate werden in Form von Forschungsarbeiten präsentiert und helfen, die Anwendbarkeit von Online-Conversion-Algorithmen in der realen Welt zu verbessern. Wir schließen die Arbeit mit einer Diskussion über eine Reihe offener Fragen, welche neue Forschungsrichtungen für die Zukunft eröffnen.

Table of Contents

List of Tables

List of Figures

1
Introduction

With rapid development of information technology, new possibilities emerged for individual as well as corporate business users. One such area which observed considerable growth in the aftermath of digital outbreak is the electronic financial market. The main objective of a financial market is to bring buyers and the sellers closer [28, 29]. The introduction of high speed communication networks which also proves to be highly reliable, has provided a novel marketplace for sellers and buyers in the form of e-financial markets [29, 40]. An example of such venture is *Forex - Foreign Currency Exchange*. In *forex* a player has a fixed amount of currency D and wants to exchange it to a desired currency Y. The purpose of the player may vary as she can do it for profit or for personal need. Irrespective of the purpose, the objective of the player is to maximize the amount of Y. In an e-commerce domain, the player is interested in the automation of the process that is carried out without (or with minimum) human intervention. Thus the need for an algorithm arises. In simple words, the player is interested to convert when the exchange rate of D to Y is the highest. However, there are some real issues that need to be considered when an algorithm is designed for the *forex* problem. For example, the player (and therefore the algorithm) has no knowledge about the future prices. Similarly, if the player rejects an exchange rate, she cannot revisit the past and accept the exchange rate as the offer expires after a fixed interval of time, i.e., all decisions are irrevocable.

Theoretical computer science answer to *forex* problem is in the form of *online algorithms* [27]. Online algorithms are designed based on rigorous mathematical proofs. The resultant algorithms are therefore guaranteed to be valid and are ensured to work as described. For the design of such algorithms, assumptions are made. These assumptions are necessary as otherwise the mathematical formulation of problems is not possible. If the assumptions are not valid, the results and proofs are invalidated as well. A natural question thus arises “Can we circumvent the assumptions?”. Unfortunately, the answer to the questions is not entirely to our liking as assumptions are like “necessary evils”. We do not like them but we cannot do away with them entirely either. For example, El-Yaniv et al. [27] assumed a priori information about the minimum and the maximum offered price in order to design an algorithm with bounded competitive ratio. The assumption seems unrealistic, but without this assumption, it is not possible to derive an online algorithm with bounded competitive ratio. Therefore, assumptions are important to formulate a problem via a mathematical model which in turn provide a well defined algorithm for the problem. The deviation of real world scenarios from the assumed settings of model can render the results invalid. Thus a gap exists between the formulation and solution of problems via mathematical models and their application in real world.

We consider *online conversion* problem [27] - a generalization of *forex*. We analyze the applicability of online algorithms in real world and discuss measures to improve the adaptability of such algorithms in practice. In the following, we present online conversion problem and discuss the various types of the problem. We then proceed to present open questions and discuss our approach and the obtained results.

1.1 Online Conversion Problem

An online conversion problem deals with the scenario of converting an asset D into another asset Y with the objective to obtain the maximum amount of Y after time T. The process can be repeated in both directions, i.e., converting asset D into asset Y , and Y back to asset D. In a typical problem setting, on each day t, the player is offered a price q_t to convert D to Y, the player may accept the price q_t or may decide to wait for a better price. The game ends when the player converts whole of the asset D to Y. The player is not restricted to maximize the terminal wealth in form asset Y but she can also maximize the terminal wealth in D. This allows the conversion from D to Y and Y back to D. Formally, we have the following problem setting;

Problem Setting:

Consider a player who wants to convert an asset D into another asset Y. Assume that the player starts with $D_0 = 1$ and Y_0=0. At each time $t = 1, 2, \ldots, T$ the player is offered a price q_t, and must immediately make an irrevocable decision whether to accept the offered price q_t or not. If the player decides to accept the price, she can convert a portion or the whole amount of asset D at the offered price q_t. The game ends when the player has converted D completely into Y. If there is still some amount of asset D remaining on the last day T, it must be converted at the last offered price q_T which might be the worst (lowest) offered price.

Based on the objective of the player, a conversion problem is classified either as *uni-directional* or *bi-directional*.

Definition 1.1 *Uni-directional Conversion;*
In a uni-directional conversion problem the online player is allowed to convert D into Y only and conversion back from Y into D is forbidden. The online player may either perform a max-search or min-search, depending on the objective function of the player.

Max-search is synonym for selling, i.e., a profit maximization problem whereas min-search refers to buying and can be classified as cost minimization problem.

Definition 1.2 *Bi-directional Conversion;*
In a bi-directional conversion problem the online player converts D into Y back and forth with the objective to maximize the terminal wealth D.

A variety of online algorithms are proposed for conversion problem [16, 20, 21, 27, 36, 39, 47, 69]. These algorithms can broadly be categorized in two classes, based on the amount of wealth s_t they invest at any given time t. The two classes are *non-preemptive* and *preemptive* algorithms and are defined as follows;

Definition 1.3 *Non-preemptive Algorithm;*
Non-preemptive algorithm invests either none or all when offered a price q_t, i.e., $s_t \in \{0, 1\}$.

Definition 1.4 *Preemptive Algorithm;*
Preemptive algorithm divides the wealth into small portions and based on pre-defined criterion invests a portion of wealth when offered a price q_t, i.e., $s_t \in [0, 1]$.

Optimum offline algorithm is used as benchmark to measure the performance of an online algorithm. Unlike an online algorithm, optimum offline algorithm OPT knows the entire input sequence in advance and always makes an optimum decision. Optimum offline algorithm is *non-preemptive* as it invests at one point of time. The performance of an online algorithm is compared against the optimum offline algorithm and the resultant ratio is called "competitive ratio". Competitive ratio is defined as following;

Definition 1.5 *Let ON be an online algorithm for some maximization problem P and $\mathcal{I}$ be the set of problem instances. Let $ON(I)$ be the performance of ON on* *input sequence I and $OPT(I)$ be the performance of optimum offline algorithm. The algorithm ON is c-competitive if $\forall I \in \mathcal{I}$*

$$ON(I) \geq \frac{1}{c} OPT(I). \tag{1.1}$$

Online algorithms for conversion problems are designed with the objective to achieve a lower competitive ratio [16, 20, 26, 27, 25, 39]. Despite the wide spread use of competitive analysis as a standard tool for the design and analysis of online algorithms for conversion problems, there is a considerable gap between the theoretical settings and real world scenario. A major drawback is the dependence of competitive analysis on single input instance which may occur very rarely in real world. It means that in real world it is highly unlikely (though not impossible) that worst case will occur and measuring the performance of an online algorithm based on single input instance is not an "ideal" choice.

1.2 Research Questions and Contributions

We focus on online uni-directional conversion problems under the competitive analysis paradigm. We contribute towards the design and analysis of algorithms for online conversion problems by identifying a number of open questions from the literature and answer them with the objective to improve the applicability of online conversion algorithms in real world. In the following, we present the research questions and short description of our results.

Question 1 *Is the competitive ratio a coherent measure of risk?*

A variety of methods are used to measure the risk of an investment strategy such as standard deviation (σ), value at risk (VaR) and conditional value at risk ($CVaR$). However, some of these measures are not consistent with basic risk paradigm. A risk measure should ideally satisfy the basic properties of risk. The fulfillment of these properties makes a risk measure suitable for use in real world. For example, one such desired property is *diversification*, referring to the phenomenon of investing in more than one assets to reduce the risk. If an investor diversify by splitting the total wealth and invests in two assets, then the risk measure should report that the combined risk of the portfolio is no more than the sum of individual risks. However, not all risk measures satisfy this property. VaR is one such example of risk measure which does not conform to the risk principle of *diversification*. In order to define the desirable properties of a risk measure, Artzner et al. [7] introduced an axiomatic definition of risk measure. Any risk measure that satisfies the axioms is termed as "coherent risk measure". Each axiom refers to one desirable property of risk measure. We consider competitive ratio as measure of risk and show that it satisfies all the required axioms of Artzner et al. [7].

Publication 1 I. Ahmad, E. Mohr and G. Schmidt: *Competitive Ratio as Coherent Measure of Risk,* Selected Papers of the Annual International Conference of the German Operations Research Society (GOR e.V.), University of Hanover, September 4-7, 2012, Springer-Verlag Berlin Heidelberg 2013, in press.

Question 2 *How to evaluate the performance of a selected set of algorithms on the real world data and identify the gap between theoretically guaranteed and experimentally achieved performance?*

We answer the question by considering a set of algorithms for uni-directional conversion problem and evaluate them on two different sets of real world as well as synthetically produced data sets. The real world data considered is daily closing prices of $DAX30$ and $S\&P500$ indices for ten years from 1-Jan-2001 to 31-Dec-2010. In addition, we employ bootstrap method to generate synthetic data sets. In order to simulate a real world scenario, we consider transaction cost of 0.025% of the volume transacted. We observe a considerable gap between theoretically guaranteed worst case and experimentally achieved competitive ratios. Algorithms perform much better than the worst case competitive ratio guarantees. It was also found that algorithms that invest prudently by taking into account the offered price perform better than those who do not consider the offered price. We use variance as measure to calculate the consistency of performance of algorithms.

Publication 2 I. Ahmad and G. Schmidt: *An Experimental Analysis of Online Unidirectional Conversion Problem,* In: C. Huemer and P. Lops (Eds.): EC-Web 2012, Lecture Notes in Business Information Processing, Vol. 123, pp. 176-187, 2012. Springer-Verlag Berlin Heidelberg 2012.

Question 3 *How can we generate test instances that truly represent both worst case as well as non-worst case input?*

The classical backtesting approach to validate algorithms on real data from the past with the assumption that if an algorithm performs better on the past data will perform well in the future as well is fraught with perils. The main reason is the limited availability of past data and the data snooping bias [15]. This originates the need of synthetic data sets. Bootstrap methods are used to generate test instances, however, the bootstrap method inherits one basic flaw from backtesting. The bootstrap data is generated from the real data. The limited availability of the real world data means that rare market events such as periods of unusually high or low returns and market crashes are not sufficiently represented in bootstrap data as well. Extreme Value Theory (EVT) deals with the events that deviate from the natural course of action and is used in a number of disciplines to deal with rare and extreme events. We use the EVT approach to generate test instances and evaluate a selected set of non-preemptive uni-directional algorithms. We report our findings on the basis of disparity between the observed competitive ratio and expected worst case competitive ratio.

Publication 3 E. Mohr, I. Ahmad and G. Schmidt: *Solving Uni-directional Conversion Problems by Online Algorithms under an Extreme Value Distribution.* DSOR Contributions to Information Systems, Band 8: Applied Mathematical Optimization and Modelling, L. Suhl, G. Mitra, C. Lucas, A. Koberstein, L. Beckmann (Eds.), pp. 459-470, 2012.

Question 4 *How to design risk aware reservation price algorithms?*

Online algorithms for conversion problems are designed based on competitive analysis. Competitive analysis assumes worst case future and the resultant algorithms are designed with guaranteed worst case performance. Therefore, online algorithms are risk averse in nature but in a real world scenario, investors want the flexibility to manage their risk level. Al-binali [4] extended the classical competitive ratio to risk-reward framework. We design reservation price algorithms that takes into account the risk level of the player and thus provide the flexibility to manage the risk. The proposed algorithms achieve a better (lower) competitive ratio than the reservation price algorithm of El-Yaniv et al. [27] when an improved (positive) outcome is observed. Similarly, the algorithms maintain a worst case competitive ratio in case the observed outcome is not favorable.

Publication 4 I. Ahmad and G. Schmidt: *Risk Aware Reservation Price Algorithms,* Submitted to: *OMEGA* - The International Journal of Management Science.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we discuss different measures for the performance evaluation of online algorithms. We discuss both theoretical measures (competitive analysis, Bayesian analysis and risk reward framework etc) as well as experimental measures (backtesting, bootstrapping, EVT approach). In addition, a short description of different risk measures is also presented. Chapter 3 presents a literature review of state of the art. The classification of online conversion algorithm based on nature of search and the amount of wealth invested is discussed. Chapter 3 answers Question 1 by discussing competitive ratio as coherent measure of risk. Chapter 5 answers the Question 2 by discussing the results of an extensive experimental study. Chapter 6 is based on Question 3 and provides methodological solution for generating test instances using EVT approach. Chapter 7 discusses Question 4 and presents risk aware reservation price algorithms for uni-directional and bi-directional conversion problems. It is shown that the proposed algorithms achieve a competitive ratio better than guaranteed competitive ratio for an improved outcome whereas maintain a bounded worst case competitive ratio for a worst outcome. Chapter 8 concludes the work by discussing a number of open questions for future research directions.

2 Performance Evaluation and Risk Measures

In this chapter, we discuss how to evaluate the performance of online algorithms and cover both theoretical and experimental measures used to evaluate the performance of online algorithms. As conversion problems are studied in the context of financial trading, where risk plays an important role, we cover different measures of risk as well.

2.1 Performance Evaluation Measure

Online algorithms are evaluated by a variety of methods. Broadly we can classify them into two main categories *i*) *theoretical measures* and *ii*) *experimental measures*. The theoretical measures evaluate the performance of online algorithms using rigorous mathematical techniques [14, 62] whereas the experimental measures use data (both real world and synthetic) to evaluate the performance of online algorithms [2, 37, 51, 59]. In the following we discuss the theoretical as well as experimental measures. As in this thesis, we will mostly deal with the worst case competitive analysis, risk-reward framework and experimental evaluation, therefore we will describe the aforementioned measures in greater detail. For the sake of completeness other performance evaluation measures will be presented in brief.

2.1.1 Theoretical Evaluation

A variety of literature discusses theoretical performance evaluation measures of online algorithms [27, 33, 57, 62]. We provide an overview of a subset of evaluation methods from the literature.

Bayesian Analysis

Bayesian analysis assumes that the player knows the probability distribution from which the input is drawn [57]. The online algorithm is analyzed with respect to the optimum offline algorithm assuming the known probability distribution. Likewise, the result of the analysis is valid as long as the input is limited to the assumed probability distribution. If the input is drawn from outside the assumed probability distribution the performance measure is rendered invalid [49].

The online conversion problem is studied under *Bayesian* analysis by a number of authors including Chou et al. [17] and Fujiwara et al. [33] among others. The work of Chou et al. [17] is based on the *statistical adversary* model introduced by Raghavan [57]. Chou et al. [17] strategy performs better than the standard benchmark of buy-and-hold. Fujiwara et al. [33] considered the *threat* based strategy of El-Yaniv et al. [27] and presented average case threat based algorithms based on different objective functions. The objective functions considered were $E[OPT/ON]$ and $E[OPT]/E[ON]$. Fujiwara et al. [33] showed that changing the performance measures leads to different optimal strategies.

Bayesian analysis provides good performance measure under "typical input" assuming the knowledge of probability distribution of the input data. However, the knowledge of exact probability distribution itself is hard to know in a real world. Similarly, it is realistically not possible to model the input on some probability distribution. Even if the input follows a particular distribution, the construction of a stochastic model that represents the underlying probability distribution is hard to envisage [49]. Bayesian analysis relies heavily on known probability distribution and any deviation of input data from the assumed distribution will nullify the results.

Worst Case Competitive Analysis

The worst case competitive analysis addresses the drawback of Bayesian analysis by circumventing the need of known probability distribution. The worst case competitive analysis does not assume any known probability distribution of input data [27, 62] and the adversary is free to draw the input data without the limitation of restricting to a specific probability distribution. This leads to simplified assumptions and easy analysis of algorithms. The performance of online algorithm is measured against an optimum offline algorithm with the objective to find the worst performance ratio of online algorithm to that of optimum offline algorithm. Formally we define the worst case competitive analysis as given in [27];

Definition 2.1 *Worst Case Competitive Analysis*
Let $\mathcal{P} = (\mathcal{I}, F, U)$ be a profit maximization problem. $\mathcal{I}$ be the set of all possible inputs; $\forall I \in \mathcal{I}$, $F(I)$ is the set of feasible outputs; U is a utility function such that $\forall I$ and $O \in F(I)$, $U(I, O) \in \mathbb{R}$. Consider an algorithm ON for $\mathcal{P}$. Given an input $I \in \mathcal{I}$, ON computes a feasible output $O \in F(I)$. $ON(I) = U(I, O)$ denotes the performance of ON on $I \in \mathcal{I}$. Each input I can be represented as a finite sequence such that $I = i_1, i_2, \ldots, i_T$ and a feasible output O can also be represented as finite sequence $O = O_1, O_2, \ldots, O_T$.

ON computes online if for each $j = 1, \ldots, T-1$, ON must compute o_j before i_{j+1} is revealed. ON is called c-competitive if $\forall I \in \mathcal{I}$,

$$ON(I) \geq \frac{1}{c} OPT(I). \tag{2.1}$$

$OPT(I)$ is the performance of optimum offline algorithm on I and T is the length of the input sequence.

Online conversion algorithms are mainly studied under the worst case competitive analysis. Since the seminal work of El-Yaniv et al. [26], there is a considerable body of literature devoted to online conversion algorithms under the worst case competitive analysis [16, 20, 47, 69, 70].

Although the worst case competitive analysis is a strong tool for the design and analysis of online algorithms, it suffers from a number of drawbacks. For instance, only a single input sequence determines the performance ratio of an algorithm. In addition, if two algorithms have the same worst case competitive ratio, it is hard to distinguish which algorithm is superior? For example, Torng [67] showed for cache problem that every marking algorithm is k-competitive (k being the size of cache). However, a number of studies [5, 54] shown that empirical performance differs widely from the theoretical bounds. *Least-Recently-Used* (LRU) outperforms *Flush-When-Full* (FWF) and *First-In-First-Out*($FIFO$) and the empirical competitive ratio of LRU was found significantly smaller than k.

Another significant limitation of the worst case competitive analysis in relation to online conversion algorithm is the lack of risk management [4]. Worst case competitive analysis assumes a worst case input in future and attempts to safeguard itself against the worst possible scenario. This leads to a conservative approach, i.e., in term of online conversion algorithms, the competitive analysis omits the risk. The approach is thus based on risk-mitigation rather than risk-management. But in a real world the player (investor) is more interested to manage the risk rather than mitigate it and the classical worst case competitive analysis fails to address it. Al-binali [4] extended the classical competitive analysis to incorporate risk and reward. The approach enables the player to manage her risk level. In the following we discuss how competitive analysis is extended to incorporate risk management.

Risk Reward Framework

In order to address the risk management aspect of online conversion algorithms, Al-binali [4] extended the classical competitive analysis approach to a risk-reward framework. Al-binali argued that algorithms developed under the competitive analysis paradigm are based on risk-mitigation principle and are therefore risk averse in nature [4]. For a decision making problem such as currency conversion, there are two possible ways, to take i)a risk-free decision and ii)a risky decision. Former leads to a fixed outcome where as the outcome of the latter varies. In latter, the player can either achieves a better performance than risk-free decision if the outcome of the risk is positive, otherwise a a sub-optimal performance than guaranteed by risk-free decision is observed. Fig. (2.1) is a schematic representation of the possible outcomes. In terms of competitive ratio, the risk-free decision will achieve an optimal worst case competitive ratio. For the risky decision, the achieved competitive ratio can either be better (lower) than the optimal competitive ratio (when the outcome is positive) or a sub-optimal competitive ratio is achieved (when the outcome is negative). Consider an online algorithm A for

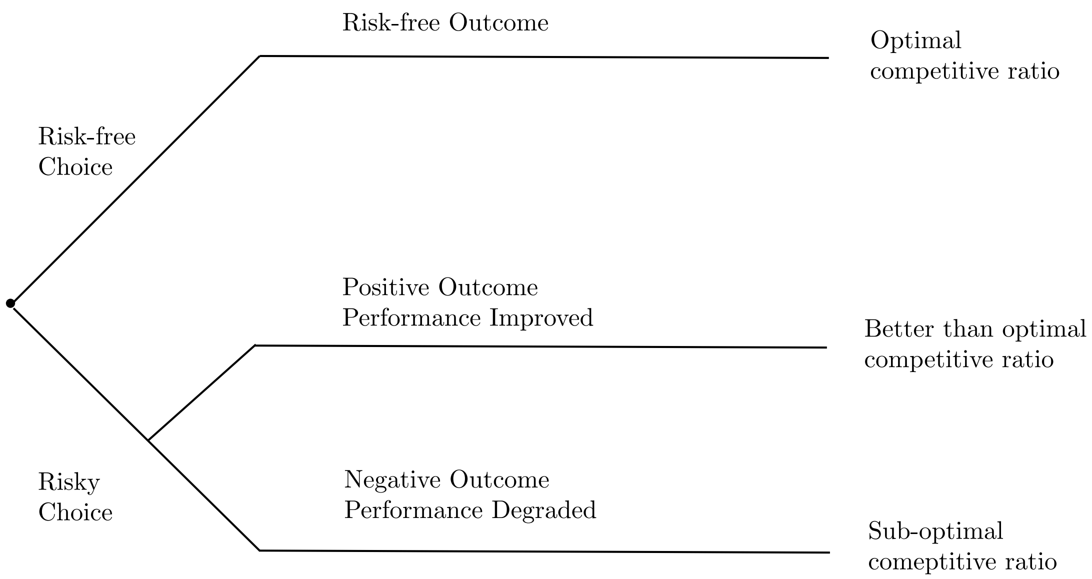


Figure 2.1: Schematic view of risk reward framework [4]

a profit maximization problem P. For the set of all problem instances $\mathcal{I}$, the competitive ratio achieved by A is;

$$c_A = sup_{I \in \mathcal{I}} \frac{OPT(I)}{A(I)}, \tag{2.2}$$

where OPT is optimum offline algorithm. The optimal competitive ratio for the same problem P is;

$$c^* = \inf_A c_A. \tag{2.3}$$

Now we define *risk*, *forecast* and *reward* - the three pillars of Al-binali's framework.

Definition 2.2 *Risk*
Risk is the maximum opportunity cost that algorithm A may occur over the optimal online algorithm, i.e.,

$$R = \frac{c_A}{c^*}. \tag{2.4}$$

Definition 2.3 *Forecast*
A forecast is a subset of problem instances, i.e., $F \in \mathcal{I}$. An example of forecast is that after time t^* the offered price will be at least M'.

A forecast $f \in F$ can either be true or false. A true forecast results in a positive outcome and a false forecast results in a negative outcome (see Fig. 2.1). The reward is based on the outcome of the forecast. If the forecast is true only then we can calculate the reward [4].

Definition 2.4 *Reward*
Reward of A is an improvement in competitive ratio over the optimal competitive ratio, i.e.,

$$f_A = \frac{c^*}{\hat{c}_A}. \tag{2.5}$$

Note that $\hat{c}_A$ is the competitive ratio achieved by A when the forecast is true. Let $\sigma \in F$ be the input instances where the forecast is true, then;

$$\hat{c}_A = \sup_{\sigma \in F} \frac{OPT(\sigma)}{A(\sigma)}.$$

Recall that α is the risk level of the player, we define a set of algorithm S_α that respects the player's level of risk as;

$$S_\alpha = \{A | c_A \leq \alpha c^*\}.$$

In other words, S_α is the set of risk aware algorithms that achieve a competitive ratio no worse than αc^*. The objective is to design online risk aware algorithms where the player is assumed to have knowledge of the forecast F. The restriction is that any risk-aware algorithm $\hat{A} \in S_\alpha$ with a risk level α must achieve a competitive ratio no worse than αc^* (even if the forecast is false).

Bijective Analysis

In competitive analysis, we compare the performance of an online algorithm against an optimum offline algorithm. Online algorithm does not have any knowledge about the future inputs whereas optimum offline algorithm knows the entire input sequence in advance. The decision of the offline algorithm is always optimum for the conversion problem. Therefore, comparing an online algorithm against an optimum offline algorithm produces highly skewed results. Competitive ratio is of minimal help when comparing two algorithms A and B that have the same worst case competitive ratio. In addition, while comparing A and B, it is redundant to compare A and B with optimum offline algorithm. Bijective analysis [6] omits the need of an optimum offline algorithm and provides a mechanism for the direct comparison of A and B. The intuition behind bijective analysis is to compare the performance of two algorithms A and B on all possible inputs of a specific length. This approach does not rely on a single worst case but considers all inputs of a certain length. Let $\mathcal{I}_T$ be the set of all input sequences of length T. We use the formal definition given by Angelopoulos et al. [6] modifying it for a maximization problem.

Definition 2.5 *Bijective Analysis*
An online algorithm A is no worse than an online algorithm B if there exists an integer $n \geq 1$ such that for each $T \geq n$, there is a bijection $b : \mathcal{I}_T \leftrightarrow \mathcal{I}_T$ satisfying $A(I) \geq B(b(I))$ for each $I \in \mathcal{I}$. This is denoted by $A \preceq_b B$, otherwise $A \npreceq_b B$. Likewise A and B are equal under bijective analysis if $A \preceq_b B$ and $B \preceq_b A$. This is denoted by $A \equiv_b B$. A is better than B if $A \preceq_b B$ and $B \npreceq_b A$ denoted by $A \prec_b B$.

Random Order Analysis

Kenyon [41] proposed a new analysis tool for online algorithms called *random order analysis*. The analysis method is proposed to address the drawbacks of the worst case competitive analysis and average-case analysis. Kenyon argued that worst case sequences are very "contrived" i.e., they are artificially created and do not represent the real world inputs. Kenyon [41] considered the *bin packing* problem and remarked that the performance of *Best-Fit* algorithm is much better than given by the worst case competitive ratio. Thus the worst case competitive analysis is too strict of a measure to replicate real world. Similarly average-case analysis is criticized for being too lenient and hugely dependent on particular distribution. Real life distributions are not as "nice" as the uniform distribution [41] and therefore the results of average-case analysis are skewed against real world. Random order analysis computes the average performance behavior of online algorithms by considering the expected results of a random ordering of an input sequence. The result obtained is compared to the performance of OPT on the same input sequence. For online conversion problem, we present the definition as given by Boyar et al. [14].

Definition 2.6 *Random Order Analysis*
The random order ratio $RC(A)$ of an online conversion algorithm A is

$$RC(A) = \limsup_{n\to\infty} \frac{OPT(I)}{E_\sigma[A(\sigma(I))]}, \tag{2.6}$$

where $\sigma(I)$ is the permuted list.

The Diffuse Adversary Model

In order to avoid the pitfalls of Bayesian analysis, competitive ratio circumvent the need of known distribution of input. Koutsoupias and Papadimitriou [42] argued that by omitting the reliance on known distribution and absolutely no knowledge of input, competitive analysis takes the the argument way too far. They criticized the assumption of competitive analysis of "knowing nothing" by calling it far from reality. The argument is based on the fact that in practice, we know or can learn "something" about the distribution. Therefore such powerlessness is unrealistic [42].

Koutsoupias and Papadimitriou [42] presented the *diffuse adversary model*. In relation to competitive analysis, the presented model removes the assumption that nothing is known about the distribution. At the same time, it also avoid the pitfall of knowing everything about it. The diffused adversary selects a probability distribution δ from a family of distributions Δ. Δ represents all acceptable probability distributions and is not restricted to a specific probability distribution (as in average-case analysis). An algorithm A is c-competitive against a known Δ class of input distributions such that

$$\mathbb{E}_\delta[A(I)] \geq \frac{1}{c}\mathbb{E}_\delta[OPT(I)]. \tag{2.7}$$

Other Measures

Beside the discussed measures for the design and analysis of online algorithms, a number of other measures are discussed in the literature as well. The complete review is beyond the scope of this work. Some of these measures are *Smoothed Complexity* [63], *Relative Worst Order Analysis* [13], *Relative Interval Analysis* [23] and *the Max/Max Ratio* [11].

2.1.2 Experimental Evaluation

Theoretical measures provide a good indication about the performance level of an algorithm. As these measures are based on solid mathematical foundations, they are valid under the assumed settings. For example, if an online maximization algorithm is shown to be c-competitive under the worst case competitive analysis, it is guaranteed to return at least $1/c$ of the total return of the optimum offline algorithm on all input instances. However, the main purpose of algorithm design and analysis is to facilitate the real world applications. For example a scheduling algorithm is of little use if it does not work in the real world scenario. Further, the design of algorithms is often based on simplistic assumptions which may or may not hold entirely true in real world. For instance, for online uni-directional conversion problems, it is often assumed that there is no transaction cost [27]. In addition, the player is also assumed to have knowledge about the lower and upper bound of offered prices [25, 27]. These assumptions are very unrealistic as in real world every transaction has an associated cost. Similarly, the exact knowledge of lower and upper bounds of prices is also not realistically possible. Therefore, it is possible that an online algorithm which is guaranteed to be c-competitive in the worst case scenario achieves an experimental case competitive ratio worse than c. In uni-directional conversion problem, this is possible in scenario when the player has error-prone estimates of lower and upper bounds of prices. Therefore, experimental evaluation is necessary to evaluate the algorithms in a real world where the settings can deviate to a greater extent than the assumed settings in theory.

Beielstein et al. [9] observed that algorithms are mathematical abstractions and only a formal approach for the analysis of algorithms is appreciated in computer science community. The emphasize is more on the theoretical analysis of algorithms and theoretical findings are not often tested with experimental results [9]. Beielstein et al. [9] put forth the idea that theoretical and experimental results can co-exist and can inspire each other. The suggested approach is useful in many ways, for instance an experimental evaluation of algorithms will highlight the gap between theory and practice. In addition, the results of the experimental approach can show that not all assumptions are valid and can motivate the design of algorithms to address the needs of a real world. Thus a dual approach of theoretical analysis followed by experimental evaluation will ensure that algorithms are properly tested both on standard mathematical models as well as in the real world scenarios.

There are a variety of methods to evaluate the algorithms via experimental methods. In the following, we discuss some of the important techniques in relation to online conversion algorithms.

Backtesting

Backtesting refers to testing the algorithms on real world historical data set. The idea is to perform the experiments on the real world data from the past and observe the performance of algorithm [49, 53]. The concept is based on the premise that if an algorithm has performed well on the past day, it is more probable that the same algorithm will perform better in the future as well. However, a good performance on past data is not a guarantee of future success [53].

A number of studied evaluated the performance of online algorithms for conversion problems using the backtesting approach. Schmidt et al. [59] evaluated two online algorithms (namely reservation price policy [25] and threat based algorithm [27]) on the real world data of $DAX30$ (1998-2007). The online algorithms are compared against optimal offline algorithm, average price trading policy and buy-and-hold. The study focused on the experimentally achieved competitive ratio of online algorithms. However, the study is restricted to the real world data only and no experiments are conducted on synthetically produced data sets. The work assumes no transaction cost which is one of the main limitation. In addition, the study does not record the number of transactions as well.

Mohr and Schmidt [51, 52] compared the performance of reservation price algorithm (called as *Market Timing*) against buy-and-hold, rand (a randomized algorithm [51, 52]) and optimum offline algorithm (called as *Market*). The study is conducted on $DAX30$ for the year 2007 only (1.Jan.2007 to 31.Dec.2007). The studies considered transaction cost of 0.0048% of the market value. The drawback of the studies is that only the real world data is considered (no synthetic data) and the number of algorithm are very limited as well.

Iqbal et al. [37] answered the question "*Can online trading algorithms beat the market?*". The purpose of the study is to investigate if the online trading algorithms can outperform the benchmark algorithm buy-and-hold. The study considers both non-preemptive and preemptive algorithms and execute them on the real world data of $DAX30$ over ten years of prices (1.Jan.2001 to 31.Dec.2010). The set of preemptive algorithms include reservation price policy [25], online difference maximization [39] and the benchmark buy-and-hold. For preemptive algorithm, the authors chose the threat based policy [27], multi-reservation price policy [47] and dollar average strategy. A transaction cost of 0.0048% of volume traded is considered. The selected set of algorithms are evaluated on a number of criteria which include geometric returns, average period returns and experimentally achieved competitive ratio. The study also focused on recording the number of transactions. It was found that for DAX30 data set, the online algorithms were able to the beat market. However, the study is limited to only DAX30 and no synthetic data was considered.

Other miscellaneous studies includes Chen et al. [16] and Hu et al. [36]. Both of these studies are limited to comparing their proposed online algorithms with buy-and-hold only. Beside these studies, a more in depth analysis of *heuristic trading algorithms* is performed in literature. However, we limit our focus to online conversion algorithms under competitive analysis, therefore, the details of heuristic trading algorithms are beyond the scope of this work. We refer the reader to [15, 44, 58, 60, 66].

The experimental evaluation of online trading algorithms suffers from a major drawback of "data snooping". Data snooping is the phenomenon of using a given set of data more than once to make deductions or inferences [64]. In their seminal work, Brock et al. [15] has warned of the danger of data snooping. All the experiments performed in [37, 51, 52, 59] uses the same data set ($DAX30$). The data set slightly changes with Iqbal et al. [37] considers ten years data from 1.Jan.2001 to 31.Dec.2010, where as Schmidt et al. [59] considers $DAX30$ data from 1.Jan.1998 to 31.Dec.2007. The studies [37, 51, 52, 59] do not use any other data sets to validate their findings. Similarly no synthetically produced data is used for evaluation. The reliance on single data set and the lack of synthetic data are, therefore, the main drawbacks of these studies.

The experimental analysis has mostly focused on bi-directional conversion (or trading) and there is no evidence of any study related to the experimental evaluation of uni-directional conversion problem. Although, bi-directional conversion might seem just an extension of uni-directional conversion, the experimental studies mostly used uni-directional conversion algorithms for testing trading strategies. For example, the reservation price algorithm of El-Yaniv [25] is specifically design for uni-directional conversion, however it is used in [37] and [59] for trading purposes.

The Bootstrap Method

As the real world data alone is not sufficient to evaluate online algorithms via experimental analysis, the need for synthetic data arises. The synthetic data can be useful in situations where the original real world data is not sufficiently large and/or we want to avoid the 'data snooping' pitfall. *Bootstrap* procedure is one of the techniques to produce synthetic data [45]. The procedure considers an original sample of data and and generates new synthetic data from the original sample. The number of required samples is user dependent, however, the size of each sample is the same as that of the original sample.

We employ "*Block Bootstrap (BB)*" technique in Ahmad and Schmidt [2] and limit our review of bootstrapping techniques to it only. For other variant of bootstrapping, we refer the reader to [12, 24, 31, 46, 56, 68]. We use the definition of Lahiri [45] to define BB. Let $\mathbb{X}_n = \{X_1, X_2, \ldots, X_n\}$ be the original time series.

Definition 2.7 *Block Bootstrap*

Let l be the length of the expected block size for the block boot strap method, such that $1 < l < n$. Given $\mathbb{X}_n$, a new time series $\{X_{0i}\}_{i\geq 1}$ is generated by periodic extension where for $i \geq 1$, $X_{0i} = X_j$ if $i = mn + j$ for some integers $m \geq 0$ and $1 \leq j \leq n$.

Let $\mathcal{B}(i,k) = (X_{0i}, \ldots, X_{0,i_k-1})$, $i \geq 1, k \geq 1$ defines the blocks of length k on the time series $X_{01}, X_{02}, \ldots$. Different versions of block bootstrap methods can be obtained by re-sampling from all observable blocks $\{\mathcal{B}(i,k) : i \geq 1, k \geq 1\}$.

Bootstrap of time series data posses a far greater challenge than ordinary data [61]. For instance, unlike other data, time series data has serial dependence, i.e., observation X_t is co-related with X_{t-1} and X_{t+1}. Therefore, considering each observation X_t independently may produce erroneous data samples. Consider the example of daily exchange rate offer over a period of one year. Assume that the minimum exchange rate during the time period is $m = 10$ and the maximum exchange rate during the same time period is $M = 100$. Taking each observation independently may result m followed by M or vice versa, which is highly unlikely in a real world scenario. Therefore, instead of individual observations, a continuous block of observations are drawn in order to preserve the underlying relationship between data [61].

Extreme Value Theory (EVT)

Scenario generation (or test case generation) is a critical issue for evaluation of algorithms. The scenario generation must satisfy (almost) all the possible situations. For example, for a trading algorithm, the scenario generation must encapsulate not only the returns of normal trading days but that of the rare events such as the days where a player achieves higher returns than the average as well as where the player observes far greater losses than expected.

Financial markets are highly volatile and exhibit an uncertain and unpredictable behavior. The uncertain and volatile behavior makes it challenging to model the behavior of these markets. Therefore, the quest for identifying the exact modeling of stock price has produced a set of different results [19]. The Black-Scholes model is among several such attempts. The model is based on the normal distribution assumption of returns and a Geometric Brownian Motion for the stock price. The underlying assumption of normal distribution has left an inherent flaw in the Black-Scholes model . The assumed distribution omits the modeling of rare events such as the possibility of greater losses and stock market meltdown. In order to encapsulate the rare extreme events, "Extreme Value Theory" (EVT) is used in literature [19]. EVT has several applications in the fields of hydrology, meteorology,geology, insurance, finance, structural engineering, telecommunications and bio-statistics [10, 22]. In financial markets EVT can be use to model the extreme nature such as the behavior of M and m, referred to as tails of the distribution [8]. In order to find the correct limit distribution of the lower and upper bounds of prices in online conversion problem, two classes of EVT are used, namely generalized extreme value (GEV) distribution [38] and generalized Pareto distribution (GPD) [55]. GEV encapsulates three standard extreme value distributions: Gumbel (type I), Fréchet (type II) and Weibull (type III). For online conversion problems in domain of finance such as currency conversion GEV is preferred to GPD for a number of reasons, including the availability of the minimum and maximum prices as the only reliable measurements. Other factor includes the better computational cost of GEV [22]. The better computational cost is valuable as a high number of time series has to be analyzed in order to avoid the data snooping bias. Let $\mathbb{X} = \{x_1, x_2, \ldots, x_n\}$ be sample data of size n. We present a basic set of definitions used later to describe the different types of distributions.

Definition 2.8 *Arithmetic Mean*
Arithmetic mean is computed by first summing all elements of a times series and dividing it by the number of observations. Let μ represents the arithmetic mean, then

$$\mu = \frac{1}{n}\sum_{t=1}^{n} x_t. \tag{2.8}$$

Definition 2.9 *Standard Deviation*
Standard deviation refers to the average variation or dispersion from the arithmetic mean. Let σ be the standard deviation, then

$$\sigma = \sqrt{\frac{1}{n}\sum_{t=1}^{n} (x_t - \mu)^2}. \tag{2.9}$$

Definition 2.10 *Skewness*
Skewness describes the symmetry of a probability distribution. Let γ be the skewness of a probability distribution, then

$$\gamma = \frac{n}{(n-1)(n-2)}\sum_{t=1}^{T}\left(\frac{x_t - \mu}{\sigma}\right)^3. \tag{2.10}$$

Definition 2.11 *Kurtosis*
Kurtosis describe the "peakedness" of a probability distribution. Let β represents the kurtosis, then

$$\beta = \left[\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{t=1}^{n}\left(\frac{x_t - \mu}{\sigma}\right)^4\right] - \frac{3(n-1)^2}{(n-2)(n-3)}. \tag{2.11}$$

The three different types of GEV distribution can be distinguished from one other based on the value of μ (location parameter), σ (scale parameter) and γ (shape parameter), using;

$$G(x;\sigma,\gamma,\mu) = \begin{cases} \exp\left(-\{1+\gamma\frac{x-\mu}{\sigma}\}^{-1/\gamma}\right), & \text{if } 1+\frac{x-\mu}{\sigma} > 0, \gamma \neq 0, \\ \exp\left(-\exp\{-\frac{x-\mu}{\sigma}\}\right), & \text{if } x \in \mathbb{R}, \gamma = 0. \end{cases} \tag{2.12}$$

Fixing $\mu \in \mathbb{R}$ and $\sigma > 0$, we can differentiate between different types of GEV distribution based on the value of γ. $\gamma = 0$ corresponds to type I distribution, $\gamma > 0$ corresponds to type II and $\gamma < 0$ corresponds to type III distribution.

Synthetic data can be produced by analyzing the given data set and using EVT to estimate μ, σ and γ. The weighted movement method of Hosking et al. [35] or likelihood-based procedures [18] can be used to estimate the parameters. Using the estimated parameters, the synthetic data (test instances) can be generated for the required distribution type (type I, II or III) [1].

2.2 Risk Measures

Risk can be defined as the possibility that the actual outcome (or return) of a strategy may be different from the expected outcome (return). Online algorithms for conversion problems are designed based on unknown future and thus the decisions are based on uncertainty in the future ahead. The online algorithms under competitive analysis are designed to avoid the risk taking aspect of decision making and provide a guaranteed solution in unforeseen scenarios. However, in real world, risk is an unavoidable phenomenon and investors want the flexibility to measure and manage their risk. A variety of measures exist in the literature to measure the risk in decision making such as "*Standard deviation*", "*Value at Risk*" and "*Conditional Value at Risk*" et cetera.

2.2.1 Standard deviation

Standard deviation measures how much the actual return can vary from the expected returns? Standard deviation can be calculated based on the past observed performance or using the probable future returns. If standard deviation is based on the historical returns over a specific period of length, we find the mean of the observed return and then calculate the average deviation from the mean. The resultant value reflects the possibility of future returns deviation from the expected returns. An alternate approach to measure the risk in form of standard deviation

[1]For implementation purposes, the 'fExtreme' package of statistical software R (www.r-project.org) can be use to generate test instance of a specific distribution type (I,II,III)

is based on the probabilities of future returns. We elaborate this with the help of an example.

Example 2.1 *Consider an asset "A" has the following possible returns in the future with the associated probabilities. Assume that the expected return $E[R]$ is 20%.*

Block i	Probability ρ_i	Return R_i
1	10%	-5%
2	15%	10%
3	25%	15%
4	30%	25%
5	15%	30%
6	5%	40%

Table 2.1: Probabilities of expected returns

Standard deviation (σ) is calculated by taking square root of the variance (V). We calculate the variance using the following formula.

$$V = \sum_{i=1}^{N} \rho_i \left(R_i - E[R]\right)^2 \tag{2.13}$$

Where;
N = The number of blocks.
ρ_i = The probability of return of block i.
R_i = Return in block i.
$E[R]$ = Expected return of the algorithm.
The standard deviation (σ) is calculated by taking the square root of variance, i.e.,

$$\sigma = \sqrt{V} \tag{2.14}$$

Using Eq. (2.13), we calculate the variance as following;

$$\begin{aligned} Var =& 0.1(-0.05 - 0.2)^2 + 0.15(0.1 - 0.2)^2 + 0.25(0.15 - 0.2)^2 \\ &+ 0.30(0.25 - 0.2)^2 + 0.15(0.30 - 0.2)^2 + 0.05(0.4 - 0.2)^2 \\ &= 0.012625 \end{aligned}$$

The standard deviation σ is

$$\sigma = \sqrt{V} = \sqrt{0.012625} = 0.112361$$

2.2.2 Value at Risk (VaR)

Standard deviation provides a simple and elegant way to measure the risk; however, the definition of the standard deviation has a basic problem. Standard deviation is a measure of the average distance of each observation from the mean of the observations. If an asset return is increasing over time or jumps to a higher level, it negatively affects the associated risk of the asset. Although investors will be happy to observe the sudden increase in returns, the standard deviation of the asset increases and thus the potential risk in term of standard deviation. Value at Risk (VaR) is used as an alternative measure to measure the risk. Let $\mathbb{X}$ be the financial position of an asset. The VaR at $\alpha\%$ confidence level is defined as the risk of $\mathbb{X}$ as the amount that can be lost with probability no more than $\alpha\%$ over a given time period of fixed length [43]. For a general representation, we use the representation given in [43]. Let α be the confidence level, VaR as the α-quantile of the probability distribution $F_{\mathbb{X}}$ of $\mathbb{X}$ as;

$$VaR_\alpha(\mathbb{X}) = -\inf\{z|\mathbb{P}\{\mathbb{X} \leq z\} > \alpha\}. \tag{2.15}$$

Despite its widespread use in financial institutions [34], VaR suffers from a number of short comings. From methodological point of view, VaR does not consider losses beyond the α-quantile level [43]. Similarly, $VaR_\alpha(\mathbb{X})$ is discontinuous with respect to α, i.e., a small change in α results significant variation in the risk estimates of VaR. A major drawback of the VaR is its inability to conform with the fundamental risk principle of *diversification* [7].

2.2.3 Conditional Value at Risk (CVaR)

In order to address the drawbacks of VaR, an alternate measure of risk called *conditional value at risk* ($CVaR$) is used. $CVaR$ reports the expected return in the worst $\alpha\%$ of the cases. For a random payoff $\mathbb{X}$ (having a continuous distribution), $CVaR$ is defined as [43];

$$CVaR_\alpha(\mathbb{X}) = -E[\mathbb{X}|\mathbb{X} \leq -VaR_\alpha(\mathbb{X})]. \tag{2.16}$$

where α is the confidence level and $VaR_\alpha(\mathbb{X})$ is the variance of $\mathbb{X}$ with confidence level α. In other words, we can say that $CVaR_\alpha(\mathbb{X})$ is the conditional expectation losses that exceed the $VaR_\alpha(\mathbb{X})$. $CVaR$ is also known as *expected shortfall* (ES), *average value at risk* ($AVaR$) and *expected tail loss* (ETL).

Example 2.2 *Consider Table 2.2 as possible returns of an asset after a fixed time period. Table 2.3 is the resultant table of $CVaR$ for different confidence level α.*

Probability ρ_i	Return R_i
5%	-100%
15%	0%
30%	20%
20%	30%
15%	40%
10%	45%
5%	55%

Table 2.2: Probabilities and expected returns

α	$CVaR_\alpha$	Calculations
10%	-50%	$[0.05(-100) + 0.05(0)] / 0.1$
25%	-16%	$[0.05(-100) + 0.15(0) + 0.05(20)] / 0.25$
35%	-5.715%	$[0.05(-100) + 0.15(0) + 0.15(20)] / 0.35$
60%	6.667%	$[0.05(-100) + 0.15(0) + 0.30(20) + 0.1(30)] / 0.60$
100%	0.25%	$[0.05(-100) + 0.15(0) + 0.30(20) + 0.2(30)$ $+0.15(40) + 0.10(45) + 0.05(55)] / 1.0$

Table 2.3: Calculating CVaR

2.2.4 Coherent Measure of Risk

Markowitz introduced the notion of measuring the risk of an investment by calculating the deviation from the mean of the distribution [65]. The idea lead to the development of new risk measures such as VaR. However, VaR and other measures of risk have certain limitations and were at odds against the well-established principles of risk. For example, VaR does not adhere to the basic risk principle of *diversification*. Diversification refers to the concept of investing in a portfolio of assets rather than a single asset. The whole wealth is thus invested in more than one assets and the combined risk level is reduced. VaR also does not take into account any risk beyond the α quantile. Further, the risk models developed under the Markowitz risk-reward framework are application driven, resulting in some risk models lacking the fundamental features of risk paradigm [43]. In order to standardize Artzner et al. [7] presented an axiomatic set of rules that a measure of risk must satisfy to be in line with the basic risk paradigm. Any measure of risk that satisfy the Artzner et al. [7] axiomatic definition is called as *Coherent Measure of Risk*.

Artzner et al. [7] used an axiomatic approach to determine a set of properties that a "good" measure of risk must satisfy [43, 65]. Let, X be any given (or desired) position of an investor when he invests in a certain asset (or portfolio of assets) and ρ be a risk measure. $\rho[X]$ is assigned by ρ to X and represents the risk of position X as measured by ρ. In other words, $\rho[X]$ can be interpreted as the minimum extra cash (investment) required by the investor to achieve X at some point of time in future [7].

Definition 2.12 *A risk measure ρ assigns a random variable X a non-negative real number R, i.e., $\rho : X \to R$.*

Here X is a random variable representing the position or return and viewed as an element of a linear space $\mathcal{X}$ of a measurable function, defined over an appropriate sample space. According to [3, 7, 65] a function $\rho : \mathcal{X} \to R$ is coherent measure of risk if it satisfies the following set of axioms.

Axiom 2.1 *Translation Invariance*
For all $X \in \mathcal{X}$, risk free returns r and $\alpha \in \mathcal{R}$;

$$\rho[X + r] = \rho[X] - \alpha. \tag{2.17}$$

Axiom 2.1 means that the addition of risk free investment reduces the risk. In other words, if a position X with a risk level $\rho[X] > 0$ is not acceptable, it can be made acceptable by the addition of a risk free return r such that the risk free return balances the $\rho[X]$, i.e., $\rho[X + \rho[X]] = 0$.

Axiom 2.2 *Subadditivity*
For all $X_1, X_2 \in \mathcal{X}$,

$$\rho[X_1 + X_2] \leq \rho[X_1] + \rho[X_2]. \tag{2.18}$$

Axiom 2.2 is based on the notion that "a merger does not create extra risk" [7]. Alternatively it can also be described as "diversification does not hurt". Consider a situation, where a risk measure fails to follow the subadditivity property. This means that an investor will prefer to invest in two assets separately (from two different accounts) rather than investing in a portfolio of same two assets via a single account [7].

Axiom 2.3 *Positive Homogeneity*
For all $X \in \mathcal{X}$, if $\lambda \geq 0$ then

$$\rho[\lambda X] = \lambda \rho[X]. \tag{2.19}$$

Positive homogeneity states that if a position X is changed by λ then the associated risk must also increase by the same factor, i.e., risk scales proportionally with the size of the position [48].

Axiom 2.4 *Monotonicity*
For all $X_1, X_2 \in \mathcal{X}$, if $X_1 \geq X_2$

$$\rho[X_1] \geq \rho[X_2]. \tag{2.20}$$

The axiom implies that if X_1 is greater than X_2, the associated risk of X_1 must also be greater than corresponding risk of X_2.

Based on these axioms, a number of further studied has been performed [1, 30, 32, 65]. These studies focused on the extension of Artzner et al. [7] axiomatic definition of coherent measures of risk. Föllmer and Schied [30] argued that in contrast to positive homogeneity (Axiom 2.3) of Artzner et al. [7], the risk of a position might increase in a non-linear way with the size of the position. Föllmer and Schied [30] relaxed the positive homogeneity and subadditivity constraints to a weaker property of "convexity", modeled as;

$$\rho[\lambda X + (1-\lambda)Y] \leq \lambda\rho[X] + (1-\lambda)\rho[Y] \quad \text{for } \lambda \in [0,1]. \tag{2.21}$$

Eq. (2.21) states that the risk associated with a diversified position $\lambda X + (1-\lambda)Y$ is no more than the weighted average of individual risks. Independently of [30], Frittelli and Gianin [32] also discussed the convex risk measure by weakening the axioms of coherence of Artzner et al. [7].

3

Online Algorithms for Conversion Problems

Summary of Results

A variety of online algorithms are presented in the literature to address online conversion problems. The problem is covered from different perspectives and on the basis of various objective functions. Similarly the assumptions and settings of each problem differ. For instance, some algorithms are designed based on the assumption that expected lower and upper bounds of offered prices are known to the player [27], whereas other algorithms assume the knowledge of upper bound of prices and length of the investment horizon [20]. A number of algorithms invest all wealth at one point of time [25, 39] whereas others divide the whole wealth in parts and invest little by little [16]. Another discrepancy in the existing work is the non-coherent and non standardized terminology to describe the ideas and assumptions.

In this work, we attempt to unify online algorithms for conversion problems under a unified and coherent set of nomenclature. We classify them based on the nature of search, the nature of investment strategy and the a-priori assumptions. Further, we provide a comprehensive review of the literature addressing online conversion problems. The survey of the literature covers both uni-directional (Section 3.3) and bi-directional (Section 3.4) conversion algorithms. We restrict the literature review to competitive search algorithms in the context of online financial markets. Further applications like algorithmic trading and online auctions are not considered.

Publication

I. Ahmad, E. Mohr and G. Schmidt: *Online Algorithms for Conversion Problems,* Submitted to Surveys in Operations Research and Management Science.
Status : Under first revision.

Online Algorithms for Conversion Problems

Iftikhar Ahmad[a], Esther Mohr[a,*], Günter Schmidt[a,b]

[a] *Saarland University, P.O. Box 151150, D-66041 Saarbrücken, Germany*
Phone +49-681-302-4559, Fax +49-681-302-4565
[b] *University of Capetown, Rondebosch 7701, Cape Town, South Africa*

Abstract

This paper surveys the literature devoted to online algorithms for conversion problems. We attempt to unify the terminology and the notation, while introducing the existing fruitful results. A novel unique classification scheme is provided, and existing algorithms are reviewed and classified. A number of questions that are still unanswered and require further consideration are highlighted.

Keywords: Classification Scheme, Survey, Online Conversion Problem, Online Algorithms, Competitive Analysis, Trading Algorithms

---

1. Introduction

1.1. Motivation

A great deal of literature is devoted to the study of online algorithms for conversion problems. In addressing the problem, various aspects are covered and many different settings are assumed [1]. In addition, the terminology used is not coherent and standardized. The great variety of existing algorithms, and the non-adherence to standards might lead to misconception on part of the reader. As each author assumes different problem settings, assumptions and nomenclature it is difficult to evaluate the suggested algorithms on existing methods, or to compare them on a mutual basis. We

*Principal corresponding author
Email addresses: `ia@orbi.uni-saarland.de` (Iftikhar Ahmad), `em@orbi.uni-saarland.de` (Esther Mohr), `gs@orbi.uni-saarland.de` (Günter Schmidt)

Preprint submitted to Surveys in Operations Research and Management Science January 22, 2013

attempt to unify the terminology and the notation, while introducing the existing fruitful results. We restrict the literature review to online algorithms in the context of conversion in financial markets, i.e. the search for best prices in order to buy and/or sell assets. Further applications like algorithmic trading, market clearing and online auctions are not considered; the reader is referred to [2, 3, 4]. Moreover, we limit to works analyzing an online algorithms' worst-case performance. Works considering other performance bounds are excluded; the reader is referred to [5] and the references therein.

1.2. Basic Definitions

A conversion problem deals with the scenario of converting an asset D into another asset Y with the objective to get the maximum amount of Y after time T. The process of conversion can be repeated in both directions, i.e. converting asset D into asset Y, and asset Y back into asset D.

In a typical problem setting, an investment horizon is considered and possibly divided into $n = 1, \ldots, N$ subsets. Each subset is comprised of $t = 1, \ldots, T$ data points, e.g. days. On each day t, an algorithm ALG is offered a price q_t to convert asset D into asset Y, and ALG may accept the price q_t or may decide to wait for a better price. The game ends either when ALG converts whole of the asset D into Y, or on the last day T where q_T must be accepted.

Based on the context of decision making, algorithms can broadly be classified in two categories. In an offline scenario full information about the future is assumed, and so an optimal offline algorithm (OPT) is carried out. In an online scenario at each point of time ALG must take a decision based only on past information, i.e. with no knowledge about the future. Typically, the quality of ALG is determined by the ratio between the result generated by ALG and OPT. This technique is called competitive analysis and the resulting worst-case competitive ratio c is an information theoretic measure [6].

We use the formal definition of the competitive ratio of [7, p. 104]. Let $\mathcal{P} = (\mathcal{I}, F, U)$ be a profit maximization problem, where $\mathcal{I}$ is a set of possible inputs, and for each $I \in \mathcal{I}$, $F(I)$ is the set of feasible outputs. U is a utility function such that for all I and $O \in F(I)$, $U(I, O) \in \mathfrak{R}$. Consider any algorithm ALG for problem $\mathcal{P}$. Given any input I, ALG computes a feasible output $O \in F(I)$. The return of ALG on instance I we denote by $ALG(I) = U(I, O)$. Typically, each input can be represented as a finite sequence I with $t = 1, \ldots, T$ elements, and a feasible output can also be represented as a finite sequence with T elements.

An algorithm ALG computes *online* if for each $t = 1, \ldots, T-1$, ALG must compute an output for t before the input for $t+1$ is given. An algorithm OPT computes *offline* if it can compute a feasible output given the entire input sequence I in advance. By definition, for each I the return of OPT is $OPT(I) = \sup_{O \in F(I)} U(O, I)$. An online algorithm is c-competitive (attains a competitive ratio c) if for any input $I \in \mathcal{I}$

$$ALG(I) \geq \frac{1}{c} \cdot OPT(I). \tag{1}$$

Any c-competitive algorithm is guaranteed at least the fraction $1/c$ of the optimal offline return $OPT(I)$ no matter how unfortunate or uncertain the future will be.

We limit to online conversion algorithms (ON) and consider conversion as a maximization problem, i.e. $c \geq 1$. The smaller c the more effective is ON. Based on their design pattern ON can broadly be classified as:

1. Guaranteeing conversion algorithms – developed to give a performance guarantee under worst-case conditions, and evaluated analytically using competitive analysis [8].

2. Heuristic conversion algorithms – developed to achieve a preferably high average-case performance, and evaluated experimentally mainly based on data from technical analysis [9].

Both classes work without any knowledge of future input, and only differ in their design pattern. Thus, both classes are referred to as online conversion algorithms.

The remainder of this paper is organized as follows. In the next section we provide a novel scheme to classify online algorithms for conversion problems based on the problem setting they are using, and define a standard nomenclature for the terms used in the work related. In Section 3 and 4 we introduce and classify existing results. In addition, we show how heuristic conversion algorithms can be evaluated using competitive analysis. The worst-case competitive ratio of three well known algorithms from the literature, namely *Moving Average Crossover* (MA), *Trading Range Breakout* (TRB), and *Momentum* (MM) is derived. The lower bound proof is the main technical contribution. The paper concludes presenting open questions and potential future research directions.

2. Classification Scheme

Our proposed novel classification scheme is based on two pillars, a) the nomenclature – a standardized set of definitions, and b) the classification factors – parameters that affect the class of problems. First we give some basic definitions, and second three factors relevant for our novel scheme are given.

2.1. Nomenclature

The standard nomenclature defines the terms used in relation to online conversion problems. The objective is to adhere to a standard set of definitions, and to avoid ambiguity.

i. *Transaction*: A transaction is either the selling *or* the buying of an asset.

ii. *Trade*: Each trade consists of two transactions; buying *and* selling. The number of trades is p, with $i = 1, \ldots, p$

iii. *Investment Horizon*: The total time duration in which all trades must be carried out. The investment horizon can be divided into $n = 1 \ldots, N$ subsets of length $t = 1, \ldots, T$.

iv. *Duration (T)*: The length (finite number of data points) of subset n, e.g. seconds, minutes or days in the discrete case.

v. *Threat Duration (k)*: The number of data points $k \leq T$ after which the offered price might drop to some minimum level and stays there until T.

vi. *Length (l)*: The length of an increasing sequence of prices, with $l \leq T$.

vii. *Offered Price (q_t)*: Possible transaction prices presented to the player at time t from subset n, i.e. $n = \{q_1, q_2, \ldots, q_T\}$.

viii. *Upper Bound (M)*: The (predicted) upper bound on the q_t in subset n.

ix. *Lower Bound (m)*: The (predicted) lower bound on the q_t in subset n.

x. *Fluctuation Ratio (ϕ)*: The (predicted) maximum fluctuation of the q_t in subset n, calculated by M/m.

xi. *Return Bound (b)*: A set of constants limiting the maximum fluctuation between prices, e.g. q_{t-1} and q_t.

xii. *Return Function ($f(q_t)$)*: The return r_t for accepting a price q_t is a (family of) function(s) of the price. For example accepted price minus the accumulated sampling costs for observing the q_t in subset n.

xiii. *Risk-tolerance Factor (a)*: The acceptable level of risk the player is willing to take for a higher return.

xiv. *Uni-directional search (uni)*: Searching for maximum (max-search) *or* minimum (min-search) price(s) to carry out either a selling *or* a buying transaction within one subset n.

xv. *Bi-directional search (bi)*: Searching for maximum (max-search) *and* minimum (min-search) price(s) to carry out both a buying *and* a selling transaction within one subset n, i.e. bi-directional search is synonym for trading.

xvi. *Preemptive conversion (pmtn)*: Search for more than one price in each subset n in order to convert D (Y). ON is allowed to convert sequentially in parts at different prices q_t, i.e. the whole available amount is converted 'little by little'. Typically, the number of prices considered for conversion is determined by ON. Except in one special case where ON desires to convert at a specific number of prices, denoted by u. This referred to as u-preemptive (u-*pmtn*).[1]

xvii. *Non-Preemptive conversion (non-pmtn)*: Search for one single price in each subset n in order to convert D (Y). ON is allowed to convert 'all or nothing', i.e. the whole available amount is converted at one price q_t.[2] Non-preemptive conversion is a special case of preemptive conversion.

xviii. *Amount to be converted (s_t)*: The fraction of the whole amount available to be converted at price q_t, with $0 \leq s_t \leq 1$. In the preemptive case $s_t \in [0, 1]$ while in the non-preemptive case $s_t \in \{0, 1\}$.

[1] In the work related algorithms for preemptive conversion are denoted as constant rebalancing algorithms, dollar-cost averaging or threat-based algorithms.

[2] In the work related algorithms for non-preemptive conversion are denoted as reservation price algorithms.

2.2. Classification Factors

By using the above definitions, the following factors are used to classify online conversion problems:

1. *Nature of search*

   i. *uni*: The player converts asset D into another asset Y, but conversion back from Y into D is forbidden. There is no restriction on the number of transactions.

   ii. *bi*: The player converts asset D back and forth, i.e. D into Y and Y back into D, etc. There is no restriction on the number of transactions.

2. *Nature of conversion*

   i. *non-pmtn*: Search for one single price in each subset n to convert asset D. Typically, the whole amount available is converted in one single transaction, i.e. $s_t \in \{0, 1\}$.

   ii. *pmtn*: Search for more than one price in each subset n to convert asset D. Typically, only a fraction of the whole amount available is converted in one transaction, i.e. $s_t \in [0, 1]$.

3. *Given information*
   The (collection of) possible information (about the future) given to ON a-priori. Parameters assumed to be known are

   i. upper bound M,

   ii. lower bound m,

   iii. fluctuation ratio ϕ,

   iv. duration T,

   v. threat duration $k \leq T$,

   vi. length l,

   vii. return bound b,

   viii. return function $f(q_t)$,

   ix. risk tolerance a.

Based on the three factors *nature of search|nature of conversion|given information* we classify the existing work related in the following. We give the worst-case competitive ratios c as well as the amount to be converted s_t – the essential parameters for conversion.

3. Uni-directional Search

The main focus of the work related is on uni-directional search. We relate our discussion w.l.o.g. to max-search, and classify these problems in two main categories based on the amount to be converted (s_t).

3.1. Uni-directional Non-preemptive Conversion

The player is allowed to convert D into Y in one single transaction, based on a pre-calculated reservation price (RP). ON concerning this scenario is either based on one single RP, denoted by q^*, or on a time varying RP, denoted by q_t^*. In both cases, each q_t is checked against the pre-calculated RP: If $q_t \geq (\leq)$ RP then q_t is accepted, and search is closed. Otherwise the search continues until the last price q_T. At this point, asset D must be converted at price q_T, which might be m in the worst-case.

Problems from the literature addressing the uni-directional non-preemptive scenario are discussed in the following.

3.1.1. Problem: uni|non-pmtn|M,m

[10] provide an algorithm called 'Reservation Price Policy' (RPP). Let the RP be q^*.

Algorithm 1.

Rule (1). Accept the first price greater than or equal to q^*.

Rule (2). If no $q_t \geq q^*$ *occurs, the player must accept the last price* q_T*, which is possibly* m.

Rule (2) only holds if the computed RP is too high, i.e. all $q_t < q^*$. A clever adversary with complete knowledge of the future, including the RP, can use this information making the player perform worse.

Theorem 1. *Let* M *and* m *be given. Then the worst-case competitive ratio equals [10, p. 35]*

$$c(m, M) = \sqrt{M/m}. \tag{2}$$

Proof.

Case 1: If q^* is too low, then the adversary provides an input sequence in such format that the 'maximum possible price' $q_{max} \in [q^*, M]$, and thus the player may suffer from the so called 'too early error': The player could have achieved M but gets q^* in the worst-case. The competitive ratio achieved thus will be $c_1 = M/q^*$.

Case 2: If q^* is too high, then the adversary provides an input sequence in such format that $q_{max} \in [m, q^*]$, and thus the player may suffer from the 'too late error': The player could have achieved q^*, and gets m in the worst-case. The competitive ratio achieved thus will be $c_2 = q^*/m$.

The player must choose a q^* while balancing the two errors, i.e. to ensure that

$$\begin{aligned} c_1 &= c_2 \\ M/q^* &= q^*/m \\ q^* &= \sqrt{M \cdot m} \end{aligned} \tag{3}$$

As OPT will choose M we get an overall competitive ratio of $\sqrt{M/m}$. ■

[11] extend Algorithm 1 and present an algorithm using two RP to sell a finite number of (stored) products. The values m, M and the storage limitation K are assumed to be known. Each time t the player receives one new product with price q_t, and may either sell it together with some stored products if $q_t \geq q^*$ or save it in the storage and sell none [11, p. 931].

3.1.2. Problem: uni|non-pmtn|M,T

[12] derive a q^* for Algorithm 1. The model assumes that $q_t \in [M/T, M/t]$ with $t = 1 \ldots T$ [12, p. 622]. Let $q_{max} \leq M$ be the highest price selected by the adversary. Again, if no $q_t \geq q^*$ occurs, the player must accept the 'minimum possible price' $q_{min} \geq m$ in the worst-case. This happens if the computed RP is too low, or the computed RP is too high.

Theorem 2. *Let M and T be given. Then the worst-case competitive ratio equals [12, p. 625]*

$$c(M,T) = \sqrt{T}. \tag{4}$$

Proof.

Case 1: If q^* is too high, the adversary will choose $q_{max} < q^*$. As no offered price q_t will satisfy the condition $q_t \geq q^*$ during T, the player must accept $q_{min} = M/T$ on day T in the worst-case. Thus, the competitive ratio in this case equals

$$\begin{aligned} c_1 &= \frac{q_{max}}{(M/T)} \\ &< \frac{q^*}{(M/T)}. \end{aligned} \tag{5}$$

Case 2: If q^* is too low, the adversary will offer q^* as the first price q_1. The player will accept q_1, and the game ends. Afterwards, the adversary increases the prices up to $q_{max} = M$. Thus, the competitive ratio in this case equals

$$c_2 = M/q^*. \tag{6}$$

The player must choose a q^* while balancing c_1 and c_2, i.e. to ensure that

$$\begin{aligned} c_1 &= c_2 \\ \frac{q_{max}}{(M/T)} &= M/q^* \\ q^* &= M/\sqrt{T}. \end{aligned} \tag{7}$$

As OPT will choose M we get an overall competitive ratio of $\sqrt{T}$. ■

3.1.3. Problem: uni|non-pmtn|M,m,$f_t(q')$

[13] extend Algorithm 1 by introducing sampling costs for observing prices. Let q' be the accepted price. Choosing q' results in some return, which is modeled by a family of functions $f_t(q')$. The basic assumptions of [13] are:

i. $f_t(q')$ with $t = 1, 2, \dots, T$ is continuous and increasing in q'.

ii. For any q' the return is the higher the earlier q' is accepted, as less sampling costs occur: $f_1(q') \geq f_2(q') \geq \cdots \geq f_T(q') > 0$.

iii. After accepting one specific q' the game ends.

In contrast to [10, 12] the considered RP varies with time, and thus is denoted by q_t^*.

Algorithm 2.

Rule (1). Accept the first price greater than or equal to q_t^*.

Rule (2). If no $q_t \geq q_t^*$ occurs, the player must accept the last price q_T, which is possibly m resulting in $f_T(m)$.

The RP is derived in the following. [13] focus on the case where $f_{t+1}(M) > f_t(m)$ for $t \in [1, T-1]$, because if $f_{t+1}(M) \leq f_t(m)$ the game ends on or before day t as the player achieves a return of $f_j(q_j) \geq f_t(m)$ when accepting q_j at day $j \in [1, t]$.

For $T = 1$ the unique price $q_1 = q'$ with the same return is accepted. Thus, the case where $T \geq 2$ is of main interest. For each (unknown) duration $L \in [1, T]$ let

$$Z_L = \min\left\{\left\{\max\left\{\frac{f_{t+1}(M)}{f_t(m)}, \sqrt{\frac{f_2(M)}{f_t(m)}}\right\}, t = 1, \ldots, L-1\right\}, \sqrt{\frac{f_2(M)}{f_L(m)}}\right\} \tag{8}$$

with $Z_L \geq 1$ since $f_{t+1}(M) > f_t(m)$, and $f_2(M) > f_L(m)$. Let

$$L' = \max\left\{L | L = \arg\max_{2 \leq L \leq T} Z_L\right\}. \tag{9}$$

This means that $Z_{L'} \geq Z_L$ for every $L \in [2, T]$. By definition of $Z_{L'}$ there exists a natural number x, such that

$$\begin{aligned} Z'_{L'} &= \frac{f_{x+1}(M)}{f_x(m)} \text{ for } x \leq L'-1, \text{ or} && (10) \\ Z''_{L'} &= \sqrt{\frac{f_2(M)}{f_x(m)}} \text{ for } x \leq L', \\ \text{with} & \\ Z_{L'} &= \min\left\{Z'_{L'}, Z''_{L'}\right\}. \end{aligned}$$

From (10) q_t^* is derived by the following cases:

Case 1: $Z_{L'} = Z'_{L'}$. For $t \in [1, x]$ let q_t^* either be the solution of

$$\begin{aligned} Z_{L'} f_t(q_t^*) &= f_{t+1}(M), \text{ or} && (11) \\ q_t^* &= m \text{ if no solution exists.} \end{aligned}$$

Case 2: $Z_{L'} = Z''_{L'}$. Let $t^* = \max\{t | f_{t+1}(M) \geq \sqrt{f_2(M) \cdot f_x(m)}\}$.

Case 2.1: For $\min\{t^*, x-1\} < t \leq x$,

$$q_t^* = m. \tag{12}$$

Case 2.2: For $1 \leq t < \min\{t^*, x-1\}$ let q_t^* be either the solution of

$$\begin{aligned} Z_{L'} f_t(q_t^*) &= f_{t+1}(M), \text{ or} \\ q_t^* &= m \text{ if no solution exists.} \end{aligned} \tag{13}$$

Theorem 3. *Let M, m and $f_t(q')$ be given. Then the worst-case competitive ratio equals $Z_{L'}$ [13, p. 196].*

For the proof, discussing several cases and worst-case time series, the reader is referred to [13, Section 4.2].

For the problem considering different return functions, an extension of the current work can possibly be to design randomized algorithms to achieve a smaller competitive ratio.

3.1.4. Problem: uni|non-pmtn|M,m,T,$f_t(q')$

[13] derive a second q^* for Algorithm 2 based on the additional knowledge of T. The assumptions are identical, only the calculation of the RP differs. For each (known) duration T, let

$$Z = \min\left\{\left\{\max\left\{\frac{f_{t+1}(M)}{f_t(m)}, \sqrt{\frac{f_2(M)}{f_t(m)}}\right\}, t = 1, \ldots, T-1\right\}, \sqrt{\frac{f_2(M)}{f_T(m)}}\right\} \tag{14}$$

with $T \geq 1$ as $f_{t+1}(M) > f_t(m)$ and $f_2(M) > f_t(m)$. By definition of Z there exists a natural number y, such that

$$\begin{aligned} Z' &= \frac{f_{y+1}(M)}{f_y(m)} \text{ for } y \leq T-1, \text{ or} \\ Z'' &= \sqrt{\frac{f_2(M)}{f_y(m)}} \text{ for } y \leq T, \text{ or} \\ Z &= \min\{Z', Z''\}. \end{aligned} \tag{15}$$

From (15) q_t^* is derived by the following cases:
Case 1: $Z = Z'$. For $t \in [1, y]$ let q_t^* either be the solution of

$$\begin{aligned} Z f_t(q_t^*) &= f_{t+1}(M), \text{ or} \\ q_t^* &= m \text{ if no solution exists.} \end{aligned} \tag{16}$$

Case 2: $Z = Z''$. Let $t^* = \max\{t | f_{t+1}(M) \geq \sqrt{f_2(M) \cdot f_y(m)}\}$.

Case 2.1: For $\min\{t^*, y-1\} < t \leq y$,

$$q_t^* = m. \tag{17}$$

Case 2.2: For $1 \leq t < \min\{t^*, y-1\}$ let q_t^* be either the solution of

$$\begin{aligned} Z f_t(q_t^*) &= f_{t+1}(M), \text{ or} \\ q_t^* &= m \text{ if no solution exists.} \end{aligned} \tag{18}$$

Theorem 4. *Let M, m, T and $f_t(q')$ be given. Then the worst-case competitive ratio equals Z [13, p. 194].*

For the proof, discussing several cases and worst-case time series, the reader is referred to [13, Section 3.2].

3.1.5. Problem: uni|non-pmtn|b, T

[14] use the framework of the bounded daily return model, namely return bound b, and consider a set $b = \{\Theta_1, \Theta_2\}$ assuming for each $q_t \in [1, T-1]$ the $q_{t+1} \in [q_t\Theta_1, q_t\Theta_2]$ where $0 < \Theta_1 \leq \Theta_2$. Further, [14] focus on the case where $0 < \Theta_1 < 1 < \Theta_2$. The following algorithm [14, p. 162] is given:

Algorithm 3.

Rule (1). Accept the t^*-th price with return r_{t^*}.

Rule (2). If no such price occurs, the player must accept the last price q_T, which is possibly m.

With $t^* = \arg\min_{t^* \in [1,T]} \{c_t\}$ and $c_t = \max\left\{\Theta_1^{1-t}, \Theta_2^{T-t}\right\}$ is a time-varying competitive ratio recalculated for each t.

Theorem 5. *Let Θ_1, Θ_2 and T be given. Then the worst-case competitive ratio equals [14, p. 164]*

$$\begin{aligned} c_T(\Theta_1, \Theta_2) &= \frac{\max\left\{\Theta_1^{1-t^*} r_{t^*}, \Theta_2^{T-t^*} r_{t^*}\right\}}{r_{t^*}} \\ &= \max\left\{\Theta_1^{1-t^*}, \Theta_2^{T-t^*}\right\}. \end{aligned} \tag{19}$$

For the proof the reader is referred to [14, Theorem 1].

3.2. Uni-directional Preemptive Conversion

In uni-directional preemptive conversion D can be converted sequentially in parts, i.e. $s_t \in [0, 1]$. The only restriction is that within subset n the player must convert D into Y completely, i.e. $\sum_{t=1}^{T} s_t = 1$.

A great deal of literature addresses the problem *uni|pmtn*. [15, 7, 16] introduce a genre of algorithms based on the assumption that there exists a threat that at some stage during the time interval, namely on day $k \leq T$, the offered price $q_t \in [m, M]$ will drop to some minimum level, and will remain there until the last day T. The algorithm proposed is commonly referred to as the threat-based, and the basic rules are [7, p. 109]:

Algorithm 4.

Rule (1). Consider a conversion from D into Y only when the current price is the highest seen so far.

Rule (2). Whenever converting D into Y, convert just enough D to ensure that a competitive ratio c would be obtained if an adversary dropped the price to the 'minimum possible price', and kept it there throughout the game.[3]

Rule (3). On the last trading day, all remaining D must be converted into Y, possibly at the 'minimum possible price'.

[15, 7, 16] discuss five variants of Algorithm 4, each assuming a different knowledge about the future. [15, 7] consider:

- known duration T with m and M known,
- unknown duration with m and M known,
- known duration T with known ϕ,
- unknown duration with known ϕ,

and [16] consider:

- unknown duration with initial price q_1, m and M known.

[3]The 'minimum possible price' is defined with respect to the information known to the player. Which is m if m is known and is q_t/ϕ if only $\phi = M/m$ is known, and q_t is highest price seen so far.

For each of the five variants [15, 7, 16] give a competitive ratio based on the assumed a-priori information. Based on these parameters, ON determines s_t such that c holds in the worst-case. [17] conclude that this ratio seems to be appropriate to generate a reasonable ordering of the algorithms investigating the worth of future information available.

3.2.1. Problem: uni|pmtn|M,m,k

In this variant of Algorithm 4 it is assumed that m and M, as well as $k \leq T$, are known a-priori [7].

Theorem 6. *Let m, M and T be given. Then $c_T(m, M)$ is the worst-case competitive ratio, and is the solution, c, of [7, (26), p. 118]*

$$c = T \cdot \left[1 - \left(\frac{m \cdot (c-1)}{M - m}\right)^{1/T}\right]. \tag{20}$$

For the known duration case $c_T(m, M)$ is the best competitive ratio that an *uni|pmtn* algorithm can achieve [7, Theorem 5, p. 118]. For the proof the reader is referred to [7].

Let y_t be the number of Y accumulated, and d_t the amount of D remaining after day t. ON starts with $d_0 = 1$ of D, and $y_0 = 0$ of Y. Then the amount to be converted [7, p. 111]

$$s_t = d_{t-1} - d_t \Rightarrow d_t = d_{t-1} - s_t, \tag{21}$$

and

$$q_t \cdot s_t = y_t - y_{t-1}. \tag{22}$$

From *Rule (2)* of Algorithm 4 we know [7, (5), p. 112]

$$y_t + d_t \cdot \text{`minimum possible price'} = \frac{q_t}{c}, \tag{23}$$

and

$$y_{t-1} + d_{t-1} \cdot \text{`minimum possible price'} = \frac{q_{t-1}}{c}. \tag{24}$$

For $t > 1$ by subtracting (24) from (23) and applying (21) and (22) we get [7, (10), p. 112]

$$s_t = \frac{1}{c} \cdot \frac{q_t - q_{t-1}}{q_t - m}. \tag{25}$$

3.2.2. Problem: uni|pmtn|M,m

In this variant of Algorithm 4 the number of days $k \leq T$ is not given to the player. An arbitrarily large number of days $T \to \infty$ must be considered in the worst-case. From [7, p. 121] we know

$$c_\infty(m, M) = \lim_{T\to\infty} c_T(m, M). \tag{26}$$

Theorem 7. *Let m and M be given. Then $c_\infty(M, m)$ is the worst-case competitive ratio, and is the solution, c, of [7, (29), p. 122]*

$$c = \ln \frac{M - m}{m \cdot (c - 1)}. \tag{27}$$

For the proof the reader is referred to [7]. In order to meet the ratio c the d_t must be determined such that the whole (remaining) amount of D is converted in case the highest possible price M occurs on day t. From this follows [15, p. 329, Case 1]

$$d_t = 1 - \frac{1}{c} \cdot \ln \frac{M - m}{m \cdot (c - 1)} \tag{28}$$

with $s_t = d_{t-1} - d_t$ and $d_0 = 1$.

[16] improve the lower bound given in (27). The claim is that a player using the Algorithm 4 assumes a much greater threat than actually faced by the player. It is shown that Algorithm 4 does not convert unless the price is as large as $c \cdot m$, i.e. the threat is at most $c \cdot m \geq m$, and shall not go beyond this point. Thus, from (27) we get

$$c = \ln \frac{\frac{M}{c \cdot m} - 1}{c - 1}, \tag{29}$$

and *Rule (2)* of Algorithm 4 is modified by [16]: the 'minimum possible price' is defined with respect to the information known to the player. Which is cm if m is known, i.e. replacing m by cm. For the proof the reader is referred to [16, Theorem 4].

3.2.3. Problem: uni|pmtn|M,m,q_1

In this variant of Algorithm 4 an arbitrary number of days $T \to \infty$ must be considered as $k \leq T$ is not known.

Theorem 8. *Let m, M and q_1 be given. Then $c_\infty(m, M, q_1)$ is the worst-case competitive ratio, and is the solution, c, of [16, p. 29] and [15, p. 329, Case 1 and 2]*

$$c = \begin{cases} \ln \frac{\frac{M}{m}-1}{c-1} & q_1 \in [m, c \cdot m] \\ 1 + \frac{q_1 - m}{q_1} \cdot \ln \frac{M-m}{q_1-m} & q_1 \in [c \cdot m, M]\,. \end{cases} \tag{30}$$

Hence (27) only holds for the case q_1 is unknown, or $q_1 \leq c \cdot m$. In this case the pessimistic assumption $q_1 = m$ must be made, cf. [15, p. 329].

Further, depending on the value of q_1 the amount of D remaining equals [15, p. 329, Case 1 and 2]

$$d_t = \begin{cases} 1 - \frac{1}{c} \cdot \ln \frac{q_t - m}{c \cdot m - m} & q_1 \in [m, c \cdot m] \\ \frac{q_1 - (q_1/c)}{q_1 - m} - \frac{1}{c} \cdot \ln \frac{q_t - m}{q_1 - m} & q_1 \in [c \cdot m, M] \end{cases} \tag{31}$$

with $s_t = d_{t-1} - d_t$ and $d_0 = 1$.

3.2.4. Problem: uni|pmtn|ϕ,k

In this variant of Algorithm 4 the information about the 'minimum possible price' available to the player varies online: at the t-th day the minimum possible price is $\frac{q_t}{\phi}$.

Theorem 9. *Let ϕ and T be given. Then $c_T(\phi)$ is the worst-case competitive ratio, and equals [7, Theorem 6, p. 126]*

$$c_T(\phi) = \phi \cdot \left(1 - \frac{(\phi - 1)^T}{(\phi^{T/(T-1)} - 1)^{T-1}}\right). \tag{32}$$

The worst-case competitive ratio in (32) can be derived as in the analysis of *Problem: uni|pmtn|M,m,k* by specializing to the case in which the 'minimum possible price' is $\frac{q_t}{\phi}$ [7, p. 122]. For the proof the reader is referred to [7].

3.2.5. Problem: uni|pmtn|ϕ

In this variant of Algorithm 4 an arbitrarily large number of days $T \to \infty$ must be considered as $k \leq T$ is not given. Thus, [7, p. 126] define

$$c_\infty(\phi) = \lim_{T\to\infty} c_T(\phi). \tag{33}$$

Theorem 10. *Let ϕ be given. Then $c_\infty(\phi)$ is the worst-case competitive ratio, and equals [7, p. 126]*

$$c_\infty(\phi) = \phi - \frac{\phi - 1}{\phi^{1/(\phi-1)}}. \tag{34}$$

For the proof the reader is referred to [7].

It remains to compute s_t for *Problem: uni|pmtn|ϕ,k* and *Problem: uni|pmtn|ϕ*. For both [15, 7] observe that the minimum price offered to ON on day t is at least $\frac{q_t}{\phi}$. By replacing m by $\frac{q_t}{\phi}$ from (23) we know

$$\begin{aligned} y_t + d_t\frac{q_t}{\phi} &= \frac{q_t}{c} \\ \Rightarrow d_t &= \phi(\frac{1}{c} - \frac{y_t}{q_t}). \end{aligned} \tag{35}$$

From [7, (4), p. 111], we get [7, (31), p. 122]

$$s_t = \frac{q_t - c(y_{t-1} + d_{t-1} \cdot q_t/\phi)}{c(q_t - q_t/\phi)}. \tag{36}$$

3.2.6. Problem: uni|pmtn|M(t),m,T,l

This variant of Algorithm 4 is based on the assumption that the upper bound is not constant but varies with time: $M(t) = M/t$. Let $l \leq T$ be the length of an increasing sequence of prices $m \leq q_1 < q_2 \cdots < q_l \leq M$. For example $n = \{1, 2, 4, 3, 7, 5, 6\}$ $(T = 7)$, then $l = 4$ with prices $\{1, 2, 4, 7\}$.

Theorem 11. *Let $M(t)$, m, l and T be given. Then the worst-case competitive ratio equals [12, (7), p. 663]*

$$c(M(t), m, T, l) = \max_{l=2,\dots,T} \left\{ c | c = l \left(1 - \left(\frac{c-1}{\frac{M(l)}{m} - 1} \right)^{1/l} \right) \right\} \tag{37}$$

where q_t is modeled as $m \leq q_t \leq M(t)$, and $M(t)$ is a decreasing function of time (constant m).

Let s_t be the amount to be converted at time t, then [12, (9), p. 634]

$$s_t = \begin{cases} \frac{1}{c}\left(\frac{q_t - cm}{q_t - m}\right) & t = 1 \\ \frac{1}{c}\left(\frac{q_t - q_{t-1}}{q_t - m}\right) & t \in [2, T]. \end{cases} \tag{38}$$

3.2.7. Problem: $uni|pmtn|b, T$

[18] use the framework of the bounded daily return model, namely return bound b, and assume a set $b = \{\alpha, \beta\}$. The two fixed constants α and β ($\alpha, \beta > 1$) determine the range of q_t with $q_t \in \left[\frac{q_{t-1}}{\beta}, \alpha \cdot q_{t-1}\right]$. The following buy-and-hold algorithm is given [18, p. 448]:

Algorithm 5.

Rule (1). For each observed q_t convert D into Y according to $s_t \in [0, 1]$.

Rule (2). On the last trading day, all remaining D must be converted into Y, possibly at m.

Based on a worst-case input sequence scenario [18, p. 543]

$$s_t = \begin{cases} \frac{\alpha(\beta-1)}{T\alpha\beta - (T-1)(\alpha+\beta) + (T-2)} & t = 1 \\ \frac{(\alpha-1)(\beta-1)}{T\alpha\beta - (T-1)(\alpha+\beta) + (T-2)} & t \in [2, T-1] \\ \frac{\beta(\alpha-1)}{T\alpha\beta - (T-1)(\alpha+\beta) + (T-2)} & t = T. \end{cases} \tag{39}$$

Theorem 12. *Let α, β and T be given. Then the worst-case competitive ratio equals [18, p. 454]*

$$c_T(\alpha, \beta) = \frac{T \cdot \alpha\beta - (T-1)(\alpha+\beta) + (T-2)}{\alpha\beta - 1}. \tag{40}$$

For the proof the reader is referred to [18, Theorem 3.4].

An open question is to replace the constants α and β by a (known) probability distribution. Further, the scenario of a continuous cash (wealth) flow could be investigated.

[19] use Algorithm 5 but consider one fixed constant $\gamma < 1$, i.e. set $b = \{\gamma\}$. For example if $\gamma = 0.1$ the maximum possible price change is

10%. Here, each current price q_t depends on yesterday's days price q_{t-1} where $q_t \in [(1-\gamma)q_{t-1}, (1+\gamma)q_{t-1}]$. [19, p. 225] present two strategies to determine s_t, namely the *Static Mixed Strategy* and the *Dynamic Mixed Strategy*.

i. Static Mixed Strategy: Based on a worst-case input sequence scenario [19, (9), p. 228]

$$s_t = \begin{cases} \frac{1+\gamma}{(T-1)\gamma+2} & t = 1 \\ \frac{\gamma}{(T-1)\gamma+2} & t \in [2, T-1] \\ \frac{1}{(T-1)\gamma+2} & t = T. \end{cases} \tag{41}$$

with $\sum_{t=1}^{T} s_t = 1$.

Theorem 13. *Let γ and T be given. Then the worst-case competitive ratio equals [19, p. 227]*

$$c_T(\gamma) = 1 + \frac{\gamma}{2}(T-1). \tag{42}$$

For the proof the reader is referred to [19, Theorem 1].

ii. Dynamic Mixed Strategy: Based on the remaining number of days $T' = T - t + 1$ in subset n [19, (12), p. 229]

$$s_t = \begin{cases} \left(\frac{1+\gamma}{(T'-1)\gamma+2}\right) d_{t-1} & t = 1 \\ \left(\frac{\gamma}{(T'-1)\gamma+2}\right) d_{t-1} & t \in [2, T'-1] \\ \left(\frac{1}{(T'-1)\gamma+2}\right) d_{t-1} & t = T' \end{cases} \tag{43}$$

where d_{t-1} the amount of D (capital) remaining after day $t-1$ with initial amount $d_o = 1$ and $\sum_{t=1}^{T} s_t = 1$.

Theorem 14. *Let γ and T' be given. Then the worst-case competitive ratio equals [19, p. 229]*

$$c_{T'}(\gamma) = 1 + \frac{\gamma}{2}(T'-1). \tag{44}$$

For the proof the reader is referred to [19, Theorem 2]. Note that (44) only holds when T is not extended to infinity. Therefore, a strategy for $T \to \infty$ is an open question. [20] study the same setting in a framework with commission and interest rate, and [21] conversion against a weak statistical adversary.

Remark 1. *Under worst-case assumptions the competitive ratios $c_T(\alpha, \beta)$ (of [18]) and $c_T(\gamma)$ (Static Mixed Strategy of [19]) are identical.*

Using the bounded daily return model [18] assume

$$\frac{q_{t-1}}{\beta} \leq q_t \leq \alpha q_{t-1}, \tag{45}$$

whereas [19] assume

$$(1-\gamma)q_{t-1} \leq q_t \leq (1+\gamma)q_{t-1}. \tag{46}$$

Proof.
Let T and $b = \{\alpha, \beta, \gamma\}$ be given. Using elementary calculus from (45) and (46) we obtain

$$\begin{aligned} \beta &= \frac{1}{1-\gamma} \text{ and,} \\ \alpha &= (1+\gamma). \end{aligned} \tag{47}$$

By applying (47) to (40) we get

$$\begin{aligned} c_T(\alpha, \beta) &= 1 + \frac{T\gamma}{2} - \frac{\gamma}{2} \\ &= 1 + \frac{\gamma}{2}(T-1) \\ &= c_T(\gamma) \end{aligned} \tag{48}$$

where $\gamma < 1$. ∎

Further, the worst-case competitive ratio of the *Dynamic Mixed Strategy* of [19], $c_{T'}(\gamma)$, can also be represented in terms of α and β, using

$$\gamma = \frac{2(\alpha\beta - \alpha - \beta + 1)}{\alpha\beta - 1} \tag{49}$$

where $\alpha, \beta > 1$.

[14] use Algorithm 5 and consider a set $b = \{\Theta_1, \Theta_2\}$ assuming for each $q_t \in [1, T-1]$ the $q_{t+1} \in [q_t\Theta_1, q_t\Theta_2]$ where $0 < \Theta_1 \leq \Theta_2$. Further, [14] focus on the case where $0 < \Theta_1 < 1 < \Theta_2$.

Let $\frac{s_1}{\Theta_2^{t-1}} + \frac{s_2}{\Theta_2^{t-2}} + \cdots + \frac{s_{t-1}}{\Theta_2} + s_t + s_{t+1}\Theta_1 + \cdots + s_T\Theta_1^{T-t}$ be $F_t(s_1, \ldots, s_T)$. To compute s_t the following linear program with variables $\{c, s_1, \ldots, s_T\}$ must be solved for $t \in [1, T]$:

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & F_t(s_1, \ldots, s_T) \geq \frac{1}{c} \\ & \sum_{t=1}^{T} s_t \leq 1 \\ & s_t \geq 0 \end{aligned} \tag{50}$$

The optimal solution is $\{c^*, s_1^*, \ldots, s_T^*\}$, i.e. the amount to be converted by Algorithm 5 on each day t equals s_t^* [14, p. 164].

Theorem 15. *Let Θ_1, Θ_2 and T be given. Then the worst-case competitive ratio equals [14, p. 164]*

$$\begin{aligned} c_T(\Theta_1, \Theta_2) &= \frac{1}{F_t(s_1^*, \ldots, s_T^*)} \\ &= c^*. \end{aligned} \tag{51}$$

For the proof the reader is referred to [14, Theorem 3].

3.2.8. Problem: uni|pmtn|M,m,a

The threat-based algorithm is risk-averse: Algorithm 4 attempts to safe guard against a clever adversary who might drop the price to the lowest level and keep it there for the rest of the game. [22] include a flexible risk management mechanism to competitive analysis. The main idea of the proposed framework is to allow the player to manage risk for some kind of reward. Let the risk-tolerance factor $a \in [1, c]$, where a can be controlled by the player and c is the worst-case competitive ratio $c_\infty(M, m)$ of Algorithm 4 assuming M and m known, cf. (27).

A forecast is the minimum/maximum value of a future price that is *at least* expected in subset n. If, for example, the forecasted upper bound is M_1 then all possible maxima $q_{max} \in [M_1, M]$. The work of [22] limits to max-search, and is based on the simple assumption that a forecast can either be true or false.

If the forecast comes true, then a competitive ratio $c_1 = \frac{c}{a}$ better (smaller) than c is achieved. This is optimal under the following condition: If the forecast does not come true, the worst-case competitive ratio is not worse than $c \cdot a$, i.e. c holds, where $c_1 \leq c$. This results in an algorithm with bounded loss within a pre-specified risk tolerance.

Let the initial price $q_0 \geq c \cdot m$, and $M_1 \leq M$ is the possible upper bound. The algorithm proposed is commonly referred to as the risk-rewarded threat-based. *Rule (2)* of Algorithm 4 is extended by two stages [22, p. 106]:

Algorithm 6.

Rule (1). Consider a conversion from D into Y only when the current price is the highest seen so far.

Rule (2). Whenever you convert D into Y, convert just enough D to ensure that a competitive ratio c would be obtained if an adversary dropped the price to m, and kept it there throughout the game, where

$$c = \begin{cases} a \cdot c, q_t \in [q_0, M_1] & \textit{Stage (1)} \\ c_1, q_t \in [M_1, M] & \textit{Stage (2)} \end{cases} \tag{52}$$

with $c_1 \leq c$.

Rule (3). On the last trading day, all remaining D must be converted into Y, possibly at m.

In Stage (1) Algorithm 6 converts under the threat that the forecast is incorrect, i.e. converts just enough D into Y to guarantee a competitive ratio of at most $a \cdot c$; if $a = 1$ the risk-averse player achieves c of Algorithm 4. This is the conservative stage, 'saving' dollars for when the forecast comes true. Then Stage (2) begins: First, the new minimum achievable competitive ratio c_1 is computed. Second, Algorithm 6 converts just to enough ensure c_1 under the threat that the price drops to m and remains there until T. Algorithm 6 has 'more' D left to spend than Algorithm 4, and will convert at higher prices. Thus on day T, the accumulated Y are greater.

Theorem 16. *Let M,m and a be given. In case the forecast comes true, the worst-case competitive ratio equals [23, (19), p. 414]*

$$c_1 = \frac{M_1 - m}{(M_1 - m)\left(1 - \frac{1}{ac} \ln \frac{M_1 - m}{acm - m}\right) + \frac{M_1}{ac}} \left(\frac{M_1}{M_1 - m} + \ln \frac{M - m}{M_1 - m} \right). \tag{53}$$

For the proof the reader is referred to [23, Section 3.2]. Note that in case of incorrect forecasts the worst-case competitive ratio equals $a \cdot c$ with c from (27).

It remains to compute the amount to be converted s_t for Stage (1) and Stage (2). Let $t^* \in [1, T]$ be the first day where the forecast comes true. Then [22, p. 107]

$$s_t^2 = \begin{cases} \frac{d_o}{q_t - m}\left(\frac{q_t}{c_1} - \frac{q_{t-1}}{a \cdot c}\right), & t = t^* \\ \frac{d_o}{c}\frac{q_t - q_{t-1}}{q_t - m}, & \text{else with } c \text{ from (52).} \end{cases} \tag{54}$$

[23] generalize the work of [22] in two ways: *i*) [22] limits a forecast to the assumption that the price will reach at least some level. [23] also allow the opposite, i.e. for example if the forecasted lower bound is m_1 then all possible minima $q_{min} \in [m, m_1]$. *ii*) Algorithm 6 is based on the scenario where one single forecast is assumed. [23] provide a scheme which enables to include several forecasts. ON can 'update' a forecast by a second forecast, etc. Results show that the suggested algorithms are not optimal for the entire investment horizon considered but for certain subsets n. In this regard, [24] show c to be a (so far not established) coherent risk measure. Using c the risk associated can be monitored, as it measures the expected regret [25].

However, in practice, forecasts often have an associated probability ρ to come true. An open question is to represent the reward as a function of ρ.

3.2.9. Problem: uni|u-pmtn|M,m

Typically, the number of prices considered for preemptive conversion within subset n is determined by ON. [26] consider the special case where the player desires to convert u units of asset D into asset Y at u prices, referred to as u-preemptive (u-*pmtn*). The proposed algorithm is based on two pre-calculated reservation prices (RP): One for buying and one for selling. Let $q_i^* = (q_1^*, q_2^*, \dots, q_u^*)$ be the RP where $i \in [1, u]$, equaling

$$q_i^* = \begin{cases} m\left[1 + (c-1)\left(1 + \frac{c}{u}\right)^{i-1}\right], & \text{for max-search} \\ M\left[1 - \left(1 - \frac{1}{c}\right)\left(1 + \frac{1}{u \cdot c}\right)^{i-1}\right], & \text{for min-search} \end{cases} \tag{55}$$

and c equals the worst-case competitive ratio [26, (5), p. 316 resp. (8), p. 318].

Algorithm 7.

Rule (1). For each unit i accept the first price greater (smaller) than or equal to q_i^* for selling (buying).

Rule (2). On the last trading day, all remaining units of D must be converted into Y, possibly at m.

For $i \geq 1$ ON accepts u different RP which are at least q_i^*. Note that $i = 1$ is a special case where Algorithm 7 equals Algorithm 1: The first price which is at least q_1^* is accepted and the game ends.

Theorem 17. *Let m, M and $u \in \mathbb{N}$ be given with $\frac{M}{m} = \phi$. Then $c_\infty(\phi, u)$ is the worst-case competitive ratio for max-search, and is the solution, c, of [26, (1), p. 313]*

$$\frac{(\phi - 1)}{(c - 1)} = \left(1 + \frac{c}{u}\right)^u. \tag{56}$$

Theorem 18. *Let m, M and $u \in \mathbb{N}$ be given with $\frac{M}{m} = \phi$. Then $c_\infty^{min}(\phi, u)$ is the worst-case competitive ratio for min-search, and is the solution, c, of [26, (2), p. 314]*

$$\frac{\left(1 - \frac{1}{\phi}\right)}{\left(1 - \frac{1}{c}\right)} = \left(1 + \frac{1}{c \cdot u}\right)^u. \tag{57}$$

For the proofs the reader is referred to [26].

3.2.10. Problem: uni|u-pmtn|M,m,T

[27] extend the work of [26] by assuming the additional knowledge of T. Algorithm 7 is allowed to convert at most one unit at one price. In the model of [27] the player is allowed to convert up to u units at one price. The proposed (max-search) algorithm is based on the idea to first specify u, and then to determine the units $u \geq 1$ to be converted at time t. Let $q_i^* = (q_0^*, q_1^*, \ldots, q_u^*)$ be the RP where $i \in [0, u]$. Then the number of units is determined by the relationship between q_t and the matching q_i^* [27, DET, p. 679]. Let the RP be

$$q_i^* = \left[\left(1 + \frac{c}{u}\right)^{i-1}(c - 1) + 1\right], \tag{58}$$

and at time t price q_t is observed. Let the observed prices $q_t \in \left[q_i^*, q_{i+1}^*\right)$.

Algorithm 8.

Rule (1). If $q_i^* > \max\{q_1, \ldots, q_{t-1}\}$ *convert just enough units of* D *into* Y *that the accumulated amount of units converted is exactly* i*.*

Rule (2). If $q_i^* \leq \max\{q_1, \ldots, q_{t-1}\}$ *do not convert.*

Rule (3). On the last trading day, all remaining units of D *must be converted into* Y*, possibly at* m*.*

Assume Algorithm 8 has converted exactly $j \in [1, u]$ units at time $T-1$. From *Rule (1)* follows that then there has been a $q_t \in \left[q_j^*, q_{j+1}^*\right)$ while all other $q_t < q_{j+1}$, cf. [27, Table 1]. At least one of the j units is converted at a price of at least q_j^*, thus the overall amount converted is at least $\sum_{i=0}^{j} q_i^*$ for $j \leq u$ assets [27, Lemma 1].

Theorem 19. *Let* m*,* M*,* T *and* $u \in \mathbb{N}$ *be given. Then* $c_\infty(m, M, u)$ *is the worst-case competitive ratio for max-search, and is the solution, c, of [27, p. 680]*

$$c = \begin{cases} \max\left\{\frac{uq_{j+1}^*}{\sum_{i=0}^{j} q_i^* + (u-j)m}, \frac{uM}{\sum_{i=0}^{j} q_i^* + (u-j)M}\right\}, u < T \\ \min_{j \in [T-1, u]} \max\left\{\frac{uq_j^*}{\sum_{i=1}^{j} q_i^* + (u-j)m}, \frac{uM}{\sum_{i=1}^{j} q_i^* + (u-j)M}\right\}, u \geq T. \end{cases} \tag{59}$$

For the proof the reader is referred to [27, Theorem 1 and 2].

[28] investigate the practical applicability of the reviewed uni-directional algorithms of [10, 12, 7, 18, 19, 26] in an experimental study. Results show that the algorithms of [7] outperform the other algorithms on real world datasets (DAX30 and S&P500).

4. Bi-directional Search

Only a small amount of work is related to bi-directional search. The player is allowed to convert D back and forth. We assume w.l.o.g. that the objective is to maximize the amount of D at day T. We classify the problems in two main categories based on the amount to be converted (s_t).

4.1. Bi-directional Non-Preemptive

The player is allowed to convert D (Y) in one single transaction, based on a pre-calculated reservation price (RP). ON concerning this scenario are either based on one single RP, denoted by q^*, or on a time varying RP, denoted by q_t^*. In both cases, each q_t is checked against the pre-calculated

RP: If $q_t \geq (\leq)$ RP then q_t is accepted, and search is closed. This implies exactly two transactions (one single trade). Search continues until the last price q_T. At this point of time Y must be converted into D, possibly at price m. Problems from the literature addressing the *bi|non-pmtn* scenario are discussed in the following.

4.1.1. Problem: bi|non-pmtn| –

[1] address a problem closely related to the secretary problem, cf. [29, 30]. The aim is to select pairs of prices in such way as to maximize their difference in ranks. For each price a x_t $(t = 1, \dots, T)$ is computed representing the rank of q_t in the already observed sequence of prices. The x_t form a permutation of subset $\{1, 2, \dots, T\}$. The aim is to achieve a possibly high profit while maximizing the difference (in ranks) between buying and selling price(s). Two scenarios are addressed, *i*) single high/low pair selection (one trade) and *ii*) multiple high/low pair selections ($p > 1$ trades). We limit to *i*) as to solve *ii*) the algorithm for *i*) is repeated p times. A low (high) selection refers to picking a buying (selling) price.

Let $q_t^*(H)$ be the RP for high selection, equaling [1, p. 3]

$$q_t^*(H) = \left\lceil \frac{t+1}{n+1} X_{t+1}^* \right\rceil \tag{60}$$

where X_t^* is the expected final rank of a high selection if an optimal strategy for high selection is followed starting at time t [1, p. 4]

$$X_t^* = \begin{cases} \frac{T+1}{2} & t = T, \\ \frac{q_t^*(H)-1}{t} \left(X_{t+1}^* - \frac{T+1}{2(t+1)} q_t^*(H) \right) + \frac{T+1}{2} & t < T. \end{cases} \tag{61}$$

Note that all permutations of final ranks are equally likely. Let $q_t^*(L)$ be the RP for low selection, equaling [1, p. 4]

$$q_t^*(L) = \begin{cases} 0 & t = T, \\ \left\lfloor \frac{t+1}{T+1} \left(X_{t+1}^* - R_{t+1}^* \right) \right\rfloor & t < T \end{cases} \tag{62}$$

where R_t^* is the expected high/low difference (return) if an optimal strategy for making the high/low selection(s) is followed starting at time t:

$$R_t^* = \begin{cases} 0 & t = T, \\ R_{t+1}^* + \frac{q_t^*(L)}{t} \left(X_{t+1}^* - R_{t+1}^* - \frac{T+1}{i+1} \frac{q_t^*(L)+1}{2} \right) & t < T. \end{cases} \tag{63}$$

[1, Section 2] provide the following algorithm:

Algorithm 9.

Rule (1) – Low Selection. Select q_t *at time* t *iff* $x_t \leq q_t^*(L)$.

Rule (2) – High Selection. Select q_t *at time* t *iff* $x_t \geq q_t^*(H)$.

Rule (3). If no selection is made until T, q_T *has to be accepted with rank* $X_T^* = \frac{T+1}{2}$.

Theorem 20. *The worst-case competitive ratio equals*

$$c = \begin{cases} 1, & \text{for single pair selection,} \\ \frac{4}{3}, & \text{for multiple pair selection.} \end{cases} \tag{64}$$

For the proof the reader is referred to [1, Section 3].

An open question is to investigate maximizing quantities (volumes) instead of differences in ranks.

As stated in Section 1, heuristic conversion algorithms are commonly evaluated through experiments (simulation) using historical data. However, they work without any knowledge of future input. In the following we present the competitive analysis of three well known and widely used heuristic conversion algorithms [31], namely *Moving Average Crossover* (MA), *Trading Range Breakout* (TRB), and *Momentum* (MM).

Let $ON \in \{MA, TRB, MM\}$, and assume for each i-th trade a worst-case time series containing only minimum prices $m(i)$ and maximum prices $M(i)$. At best ON buys at price $m(i)$, and sells at price $M(i)$ resulting in an optimum return $OPT = M(i)/m(i)$. But in the worst-case ON achieves a return of $ON = m(i)/M(i) = 1/OPT$.

Theorem 21. *The competitive ratio of* MA, TRB *and* MM *equals (cf. (1))*

$$c = \prod_{i=1}^{p} \left(\frac{M(i)}{m(i)} \right)^2, \tag{65}$$

and in case $m(i)$ *and* $M(i)$ *are constant*

$$c = \left(\frac{M}{m} \right)^{2p}. \tag{66}$$

To prove (65) we assume $p = 1$, i.e. ON is allowed to trade once.

Algorithm 10.

Rule (1). Buy on day t if $MA(S)_t > uB(L)_t$ and $MA(S)_{t-1} \leq uB(L)_{t-1}$.

Rule (2). Sell on day t if $MA(S)_t < lB(L)_t$ and $MA(S)_{t-1} \geq lB(L)_{t-1}$.

Where $MA(S)_t$ is a short moving average, $MA(L)_t$ a long moving average ($S < L$), and the value $n \in \{L, S\}$ defines the number of previous data points (days) considered to calculate $MA(n)_t = \frac{\sum_{i=t-n+1}^{t} q_i}{n}$. Prices q_t are lagged by bands, the upper band is $uB(L)_t = MA(L)_t \cdot (1+b)$, and the lower band is $lB(L)_t = MA(L)_t \cdot (1-b)$ with $b \in [0.00, \infty]$.

Assume a worst-case time series $\{m, \ldots, m, M, m, \ldots, m\}$. Hence, prices $q_1, \ldots, q_{t^*-1} = m$, $q_{t^*} = M$, and $q_{t^*+1}, \ldots, q_T = m$. Let $S = 1$, $L \leq (t^* - 1)$ and $b = 0.00$. We have to show that Algorithm 10 buys on day t^* at price $q_{t^*} = M$ and sells on day $t^* + 1$ at price $q_{t^*+1} = m$.

Proof.

Rule (1).:

$$\begin{aligned} MA(1)_{t^*} &= q_{t^*} = M \\ &> uB(t^*-1)_{t^*} = MA(t^*-1)_{t^*} = \underbrace{\frac{(t^*-2)m + M}{(t^*-1)}}_{<M} \end{aligned} \tag{67}$$

and

$$\begin{aligned} MA(1)_{t^*-1} &= q_{t^*-1} = m \\ &\leq uB(t^*-1)_{t^*-1} = MA(t^*-1)_{t^*-1} = \underbrace{\frac{(t^*-1)m}{(t^*-1)}}_{=m}. \end{aligned} \tag{68}$$

Rule (2).:

$$\begin{aligned} MA(1)_{t^*+1} &= q_{t^*+1} = m \\ &< lB(t^*-1)_{t^*+1} = MA(t^*-1)_{t^*+1} = \underbrace{\frac{(t^*-3)m + M + m}{(t^*-1)}}_{>m} \end{aligned} \tag{69}$$

and

$$\begin{aligned} MA(1)_{t^*} &= q_{t^*} = M \\ &\geq lB(t^*-1)_{t^*} = MA(t^*-1)_{t^*} = \underbrace{\frac{(t^*-2)m + M}{(t^*-1)}}_{<M}. \end{aligned} \tag{70}$$

Taking these decisions into account Algorithm 10 achieves a return of m/M in the worst-case resulting in $c = \left(\frac{M}{m}\right)^2$. ■

Algorithm 11.

Rule (1). Buy on day t if $q_t > uB(n)_t$ and $q_{t-1} \leq uB(n)_{t-1}$.

Rule (2). Sell on day t if $q_t < lB(n)_t$ and $q_{t-1} \geq lB(n)_{t-1}$.

Where lower band $lB(n)_t = q_t^{min}(n) \cdot (1-b)$ with $q_t^{min}(n) = \min\{q_i | i = t-n, \ldots, t-1\}$, and upper band $uB(n)_t = q_t^{max}(n) \cdot (1-b)$ with $q_t^{max}(n) = \max\{q_i | i = t-n, \ldots, t-1\}$ For $b \in [0.00, \infty]$. The value $n < t$ is the number of previous data points (days) considered by TRB.

Assume a worst-case time series $\{m+\epsilon, \ldots, m+\epsilon, M, m, \ldots, m\}$. Hence, prices $q_1, \ldots, q_{t^*-1} = m+\epsilon$, $q_{t^*} = M$, and $q_{t^*+1}, \ldots, q_T = m$. Let $n \leq (t^*-2)$, and $b = 0.00$.

Algorithm 12.

Rule (1). Buy on day t if $MM_t(n) \geq 0$ and $MM_{t-1}(n) < 0$.

Rule (2). Sell on day t if $MM_t(n) \leq 0$ and $MM_{t-1}(n) > 0$.

Where the momentum $MM_t(n) = q_t - q_{t-n+1}$, and $n \leq t$ is the number of previous data points (days) considered.

Assume a worst-case time series $\{m+\epsilon, m, \ldots, m, M, m, \ldots, m\}$. Hence, prices $q_1 = m+\epsilon$, $q_2, \ldots, q_{t^*-1} = m$, $q_{t^*} = M$, and $q_{t^*+1}, \ldots, q_T = m$. Let $n \leq (t^*-1)$ and $0 < m < M$.

We again have to show that Algorithm 11 and 12 buy on day t^* at price $q_{t^*} = M$ and sell on day t^*+1 at price $q_{t^*+1} = m$. The proofs are not given as they can be done in the same manner as for Algorithm 10, cf. [32].

4.1.2. Problem: bi|non-pmtn|M,m

[33] extend the uni-directional RPP of [10] (cf. *Problem: uni|non-pmtn|M,m*) to buying and selling within each subset n, i.e. introduce a rule for min-search.

Algorithm 13. *Buy at the first price smaller or equal, and sell at the first price greater or equal to reservation price $q^* = \sqrt{M \cdot m}$.*

Theorem 22. *Let M and m be given. Then for constant bounds the worst-case competitive ratio equals*

$$c = \left(\frac{M}{m}\right)^p, \tag{71}$$

otherwise

$$c = \prod_{i=1}^{p} \left(\frac{M(i)}{m(i)}\right) \tag{72}$$

with upper (lower) bounds $M(i)$ $(m(i))$ and overall number of trades p $(i = 1, \ldots, p)$.

For the proof the reader is referred to [32, Theorem 2].

4.2. Bi-directional preemptive

In bi-directional preemptive conversion D can be converted back and forth sequentially in parts, i.e. $s_t \in [0, 1]$. The only restriction is that within each subset n the player must convert D into Y completely, i.e. $\sum_{t=1}^{T} s_t = 1$.

4.2.1. bi|pmtn|M,m

[15] suggest an algorithm that divides the sequence of prices into upward and downward runs and then repeats Algorithm 4. Asset D is converted into Y (max-search) if the price is on an upward trend (run). Y is converted into D (min-search) if the price is on a downward trend (run). Let there be $p/2$ upward runs and $p/2$ downward runs.

Theorem 23. *Let M and m be given. Then for constant bounds $c^p_\infty(M, m)$ is the worst-case competitive ratio, and is the solution, c, of (cf. (27))*

$$c = \left(\ln \frac{(\frac{M}{m} - 1)}{(c - 1)}\right)^p \tag{73}$$

with overall number of trades p $(i = 1, \ldots, p)$.

For the proof the reader is referred to [15, Section 4].

[16] show that Algorithm 4 does not convert unless the price is as large as $c \cdot m$, i.e. the threat is at most $c \cdot m \geq m$, and shall not go beyond this point.

Theorem 24. *Let M and m be given. Then for constant bounds $c^p_\infty(M, cm)$ is the worst-case competitive ratio, and is the solution, c, of (cf. (29))*

$$c = \left(\ln \frac{(\frac{M}{c \cdot m} - 1)}{(c-1)} \right)^p \tag{74}$$

with number of trades p ($i = 1, \ldots, p$).

Again, *Rule (2)* of Algorithm 4 is modified, replacing m by cm. For the proof the reader is referred to [16, Theorem 7].

Note that the bi-directional variant of Algorithm 4 is not optimal. The challenge of designing an optimal algorithm for bi-directional search remains [16, p. 33].

[34] investigate the practical applicability of the reviewed bi-directional algorithms of [33, 15, 1, 26] in an experimental study. Buy-and-hold (BH) as well as dollar-cost averaging (DCA) [35] are used as benchmark. Results show that the algorithms can beat the market, i.e. outperform BH. The results for DCA are inconsistent.

Figure 1 gives an overview on the reviewed existing work on online algorithms for conversion problems. The worst-case competitive ratios c depend on the parameters assumed to be known a-priori to ON.

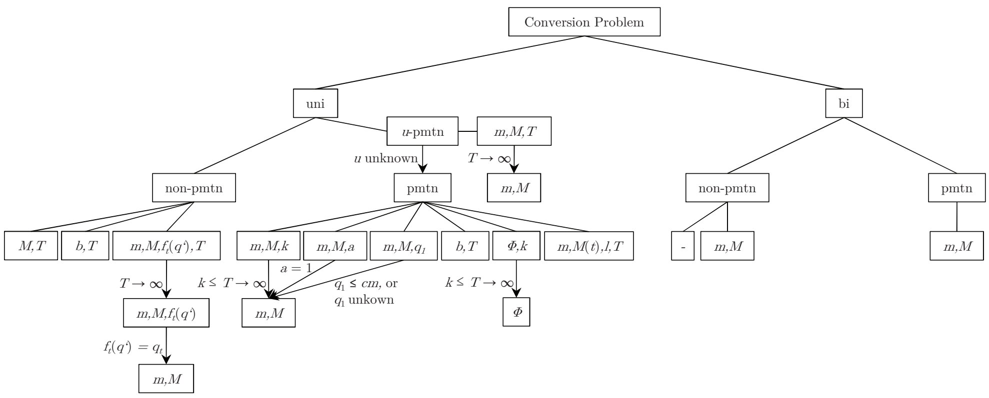


Figure 1: Conversion problems addressed

5. Conclusions

This survey provides a novel scheme to classify online algorithms for conversion problems based on their problem setting, and defines a standard nomenclature for the terms used. The three factors *nature of search|nature of conversion|given information* are used to classify existing work related and the essential parameters for conversion are given: the worst-case competitive ratio c as well as the amount to be converted s_t. In addition, we show how heuristic conversion algorithms can be evaluated using competitive analysis. Though a considerable amount of work addresses online algorithms for conversion problems, a number of questions are still unanswered and require further consideration.

The review highlights a number of open questions that need further consideration. Particularly, the applicability of the reviewed algorithms to practical problems is to be verified. The gap between a theoretically guaranteed and an experimentally achieved competitive ratio is to be investigated. A significant drawback of preemptive algorithms is the large number of conversions that might be carried out, namely T in the worst-case. The risk of a high number of transactions is not a feasible option when applying these algorithms as each conversion has an associated fee. Therefore, designing a preemptive algorithm that aims to reduce the number of conversions while maintaining a certain competitive ratio is an open problem. Online algorithms require information about the future. These parameters assumed to be known a-priori might either not be available or bound to errors in estimates. It is of interest to analyze online conversion algorithms under incorrect estimates addressing theoretical and practical aspects.

References

[1] K.-Y. Kao and S.R. Tate. On-line difference maximization. *SIAM J. on Discret. Math.*, 12(1):78–90, 1999.

[2] A. Blum, T. Sandholm, and M. Zinkevich. Online algorithms for market clearing. *J. ACM.*, 53(5):845–879, September 2006.

[3] K.L. Chang and A. Johnson. Online and offline selling in limit order markets. In C. Papadimitrou and S. Zhang, editors, *Internet and Network Economics*, volume 5385 of *Lecture Notes in Computer Science*, pages 41–52. Springer, 2008.

[4] R. Kleinberg. A multiple-choice secretary algorithm with applications to online auctions. In *Proceedings of the sixteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 630–631, 2005.

[5] W.M. Koolen and S. de Rooij. Switching investments. *Theor. Comput. Sci.*, 2012. Special Issue on Algorithmic Learning Theory (ALT 2010).

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

[7] R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Optimal search and one-way trading algorithm. *Algorithmica.*, 30(1):101–139, 2001.

[8] R. El-Yaniv, R. Kaniel, and N. Linial. Competitive optimal on-line leasing. *Algorithmica.*, 25(1):116–140, 1999.

[9] B. Vanstone and G. Finnie. An empirical methodology for developing stockmarket trading systems using artificial neural networks. *Expert. Syst. with Appl.*, 36(3, Part 2):6668–6680, April 2009.

[10] R. El-Yaniv. Competitive solutions for online financial problems. *ACM Comput. Surv.*, 30(1):28–69, 1998.

[11] W. Zhang, Y. Xu, F. Zheng, and Y. Dong. Online algorithms for the multiple time series search problem. *Comput. Oper. Res.*, 39(5):929–938, May 2012.

[12] P. Damaschke, P.H. Ha, and P. Tsigas. Online search with time-varying price bounds. *Algorithmica.*, 55(4):619–642, 2009.

[13] Y. Xu, W. Zhang, and F. Zheng. Optimal algorithms for the online time series search problem. *Theor. Comput. Sci.*, 412(3):192–197, January 2011.

[14] W. Zhang, Y. Xu, F. Zheng, and Y. Dong. Optimal algorithms for online time series search and one-way trading with interrelated prices. *J. Comb. Optim.*, 23(2):159–166, February 2012.

[15] R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Competitive analysis of financial games. In *IEEE Symposium on Foundations of Computer Science*, pages 327–333, 1992.

[16] E. Dannoura and K. Sakurai. An improvement on El-Yaniv-Fiat-Karp-Turpin's money-making bi-directional trading strategy. *Inf. Process. Lett.*, 66(1):27–33, 1998.

[17] E. Mohr and G. Schmidt. How much is it worth to know the future in online conversion problems? *Discret. Appl. Math.*, 2013. to appear.

[18] G.-H. Chen, M.-Y. Kao, Y.-D. Lyuu, and H.-K. Wong. Optimal buy-and-hold strategies for financial markets with bounded daily returns. *SIAM J. on Comput.*, 31(2):447–459, 2001.

[19] S. Hu, Q. Guo, and H. Li. Competitive analysis of on-line securities investment. In N. Megiddo, Y. Xu, and B. Zhu, editors, *Algorithmic Applications in Management*, volume 3521 of *Lecture Notes in Computer Science*, pages 224–232. Springer, 2005.

[20] Z. Zhijun, X. Yinfeng, and J. Jinhu. Competitive analysis of one-way trading with interest and transaction cost. *Forecast.*, 22(4):51–56, 2003.

[21] A. Chou, J. Cooperstock, R. El-Yaniv, M. Klugerman, and T. Leighton. The statistical adversary allows optimal money-making trading schemes. In *Proceedings of the sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 467–476, 1995.

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

[23] K. Iwama and K. Yonezawa. Using generalized forecasts for online currency conversion. In T. Asano, H. Imai, D.T. Lee, S.-I. Nakano, and T. Tokuyama, editors, *Computing and Combinatorics*, volume 1627 of *Lecture Notes in Computer Science*, pages 409–421. Springer, 1999.

[24] I. Ahmad, E. Mohr, and G. Schmidt. Competitive ratio as coherent measure of risk. In *Operations Research Proceedings 2012*. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), University of Hannover, September 4-7, 2012, Springer, 2012. in press.

[25] P. Krokhmal, M. Zabarankin, and S. Uryasev. Modeling and optimization of risk. *Surv. Oper. Res. Manag. Sci.*, 16(2):49–66, 2011.

[26] J. Lorenz, K. Panagiotou, and A. Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica.*, 55(2):311–328, 2009.

[27] W. Zhang, Y. Xu, F. Zheng, and M. Liu. Online algorithms for the general k-search problem. *Inf. Process. Lett.*, 111(14):678–682, July 2011.

[28] I. Ahmad and G. Schmidt. An experimental analysis of online unidirectional conversion problem. In Ch. Huemer and P. Lops, editors, *E-Commerce and Web Technologies*, volume 123 of *Lecture Notes in Business Information Processing*, pages 176–187. Springer, 2012.

[29] P.R. Freeman. The secretary problem and its extensions: A review. *Int. Stat. Rev.*, 51(2):189–206, August 1983.

[30] T.S. Ferguson. Who solved the secretary problem? *Stat. Sci.*, 4(3):282–296, 1989.

[31] E. Mohr, G. Schmidt, and S. Jansen. Comparative study of heuristic conversion algorithms, genetic programming and return predictability on the german market. In E. Tantar, A.-A. Tantar, P. Bouvry, P. Del Moral, P. Legrand, C.A. Coello Coello, and O. Schtze, editors, *EVOLVE - A bridge between Probability, Set Oriented Numerics and Evolutionary Computation*, volume 447 of *Studies in Computational Intelligence (SCI)*, pages 393–414. Springer, 2013.

[32] E. Mohr. *Online Algorithms for Conversion Problems*. PhD thesis, Saarland University and Max Planck Institute for Informatics, July 2011.

[33] G. Schmidt, E. Mohr, and M. Kersch. Experimental analysis of an online trading algorithm. *Electron. Notes Discret. Math.*, 36:519–526, 2010.

[34] J. Iqbal, I. Ahmad, and G. Schmidt. Can online trading algorithms beat the market? An experimental evaluation. In S. Ravizza and P. Holborn, editors, *3rd Student Conference on Operational Research*, volume 22 of *OpenAccess Series in Informatics (OASIcs)*, pages 43–52, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[35] G.M. Constantinides. A note on the suboptimality of dollar-cost averaging as an investment policy. *J. Financial Quant. Analysis.*, 14(2):443–450, 1979.

4
Competitive Ratio as Coherent Measure of Risk

Summary of Results

Risk is an unavoidable phenomenon in financial markets. Therefore, it is paramount to measure the associated risk of a financial position in order to make sound investment decisions. Ever since the "Modern Portfolio Theory" (MPT) of Markowitz, there is a considerable amount of literature devoted to measure the risk of a financial position. One such measure of risk is value at risk (VAR). VAR measures potential losses with a certain confidence level over a specific period of time. However, VAR has one inherent flaw, it fails to address the basic risk principle of *diversification*. In order to properly define a risk measure, Artzner et al. [7] put forth an axiomatic definition of coherent risk measure. The idea is that a risk measure is coherent if it is in line with basic risk principles defined in the form of axioms. The axioms were named as *Translation Invariance*, *Subadditivity*, *Positive Homogeneity*, and *Monotonicity*.

In this work, we consider the competitive ratio as risk measure and show that it satisfies all the required axioms of coherence. We discuss risk management in online conversion problems, and conclude that competitive ratio can be used as a risk measure as it is sensitive to diversification and takes into account worst case scenarios like market crashes or situation of extreme rarity.

Publication

I. Ahmad, E. Mohr and G. Schmidt: *Competitive Ratio as Coherent Measure of Risk,* Selected Papers of the Annual International Conference of the German Operations Research Society (GOR e.V.), University of Hannover, September 4-7, 2012, Springer-Verlag Berlin Heidelberg 2013.
Status : Accepted, in press.

Competitive Ratio as Coherent Measure of Risk

Iftikhar Ahmad and Esther Mohr and Günter Schmidt

Abstract A risk measure determines the quantity of an asset that needs to be kept in reserve in order to make the risk taken by an investor acceptable. In the last decade coherent measures of risk meeting a set of four desirable properties gain in importance. We prove the *Competitive Ratio* to be coherent since it satisfies the four required axioms. We explain risk management in online conversion problems, and show how the *Competitive Ratio* can be used to manage the risk.

1 Introduction

Investors in financial markets are naturally exposed to risk. It is therefore useful to quantify the risk of a financial position in order to decide if it is acceptable or not. Among several risk measures proposed in literature *Value-at-Risk* (*VaR*) and coherent risk measures are most commonly used [1–3]. *VaR* is the most popular risk measure, especially in practice, but there are several criticisms (see, for example, [4]). The main critics are that (1) *VaR* is not sensitive to diversification, and (2) *VaR* disregards any loss beyond the *VaR* level. This led [1] to introduce an axiomatic definition of coherent measures of risk. The *VaR* turns out to be not coherent since it does not decrease when an investor diversifies [2, p. 10].

Recently only *Expected Shortfall* is suggested as practicable and sound alternative to *VaR* as it is coherent and takes into account losses beyond the *VaR* level [3, p. 1519]. We prove the *Competitive Ratio* to be a further alternative as it satisfies the axioms of a coherent measure of risk.

The rest of the paper is organized as follows. Section 2 defines the concepts of online algorithms and risk management. In Section 3 it is shown that the *Competitive Ratio* is a coherent measure of risk. Section 4 concludes the paper.

Iftikhar Ahmad, Esther Mohr
Saarland University, P.O. Box 151150, D-66041 Saarbrücken, Germany, e-mail: `{ia,em}@orbi.uni-saarland.de`

Günter Schmidt
University of Cape Town, Department of Statistical Sciences, South Africa

2 Online Algorithms and Risk Management

In their work [5] present a basic risk paradigm, and classify action under two categories, (1) a risk-free action which produces a certain outcome, and (2) a risky action where the outcome is not certain. Based on [5] we interpret the risk-free action as the one which guarantees a specific return. Similarly, a risky action does not guarantee some return.

Competitive Analysis

Competitive analysis is the main tool when analyzing online algorithms, and compares the performance of an online algorithm (*ON*) to that of an adversary, the optimal offline algorithm (*OPT*).

ON computes *online* if for each $t = 1,\dots,T-1$, *ON* must compute an output for t before the input for $t+1$ is given. *OPT* computes *offline* if it can compute a feasible output given the entire input sequence I in advance. By definition, the return of *OPT* is $OPT(I) = \sup_{O\in F(I)} U(O,I)$, where $\mathscr{I}$ is a set of possible inputs I, and $F(I)$ is the set of feasible outputs O. U is a utility function such that for all I and $O \in F(I)$, $U(I,O) \in \mathbb{R}$. *ON* is c-competitive if for any input $I \in \mathscr{I}$ [6, p. 104]

$$ON(I) \geq \frac{1}{c} \cdot OPT(I). \tag{1}$$

Any c-competitive *ON* is guaranteed at least the fraction $\frac{1}{c}$ of the optimal offline return $OPT(I)$ no matter how (un)fortunate the future will be. We consider a maximization problem, i.e. $c \geq 1$. The smaller c the more effective is *ON*.

If the *Competitive Ratio* is related to a *performance guarantee* it must be a worst-case measure. This certain (risk-free) value is denoted as worst-case competitive ratio c throughout the paper. When c is derived it is commonly assumed *ON* is confronted with the worst possible sequence of prices for each $I \in \mathscr{I}$, and thus achieves the worst possible return r_{ON} on each I. Whereas *OPT* achieves the best possible return r_{OPT}.

But in a real world scenario, an investor in a financial market might be willing to take some risk in order to gain a return $X \in [r_{ON}, r_{OPT}]$. To incorporate risk management to online algorithms [7] proposed a risk-reward framework based on the competitive ratio. The framework allows an investor to take risk for a (possibly) higher return (lower competitive ratio). The model is based on forecasts on future price movements. In case the forecast is true the investor obtains a competitive ratio $c_1 < c$. However, in case the forecast is not true the investor obtains competitive ratio $c_2 > c$. Further, assume the risk taken can be controlled by a certain factor. Let the acceptable level of risk for an investor be $a \in [1,c]$. Hence, a defines the minimum and maximum bound of returns where $a = 1$ reflects no risk and $a = c$ the maximum risk. If the forecast is true, the investor can thus achieve a competitive ratio of $c_1 = c/a$. If the forecast is not true, the investor is guaranteed a competitive ratio not worse than $c_2 = a \cdot c$. Note that even if the assumptions based on which the investor is willing to take some risk are not true, the worst (possible) competitive ratio c_2 is still guaranteed.

Let the desired return of an investor be X, and the associated level of risk to achieve X be a. Consider an investor is willing to take some risk a to achieve a return $X \in [r_{ON}, r_{OPT}]$. Then the desired competitive ratio equals (cf. (1))

$$c_X \leq \frac{r_{OPT}}{X}, \tag{2}$$

and the level of risk the investor is willing to take equals

$$\begin{aligned} a &= \frac{X}{r_{ON}} \\ &= \frac{c}{c_X}, \end{aligned} \tag{3}$$

where $a \in [1, c]$. Thus, the resultant investor return can vary in a range of $[\frac{r_{ON}}{a}, a \cdot r_{ON}]$.

Coherent Risk Measures

A risk measure determines the quantity of an asset that needs to be kept in reserve in order to make the risk taken by an investor acceptable. The notion of coherent risk measures arose from an axiomatic approach for quantifying the risk of a financial position, presented in the seminal paper of [1].

Definition 1. A risk measure ρ assigns a random variable X a non-negative real number R, i.e. $\rho : \mathscr{X} \to R$.

Consider a random return X viewed as an element of a linear space $\mathscr{X}$ of measurable functions, defined on an appropriate sample space. According to [1, 2, 8] a function $\rho : \mathscr{X} \to R$ is said to be a coherent risk measure for X if it satisfies the following set of axioms.

Axiom M: Monotonicity. For all random returns $\{X_1, X_2\} \in \mathscr{X}$, if $X_1 \geq X_2$ then

$$\rho[X_1] \geq \rho[X_2]. \tag{4}$$

Monotonicity implies that if a random return X_1 is always higher than a random return X_2, then the risk of X_1 should be greater than the risk of X_2.

Axiom S: Subadditivity. For all random returns $\{X_1, X_2\} \in \mathscr{X}$

$$\rho[X_1 + X_2] \leq \rho[X_1] + \rho[X_2]. \tag{5}$$

Subadditivity implies that the risk of two investments together cannot get any worse than adding the two risks separately.

Axiom PH: Positive Homogeneity. For all random returns $X \in \mathscr{X}$, if $\lambda > 0$ then

$$\rho[\lambda X] = \lambda \rho[X]. \tag{6}$$

Positive Homogeneity implies that if a random return X is increased by λ, then the risk associated is also increased by λ.

Axiom TI: Translational Invariance. For all random returns $X \in \mathscr{X}$, risk-free returns r, and $\alpha \in \mathbb{R}$

$$\rho[X + r] = \rho[X] - \alpha. \tag{7}$$

Translational Invariance implies that by adding a risk-free return r to a random return X the risk associated decreases by α.

A widely used coherent (and moreover a spectral) measure of financial portfolio risk is the *Conditional VaR* (for a formal definition see [8, p. 227]). The *Conditional VaR* defines the expected loss of portfolio value given that a loss is occurring at or below a certain quantile-level q. Its effectiveness, however, depends on the accuracy of estimation [4, p. 999]. In contrast, the (desired) competitive ratio does not depend on any estimates (cf. (1) and (2)).

3 Competitive Ratio as Coherent Measure of Risk

While considering *Competitive Ratio* as coherent risk measure, it is pertinent to note that the nature of the *Competitive Ratio* varies a great deal from the *Expected Shortfall*. The *Competitive Ratio* quantifies the maximum regret (possible loss) under worst-case assumptions. The *Expected Shortfall* quantifies the expected return in the worst $q\%$ of the cases.

Let us consider *ON* with return r_{ON}, and *OPT* with return r_{OPT}. As r_{ON} is risk-free, *ON* is guaranteed to achieve minimum r_{ON}. Further, consider an investor is willing to take some risk $a \geq 1$ for a higher reward, and wants to achieve a return $X \geq r_{ON}$. Then, the desired competitive ratio equals $c_X = \frac{r_{OPT}}{X}$ (cf.(2)), and the level of risk to achieve X, $\rho[X]$, equals $a = \frac{c}{c_X}$ (cf. (3)).

Axiom M: Monotonicity. From (4) we know for higher returns an investor has a higher risk level, and potentially greater losses: If desired return X_1 is greater than desired return X_2 $\forall X_1, X_2 \in [r_{ON}, r_{OPT}]$, then the associated risk (and thus the potential loss) of X_1 will be at least as high as that of X_2.

Proof. If $X_1 = X_2$, it is trivial to show that $\rho[X_1] = \rho[X_2]$. If $X_1 > X_2$, using (3), we get

$$\begin{aligned} \frac{X_1}{r_{ON}} &> \frac{X_2}{r_{ON}} \qquad (8) \\ \frac{c}{c_{X_1}} &> \frac{c}{c_{X_2}} \\ \rho[X_1] = a_1 &> a_2 = \rho[X_2]. \qquad \square \end{aligned}$$

Axiom S: Subadditivity. From (5) we know diversification never increases risk: If an investor wants to achieve a higher return $(X_1 + X_2)$ such that $r_{ON} < (X_1 + X_2) \leq r_{OPT}$, then risk associated is never greater than the sum of the individual risks associated with X_1 and X_2.

Proof. For $\{X_1, X_2\} \in \mathscr{X}$, using (3), we get

$$\begin{aligned} \rho[X_1 + X_2] &= a_1 + a_2 \\ &= \frac{c}{c_{X_1}} + \frac{c}{c_{X_2}} \\ &= \rho[X_1] + \rho[X_2]. \quad \square \end{aligned} \tag{9}$$

Axiom PH: Positive Homogeneity. From (6) we know that if a random return X is increased, then the risk associated is increased by the same factor: For a desired return λX, such that $r_{ON} < \lambda X \leq r_{OPT}$, the risk associated with λX is λ times greater than the associated risk for X.

Proof. For $X \in \mathscr{X}$, if $\lambda > 0$, from (3) we get

$$\begin{aligned} \rho[\lambda X] &= \lambda a \\ &= \lambda \left(\frac{c}{c_X} \right) \\ &= \lambda \rho[X]. \quad \square \end{aligned} \tag{10}$$

Axiom TI: Translational Invariance. From (7) we know that the introduction of a risk-free investment does never increase the level of risk.
When considering the *Competitive Ratio* as a risk measure, Axiom TI needs to be redefined. We can state that in a risk-free environment there is no additional capital requirement to assure an investment decision since there is no uncertainty. Assume an investor diversifies, and invests some amount risk-free. Then the desired return equals $Y = X + r$, and from (7) we get

$$\rho[Y] = \rho[X] - \alpha. \tag{11}$$

Further, let the risky return be r' be X, i.e. $Y = r' + r$. From (11), we get

$$\rho[Y] = \rho[r'] - \alpha. \tag{12}$$

Since a risky investment, resulting in r', never decreases the level of risk $\alpha = 0$. Thus, we have to show that

$$\rho[Y] \leq \rho[r']. \tag{13}$$

Proof. For all random returns $X \in \mathscr{X}$, risk-free returns r, and risky returns r'

$$\begin{aligned} \rho[Y] &= \rho[X + r] \\ &= \rho[r' + r] \end{aligned} \tag{14}$$

by using Axiom S

$$\rho[Y] = \rho[r'] + \rho[r].$$

As $a = 1$ reflects no risk, and r is risk-free

$$\rho[Y] = \rho[r']. \quad \square \tag{15}$$

4 Concluding Remarks

Two risk measures are well established and widely used: *VaR* and *Expected Shortfall*. As *VaR* is not coherent, it may underestimate risk under extreme asset price fluctuations or an extreme dependence structure of assets [4]. Information provided by *VaR* may mislead investors. In search for a suitable alternative to *VaR*, *Expected Shortfall* has been characterized as the smallest coherent risk measure to dominate *VaR* [3].

In this paper we showed the *Competitive Ratio* to be coherent. It is sensitive to diversification and thus also dominates *VaR*. Unfortunately, the *Competitive Ratio* is so far not established as a measure of risk, or even unknown to practitioners. But a risk measure that takes into account worst-case scenarios like crashes or situations of extreme stress on investor portfolios is essential.

We conclude that the use of a single risk measure should not dominate financial risk management, and suggest the *Competitive Ratio* as a further alternative to *VaR*. Existing coherent risk measures could complement one another to provide an effective way to facilitate a more comprehensive risk monitoring.

References

1. Artzner, P., Delbaen, F., Eber, J.M., Heath, D.: Coherent measures of risk. Mathematical Finance **9**(3) (July 1999) 203–228
2. Szegö, G.: Measures of risk. European Jounal of Operational Research **163**(1) (2005) 5–19
3. Tasche, D.: Expected shortfall and beyond. Journal of Banking and Finance **26**(7) (2002) 1519–1533
4. Yamai, Y., Yoshiba, T.: Value-at-risk versus expected shortfall: A practical perspective. Journal of Banking and Finance **29**(4) (2005) 997–1015
5. MacCrimmon, K., Wehrung, D., Stanbury, W.: Taking risks: the management of uncertainty. New York: Free Press; London: Collier Macmillan Publishers (1988)
6. El-Yaniv, R., Fiat, A., Karp, R., Turpin, G.: Optimal search and one-way trading algorithm. Algorithmica **30**(1) (2001) 101–139
7. al-Binali, S.: A risk-reward framework for the competitive analysis of financial games. Algorithmica **25**(1) (1999) 99–115
8. Ahmed, S., Cakmak, U., Shapiro, A.: Coherent risk measures in inventory problems. European Journal of Operational Research **182**(1) (2007) 226–238

5

An Experimental Analysis of Online Unidirectional Conversion Problem

Summary of Results

Online algorithms for conversion problems are studied under the worst case competitive analysis. This work presents an extensive experimental study of unidirectional conversion algorithms. The objective of the work is to investigate;

1. How algorithms perform on the real world and synthetic data sets?
2. How the experimentally observed performance differs from theoretically worst case competitive ratio?
3. Which algorithms perform better than others and what affects the performance of an algorithm?

We consider a set of non-preemptive as well as preemptive algorithms [16, 20, 25, 27, 36, 47] and experiments are conducted on the real world as well as synthetically produced data sets. For the real world data we considered the daily closing prices of $DAX30$ and $S\&P500$ indices from 1-Jan-2001 to 31-Dec-2010. The artificial data were produced from the real world data using bootstrap procedure. In order to simulate real world conditions, we consider transaction fee of 0.025% of the volume transacted.

We observe that all algorithms perform significantly better than their worst case competitive ratio. The threat based algorithm of El-Yaniv et al. [27] out performs other algorithms and constantly achieves a lower competitive ratio on both real world as well as synthetically produced data set. The main reason for the stand out performance of the threat based algorithm of El-Yaniv et al. [27] is the intrinsic investment behavior of the algorithm. The threat based algorithm does not invest at every point of time, it invests only when the current price is the highest seen so far. Similarly, the amount of wealth invested also depends on the offered price.

Publication

I. Ahmad and G. Schmidt: *An Experimental Analysis of Online Unidirectional Conversion Problem,* In: C. Huemer and P. Lops (Eds.): EC-Web 2012, Lecture Notes in Business Information Processing, Vol. 123, pp. 176-187, 2012. Springer-Verlag Berlin Heidelberg 2012.
Status : Published.

An Experimental Analysis of Online Unidirectional Conversion Problem

Iftikhar Ahmad[1,⋆] and Günter Schmidt[1,2]

[1] Chair of Information and Technology Management, University of Saarland, Saarbrücken, Germany
[2] Department of Statistical Sciences,University of Cape Town, South Africa
{ia,gs}@itm.uni-sb.de

Abstract. Financial markets are highly volatile and decision making in these markets is highly risky. With the introduction of automated trading, a number of techniques are developed to facilitate the automation of financial markets. We consider a set of preemptive as well as non-preemptive online algorithms and evaluate them on real world as well as synthetically produced data. We present extensive computational results based on the observed performance of algorithms in terms of experimentally achieved competitive ratio, number of transactions performed and consistency of the results. We also investigate the gap between the worst case competitive ratio and experimentally achieved competitive ratio and conclude that algorithms perform better than their performance guarantee suggest. We conclude by highlighting a number of open questions.

Keywords: Online algorithms, Experimental evaluation, Gap between theory and practice.

1 Introduction

With the rapid development of e-commerce technologies, a number of techniques are developed to facilitate automated trading in financial markets. We consider the methods proposed in theoretical computer science and evaluate their applicability in financial markets. These strategies are called online algorithms for conversion problem. Unlike other approaches (such as Artificial Neural Networks), online algorithms do not rely on past data. Thus the performance of online algorithms are not effected by the choice of parameters such as past data or forecasts.

In an online unidirectional conversion problem, the aim is to convert an asset D into another asset Y with the objective to maximize the amount of Y after time T. On each day t, the player is offered a price q_t; the player either accepts the offered price and converts whole/portion of her remaining wealth at offered price or alternatively rejects the offered price and waits for a better price. The game ends when the player converts her whole wealth D into Y.

⋆ Corresponding author.

C. Huemer and P. Lops (Eds.): EC-Web 2012, LNBIP 123, pp. 176–187, 2012.

A number of online algorithms are proposed to address the unidirectional conversion problem [4,5,6,7,9]. Online algorithms are evaluated using competitive ratio. Competitive ratio measures the performance of an online algorithm against optimum offline algorithm. Let ON be an online algorithm for some maximization problem P and $\mathcal{I}$ be the set of problem instances. Let $ON(I)$ be the performance of ON on input sequence I and $OPT(I)$ be the performance of optimum offline algorithm. The algorithm ON is *c-competitive* if $\forall\ I \in \mathcal{I}$

$$ON(I) \geq \frac{1}{c} \cdot OPT(I). \tag{1}$$

1.1 Motivation

Although a number of solutions are proposed to solve *unidirectional conversion* problems [4,5,6,7,9], there are very few studies to investigate the applicability of these solutions to real world problems - for instance trading in financial markets. Similarly, the variety of solutions proposed are all based on different assumptions such as a priori knowledge about the upper bound of offered prices or fluctuation ratio etc. Chen et al. [4] and Hu et al. [7] compared their proposed solutions to classical buy and hold and dollar average strategy. Mohr and Schmidt [10] compared only a single online algorithm to classical techniques like moving average and buy and hold. To our knowledge, there is no comprehensive study in literature that investigates the applicability of online algorithms to real world problems.

Our aim is to conduct an extensive experimental study to evaluate the performance of online algorithms for *unidirectional conversion* problem and report the findings based on the competitive ratio. Our focus is to find out, how these algorithms fit in the real world scenario. We will identify a set of algorithms that performs better than others and will reason about their performance edge. Moreover we identify a set of problems that needs to be addressed in order to improve the applicability of *online conversion* algorithms in real world applications.

2 Related Work

Experimental analysis of online conversion algorithms has not received much attention, so far. Mohr and Schmidt [10] investigated the empirical and worst case performance of reservation price policy [6] and compared it with buy and hold. Chen et al. [4] and Hu et al.[7] compared their proposed solutions to buy and hold and dollar average strategy. Schmidt, Mohr and Kersch [12] compared threat based algorithm of El-Yaniv et al. [6] to reservation price algorithm, average price algorithm and buy and hold.

In contrast to experimental study of online algorithms, there is a significant amount of experimental studies on heuristic trading algorithms like Moving Average Crossover and Trading Range Breakout (TRB). Buy and hold (BH) is used as benchmark in these studies. Brock et al. [2] conducted an extensive experimental study of Dow Jones Industrial Index (DJIA) from 1897 to 1986. They introduced Moving Average Cross over and Trading Range Breakout (TRB), which are of great interest in literature. They compared the returns of buy (sell) signal on *DJIA* to that generated by autoregressive (AR), generalized autoregressive conditional heteroskedasticity in mean (GARCH-M) and an exponential GARCH. The results found that technical trading rules are superior to BH, AR(1), generalized autoregressive conditional heteroskedasticity in mean (GARCH-M) and an exponential GARCH. Kwon and Kish [8] extended Brock et al. [2] work by studying the predictive ability of Variable Moving Average (VMA), Fixed Moving Average (FMA) and TRB on New York Stock Exchange as well as NASDAQ indices. Other related works include [3,11,13,14].

3 Basic Definitions

We define a set of standard definitions which are used in the remaining of this paper:

i. *Duration (T)*: The length of the investment horizon in which all transactions must be carried out.
ii. *Upper Bound (M)* : The upper bound of prices in the investment horizon.
iii. *Lower Bound (m)* : The lower bound of prices in the investment horizon.
iv. *Fluctuation Ratio* (ϕ): The predicted maximum fluctuation of prices that can possibly be observed during the time interval, calculated by M/m.
v. *Threat Duration (k)* : Number of days after which the adversary may drop the offered price to some minimum level m and will keep the offered price at minimum level for the rest of the investment horizon, $k \leq T$.
vi. *Amount converted* (s_t) : Specifies which fraction of the amount available is converted at price q_t on day t, $0 \leq s_t \leq 1$.
vii. *Price Function* ($g(q_t)$) : Models price q_t based on some predefined function, e.g., the current price q_t is a function of previous price, i.e; $q_t = g(q_{t-1})$.

4 The Implemented Algorithms

In the following, we provide an overview of the algorithms selected for our experimental study. We briefly describe the algorithms and their competitive ratios. For proof of the competitive ratio the reader is referred to the respective paper.

4.1 Unidirectional Non-preemptive

In *unidirectional non-preemptive* solution (also called as *Reservation Price* algorithms), the player converts only once in an investment horizon. The player computes a reservation price q^* and compares each offered price q_t with q^*. The player accepts the first offered price q_t which is at least q^* and converts whole of D into Y in one transaction. We consider two such algorithms for our experimental study namely *RPMm* [6] and *RPMT* [5].

Algorithm 1. *(RPMm)*
Accept the first price greater than or equal to $q^* = \sqrt{M \cdot m}$.

Theorem 1. *Algorithm 1 is* $\sqrt{M/m}$ *competitive.*

Algorithm 2. *(RPMT)*
Accept the first price greater than or equal to $q^* = M/\sqrt{T}$.

Theorem 2. *Algorithm 2 is* $\sqrt{T}$ *competitive.*

4.2 Unidirectional Preemptive

In *unidirectional preemptive* solution, the player does not convert only once in the investment horizon but depending on the priced offered q_t (or time t) converts a portion s_t of D into Y. For our experiments, we consider the algorithms proposed by El-Yaniv et al. [6], Hu et al. [7], Chen et al. [4] and Lorenz et al. [9]. We briefly describe each algorithm and the competitive ratio as follows;
El-Yaniv et al [6] Threat based Algorithm: El-Yaniv et al. [6] proposed a threat based algorithm based on the assumption that there exists a threat that on day $k \leq T$, the adversary may drop the offered price to minimum level m and keep it there for the remaining period of the investment horizon.

Algorithm 3. *The basic rules of the threat-based algorithm are:*

1. *Consider a conversion from asset D to asset Y only if the price offered is the highest seen so far.*
2. *Whenever convert asset D to asset Y, convert just enough D to ensure that a competitive ratio c would be obtained if an adversary drops the price to the minimum possible price m, and keeps it there afterwards.*
3. *On the last trading day T, all remaining D must be converted to Y, possibly at price m.*

El-Yaniv et al [6] presented four variants of Algorithm 3, each assuming different a priori knowledge. We restrict our study to two variants of Algorithm 3.

i. *Variant 1 (YFKTMm):* With known M and m

Theorem 3. *Variant 1 of Algorithm 3 has a competitive ratio c as:*

$$c = \ln\left(\frac{\frac{M}{m} - 1}{c - 1}\right). \tag{2}$$

ii. *Variant 2 (YFKTMmk) :* With known M , m and k

Theorem 4. *Variant 2 of Algorithm 3 has a competitive ratio c of:*

$$c = k\left(1 - \left(\frac{m(c-1)}{M - m}\right)^{1/k}\right). \tag{3}$$

Hu et al [7] with known $g(q_t)$ and T: Hu et al. [7] presented two algorithms to achieve optimal competitive ratio under worst case assumptions, namely *Static Mixed Strategy* and *Dynamic Mixed Strategy*, where the player has the knowledge of length of investment horizon T and price function $g(q_t)$. Hu et al. [7] assumed that the current day price q_t satisfies $(1-\gamma)q_{t-1} \leq q_t \leq (1+\gamma)q_{t-1}$, where $\gamma \leq 1$

Static Mixed Strategy: The static mixed strategy allocates the amount to be converted based on the worst-case input sequence of prices.

Algorithm 4. *(HGLSMS): Amount converted on day t is determined by the following rules;*

$$s_t = \begin{cases} \left(\frac{1+\gamma}{(T-1)\gamma+2}\right) & t = 1, \\ \left(\frac{\gamma}{(T-1)\gamma+2}\right) & t \in [2, T-1], \\ \left(\frac{1}{(T-1)\gamma+2}\right) & t = T. \end{cases} \quad (4)$$

Theorem 5. *The competitive ratio c achieved by Algorithm 4 is*

$$c = 1 + \frac{\gamma}{2}(T-1)\,. \quad (5)$$

Dynamic Mixed Strategy: The worst-case scenario does not occur that frequently as assumed by the static mixed strategy. The dynamic mixed strategy allocates s_t based on the remaining number of days T' in the time interval.

Algorithm 5. *(HGLDMS): Amount converted on day t is determined by the following rules;*

$$s_t = \begin{cases} \left(\frac{1+\gamma}{(T'-1)\gamma+2}\right) W'_t & t = 1, \\ \left(\frac{\gamma}{(T'-1)\gamma+2}\right) W'_t & t \in [2, T-1], \\ \left(\frac{1}{(T'-1)\gamma+2}\right) W'_t & t = T. \end{cases} \quad (6)$$

where W'_t denotes the remaining amount of wealth at day t and $T' = T - t + 1$.

Theorem 6. *The competitive ratio c achieved by Algorithm 5 based on the remaining number of days T' is*

$$c = 1 + \frac{(T'-1)\gamma}{2}. \quad (7)$$

Chen et al [4] with known $g(q_t)$ and T: Chen et al. [4] assume prior knowledge of the duration T, and the price function $g(q_t)$. The constants α and β ($\alpha, \beta \geq 1$) determine the prices offered on a day t, and q_t satisfies $q_{t-1}/\beta \leq q_t \leq \alpha \cdot q_{t-1}$. The algorithm and the amount invested s_t on day t is described as follows:

Algorithm 6. *(CKLW): Determine the amount to be converted at time t by the following rules*

$$s_t = \begin{cases} \frac{\alpha(\beta-1)}{T\alpha\beta-(T-1)(\alpha+\beta)+(T-2)} & t = 1, \\ \frac{(\alpha-1)(\beta-1)}{T\alpha\beta-(T-1)(\alpha+\beta)+(T-2)} & t \in [2, T-1], \\ \frac{(\alpha-1)\beta}{T\alpha\beta-(T-1)(\alpha+\beta)+(T-2)} & t = T. \end{cases} \quad (8)$$

Theorem 7. *The competitive ratio c achieved by Algorithm 6 is*

$$c = \frac{T\alpha\beta - (T-1)(\alpha+\beta) + (T-2)}{\alpha\beta - 1}. \tag{9}$$

Lorenz et al [9] with known m and ϕ: Lorenz et al [9] proposed an algorithm with known m and ϕ. We discuss the strategy for max search (selling).

Algorithm 7. *(LPS):* ***Max-search Problem:*** *At the start of the game compute reservation prices $q_i^* = (q_1^*, q_2^*, ...q_u^*)$, where $i = 1, .., u$. As the adversary unfolds the prices, the algorithm accepts the first price which is at least q_1^*. The player then waits for the next price which is at least q_2^*, and so on. If there are still some units of asset left on day T, then all remaining units must be sold at the last offered price, which may be at the lowest price m.*

$$q_i^* = m\left[1 + (c-1)\left(1 + \frac{c}{u}\right)^{i-1}\right]. \tag{10}$$

Where c is the competitive ratio for the max-search problem.

Theorem 8. *Let $u \in N$, $\phi > 1$, there exists a c-competitive deterministic algorithm for u max-search problem where $c = c(u, \phi)$ is the unique solution of*

$$\frac{(\phi-1)}{(c-1)} = \left(1 + \frac{c}{u}\right)^u. \tag{11}$$

5 Experiments

We consider the set of algorithms as described in Section 4 and execute them on two different types of dataset, real world data and synthetic data (bootstrap data). We evaluate performance and the consistency of performance. The performance of algorithms is measured in terms of competitive ratio and variance of competitive ratio is used as consistency measure. We also record the number of transactions performed by each algorithm. In the following we describe the dataset, experimental settings and results.

5.1 Dataset

We consider the following two types of datasets for our experiments.

Real World Data: Two datasets *DAX30* (1.1.2001 to 31.12.2010) and *S&P500* (1.1.2001 to 31.12.2010) are considered.

Synthetic Data: We employed bootstrap method to generate additional datasets. Bootstrap is useful technique to produce additional data where original data sample size is small [14]. Using the moving block bootstrap, we generated 15 additional samples for each year for DAX30 and S&P500 (2001-2010) datasets. So, for each dataset, we generate 150 synthetic time series.

5.2 Experimental Settings

Each algorithm is executed on yearly data of *DAX30* and *S&P500*. Competitive ratio for all algorithms on yearly data is calculated. Variance is used as consistency measure. Further, we recorded the number of transactions by each algorithm, as in real world each transaction has an associated cost, it will be helpful to identify if an algorithm performing well on the basis of competitive ratio also has fewer number of transaction or vice versa.

Assumptions. For the sake of simplicity, we make the following assumptions.

i. Each transaction has associated cost of 0.025% of volume transacted.
ii. The yearly interest rate is zero.
iii. The prices considered are all closing day prices.
iv. Any amount of wealth left on last trading day is converted at the last day offered price. This is inline with the rules of threat based algorithms [6] and reservation price algorithms [5,6].
v. For algorithm $YFKTMmk$, we consider $k = T$, similarly for LPS, we assume $u = T$.
vi. For all algorithms, required a priori parameters such as m, M and/or ϕ etc are derived from the time series before execution of algorithm begins. This is inline with the working of algorithms as every algorithm assumes the exact a priori information about the future.

5.3 Results

Real World Data
DAX30 (2001-2010)
Table 1 summarizes the results for the *DAX30* and *S&P500* datasets for the years 2001 to 2010. The column *"Ave CR"* represents the average of the competitive ratio calculated over the yearly data for *DAX30* (2001-2010) and the column *"Var"* shows the variance of the competitive ratio. An average competitive ratio closer to 1 reflects the better performance of algorithms while a low variance shows the consistency of the algorithm.

In our experiments, we observed that unidirectional preemptive algorithm $YFKTMm$ suggested by El-Yaniv et al [6] performs the best among all set of algorithms considered with an average competitive ratio of 1.0873. The worst performance is observed for $HGLSMS$ [7] with an average competitive ratio of 1.1771. The most consistent algorithm is $YFKTMm$ which has a variance of $2.91 * 10^{-3}$, whereas the most inconsistent performance behavior is observed for LPS which has a variance of $12.79 * 10^{-3}$. The next closest (worst) algorithm in terms of consistency is reservation price algorithm $RPMT$ by Damaschke et al [5] with variance of $10.13 * 10^{-3}$.

S&P500 (2001-2010)
The unidirectional preemptive algorithm $YFKTMm$ of El-Yaniv et al. [6] with known M and m performance is found the best with an average competitive ratio of 1.0606 and is the most consistently performing algorithm as well with

Table 1. Avg CR ,Variance and number of transactions on real world data

$Algorithm$	$DAX30$			$S\&P500$		
	$AveCR$	$Var(10^{-3})$	$\#Tx$	$AveCR$	$Var(10^{-3})$	$\#Tx$
RPMm	1.1178	6.32	1	1.0844	4.28	1
RPMT	1.1848	10.13	1	1.1297	5.59	1
YFKT Mm	1.0873	2.91	24.2	1.0606	2.14	23
YFKT Mmk	1.1651	3.46	24.6	1.1256	2.14	23.6
HGL SMS	1.1771	6.34	254.2	1.1192	2.86	251.5
HGL DMS	1.1729	7.53	254.2	1.1162	3.23	251.5
CKLW	1.1766	6.47	254.2	1.1192	2.87	251.5
LPS	1.1532	12.79	158	1.099	4.18	150.5

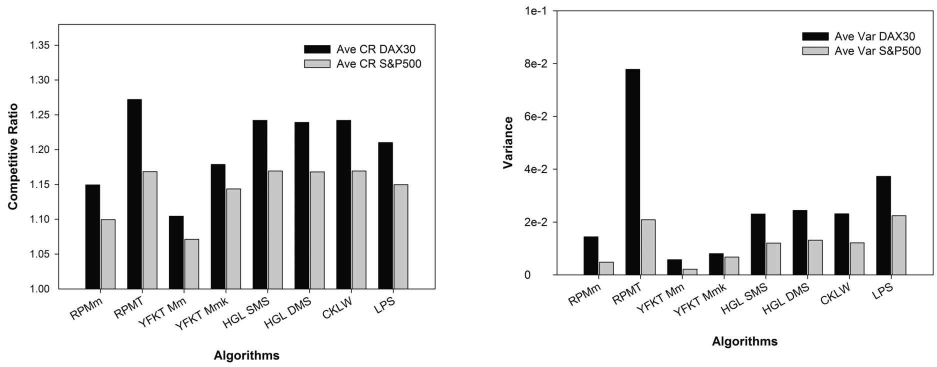
variance of $2.14 * 10^{-3}$. The worst competitive ratio is observed for $CKLW$ [4] and $HGLSMS$ [7] with an average competitive ratio of 1.1192. The most inconsistent algorithm is reservation price policy $RPMT$ of Damaschke et al. [5] with variance of $5.59 * 10^{-3}$. Table 1 summarizes the results.

Synthetic Datasets: Table 2 summarizes the results on bootstrap data. Although the individual performance on algorithm varies on different datasets, (for instance, on bootstrap DAX30 dataset, the average competitive ratio of $RPMm$ is 1.14 whereas on bootstrap S&P500 the average competitive ratio is 1.09) there is little change in overall performance order. For example, $YFKTMm$ is the best performing algorithm on both datasets. Similarly, the algorithms' behavior remains the same in terms of performance consistency as well, as depicted by Fig. 1(b). The performance consistency of $YFKTMm$ is found the best among all algorithms on both synthetic datasets, whereas the performance of $RPMT$ is found the most inconsistent. Fig. 1 depicts the performance and consistency pattern of algorithms on bootstrap data.

Number of Transactions: As each transaction has an associated cost, thus an algorithm with large number of transactions may not be a viable option. We discuss the number of transactions for each algorithm on both DAX30 and

Table 2. Avg CR, Variance and number of transactions on bootstrap data

$Algorithm$	$DAX30$			$S\&P500$		
	$AveCR$	$Var(10^{-3})$	$\#Tx$	$AveCR$	$Var(10^{-3})$	$\#Tx$
RPMm	1.1495	14.4	1	1.099	4.8	1
RPMT	1.3045	81.8	1	1.1982	22.1	1
YFKT Mm	1.1044	5.7	21.25	1.071	2.0	18.48
YFKT Mmk	1.1784	8.05	21.6	1.1436	6.77	18.94
HGL SMS	1.2420	23.0	254.1	1.169	12	251.5
HGL DMS	1.2391	24.4	254.1	1.167	13.1	251.5
CKLW	1.2420	23.1	254.1	1.17	12.1	251.5
LPS	1.2099	37.3	135.96	1.149	22.4	130.67



(a) Performance - bootstrap data (b) Consistency - bootstrap data

Fig. 1. Performance and consistency on bootstrap data

S&P500. Table 1 indicates that the non-preemptive algorithms carry only a single transaction in each investment horizon. This holds true for all unidirectional non-preemptive strategies as they convert at a single point of time in the investment horizon, but not for unidirectional preemptive solutions, where the conversion amount is calculated based on the price offered and time in the investment horizon. Table 1 reflects that for DAX30, the algorithm $YFKTMm$ by El-Yaniv et al. [6] has the least number of transactions in all unidirectional preemptive solutions, as the algorithm only invests when the price offered is the highest seen so far, thus the algorithm does not convert at all offered prices but does so on local maxima. HGL [7] and CKLW [4] have the highest number of transactions, which is the same as the number of days in the investment horizon, as they invest on each day of the investment horizon. The same pattern is found when transactions on bootstrap data are considered. Table 2 also summarizes the resultant number of transactions on bootstrap data.

6 Discussion

From the outcome of the experiments and based on the criterion of competitive ratio, we observe that unidirectional preemptive algorithm $YFKTMm$ [6] performs better than other algorithms. On DAX30 and S&P500 datasets,$YFKTMm$ performs 6% and 4% better than the average performance of the remaining algorithms, whereas on bootstrap data the corresponding numbers are 9% and 7% respectively. Similarly, the $YFKTMm$ also proves to be more consistent in terms of variance in the competitive ratio. On DAX30 and S&P500 datasets, the variance of $YFKTMm$ is on average 62% and 67% less than the average variance of all other considered algorithms. $YFKTMm$ remains the most consistent algorithm on bootstrap data as well. There is no clear worst performing algorithm as a number of algorithms perform poorly on different datasets. For instance on DAX30 dataset, $HGLSMS$, $HGLDMS$ and $CKLW$ performance are among the worst three, they have an average competitive ratio of approximately 1.17, the same holds for S&P500. On DAX30 and S&P500 bootstrap data, the worst performing algorithm is $RPMT$ with an average competitive ratio of 1.3 and 1.19 respectively.

The reason for the better performance of $YFKTMm$ [6] is that the algorithm converts only when it finds a new maximum, this not only results in better performance but also reduces the number of transactions. Although LPS [9] also converts only when it encounters a new maximum, it is not as competitive as $YFKTMm$, this can be attributed to the amount of wealth converted. $YFKTMm$ considers the offered price q_t when calculating s_t but LPS does not consider the offered price. Another significant result, we observed, is the performance of non-preemptive algorithm of El-Yaniv et al. [6]. On dataset S&P500, the average competitive ratio of non-preemptive algorithm of El-Yaniv et al [6] is 1.0844 which is second only to $YFKTMm$, the same results holds for DAX30 dataset and for bootstrap data (DAX30, S&P500). Another aspect of the study is that intuitively, the more information available to (use by) an algorithm, the better it must perform, but this however may not happen. On both datasets DAX30 and S&P500, the preemptive algorithm $YFKTMm$ performs better than $YFKTMmk$, this can be attributed to the 'luckily behaving data' which results in better performance of $YFKTMm$.

An important consideration of any experimental study is to observe the gap between theory and practice. For all algorithms and for each yearly dataset, we calculate the worst case competitive ratio (c_{wc}) that an algorithm can achieve with the given setting and after the algorithm is executed on yearly data, we record the experimental competitive ratio (c_{ec}). For instance, consider yearly data of DAX30 for 2001 and algorithm $YFKTMm$, before the algorithm begins execution, we calculate the c_{wc} using Theorem 3, similarly when the algorithm is executed on data, we record the c_{ec} achieved by algorithm. The process is repeated for all algorithms and for all real world data of DAX30 and S&P500 datasets. We limit it only to real world data and do not include the bootstrap data as we are investigating the gap between theory and practice, hence synthetic (bootstrap) data is not considered. It can be seen from Table 3 that the algorithms suggested by El-Yaniv et al. [6] ($RPMm, YFKTMm, YFKTMmk$) have the least gap between c_{wc} and c_{ec} whereas other algorithms have considerable gap between c_{wc} and c_{ec}. For instance, $RPMT$ [5] on DAX30 dataset, has (average) c_{wc} of 387.19 whereas the c_{ec} is 1.155, this is because of the reservation price calculation of $RPMT$ (Theorem 2) which only considers M and T and not the relative fluctuation in the prices. For HGL, $CKLW$ and LPS, the gap is not as wide as of $RPMT$ but is considerably more than that of El-Yaniv et al. [6]. The gap between c_{wc} and c_{ec} of $HGLSMS$, $HGLDMS$ and $CKLW$ is based on the fact that length of investment horizon T has significant contribution in determining the worst case competitive ratio (see Theorem 5 and 7). For LPS the gap is the result of our choice of parameter u, as we consider $u = T$, thus it results in higher c_{wc}. It is interesting to see that the performance of $HGLSMS$, $HGLDMS$ and $CKLW$ does not differ a great deal, it is because of the fact

Table 3. Gap between theory and practice

Algorithm	$DAX30$		$S\&P500$	
	c_{wc}	c_{ec}	c_{wc}	c_{ec}
RPMm	1.2403	1.1178	1.1660	1.0844
RPMT	387.1989	1.1848	81.8012	1.1297
YFKTMm	1.188	1.0873	1.1408	1.0606
YFKTMmk	1.1812	1.1651	1.1382	1.1256
HGLSMS	8.3821	1.1771	6.8383	1.1192
HGLDMS	8.3821	1.1729	6.8383	1.1162
CKLW	8.3821	1.1766	6.8383	1.1192
LPS	7.6397	1.1532	6.5921	1.099

that these algorithms considers only price function ($g(q_t)$) and the length of investment horizon T. In addition, the price function considered by Hu et al. [7] is identical to that of Chen et al. [4] (only the mathematical formulation differs), thus resulting in similar performance behavior. Considering the performance of LPS, it is important to mention that for $u = 1$, the algorithm is similar to unidirectional preemptive algorithm $RPMm$ but as we consider $u = T$ thus the performance varies.

A major drawback in preemptive algorithms is the large number of transactions. On yearly real world data, with approximately 250 trading days, the least number of transactions performed by pre-emptive algorithms is 24 by $YFKTMm$. Although the number of transaction of $YFKTMm$ are less than other pre-emptive algorithms like $HGLSMS$, $HGLDMS$ and $CKLW$, which trades on every day, it still is significantly higher number when the impact of transaction cost on performance is considered. However, the ideal number of transactions per year is hard to envisage and depends on the amount of wealth available to the player.

7 Future Work and Conclusion

We presented an extensive experimental study to evaluate the applicability of online conversion algorithms in real world scenario such as trading in financial markets. We observed that although, a good number of algorithms are proposed to deal with unidirectional conversion problem, there are still a considerable number of open questions. An important factor for the designing new conversion algorithms must be to reduce the number of transactions, as in real world each transaction has an associated cost, thus reducing the number of transactions can be useful. However, the optimum number of transactions cannot explicitly be defined. Another open question will be to develop algorithms that provide risk management for the investors, as in real world the investors want to manage risk but online algorithms are designed based on risk mitigation paradigm. Al-binali [1] proposed a risk-reward framework, which can be used to incorporate risk management in online conversion problems.

References

1. Al-Binali, S.: A risk-reward framework for the competitive analysis of financial games. Algorithmica 25, 99–115 (1999)
2. Brock, W.A., Lakonishok, J., LeBaron, B.: Simple technical trading rules and the stochastic properties of stock returns. Journal of Finance 47, 1731–1764 (1992)
3. Chang, E.J., Lima, E.J.A., Tabak, B.M.: Testing for predictability in emerging equity markets. Emerging Markets Review 5(3), 295–316 (2004)
4. Chen, G.-H., Kao, M.-Y., Lyuu, Y.-D., Wong, H.-K.: Optimal buy-and-hold strategies for financial markets with bounded daily returns. SIAM Journal on Computing 31(2), 447–459 (2001)
5. Damaschke, P., Ha, P.H., Tsigas, P.: Online search with time-varying price bounds. Algorithmica 55(4), 619–642 (2009)
6. El-Yaniv, R., Fiat, A., Karp, R.M., Turpin, G.: Optimal search and one-way trading algorithm. Algorithmica 30, 101–139 (2001)
7. Hu, S., Guo, Q., Li, H.: Competitive Analysis of On-line Securities Investment. In: Megiddo, N., Xu, Y., Zhu, B. (eds.) AAIM 2005. LNCS, vol. 3521, pp. 224–232. Springer, Heidelberg (2005)
8. Kwon, K.-Y., Kish, R.J.: A comparative study of technical trading strategies and return predictability: An extension of brock, lakonishok, and lebaron (1992) using NYSE and NASDAQ indices. Quarterly Review of Economics and Finance 42(3), 613–633 (2002)
9. Lorenz, J., Panagiotou, K., Steger, A.: Optimal algorithms for k-search with application in option pricing. Algorithmica 55(2), 311–328 (2009)
10. Mohr, E., Schmidt, G.: Empirical Analysis of an Online Algorithm for Multiple Trading Problems. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) MCO 2008. CCIS, vol. 14, pp. 299–308. Springer, Heidelberg (2008)
11. Ratner, M., Leal, R.P.C.: Tests of technical trading strategies in the emerging equity markets of latin america and asia. Journal of Banking and Finance 23, 1887–1905 (1999)
12. Schmidt, G., Mohr, E., Kersch, M.: Experimental analysis of an online trading algorithm. Electronic Notes in Discrete Mathematics 36, 519–526 (2010)
13. Shen, P.: Market-timing strategies that worked. Jounal of Portfolio Management 29(2), 57–68 (2003)
14. Tabak, B.M., Lima, E.J.A.: Market efficiency of brazilian exchange rate: Evidence from variance ratio statistics and technical trading rules. European Journal of Operational Research 194, 814–820 (2009)

6
Experimental Analysis of Online Uni-directional Conversion Algorithms using EVT Approach

Summary of Results

Using the backtesting approach for evaluation of online algorithms on the past real world data has significant drawbacks. For instance, the data might be very limited to draw valid conclusions. Similarly, if an algorithm performs well on the past data, it is not a guaranteed success of algorithm on unforeseen data in the future. Therefore, the need for the synthetic data arises in order to validate the findings on a larger set of data and to draw conclusion with a reasonable confidence level.

Bootstrap method is used to generate artificial data. However, the bootstrap method depends on the original data and fails to replicate all scenarios. Similarly, the bootstrap method omits the extreme values as outliers. For instance, the bootstrap method may fail to replicate market crashes. In order to improve the scenario generation for algorithm evaluation, we recommend using Extreme Value Theory (EVT) approach. Extreme Value Theory (EVT) provides a solid probabilistic foundation for studying the distribution of extreme events in order to quantify the stochastic behavior. When applied to financial markets EVT is useful for modeling the impact of market crashes or situations of extreme stress on in-

vestor portfolios. Thus, EVT approach ensures that extreme and rare events such as market crashes (worst case scenarios) are also represented in the test instances.

We consider three sets of DAX30 index, each set representing a different market trend. The first set contains daily closing prices from 13-Mar-2003 to 10-Mar-2006 representing an increasing trend of the market. The second set representing the decreasing trend of the market contains data from 7-Mar-2000 to 6-Jun-2003. The third set exhibits "no trend" representing a time period from 2-Jan-1979 to 30-Dec-1981. After computing stylized facts for each set of data, we use the estimated parameters as input data and generate synthetic data set that fits the extreme value distribution. We then consider a set of non-preemptive algorithms namely $RP(m, M)$, $RP(M, T)$, $RP(m, M, f(q_t))$ and $RP(m, M, f(q_t), T)$ (see Chapter 3, Section 3.1.1) and study i) the worst case competitive ratio, ii) the expected experimental case competitive ratio of each algorithm. We observe that worst case competitive ratio of $RP(m, M) < RP(m, M, f(q_t))$ (and $RP(m, M, f(q_t), T)) < RP(M, T)$ whereas the experimental results observe a different ordering where the experimentally achieved competitive ratio of $RP(m, M, f(q_t))$ (and $RP(m, M, f(q_t), T)) < RP(m, M) < RP(M, T)$.

Publication

E. Mohr, I. Ahmad and G. Schmidt: *Solving Uni-directional Conversion Problems by Online Algorithms under an Extreme Value Distribution.* DSOR Contributions to Information Systems, Band 8: Applied Mathematical Optimization and Modelling, L. Suhl, G. Mitra, C. Lucas, A. Koberstein, L. Beckmann (Eds.), pp. 459-470, 2012.

Status : Published.

Solving uni-directional conversion problems by online algorithms under an extreme value distribution

Esther Mohr[a,*], Iftikhar Ahmad[a], Günter Schmidt[a,b]

[a] *Saarland University, P.O. Box 151150, D-66041 Saarbrücken, Germany*
Phone +49-681-302-4559, Fax +49-681-302-4565
[b] *University of Liechtenstein, Fürst-Franz-Josef-Strasse, 9490 Vaduz, Liechtenstein*

Abstract

Online algorithms can be used for trading financial markets such as currency conversion. We consider a set of uni-directional non-preemptive online conversion algorithms, and investigate their performance empirically. We compute stylized facts of different datasets of the German DAX30, and estimate the 'true' stochastic properties of the dataset under consideration by the weighted moment method. Using the estimated parameters as input data, we generate synthetic datasets fitting an extreme value distribution. The worst-case and the empirical-case performance of known online conversion algorithms is investigated on these 'typical datasets'. Based on the experimental data, we calculate 1) the worst-case competitive ratio c^{wc} taking the data of the problem instance into account, and 2) the expected competitive ratio $c^{ex} = E[\frac{OPT}{ON}]$ using the returns achieved by the algorithms. We report a great disparity between the worst-case and the empirical-case results, and show that the extreme value distribution is proper to model a realistic volatility.

Keywords: Online Conversion Problem, Extreme Value Distribution, Competitive Analysis, Online Algorithm, Stochastic Optimization

*Corresponding author
Email addresses: em@itm.uni-sb.de (E. Mohr), ia@itm.uni-sb.de (I. Ahmad), gs@itm.uni-sb.de (G. Schmidt)

1. Introduction

A uni-directional conversion problem deals with the scenario of converting an asset D into another asset Y with the objective to get the maximum amount of Y after time T. Non-preemptive conversion algorithms convert the whole amount available at one price q_t ($t = 1, \ldots, T$).

A number of online algorithms (ON) are proposed in the literature to solve the uni-directional (uni) non-preemptive ($npmtn$) conversion problem [1–3]. When considering the $uni|npmtn$ online conversion problem, ON does not know the whole set of prices q_t in advance. On any day t, ON must take a decision whether or not to accept q_t without any knowledge about q_{t+1}, and so on. Moreover, the decision to convert at any offered q_t is irreversible.

In competitive analysis (which was first applied to online algorithms by [4]), the performance of ON is compared with that of an optimal offline algorithm (OPT). It is assumed that OPT has full knowledge of future events, and thus acts optimally. In contrast, ON incrementally receives one observation (the input) in each time period, i.e. generates an output without any knowledge of future events. Let $\mathcal{I}$ be the set of all input instances, and let $ON(I)$ be the performance of ON on input instance $I \in \mathcal{I}$. ON is called c-competitive if $\forall \quad I \in \mathcal{I}$

$$ON(I) \geq \frac{1}{c} \cdot OPT(I). \tag{1}$$

When considering worst-case scenarios, any c-competitive algorithm is guaranteed a value of at least the fraction $\frac{1}{c}$ of the optimal offline result, no matter how uncertain the future will be [5, p. 104]. Note that within this paper we differ the worst-case competitive ratio c^{wc} and the expected competitive ratio c^{ex}.

The worst-case competitive ratio is a theoretical performance measure used in *Computer Science*. In *Finance*, to evaluate the practical applicability of ON to real world data, backtesting is used. Backtesting is the concept of taking an algorithm $X \in \{OPT, ON\}$ and going back in time in order to see what would have happened if X had been followed [6]. The assumption is that if X has (not) performed well in the past, it has a great (but not certain) chance of (not) performing well again in the future.

In the classical backtesting approach, data collection is followed by the imposition of a specific stochastic model (assumed) to fit the input data presented to X [7]. The experimental analysis that follows is focused on the parameters of that model. The objective is to compute and analyze the empirical-case performance of X under 'typical inputs' with respect to these stochastic assumptions. When considering financial data, for example, a common a-priori assumption is that data is normal distributed [8, 9, p. 123]. Unfortunately, most existing models fail to reproduce the underlying data structure [10, p. 233]. Thus, this approach is criticized from both a technical, and a conceptual perspective. Technically, for many real-life problems, an adequate stochastic model is extremely difficult or costly to devise [11, p. xxiii]. Conceptually, the validity of the conclusions on the performance of X becomes dependent on the validity of the underlying (distributional) assumptions [12]. Worse yet, the exact underlying assumptions may be unknown, or if known, untested. For instance, a great deal of effort has been invested in attempt to identify the probability distributions of currency exchange rates but there is still no evidence that an appropriate stochastic model exists [13].

As a result, some research attempts focus on *identifying* the properties of input data under consideration rather than on assuming underlying properties using statistical models. Several methods based on statistical techniques are recently applied in *Empirical Finance* [10, 14–16]. The goal is to 'let the data speak for themselves' as much as possible. This approach is also known as *Exploratory Data Analysis* (EDA) [17]. In terms of statistical methods, analysis is done by using so-called non-parametric methods which make only qualitative assumptions about the properties of the stochastic process generating the data. They do not assume a-priori that the input data belongs to any prespecified parametric family [10, p. 223]. These qualitative properties are called empirical 'stylized facts', and characterize a dataset from a statistical point of view. When considering financial datasets stylized facts are summary statistics calculated by daily logarithmic returns

$$r_t = \ln \frac{q_t}{q_{t-1}}, \tag{2}$$

where q_t equal daily (closing) prices within a time interval of length T $(t = 1, \ldots T)$ [10, p. 224]. Stylized facts at least contain 1) Arithmetic mean, 2) standard deviation, 3) skewness, and 4) kurtosis of the dataset under consideration [18, p. 1737].[1] The

[1] See [8, (5-12) to (5-14)] for a formal definition of 1) to 4).

arithmetic mean $\bar{r}_T(X)$ (of returns r_t achieved by X within T) is commonly used as the estimator for the (unknown) return to be expected from ON in the future. The standard deviation σ shows the variation from $\bar{r}_T(X)$: A low σ indicates that the observed r_t tend to be very close to $\bar{r}_T(X)$, whereas a high σ indicates that the observed r_t are spread out over a large range of values. The skewness γ measures the (a)symmetry in the probability distribution of the observed r_t. The kurtosis β measures with which probability extremely low or extremely high r_t might occur. In case $\beta > 3$ (leptokurtosis) both tails of the probability distribution are 'fat', i.e. the mass of the distribution is concentrated on the left and on the right. Relatively many high and low r_t exist. These 'fat-tailed' distributions are commonly known as extreme value distributions, and are often used to describe (or model) financial data [16, 19, 20].

2. Our contribution

Within this work we contribute to the experimental analysis of (online) conversion algorithms as follows. In contrast to the classical backtesting approach we omit the imposition of a specific stochastic model (assumed) to fit the input data presented to X. First, we compute stylized facts of different datasets of the German DAX30, i.e. using the underlying r_t we estimate the 'true' stochastic properties of the dataset under consideration. We use the weighted moment method of [19], and find that the r_t follow an extreme value distribution. Second, using the estimated parameters as input data, we generate synthetic datasets (based on original DAX30 data) also fitting an extreme value distribution. Third, we investigate the worst-case and the empirical-case performance of known ON on these 'typical datasets'. Simulation runs with four $uni|npmtn$ algorithms from the literature are performed. Based on the experimental data, we calculate 1) the worst-case competitive ratio c^{wc} taking the data of the problem instance into account, and 2) the expected competitive ratio $c^{ex} = E[\frac{OPT}{ON}]$ using the returns achieved by $X \in \{OPT, ON\}$. We report a great disparity between the worst-case and the empirical-case results.

We extend former works by using more than one time series, and thus our results are less effected by randomness in terms of 'luckily' behaving data. Further, we empirically show that an extreme value distribution is proper to model a realistic volatility. Our results are consistent with several results from the literature, an overview can be found in [20, p. 6].

3. Algorithms, Experiments and Results

We limit our experimental study to $uni|npmtn$ online conversion algorithms. Non-preemptive algorithms define limit price(s) (the market participant is willing to accept) to avoid a conversion at a price higher (lower) than a specific level. When converting assets, that is the highest (lowest) price per asset ON might accept for buying (selling) [21]. Such limit prices are commonly known as reservation prices (RP), with $RP \in \{q^*, q_t^*\}$. We consider ON that 1) solve the max-search problem (sell an asset), and 2) convert 'all or nothing', i.e. the first $q_t \geq RP$ must be accepted within T. Note that if no such q_t exists the last price q_T must be accepted.

3.1. Algorithms

Table 1 presents the four $uni|npmtn$ algorithms considered in our experiments, and their worst-case competitive ratios c^{wc} are given. The proofs for the respective c^{wc}, discussing several cases and worst-case time series, are not given here due to their length. The reader is referred to [1–3], especially the definition of α (resp. α_l) can be found in [3, p. 3, eq (1)] (resp. [3, p. 4, eq (2)]).

Algorithm	Reference	RP	c^{wc}
$RP(M,m)$	[1]	$q^* = \sqrt{M \cdot m}$	$\sqrt{\frac{M}{m}}$
$RP(M,T)$	[2]	$q^* = \frac{M}{\sqrt{T}}$	$\sqrt{T}$
$RP(M,m,f(q_t))$	[3]	$q_t^* = f_t^{-1}\left(\frac{f_{t+1}(M)}{\alpha_l}\right)$	α_l
$RP(M,m,f(q_t),T)$	[3]	$q_t^* = f_t^{-1}\left(\frac{f_{t+1}(M)}{\alpha}\right)$	α

Table 1. Overview on the algorithms considered

Solving an online problem, each decision must be made based on the already appeared data of the problem instance, and without any knowledge about future data [22]. Though, the considered ON require some a-priori knowledge about the future. In order to compute a result, some 'amount of information' (about the future) must be known to ON. The algorithms differ in their assumed a-priori knowledge. To compute a $RP \in \{q^*, q_t^*\}$

- $RP(M,m)$ requires the upper and lower bounds of prices m and M,

- $RP(M,T)$ requires M, and the length of the time interval T,
- $RP(M,m,f(q_t))$ requires m, M, and a return function $f(q_t)$,[2]
- $RP(M,m,T,f(q_t))$ requires m, M, $f(q_t)$, and T.

For the algorithms of [3] we use the $f(q_t)$ given in [23, p. 987].

3.2. Experiments

To identify the underlying distribution(s), historical DAX30 datasets are used representing 1) an increasing market, 2) a decreasing market, and 3) a market without trend. Table 2 gives an overview. Instead of assuming stochastic parameters, we

Market	Increasing	Decreasing	No trend
First day	13.03.2003	07.03.2000	02.01.1979
Last day	10.03.2006	06.03.2003	30.12.1981
Starting value	2354.31	8064.97	581.14
Final value	5804.92	2437.51	490.39

Table 2. Overview on the time series chosen for identifying distribution parameters

determine the stylized facts of each dataset by the weighted moment method of [19]. We find that the underling r_t of each dataset 1) to 3) follow an extreme value distribution. Using the stylized facts as input (to the package 'fExtremes' of the statistic software **R**) synthetic datasets also fitting an extreme value distribution are generated.[3] Representing the three different market types given in Table 2, datasets of different length $T \in \{750, 250, 100\}$ days are compiled by **R**. With this setting we run the algorithms presented in Table 1 on overall 9.000 different datasets (1.000 datasets for each market 1) to 3) and T).

The performance of the algorithms is measured by the average terminal wealth $\bar{w}_T(X)$ achieved by $X \in \{OPT, ON\}$. Let w_0 be the initial wealth to be invested by X. Further, let r_t be the daily log returns for each day X *holds* an asset; calculated

[2] The return r_t for accepting a q_t is not exactly the price itself but a function of the price. For example, the accepted q_t minus the costs for observing $T - t$ prices during T.

[3] The **R** Project for Statistical Computing: www.r-project.org

by eq (2). Daily log returns are time additive. Thus, for each of the 9.000 datasets considered the overall return achieved by X equals

$$r_T(X) = \sum_{t=1}^{T} r_t, \tag{3}$$

and thus the terminal wealth $w_T(X) = r_T(X) \cdot w_0$.

To estimate the (unknown) return to be expected from X in the future, we use the average terminal wealth $\bar{w}_T(X)$. As each X is run on $N = 1.000$ datasets ($i = 1, \ldots, N$) of equal length $T \in \{750, 250, 100\}$

$$\bar{w}_T(X) = \frac{1}{N} \sum_{i=1}^{N} w_T(X). \tag{4}$$

Further, the (unknown) return to be expected from X in the future $E[X]$ is estimated by (cf. eq (4))

$$\begin{aligned} E[X] &= \frac{\bar{w}_T(X)}{w_0} \\ &= \bar{r}_T(X), \end{aligned} \tag{5}$$

and the so-called *ratio of expectations* is calculated by $\frac{E[OPT]}{E[ON]}$ [16, p. 85]. Several authors use this ratio, cf. [24–27]. But [25] showed that $E\left[\frac{OPT}{ON}\right]$ measures performance more adequately than $\frac{E[OPT]}{E[ON]}$. Further, [16, p. 85] showed when converting 'all or nothing' the so-called *expectation of ratios* $E\left[\frac{OPT}{ON}\right]$ must be used: Instead of using averages, for each i-th dataset of length $T \in \{750, 250, 100\}$, the *empirical-case return ratio* achieved by ON (cf. eq (1) and eq (3))

$$c_T^{ex} = \frac{r_T(OPT)}{r_T(ON)} \tag{6}$$

is required. As each ON is run on $N = 1.000$ datasets ($i = 1, \ldots, N$) of equal length $T \in \{750, 250, 100\}$

$$E\left[\frac{OPT}{ON}\right] = \frac{1}{N}\sum_{i=1}^{N} c_T^{ex}. \tag{7}$$

We name $E\left[\frac{OPT}{ON}\right]$ the overall *expected competitive ratio*, and denote it by c^{ex}. Similarly, the overall *worst-case competitive ratio* c^{wc} of ON equals

$$c^{wc} = \frac{1}{N}\sum_{i=1}^{N} c_T^{wc}, \tag{8}$$

where c_T^{wc} denotes the *worst-case return ratio* for each i-th dataset of length $T \in \{750, 250, 100\}$, and is calculated using the equations given in column 4 of Table 1. The average deviation from the worst-case

$$\delta = \left(\prod_{i=1}^{9} \frac{c^{wc}}{c^{ex}}\right)^{\frac{1}{9}} \tag{9}$$

is calculated using eq (7) and eq (8).

3.3. Results

On each dataset of length T the four RP algorithms presented in Section 3.1 and OPT are run. As performance measure we consider the worst-case competitive ratio c^{wc} (cf. eq (8)), and the expected competitive ratio c^{ex} (cf. eq (7)). In addition, the average deviation from the worst-case δ is given (cf. eq (9)). Clearly, the RP algorithms cannot outperform OPT, i.e. $c \geq 1$, and $c \in \{c^{wc}, c^{ex}\}$ (cf. eq (1)). For each dataset of length T we determine the parameters required to calculate the possible c^{wc} of $ON \in \{RP(M, m), RP(M, T), RP(M, m, f(q_t)), RP(M, m, f(q_t), T)\}$. When calculating the c^{ex} the empirical-case return which actually was achieved by ON is compared to OPT. Results are shown in Tables 3 and 4.

In increasing (decreasing) markets the best-case is to convert on the last (first) day.

We observed that the $q^* = \sqrt{M \cdot m}$ of $RP(M, m)$ is too low. In an increasing market $RP(M, m)$ converts too early at a price close to the minimum m, and the worst-case almost occurs as c^{ex} is very close to c^{wc}. Similarly, due to the price movement

Algorithm		$RP(M,m)$		$RP(M,T)$		$RP(M,m,f(q_t))$ $RP(M,m,f(q_t),T)$	
Market	T	c^{wc}	c^{ex}	c^{wc}	c^{ex}	c^{wc}	c^{ex}
Decreasing	100	1.1744	1.0721	10.0000	1.0953	1.3804	1.0279
	250	1.3510	1.0980	15.8114	1.1161	1.8257	1.0627
	750	2.0595	1.1224	27.3861	1.1268	2.9083	1.1067
Increasing	100	1.1043	1.0923	10.0000	1.1757	1.2171	1.0124
	250	1.2111	1.2001	15.8114	1.4105	1.4715	1.1060
	750	1.6644	1.6529	27.3861	2.7142	2.3377	1.9620
No trend	100	1.0544	1.0303	10.0000	1.0441	1.1076	1.0022
	250	1.0936	1.0487	15.8114	1.0670	1.1954	1.0094
	750	1.1879	1.0721	27.3861	1.0902	1.4151	1.0305

Table 3. Worst-case and expected competitive ratios of the RP algorithms

Algorithm	δ
$RP(M,m)$	13.00%
$RP(M,T)$	1202.40%
$RP(M,m,f(q_t))$	40.08%
$RP(M,m,f(q_t),T)$	40.08%

Table 4. Average deviation from the worst-case

$RP(M,m)$ converts (too) early in a decreasing market which leads to $c^{ex} << c^{wc}$. Overall, in the experiments $RP(M,m)$ achieves a result δ=13.00% better than the worst-case. In practice, to calculate the q^* of $RP(M,m)$ precise estimates of M and m are required. But the better the estimates, the smaller δ – especially in an increasing market. We conclude the c^{wc} is too optimistic.

In case of $RP(M,T)$ the c^{wc} equals $\sqrt{T}$, i.e. the longer T the greater c^{wc} gets. In case of $RP(M,T)$ the longer T the greater c^{wc} gets. This finding is contradictionary to the findings in *Finance*, as longer datasets are considered to be advantageous in the sense that they generate more reliable results. Experiments show that the c^{wc} is too pessimistic as on average the c^{ex} is 1202.40% better than the worst-case. In addition the

$q^* = \frac{M}{\sqrt{T}}$ gets the smaller the longer T gets. Similarly to $RP(M,m)$ the q^* calculated by $RP(M,T)$ is too low, and $RP(M,T)$ converts too early.

Surprisingly, $RP(M,m,f(q_t))$ and $RP(M,m,f(q_t),T)$ generate identical results. The q_t^* used by both algorithms only differs in the denominator. Due to the $f(q_t)$ of [23] α_l and α are monotone increasing, i.e. when searching for the minimum only $t = 1$ must be considered. We find α_l equals α in this setting, resulting in identical q_t^*, i.e. the algorithms convert at identical prices. Similarly to $RP(M,m)$ and $RP(M,T)$ the calculated q_t^* is too low, and the algorithms convert too early. On average, the algorithms of [3] achieve a result δ=40.08% better than the worst-case. In markets without trend results are biased for all RP algorithms, and they also tend to convert too early.

4. Conclusions

In case the input data processed by an online algorithm does not represent the worst-case scenario, their performance is considerably better than the worst-case competitive ratio tells. This result is consistent with [21], as we report a great disparity between the worst-case and the empirical-case results. We could strictly order the algorithms based on their results. In terms of c^{ex} $RP(M,m,f(q_t))$ (resp. $RP(M,m,f(q_t),T)) < RP(M,m) < RP(M,T)$. In contrast, in terms of c^{wc} $RP(M,m) < RP(M,m,f(q_t))$ (resp. $RP(M,m,f(q_t),T)) < RP(M,T)$. The algorithms of [3] require a return function, and their performance strongly depends on $f(q_t)$. It would be of interest if the algorithms still outperform in case another return function is used, and whether or not the knowledge of T improves the result of $RP(M,m,f(q_t),T)$.

From the practical point of view, the considered RP algorithms are only favorable in decreasing markets as they tend to convert (too) early. All the computed RP are too small, and the so-called *too-early-error* occurs in non-decreasing markets where the underlying r_t follow an extreme value distribution. The algorithms could have achieved M but get the RP in the worst-case (cf. [1]).

We extend former works by using more than one time series, and thus our results are less effected by randomness in terms of 'luckily' behaving data. Our results show that an extreme value distribution is proper to model a realistic volatility as it includes crashes, or situations of extreme stress on investor portfolios.

References

[1] R. El-Yaniv, Competitive solutions for online financial problems, ACM Computing Surveys 30 (1) (1998) 28–69. doi:10.1145/274440.274442.

[2] P. Damaschke, P. Ha, P. Tsigas, Online search with time-varying price bounds, Algorithmica 55 (4) (2009) 619–642. doi:10.1007/s00453-007-9156-9.

[3] Y. Xu, W. Zhang, F. Zheng, Optimal algorithms for the online time series search problem, Theoretical Computer Science 412 (3) (2011) 192–197. doi:10.1016/j.tcs.2009.09.026.

[4] D. Sleator, R. Tarjan, Amortized efficiency of list update and paging rules, Communications of the ACM 28 (2) (1985) 202–208. doi:10.1145/2786.2793.

[5] R. El-Yaniv, A. Fiat, R. Karp, G. Turpin, Optimal search and one-way trading algorithm, Algorithmica 30 (1) (2001) 101–139. doi:10.1007/s00453-001-0003-0.

[6] J. Ni, C. Zhang, An efficient implementation of the backtesting of trading policies, in: Proceedings of the third international Conference on Parallel and Distributed Processing and Applications (ISPA'05), Springer, Heidelberg, 2005, pp. 126–131. doi:10.1007/11576235_17.

[7] R. Karp, On-line algorithms versus off-line algorithms: How much is it worth to know the future?, in: Proceedings of the IFIP twelveth World Computer Congress, Vol. 1, 1992, pp. 416–429.

[8] K. Spremann, Portfoliomanagement, Oldenburg Wissenschaftsverlag, 2006.

[9] W. Leigh, N. Paz, R. Purvis, Market timing: A test of a charting heuristic, Economics Letters 77 (1) (2002) 55–63. doi:10.1016/S0165-1765(02)00110-6.

[10] R. Cont, Empirical properties of asset returns: Stylized facts and statistical issues, Quantitative Finance 1 (2) (2001) 223–236. doi:10.1080/713665670.

[11] A. Borodin, R. El-Yaniv, Online Computation and Competitive Analysis, Cambridge University Press, 1998.

[12] R. El-Yaniv, R. Kaniel, N. Linial, Competitive optimal on-line leasing, Algorithmica 25 (1) (1999) 116–140. doi:10.1007/PL00009279.

[13] P. Boothe, D. Glassman, The statistical distribution of exchange rates: Empirical evidence and economic implications, Journal of International Economics 33 (3-4) (1987) 297–319. URL `http://EconPapers.repec.org/RePEc:eee:inecon:v:22:y:1987:i:3-4:p:297-319`

[14] T. Bollerslev, R. Chou, K. Kroner, ARCH modeling in finance: A review of the theory and empirical evidence, Journal of Econometrics 52 (1-2) (1992) 5–59. doi:10.1016/0304-4076(92)90064-X.

[15] W. Brock, P. de Lima, Handbook of Statistics, Vol. 14, Elsevier Science B.V., 1996, Ch. 11– Nonlinear time series, complexity theory, and finance, pp. 317–361. doi:10.1016/S0169-7161(96)14013-X.

[16] H. Fujiwara, K. Iwama, Y. Sekiguchi, Average-case competitive analyses for one-way trading, Journal of Combinatorial Optimization 21 (1) (2011) 83–107. doi:10.1007/s10878-009-9239-4.

[17] D. Hoaglin, F. Mosteller, J. Tukey (Eds.), Understanding Robust and Exploratory Data Analysis, John Wiley & Sons, Inc., 2000.

[18] W. Brock, J. Lakonishok, B. LeBaron, Simple technical trading rules and the stochastic properties of stock returns, Journal of Finance 47 (5) (1992) 1731–1764. doi:10.2307/2328994.

[19] J. Hosking, J. Wallis, E. Wood, Estimation of the generalized extreme-value distribution by the method of probability-weighted moments, Technometrics 27 (3) (1985) 251–261.

URL http://www.jstor.org/stable/1269706

[20] C. Combes, A. Dussauchoy, Generalized extreme value distribution for fitting opening/closing asset prices and returns in stock-exchange, 6, 3-26, 2006, Operational Research 6 (1) (2006) 3–26. doi:10.1007/BF02941135.

[21] G. Schmidt, E. Mohr, M. Kersch, Experimental analysis of an online trading algorithm, Electronic Notes in Discrete Mathematics 36 (2010) 519–526. doi:10.1016/j.endm.2010.05.066.

[22] A. Fiat, G. Woeginger (Eds.), Online Algorithms - The State of the Art, Vol. 1442 of Lecture Notes in Computer Science, Springer, 1998.

[23] C. Anagnostopoulos, S. Hadjiefthymiades, Delay-tolerant delivery of quality information in ad hoc networks, Journal of Parallel and Distributed Computing 71 (7) (2010) 974–987. doi:10.1016/j.jpdc.2010.11.007.

[24] L. Becchetti, Modeling locality: A probabilistic analysis of LRU and FWF, in: ESA'04, 2004, pp. 98–109. doi:10.1007/978-3-540-30140-0_11.

[25] K. Panagiotou, A. Souza, On adequate performance measures for paging, in: STOC'06, 2006, pp. 487–496. doi:10.1145/1132516.1132587.

[26] N. Naaman, R. Rom, Average case analysis of bounded space bin packing algorithms, Algorithmica 50 (1) (2008) 72–97. doi:10.1007/s00453-007-9073-y.

[27] N. Garg, A. Gupta, S. Leonardi, P. Sankowski, Stochastic analyses for online combinatorial optimization problems, in: Proceedings of the nineteenth annual ACM-SIAM Symposium on Discrete Algorithms, ACM, 2008, pp. 942–951. doi:10.1145/1347082.1347185.

7

Risk Aware Reservation Price Algorithms

Summary of Results

As risk is an unavoidable phenomenon in financial markets, an investor prefers a strategy that provides the flexibility to manage her risk level. Online algorithms under classical competitive analysis are risk averse in nature as the algorithms guarantee a worst case competitive ratio, eliminating the risk. The worst case scenario does not occur on a regular basis and the investor always has some information about the trends of the market. This information may not necessarily be true but can be used by the investor. In case the assumed information is true, the investor earns a higher reward (a lower competitive ratio), otherwise a loss is observed (a higher competitive ratio).

Al-binali [4] extended the classical competitive analysis approach to a risk-reward framework, allowing the investor to manage her risk level. We consider the non-preemptive, reservation price algorithm of El-Yaniv [25] and extend it to incorporate risk management. The proposed algorithms (for uni and bi-directional conversion) achieve an improved competitive ratio when the outcome is favorable; otherwise the observed competitive ratio is worsened. In either case, the resultant competitive ratio depends on the risk level of the player. In order to satisfy Al-binali's condition, we show that the worst case competitive ratio is not arbitrarily bad but is bounded. We also perform an experimental study to compare our proposed algorithms with non-preemptive reservation price algorithms of El-Yaniv [25]. We report that as the risk level increases the performance of algorithms is improved. However, when the risk level goes beyond 1.24, the performance degrades and thus the competitive ratio is increased.

Publication

I. Ahmad and G. Schmidt: *Risk Aware Reservation Price Algorithms,* Submitted to: *OMEGA* - The International Journal of Management Science.
Status : Submitted.

Risk Aware Reservation Price Algorithms

Iftikhar Ahmad * †1 and Günter Schmidt ‡1,2

[1]Chair of Operations Research and Business Informatics, Saarland University, P.O. Box 151150, D-66041 Saarbrücken, Germany, Phone +49-681-302-4559.
[2]Department of Statistical Sciences,University of Cape Town, South Africa.

Abstract

Online algorithms for conversion problems are designed under the competitive analysis paradigm. The drawback of competitive analysis is the lack of risk management aspect required in financial markets. We consider the reservation price algorithm of El-Yaniv et al. (2001) and extend it to incorporate risk and reward. We present risk-aware algorithms for both uni-directional conversion and bi-directional conversion problems. We show that our proposed algorithms achieve a better (lower) competitive ratio when an improved outcome is observed, otherwise a competitive ratio inferior to the optimal competitive ratio is achieved. However, the inferior competitive ratio is not arbitrarily bad and is bounded based on the risk level of the player.

Keywords: Operational/OR, Online Algorithms, Competitive Analysis, Risk Reward Framework, Risk

1 Introduction

Financial markets are highly volatile and risky. The investor (henceforth called as player) in these markets is subjected to take decisions under a high degree of uncertainty and is exposed to high level of risk. An example of such financial market is currency exchange. In order to assist the player a number of algorithms and tools have been developed to facilitate the decision making under uncertain circumstances [2, 5]. Online algorithms for conversion problems are among a number of proposed solutions to solve such problems [1, 5].

In theoretical computer science, competitive analysis is a standard tool for the design and analysis of online algorithms [2, 5]. Although competitive analysis is a strong tool, it has one inherent drawback. Online algorithms designed under competitive analysis paradigm assume a worst case scenario and attempt to achieve an optimal performance for the unforeseen worst case. This approach leads to conservative decision making which in turn results in algorithms that are based on risk mitigation rather than risk management [2]. Therefore, in real world the classical competitive analysis is not an ideal tool to design online algorithms for financial problems.

In this paper, we consider online conversion problem [5] and discuss how we can design risk-aware algorithms? The proposed solution is a simple and an elegant way of introducing risk management in reservation price algorithms. The resultant algorithms are flexible enough to provide the player an opportunity to earn higher reward as well as ensuring that performance is never degraded below a fixed threshold. To the best of our knowledge this is the first attempt to introduce risk management in reservation price algorithms. This work will bridge the gap between theoretical computer science and its application domain in business and finance.

*corresponding author
†ia@orbi.uni-saarland.de
‡gs@orbi.uni-saarland.de

The rest of the paper is organized as follows. In Section 2 we discuss the conversion problem [5] and define competitive analysis. This is followed by the introduction of risk-reward framework in Section 3. In Section 4 we present the online algorithm of El-Yaniv et al. [5] and the competitive analysis is discussed in the same section. Section 5 discusses how we can design risk-aware online algorithms for conversion problems and the competitive analysis is performed to show the reward gained by the online algorithms under the risk-reward framework. Section 6 reports the findings of an experimental study to compare our proposed algorithms with that of El-Yaniv et al. [5]. Section 7 concludes the work by highlighting the application of proposed idea and discusses some open questions.

2 Problem Formulation

In an online conversion problem, a player has a fixed amount of initial currency D at the start of the game. On each day t ($t = 1, 2, \dots, T$) the player is offered a price q_t. Without any knowledge about future prices q_j ($j = t+1, t+2, \dots, T$), the player has to make a conversion decision. The player may accept an offered price q_t and convert the whole of D to the desired currency Y or may reject q_t and wait for a better price. The decision taken is irrevocable. The game ends when the player accepts a price or on the last day T when D must be converted into Y at the last offered price q_T. The player can have two objectives, i) to maximize the amount of Y or ii) to maximize the amount of D at the end of the game. i) is called *uni-directional conversion* and forbids the conversion of Y back to D whereas ii) is called *bi-directional conversion* and allows the player to convert Y back to D. In *uni-directional conversion*, there is a single investment horizon in which all the transactions must be carried out whereas in *bi-directional conversion* the total time T is divided into two investment horizons of equal lengths $T/2$. In the first period D is converted into Y and in the second period Y is converted back into D. In either case the objective of the player is to maximize the terminal wealth in form of D or Y.

A number of online algorithms are proposed to solve online conversion problems [1, 5]. These algorithms are classified in to two classes based on the investment pattern, namely *non-preemptive* and *preemptive* algorithms. Non-preemptive algorithms invest the whole wealth at one point of time whereas preemptive algorithms divide the wealth in to small portions and invest a portion of the wealth when an investment decision is made. Assume that the total wealth of the player is normalized to 1 unit and let s_t be the amount of wealth invested at any given time t, then for non-preemptive algorithms $s_t \in \{0, 1\}$ and for preemptive algorithms $s_t \in [0, 1]$.

Competitive analysis measures the performance of an online algorithm against an optimum offline algorithm. Unlike an online algorithm, an optimum offline algorithm OPT knows the whole input in advance and always makes an optimum decision. Competitive ratio can formally be defined as [5]; let $\mathcal{P} = (\mathcal{I}, \mathcal{F}, \mathcal{U})$ be a maximization problem where $\mathcal{I}$ is the set of all possible inputs; $\forall I \in \mathcal{I}, \mathcal{F}(I)$ is the set of feasible outputs; $\mathcal{U}$ is a utility function such that $\forall I \in \mathcal{I}$ and $O \in \mathcal{F}(I), \mathcal{U}(I, O) \in \mathbb{R}$. Let ON be an online algorithm for $\mathcal{P}$, given an input I, ON computes a feasible solution $O \in \mathcal{F}$. Let $ON(I)$ denotes the performance of ON on $I \in \mathcal{I}$ and the performance of OPT on $I \in \mathcal{I}$ is denoted by $OPT(I) = sup_{O \in \mathcal{F}(I)} \mathcal{U}(I, O)$. ON is c-competitive iff $\forall I \in I$,

$$ON(I) \geq \frac{1}{c} OPT(I) \tag{1}$$

A c-competitive online algorithm for a maximization problem guarantees a return of $1/c$ to that of an optimum offline algorithm.

Al-binali [2] extended the classical approach of competitive analysis to the risk-reward framework. The framework is based on the principle that the player must be able to decide his risk tolerance level (the maximum level of risk that he is willing to take) and implement an algorithm that respects the risk tolerance level. In the next section, we present the basic risk principle and the framework of Al-binali [2].

3 Risk Reward Framework

A decision making problem can lead to two possible outcomes, i) a risk-free outcome, based on risk-free selection made by the player and ii) a risky outcome, based on the risk level of the player [2]. A risk-free outcome leads to a certain fixed outcome whereas a risky outcome can lead to one of the following two outcomes. An improved outcome - the player return is greater than the risk-free action and a worse outcome - the player return is less than the risk-free outcome. Fig. 1 is a schematic representation of the risk-free and risky outcomes. The classical competitive analysis corresponds to risk-free selection as it does not give any selection choice to the player and employs a risk-free strategy to ensure a guaranteed competitive ratio.

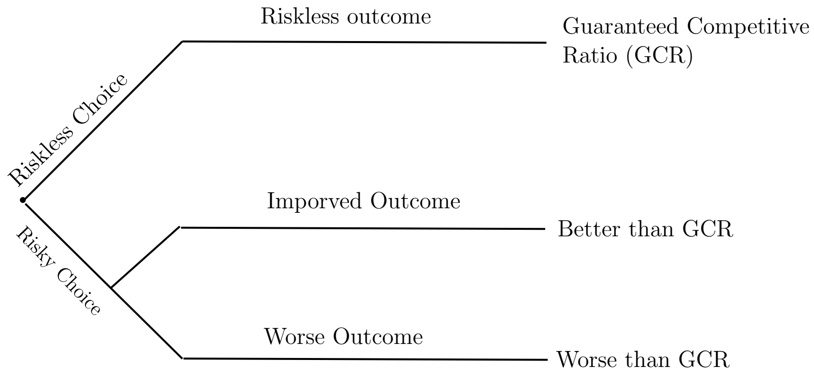


Figure 1: A view of risk and reward comprising of different outcomes based on the player's choice [2]

Al-binali [2] pointed out that classical competitive analysis is an inflexible approach for online algorithms in financial markets. The classical competitive analysis leads to the development of algorithms that reduces the uncertainty and risk. However, in the real world investors are keener to manage their level of risk. Al-binali [2] proposed a risk-reward framework which extends the classical competitive analysis to enable the design of risk-aware algorithms. Let A be an online algorithm for a maximization problem P, from Eq. (1), we know that competitive ratio c_A

$$c_A = \sup_{I \in \mathcal{I}} \frac{OPT(I)}{A(I)}. \tag{2}$$

The optimal competitive ratio for the problem is

$$c^* = \inf_A c_A. \tag{3}$$

Risk of algorithm A can be defined as

$$\frac{c_A}{c^*}.$$

Al-binali [2] referred to the risk from investor's point of view as the maximum opportunity cost that algorithm A may incur over optimal online algorithm. Let α is the risk tolerance level of the investor such that $\alpha \geq 1$, then

$$S_\alpha = \{A | c_A \leq \alpha c^*\} \tag{4}$$

be the set of all algorithms that respect the investors risk tolerance level. Based on the outcome there are two possible scenarios, i) the risk taken by the player results in an improved outcome and the performance is improved ii) the worse outcome where the performance is degraded. Therefore, based on the outcome either the player is rewarded or the performance is degraded (see Fig: 1).

For the positive outcome the reward of the investor is defined as an improvement over the optimal online competitive ratio and is given as

$$R_A = \frac{c^*}{\hat{c}_A} \tag{5}$$

Where $\hat{c}_A$ is the competitive ratio of A (without using risk tolerance level) in cases where an improved outcome is observed.

Remark 1. *The reward R_A of an online risk-aware algorithm A is bounded below by 1 and above by c^*, i.e., $R_A \in [1, c^*]$ [2].*

Remark 2. *From Eq. (4), we can see that any risk-aware algorithm must not perform worse than αc^*.*

Remark 2 upper bounds the performance of risk-aware algorithm, i.e., if a worse outcome is observed, the competitive ratio must not be greater than αc^*. All risk-aware algorithms must respect this condition.

Al-binali [2] used the framework for the design of risk aware preemptive algorithms. The work in based on the assumption that the player has some piece of information about the future in the form of forecast. For example, a forecast may state that in the future the prices will reach some minimum level. The forecast may or may not be true. The idea is to invest more prudently by investing less at the start of the investment horizon so that comparatively more wealth is left to invest when the forecast becomes true. As the framework is mainly designed for preemptive algorithms, there is no work to design risk aware non-preemptive algorithms.

Behavioral finance divides an investor into different categories based on their tendency to invest in risky assets [3]. Bailard et al. [3] classified investors into five different categories based on the personality characteristics. Bailard et al. [3] suggested that each category uses a different approach to risk and thus prefers different risk levels. The five categories are;

i *Adventurers* : The investors who are willing to take high risks. They are willing to invest in assets where the returns are expected to be high although such investment can be highly risky.

ii *Celebrity* : The people who do not have real knowledge of the market but wants to invest in assets that are popular in the market. As they have no real working knowledge, they rely on others for their investment decisions.

iii *Individualists*: The people who are confident, methodical and take their own decisions on investment strategies. They take risk where appropriate but are not tempted by higher risks for higher returns .

iv *Guardian*: The group of people who are worried about their investments and do not prefer to take risks. Their preferred choice of investment includes securities where fixed returns are guaranteed.

v *Straight Arrows* : The group of people who do not fit in the above mentioned four classes. Generally they are willing to take medium amount of risk but can be conservative in their choices as well.

The classical competitive ratio is a good tool for *guardian* as they are mainly risk averse in nature. The guaranteed competitive ratio can be an ideal tool for such class as the returns are guaranteed. Risk-reward framework can be used by *individualists.* As this group is more methodical in their decision making, and are not extravagant risk takers as well, the risk-reward framework is more suited to such individuals. The framework allows them the flexibility to set an appropriate level of risk.

We proceed to discuss the non-preemptive reservation price algorithm presented by El-Yaniv et al. [5] and discuss how to modify the algorithm for incorporating the risk-reward framework.

4 Reservation Price Algorithm (RP)

A reservation price algorithm computes a threshold price q^* and compares each offered price against q^*. For max-search (sell) the algorithm accepts a price q_t if $q_t \geq q^*$ and invests the whole wealth at one point of time. Contrary to non-preemprive algorithms, preemptive threat based algorithms [5] do not invest at one point of time but instead invest a portion of wealth when the offered price is acceptable. The criterion for price acceptability varies and the reader is referred to [5].
El-Yaniv et al. [5] presented a reservation price algorithm assuming that the player knows the value of possible lower bound m and upper bound M of prices. It is pertinent to note that if the online player does not assume the a priori knowledge of m and M, it is not possible to derive an online algorithm with bounded competitive ratio [5]. It is not mandatory that these bounds are observed, however all prices must be in this range, i.e., $q_t \in [m, M]$.

4.1 Reservation Price Algorithm for Uni-directional Conversion

Algorithm 1. $RP(Uni|m, M)$

- *Accept the first offered price q_t, such that $q_t \geq q^* = \sqrt{Mm}$.*

El-Yaniv et al. [5] showed that $RP(Uni|m, M)$ is $\sqrt{M/m}$ competitive. In order to make the paper self-contained, we present the result in Theorem 1.

Theorem 1. *$RP(Uni|m, M)$ is $\sqrt{M/m}$ competitive.*

Proof. We need to show that the reservation price q^* calculated by $RP(Uni|m, M)$ is optimal. The online player can commit either of the following two types of errors when calculating q^*.
q^* **is too low** : If the reservation price q^* calculated by the player is too low, the adversary offers q^* early in the input sequence and later raises the offered price to maximum M. The input sequence is of the following form;

$$q_1 = q^*, \ldots, q_T = M.$$

The online player could have converted at M but instead converts at q^*, the competitive ratio achieved by the online algorithm ON (Algorithm 1) is

$$\begin{aligned} c_1 &= \frac{OPT}{ON} \\ &= \frac{M}{q^*} \end{aligned} \tag{6}$$

q^* **is too high**: If the q^* is too high, the adversary offers a price sequence such that the maximum observed price is less than q^*. The input sequence is of the following form;

$$q_1 = q^* - \epsilon, \ldots, q_T = m.$$

The online player will convert on the last offered price m whereas OPT will convert on $q^* - \epsilon$. The competitive ratio achieved is

$$\begin{aligned} c_2 &= \frac{OPT}{ON} \\ &= \frac{q^* - \epsilon}{m} \quad (0 < \epsilon << 1) \\ &\leq \frac{q^*}{m} \end{aligned} \tag{7}$$

In order to calculate a balanced q^*, we use the error-balancing technique [5] such that competitive ratio in both cases holds.

$$\begin{aligned} c_1 &= c_2 \\ \frac{M}{q^*} &= \frac{q^*}{m} \\ q^* &= \sqrt{Mm} \end{aligned} \tag{8}$$

Considering $q^* = \sqrt{Mm}$, $RP(Uni|m, M)$ achieves a competitive ratio $\sqrt{M/m}$ in both "too low" and "too high" errors. Therefore, the resultant competitive ratio $c^w(Uni|m, M) = \sqrt{M/m}$. □

4.2 Reservation Price Algorithm for Bi-directional Conversion

For bi-directional conversion, we divide the investment horizon of length T in two equal parts, one each for buying and selling. In the first part we execute the algorithm for buying and in the second part we execute the algorithm for selling.

Algorithm 2. $RP(Bi|m, M)$

- *Buy: Buy at the first offered price* q_t *such that* $q_t \leq q^* = \sqrt{Mm}$.
- *Sell: Sell at the first offered price* q_t *such that* $q_t \geq q^* = \sqrt{Mm}$.

Theorem 2. *The competitive ratio achieved by* $RP(Bi|m, M)$ *is* M/m.

Proof. In order to prove the theorem, we construct a worst case sequence. As the time horizon is divided into two equal parts of length $K = T/2$, therefore in the first K days a buy transaction must be performed whereas on $K+1$ the search for a sell price begins. Following is the worst case input sequence;

$$q_1 = q^*, \ldots, q_K = m, q_{K+1} = q^*, \ldots, q_T = M.$$

ON (Algorithm 2) will buy at the first price $q_1 = q^*$ and OPT will buy at $q_K = m$, similarly, ON will sell on $q_{K+1} = q^*$ whereas OPT will sell on the maximum offered price $q_T = M$. The return of OPT (r_{OPT}) is thus M/m whereas the return of ON (r_{ON}) is $q^*/q^* = 1$. The competitive ratio achieved by the online bi-directional algorithm is

$$\begin{aligned} c^w(Bi|m, M) &= \frac{r_{OPT}}{r_{ON}} \\ &= \frac{M/m}{1} \\ &= \frac{M}{m} \end{aligned} \tag{9}$$

□

5 Risk Aware Reservation Price Algorithms

Iwama and Yonezwama [6] presented risk-aware online uni-directional currency conversion algorithms using the framework of Al-binali [2]. The work extended the threat based algorithm of El Yaniv et al. [5]. To the best of our knowledge, there is no significant work to design risk-aware non-preemptive reservation price algorithms and our work is the first such attempt.

The work of Al-binali [2] is based on *forecast*. A *forecast* is a piece of information about the future prices. The forecast can either be true or false. Instead of considering the forecast, we use a more generalized term "knowledge" of the player. Knowledge of the player may be based on some forecast mechanism or may stem from the experience of the player. Similarly it can be based on the intuition of the player as well. Knowledge is a more coherent term as it encompasses not only the available forecast but also includes others factors that are hard to quantify e.g., intuition and experience. Irrespective of the source of knowledge, the player gains the insight that the worst case scenario may not occur that frequently. The player is willing to take risk in order to achieve higher returns. If the knowledge (trend) that he predicts is true, a higher return than optimal online algorithm is gained otherwise a lower return than optimal online algorithm is observed.

In this section we extend Algorithm 1 and Algorithm 2 to incorporate risk management based on the proposed framework of Al-binali [2]. We show that competitive ratio achieved by the modified algorithms satisfy the Al-binali's condition as given in Eq. (4).

5.1 Risk Aware Uni-directional Algorithm

In the following we present a reservation price algorithm for uni-directional conversion incorporating risk-reward framework. Let $\alpha \geq 1$ be the acceptable level of risk that the player is willing to take for a better performance.

Algorithm 3. *$RP(Uni|m, M, \alpha)$*

- *Accept the first offered price q_t, such that $q_t \geq \alpha\sqrt{Mm}$.*

Theorem 3. *$RP(Uni|m, M, \alpha)$ is $\alpha\sqrt{M/m}$ competitive.*

Proof. We have already shown in Theorem 1 that $\sqrt{M/m}$ is the optimal competitive ratio; therefore, we only need to construct a worst case sequence where $RP(Uni|m, M, \alpha)$ achieves a competitive ratio of $\alpha\sqrt{M/m}$. We consider the following two cases;
Case 1 : $q_t \in [m, \alpha\sqrt{Mm} - \epsilon]$: The adversary knows that the online player is searching for a reservation price which is at least $\alpha\sqrt{Mm}$. The adversary offers a price sequence in which the maximum offered price is less than $\alpha\sqrt{Mm}$. The sequence is of the following form;

$$\alpha\sqrt{Mm} - \epsilon, \ldots, m$$

ON ($RP(Uni|m, M, \alpha)$) does not find a price which is at least $\alpha\sqrt{Mm}$ and converts on the last offered price m, whereas optimum offline algorithm OPT converts at $\alpha\sqrt{Mm} - \epsilon$. The competitive ratio is;

$$\begin{aligned} c'_1 &= \frac{OPT}{ON} \\ &= \frac{\alpha\sqrt{Mm} - \epsilon}{m} \\ &\leq \alpha\sqrt{\frac{M}{m}} \end{aligned} \tag{10}$$

Case 2 : $q_t \in [\alpha\sqrt{Mm}, M]$: In this case, the adversary offers a price sequence in which $\alpha\sqrt{Mm}$ appears early in the input sequence so that conversion is made at $\alpha\sqrt{Mm}$. Later the adversary raises the offered price to M. The worst case sequence is of the following form;

$$\alpha\sqrt{Mm}, \ldots, \ldots, M$$

The competitive ratio achieved by the online player is;

$$\begin{aligned} c'_2 &= \frac{OPT}{ON} \\ &= \frac{M}{\alpha\sqrt{Mm}} \\ &= \frac{1}{\alpha}\sqrt{\frac{M}{m}} \end{aligned} \tag{11}$$

The resultant competitive ratio $c^w(Uni|m, M, \alpha)$ can be derived from the maximum of c'_1 and c'_2. Therefore, $c^w(Uni|m, M, \alpha) = \max\{c'_1, c'_2\} = c'_1 \leq \alpha\sqrt{M/m}$. □

Case 1 represents the competitive ratio when the outcome is not improved whereas *Case 2* represent the competitive ratio when the outcome is favorable.

Corollary 1. *$RP(Uni|m, M, \alpha)$ satisfies Al-binali's condition by achieving a competitive ratio $c^w(Uni|m, M, \alpha)$ such that $c^w(Uni|m, M, \alpha) \leq \alpha c$. Here c is the competitive ratio of $RP(Uni|m, M)$ and α is the acceptable level of risk tolerance of the player.*

Proof. In order to maximize the loss of the online algorithm, the adversary offers all prices less than the reservation price $\alpha\sqrt{Mm}$. On the last day, the adversary offers the minimum price m. As the online algorithm does not find an offered price which is at least $\alpha\sqrt{Mm}$, she is forced to convert on the last offered price m. The resultant competitive ratio is $\alpha\sqrt{M/m}$. The competitive ratio is worsened by a factor α, which also bounds the loss of the online algorithm. □

Lemma 1. *When the outcome is improved, the minimum reward of $RP(Uni|m, M, \alpha)$ is α.*

Proof. If an improved outcome is observed and with a risk tolerance level of α, suggests that the algorithm observes at least one price q_t such that $q_t \geq \alpha\sqrt{Mm}$ (see Case 2). The algorithm converts on $\alpha\sqrt{Mm}$. In order to maximize the competitive ratio, the adversary will raise the price to M, the competitive ratio achieved will be $c' = M/(\alpha\sqrt{Mm}) = (1/\alpha)\sqrt{M/m}$. Recall, $c^* = \sqrt{M/m}$ is the optimal competitive ratio achieved by reservation price algorithm ($RP(Uni|m, M)$). The reward R of $RP(Uni|m, M, \alpha)$ is;

$$\begin{aligned} R &= \frac{c^*}{c'} \\ &= \frac{\sqrt{M/m}}{1/\alpha\sqrt{M/m}} \\ &= \alpha \end{aligned} \tag{12}$$

□

5.2 Risk Aware Bi-directional Algorithm

In this section, we extend $RP(Bi|m, M)$ to introduce the risk management aspect. Recall that α is the acceptable level of risk tolerance of the online player. The algorithm works as follows;

Algorithm 4. $RP(Bi|m, M, \alpha)$

- *Buy: Buy at the first offered price q_t such that $q_t \leq \frac{1}{\sqrt{\alpha}}\sqrt{Mm}$.*
- *Sell: Sell at the first offered price q_t such that $q_t \geq \sqrt{\alpha}\sqrt{Mm}$.*

Theorem 4. *$RP(Bi|m, M, \alpha)$ is $\alpha\frac{M}{m}$ competitive.*

Proof. We construct a worst case sequence to demonstrate that $RP(Bi|m, M, \alpha)$ cannot achieve a competitive ratio better than $\alpha\frac{M}{m}$. As for bi-directional, we consider an investment horizon of length T and divide it in two equal parts each of length $K = T/2$. In the first period, search for a buy price is made whereas in the second period a sell price is sought. The worst case sequence is of the following form;

$$q_1 = \frac{1}{\sqrt{\alpha}}\sqrt{Mm} + \epsilon, \ldots, q_K = M, q_{K+1} = \sqrt{\alpha}\sqrt{Mm} - \epsilon, \ldots, q_T = m. \tag{13}$$

In the first K days the online player is searching for a buy price which is not greater than $(1/\sqrt{\alpha})\sqrt{Mm}$. On the first day, the adversary offers $q_1 > (1/\sqrt{\alpha})\sqrt{Mm}$, after day 1, all the prices offered are more than q_1. On the last day K of the buying period, the adversary offers M, that must be accepted by the online player as K is the last day of the "buying period". Thus, online algorithm buys on M whereas optimum offline algorithm buys on $(1/\sqrt{\alpha})\sqrt{Mm} + \epsilon$. Likewise in the selling period, the online player searches for a price which is at least $\sqrt{\alpha}\sqrt{Mm}$. However, the adversary offers $\alpha\sqrt{Mm} - \epsilon$ on day $K + 1$ and all the subsequent prices are less than q_{K+1}. On the last day T the adversary reveals the last price m which the online player must accept. Thus the online player sells at m whereas the optimum offline algorithm sells at $\alpha\sqrt{Mm} - \epsilon$. Let r_{OPT} be the return of optimum offline algorithm and r_{ON} be the return of the online algorithm (Algorithm 4).

$$\begin{aligned} r_{OPT} &= \frac{\sqrt{\alpha}\sqrt{Mm} - \epsilon}{1/\sqrt{\alpha}\sqrt{Mm} + \epsilon} \quad (0 < \epsilon << 1) \\ &= \alpha \end{aligned} \tag{14}$$

$$r_{ON} = \frac{m}{M} \tag{15}$$

Thus the competitive ratio $c^w(Bi|m, M, \alpha)$ is;

$$\begin{aligned} c^w(Bi|m, M, \alpha) &= \frac{r_{OPT}}{r_{ON}}, \\ &= \frac{\alpha}{m/M}, \\ &= \alpha \frac{M}{m}. \end{aligned} \tag{16}$$

□

Corollary 2. *$RP(Bi|m, M, \alpha)$ respects Al-binali's condition by achieving a competitive ratio $c^w(Bi|m, M, \alpha)$ such that $c^w(Bi|m, M, \alpha) \leq \alpha c^w(Bi|m, M)$. The maximum loss is thus bounded by α.*

Proof. From the worst case sequence given in Eq. (13), the return of OPT is α where as the return of online algorithm ($RP(Bi|m, M, \alpha)$) is m/M. Therefore, the competitive ratio achieved by online algorithm is $\alpha(M/m)$. The maximum loss is therefore bounded by α observed in a case when the outcome is worsened. □

6 Experimental Evaluation

Ahmad and Schmidt [1] performed an extensive experimental evaluation of online uni-directional conversion problems. The study considered a variety of algorithms and evaluated the selected set of algorithms on real world data of DAX30 and S&P500 indices. The set of algorithms also included reservation price algorithm of El-Yaniv et al. [5] ($RP(Uni|m, M)$, Algorithm 1). However, the study did not consider any risk-aware algorithms. In the following, we consider the same experimental setting as in Ahmad and Schmidt [1] and compare $RP(Uni|m, M)$ and $RP(Bi|m, M)$ to our proposed risk-aware algorithms $RP(Uni|m, M, \alpha)$, and $RP(Bi|m, M, \alpha)$. We consider the following experimental set up.

Experimental Setup

i We consider the real world data of DAX30 and S&P500 for ten years from 1.Jan.2001 to 31.Dec.2010.

ii We consider a transaction fee (or commission) of 0.025% of the volume transacted.

iii The yearly interest rate is zero.

iv Algorithms have a priori information about m and M.

v For risk-aware reservation price algorithm $RP(Uni|m, M, \alpha)$ and $RP(Bi|m, M, \alpha)$, we consider risk level of $\alpha = \{1.1, 1.2\}$.

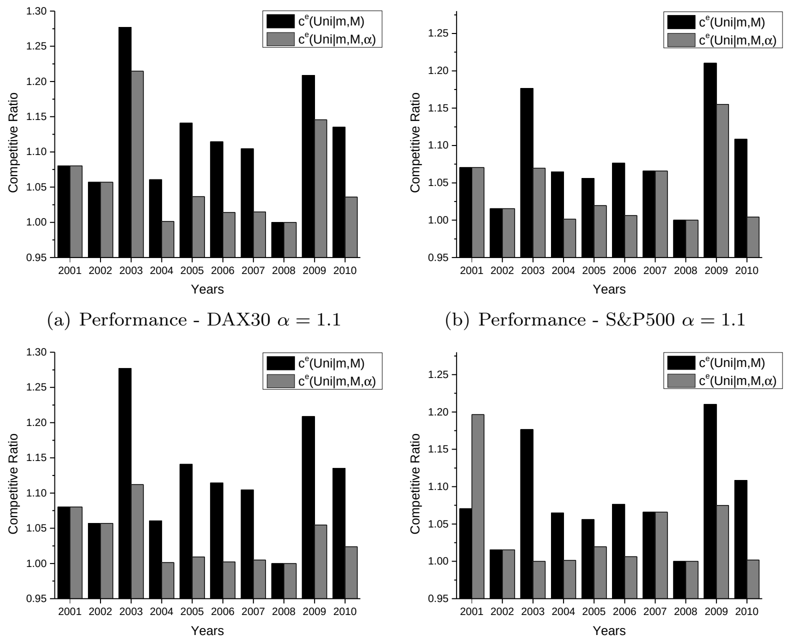
vi The experiments are run on the subset of data such that the length of each sub set corresponds to one year of investment horizon.

vii For each subset of data (of length one year), we record the worst case competitive ratio c^w and the experimentally observed competitive ratio c^e.

Results

Uni-directional Results

Considering DAX30 data set, we observed that the average performance of $RP(Uni|m, M, \alpha)$ is better than that of $RP(Uni|m, M)$. For risk level $\alpha = \{1.1, 1.2\}$, $RP(Uni|m, M, \alpha)$ achieves an average competitive ratio 1.06 and 1.03. For the same data set $RP(Uni|m, M)$ achieves a competitive of 1.1178. $RP(Uni|m, M, \alpha)$ improves the competitive ratio by 5% (for $\alpha = 1.1$)



(a) Performance - DAX30 $\alpha = 1.1$

(b) Performance - S&P500 $\alpha = 1.1$

(c) Performance - DAX30 $\alpha = 1.2$

(d) Performance - S&P500 $\alpha = 1.2$

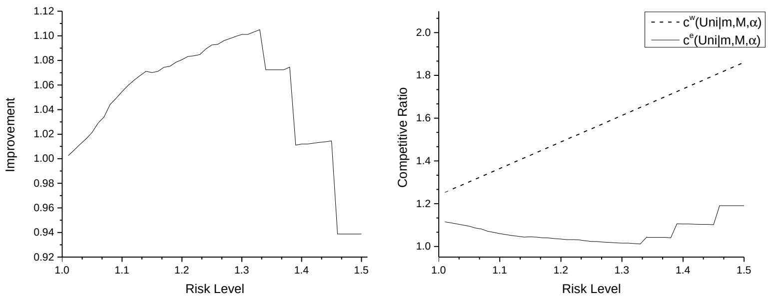
Figure 2: Performance on DAX30 and S&P500 for Uni-directional Conversion

and 8% (for $\alpha = 1.2$) respectively. $RP(Uni|m, M, \alpha)$ outperforms $RP(Uni|m, M)$ on S&P500 as well. The corresponding improvement in the competitive ratio is 3.7% (for $\alpha = 1.1$) and 3.8% (for $\alpha = 1.2$). Fig 2 is a pictorial representation of the competitive ratio over DAX30 and S&P500 data sets. It can be seen that for $\alpha = 1.1$, $RP(Uni|m, M, \alpha)$ performs better than $RP(Uni|m, M)$ on both $DAX30$ and S&P500 data sets, except for the years 2001, 2002 and 2008. Similarly, for $\alpha = 1.2$, we observe the same performance except for S&P500 where on 2001 data, $RP(Uni|m, M, \alpha)$ performs better than $RP(Uni|m, M)$.

For risk level $\alpha = \{1.1, 1.2\}$, we observe that on both data sets (DAX30 and S&P500), $RP(Uni|m, M, \alpha)$ outperforms $RP(Uni|m, M)$. We do not observe any case where the performance of $RP(Uni|m, M, \alpha)$ is inferior to that of $RP(Uni|m, M)$ except on S&P500 for $\alpha = 1.2$ and year 2002. Therefore, we considered the risk level $\alpha = \{1.01, 1.02, \ldots, 1.5\}$ and record the worst case $c^w(Uni|m, M, \alpha)$ and experimental case $c^e(Uni|m, M, \alpha)$ competitive ratios of $RP_U(m, M, \alpha)$.

For DAX30, we observe a consistent improvement in competitive ratio of $RP(Uni|m, M, \alpha)$ to that $RP(Uni|m, M)$ (see Fig 3(a)) for risk level $\alpha \in [1.01, 1.33]$. However, there is a sudden drop in the improvement factor ($\frac{c^e(m,M)}{c^e(m,M,\alpha)}$) when $\alpha = 1.34$. At $\alpha = 1.33$, the improvement factor is 1.1049 which is the maximum improvement, at $\alpha = 1.34$, the improvement factor is reduced to 1.0724 and from there on it decreases. At $\alpha = 1.46$, the improvement factor is below 1.0, which means that $RP(Uni|m, M)$ outperforms $RP(Uni|m, M, \alpha)$. Fig 3(b) models the performance of $RP(Uni|m, M, \alpha)$ based on $c^w(Uni|m, M, \alpha)$ and $c^e(Uni|m, M, \alpha)$. The worst case competitive ratio $c^w(Uni|m, M, \alpha) = \alpha\sqrt{M/m}$ steadily increases with the increase in risk level α. In contrast to $c^w(Uni|m, M, \alpha)$, $c^e(Uni|m, M, \alpha)$ decreases with increasing risk level till $\alpha = 1.34$. Afterward the gap between $c^w(Uni|m, M, \alpha)$ and $c^e(Uni|m, M, \alpha)$ is reduced for $\alpha = [1.34, 1.5]$.

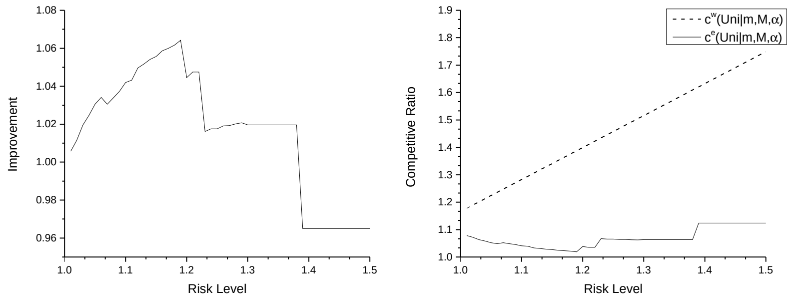
S&P500 exhibits the same behavior as DAX30. Comparing the performance of $RP(Uni|m, M, \alpha)$ to $RP(Uni|m, M)$, we observe a steady improvement for risk level $\alpha = [1.01, 1.29]$ (see Fig 4(a)). The maximum improvement is observed at $\alpha = 1.29$. The performance of $RP(Uni|m, M, \alpha)$ is inferior to $RP(Uni|m, M)$ for $\alpha = [1.39, 1.5]$. Likewise, we observe the same level of disparity



(a) DAX30: Improvement Factor

(b) DAX30: Disparity between c^w and c^e of $RP(Uni|m, M, \alpha)$

Figure 3: Improvement Factor and Disparity on DAX30 for Uni-directional Conversion



(a) S&P500: Improvement Factor

(b) S&P500: Disparity between c^w and c^e of $RP(Uni|m, M, \alpha)$

Figure 4: Improvement Factor and Disparity on S&P500 for Uni-directional Conversion

between $c^w(Uni|m, M, \alpha)$ and $c^e(Uni|m, M, \alpha)$ for S&P500. This is depicted in Fig 4(b)

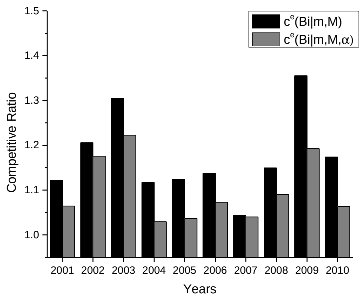
Bi-directional Results

For bi-directional scenario, we observe that our proposed algorithm $RP(Bi|m, M, \alpha)$ using the risk-reward principle performs better than $RP(Bi|m, M)$. For both data sets (DAX30, S&P500) and risk level $\alpha = \{1.1, 1.2\}$, $RP(Bi|m, M, \alpha)$ achieves an average $c^e(Bi|m, M, \alpha)$ better than $c^e(Bi|m, M)$. For data set DAX30 and $\alpha = 1.1$, we observe that $c^e(Bi|m, M, \alpha)$ is 6% better (lower) than $c^e(Bi|m, M)$, whereas for $\alpha = 1.2$ the corresponding improvement is 8%. The same pattern in improvement is observed for S&P500 data set where the improvement using $RP(Bi|m, M, \alpha)$ is 5% and 7% respectively for $\alpha = \{1.1, 1.2\}$. Fig 5 compares the performance of $c^e(Bi|m, M, \alpha)$ and $c^e(Bi|m, M)$ on DAX30 and S&P500.

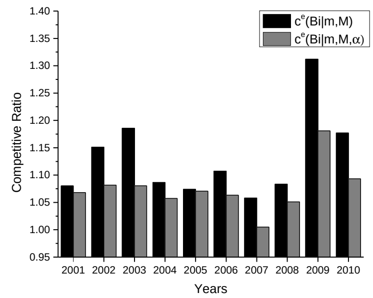
In order to observe the improvement of $RP(Bi|m, M, \alpha)$ over $RP(Bi|m, M)$ and the disparity between $c^w(Bi|m, M, \alpha)$ and $c^e(Bi|m, M, \alpha)$, we consider risk level $\alpha = \{1.01, 1.02, \ldots, 1.5\}$.

For DAX30, the improvement factor $\frac{c^e(Bi|m,M)}{c^e(Bi|m,M,\alpha)}$ and the disparity between $c^w(Bi|m, M, \alpha)$ and $c^e(Bi|m, M, \alpha)$ is summarized in Fig 6. We observe that improvement factor increases with increase in α , the maximum improvement factor 1.1 is observed at $\alpha = 1.25$. For $\alpha = \{1.25, 1.26, \ldots 1.5\}$ an irregular pattern in the performance is observed, although the main trend is decreasing after $\alpha = 1.25$ (see Fig 6(a)). We observe a wide gap between $c^w(Bi|m, M, \alpha)$ and $c^e(Bi|m, M, \alpha)$ for DAX30. The disparity is illustrated in Fig 6(b).

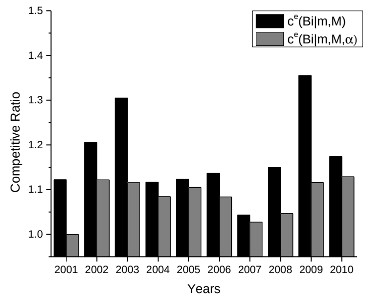
We observe an identical pattern for the improvement factor on S&P500. There is a steady increase in improvement factor as α increases. The maximum improvement is observed at $\alpha = 1.24$ after which a steady decrease in the improvement factor is observed. The same pattern for the disparity between $c^w(Bi|m, M, \alpha)$ and $c^e(Bi|m, M, \alpha)$ is observed for S&P500



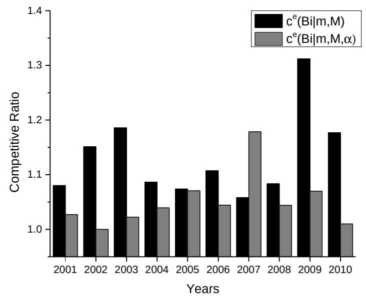
(a) Performance - DAX30 $\alpha = 1.1$



(b) Performance - S&P500 $\alpha = 1.1$



(c) Performance - DAX30 $\alpha = 1.2$

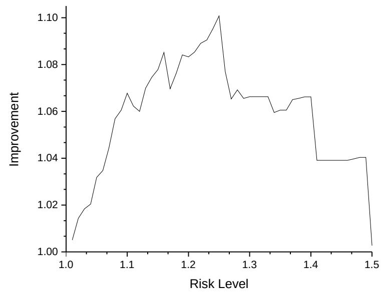


(d) Performance - S&P500 $\alpha = 1.2$

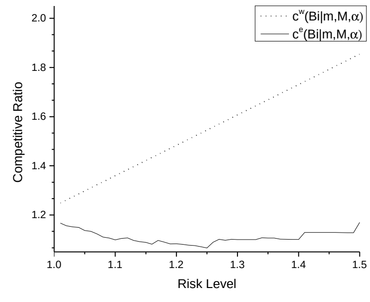
Figure 5: Performance on DAX30 and S&P500 for bi-directional conversion

as is seen on DAX30. Fig 7 illustrates the improvement factor (Fig 7(a)) and the disparity (Fig 7(b)) on S&P500 data set.

From the experimental results, we observe that the improvement factor $\frac{c^e(Bi|m,M)}{c^e(Bi|m,M,\alpha)}$, achieves its maximum value on DAX30 when $\alpha = 1.25$, for S&P500 the maximum improvement factor is observed for $\alpha = 1.24$. Although no conclusive decision can be drawn as the data is limited to two indices only, risk level $\alpha = [1.20, 1.25]$ seems to be the optimal risk level where maximum improvement can be gained.

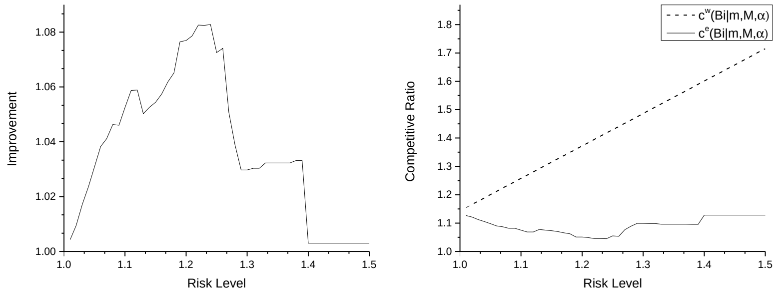


(a) DAX30: Improvement Factor



(b) DAX30: Disparity between c^w and c^e of $RP(Bi|m, M, \alpha)$

Figure 6: Improvement Factor and Disparity on DAX30 for bi-directional conversion



(a) S&P500: Improvement Factor

(b) S&P500: Disparity between c^w and c^e of $RP(Bi|m, M, \alpha)$

Figure 7: Improvement Factor and Disparity on S&P500 for bi-directional conversion

7 Conclusion

We studied online conversion problem under the competitive analysis and presented risk-aware algorithms for uni-directional conversion and bi-directional conversion problems. Our proposed approach is the first attempt to introduce risk management in non-preemptive reservation price algorithms. The proposed approach can be applied to other algorithms for conversion problems such as the algorithms proposed in technical analysis [4]. Examples of such algorithms are moving averages and trading range break out et cetera. Moving average also behaves likes a reservation price algorithm as it invests at one point of time. The other resemblance that moving average has with reservation price is inherent lack of risk management. Our proposed approach can be applied to algorithms of technical analysis [4] and other reservation price algorithms.

In this work, we considered the assumption that the outcome of the risk based strategy is either improved or worsened. The outcome is forecast dependent, which can either be true or false. It will be of interest to design reservation price risk-aware online algorithms which are based on the probability function of the forecast rather than boolean values of the forecast, i.e., we assign a probability ρ to forecast being true. For example, how the reservation price can be adjusted based on the value of ρ and what is the impact of ρ on the competitive ratio?

References

[1] I. Ahmad and G. Schmidt. An experimental analysis of online unidirectional conversion problem. In Ch. Huemer and P. Lops, editors, *E-Commerce and Web Technologies*, volume 123 of *Lecture Notes in Business Information Processing*, pages 176–187. Springer, 2012.

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

[3] T. E. Bailard, D.L. Biehl, and R.W. Kaiser. *Personal Money Management*. Chicago Science Research Associates, Inc., 5 edition, 1986.

[4] W.A. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47:1731–1764, 1992.

[5] R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Optimal search and one-way trading algorithm. *Algorithmica*, 30:101–139, 2001.

[6] K. Iwama and K. Yonezawa. Using generalized forecasts for online currency conversion. In T. Asano and et al., editors, *Computing and Combinatorics*, volume 1627 of *Lecture Notes in Computer Science*, pages 409–421. Springer, 1999.

8

Conclusion and Future Work

This chapter concludes the work by summarizing the thesis and provides future directions for new research.

8.1 Conclusion

Online algorithms for conversion problems designed under competitive analysis paradigm face a number of problems when the applicability of such algorithms in real world is considered. The objective of the work was to perform an analysis of online uni-directional conversion problems and discuss methods and measures to improve their adaptability in real world.

A drawback of using competitive analysis as a design tool for online algorithms is the lack of risk management aspect. As competitive analysis guarantees a risk free outcome in the worst unforeseen future, the resulting algorithm is risk mitigating in nature. However, in a real world, an online player will like to manage her risk level rather than mitigate it. In order to bridge this gap, we show that competitive ratio can be used as a risk measure as it satisfies all the required axioms of a *Coherent Risk Measure* suggested by Artzner et al. [7].

We evaluated a selected set of preemptive as well non-preemptive online conversion algorithms on the real world data using the backtesting technique. The objective was to find how algorithms perform on the real world data and to discuss the disparity between the worst case and experimentally achieved competitive ratio. We observed that preemptive algorithm of El-Yaniv et al. [27] with a priori knowledge of lower and upper bounds of offered prices out performs other algorithms on both $DAX30$ and $S\&P500$ data sets. The main reason for better performance of aforementioned algorithm is the investing technique. The algorithm invests only when it observes a price which is the highest seen so far. Similarly, the amount of wealth invested also depends on the offered price. This results in an improved performance of the algorithm. We also observed a significant gap between the worst case and experimentally achieved competitive ratio.

The problem with backtesting is relying on limited data and thus the results might suffer from "data snooping" bias. In order to avoid the data snooping bias, we also employed bootstrap method to generate artificial data sets for algorithm's evaluation in Ahmad and Schmidt [2]. However the bootstrap method does not replicate all possible real world scenarios. In order to address the drawback of bootstrap method, we propose to use *Extreme Value Theory* (EVT) approach to generate test instances in Mohr, Ahmad and Schmidt [50]. The proposed EVT approach encapsulates extreme scenarios such as market crash as well as normal trading scenarios.

Competitive analysis results in algorithms that are risk averse in nature. However, in real world an online player (investor) wants the flexibility to manage the risk rather than mitigate it. In order to empower the investor to manage her risk level we propose risk aware reservation price algorithms that allow the online player to manage the risk. If a positive outcome is observed, the online player achieves a better competitive ratio than the worst case competitive ratio otherwise a sub-optimal competitive ratio is observed. However, the sub-optimal competitive ratio is bounded and is not arbitrarily bad.

8.2 Future Works

Online algorithms for conversion problems are mostly designed with worst case competitive analysis [20, 47, 27]. Fujiwara et al. [33] used average case competitive analysis to design a threat based algorithm. It will be of interest to design online algorithms for conversion problems using other methods such as using diffuse adversary and comparative analysis approach [42] or using the smoothed complexity approach suggested by Spielman and Teng [63]. Boyar et al. [14] considered the reservation price algorithm of El-Yaniv et al. [27] and analyzed it under different performance evaluation measures such as average analysis, random order analysis and relative worst order analysis. A challenging research prospect will be to compare the threat based algorithm of El-Yaniv et al. [27] with that of Fujiwara et al. [33] using bijective analysis, random order analysis, relative worst order analysis and other performance measures.

Online algorithms for conversion problems are designed assuming different a priori information about the future such as information about lower and upper bound of offered prices. In real world, these informations' are based on estimated values which are error prone. As there are errors in the assumed information, it will be of interest to perform the competitive analysis of online conversion algorithms assuming errors in estimation. It should be investigated how the errors in estimation affect the guaranteed competitive ratio. Another drawback of the assumed information of lower and upper bound of prices is that bounds are assumed to be static and does not change with time. A possible research prospect will be to consider the initial estimated values and update them using a price function. As more and more prices are observed the initial estimates can be improved and the reservation price can thus be modified.

Al-binali [4] extended the classical competitive analysis approach to risk-reward framework. The objective is to allow online player the flexibility to manage her risk level for a higher reward. The idea behind the work is that worst cases do not occur frequently and in real world online player has information about future in the form of forecasts. Based on the forecast, the online player can invest prudently, i.e., investing less at the start so as to save more for later when prices will be favorable. Al-binali [4] assumed that forecasts can either be true or false, but in real world each forecast is assigned a probability. An open question is to design online algorithms where forecasts assume some probability rather than boolean values.

Lorenz et al. [47] proposed k-max search algorithm for a scenario where an online player wants to sell k units of an asset. The online player calculates k reservation prices $q_1^*, q_2^*, \ldots, q_k^*$. As the price sequence is unfolded, the online player waits for the first offered price which is at least q_1^* and sells one unit. Similarly, the online player sells another unit when the offered price is at least q_2^* and so on. The player is restricted to sell only one unit even when the observed price is good enough to warrant the selling of more units. Zhang et al. [71] extended Lorenz et al. [47] max search by allowing the player to sell more than one unit when the price is favorable. An open question will be to extend Zhang et al. [71] to incorporate risk-reward framework. The online player should be allowed to choose a risk level and depending on risk level, reservation prices and number of units sold can be modified.

Bibliography

[1] C. Acerbi. Spectral measures of risk: A coherent representation of subjective risk aversion. *Journal of Banking & Finance*, 26(7):1505 – 1518, 2002.

[2] I. Ahmad and G. Schmidt. An experimental analysis of online unidirectional conversion problem. In Ch. Huemer and P. Lops, editors, *E-Commerce and Web Technologies*, volume 123 of *Lecture Notes in Business Information Processing*, pages 176–187. Springer, 2012.

[3] S. Ahmed, U. Cakmak, and A. Shapiro. Coherent risk measures in inventory problems. *European Journal of Operational Research*, 182(1):226–238, 2007.

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

[5] S. Albers, L. M. Favrholdt, and O. Giel. On paging with locality of reference. *J. Comput. Syst. Sci.*, 70:145–175, March 2005.

[6] S. Angelopoulos, R. Dorrigiv, and A. López-Ortiz. On the separation and equivalence of paging strategies. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '07, pages 229–237, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[7] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, July 1999.

[8] T.G. Bali. The generalized extreme value distribution. *Economics Letters*, 79(3):423–427, June 2003.

[9] T. B. Beielstein, M. Chiarandini, L. Paquete, and M. Preuss, editors. *Experimental Methods for the Analysis of Optimization Algorithms*. Springer-Verlag, Berlin Heidelberg, 2010.

[10] J. Beirlant, F. Caeiro, and M.I. Gomes. An overview and open research topics in statistics of univariate extremes. *REVSTAT - Statistical Journal*, 10(1):1–31, March 2012.

[11] S. Ben-David and A. Borodin. A new measure for the study of on-line algorithms. *Algorithmica*, 11:73–91, 1994.

[12] J. Berkowitz and L. Kilian. Recent developments in bootstrapping time series. *Econometric Reviews*, 19(1):1–48, 2000.

[13] J. Boyar and L. M. Favrholdt. The relative worst order ratio for online algorithms. *ACM Trans. Algorithms*, 3(2), May 2007.

[14] J. Boyar, K. Larsen, and A. Maiti. A comparison of performance measures via online search. In J. Snoeyink, P. Lu, K. Su, and L. Wang, editors, *Frontiers in Algorithmics and Algorithmic Aspects in Information and Management*, volume 7285 of *Lecture Notes in Computer Science*, pages 303–314. Springer Berlin Heidelberg, 2012.

[15] W.A. Brock, J. Lakonishok, and B. LeBaron. Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5):1731–1764, 1992.

[16] G.-H. Chen, M.-Y. Kao, Y.-D. Lyuu, and H.-K. Wong. Optimal buy-and-hold strategies for financial markets with bounded daily returns. *SIAM Journal on Computing*, 31(2):447–459, 2001.

[17] A. Chou, J. Cooperstock, R. El-Yaniv, M. Klugerman, and T. Leighton. The statistical adversary allows optimal money-making trading schemes. In *Proceedings of the sixth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 467–476, 1995.

[18] S.G. Coles and M.J. Dixon. Likelihood-based inference of extreme value models. *Extremes*, 2(1):5–23, 1999.

[19] C. Combes and A. Dussauchoy. Generalized extreme value distribution for fitting opening/closing asset prices and returns in stock-exchange. *Operational Research*, 6(1):3–26, 2006.

[20] P. Damaschke, P.H. Ha, and P. Tsigas. Online search with time-varying price bounds. *Algorithmica*, 55(4):619–642, 2009.

[21] E. Dannoura and K. Sakurai. An improvement on El-Yaniv-Fiat-Karp-Turpin's money-making bi-directional trading strategy. *Information Processing Letters*, 66(1):27–33, 1998.

[22] J. Diebolt, A. Guillou, P. Naveau, and P. Ribereau. Improving probability-weighted moment methods for the generalized extreme value distribution. *REVSTAT - Statistical Journal*, 6(1):33–50, March 2008.

[23] R. Dorrigiv, A. López-Ortiz, and J. I. Munro. On the relative dominance of paging algorithms. *Theor. Comput. Sci.*, 410(38-40):3694–3701, September 2009.

[24] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap.* Chapman & Hall, New York, 1993.

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

[26] R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Competitive analysis of financial games. In *IEEE Symposium on Foundations of Computer Science*, pages 327–333, 1992.

[27] R. El-Yaniv, A. Fiat, R.M. Karp, and G. Turpin. Optimal search and one-way trading algorithm. *Algorithmica*, 30(1):101–139, 2001.

[28] M. Fan, J. Stallaert, and A. B. Whinston. A web-based financial trading system. *Computer*, 32(4):64 –70, 1999.

[29] M. Fan, J. Stallaerta, and A.B. Whinston. The design and development of a financial cybermarket with a bundle trading mechanism. *International Journal Electronic Commerce*, 4(1):5–22, Fall 1999.

[30] H. Föllmer and A. Schied. Convex measures of risk and trading constraints. *Finance and Stochastics*, 6(4):429–447, 2002.

[31] D. A. Freedman and S. C. Peters. Bootstrapping a regression equation: Some empirical results. *Journal of the American Statistical Association*, 79(385):97–106, 1984.

[32] M. Frittelli and E. R. Gianin. Putting order in risk measures. *Journal of Banking & Finance*, 26(7):1473 – 1486, 2002.

[33] H. Fujiwara, K. Iwama, and Y. Sekiguchi. Average-case competitive analyses for one-way trading. *J. Comb. Optim.*, 21(1):83–107, 2011.

[34] T. M. Guilderman and J. Longerstay. Riskmetrics technical document. Technical report, JP Morgan Market Risk Research Group, 1994.

[35] J.R.M. Hosking, J.R. Wallis, and E.F. Wood. Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics*, 27(3):251–261, 1985.

[36] S. Hu, Q. Guo, and H. Li. Competitive analysis of on-line securities investment. In N. Megiddo, Y. Xu, and B. Zhu, editors, *Algorithmic Applications in Management*, volume 3521 of *Lecture Notes in Computer Science*, pages 224–232. Springer, 2005.

[37] J. Iqbal, I. Ahmad, and G. Schmidt. Can online trading algorithms beat the market? An experimental evaluation. In S. Ravizza and P. Holborn, editors, *3rd Student Conference on Operational Research*, volume 22 of *OpenAccess Series in Informatics (OASIcs)*, pages 43–52, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[38] A.F. Jenkinson. The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Quarterly Journal of the Royal Meteorological Society*, 81(348):158–171, April 1955.

[39] K.-Y. Kao and S.R. Tate. On-line difference maximization. *SIAM Journal on Discrete Mathematics*, 12(1):78–90, 1999.

[40] R. J. Kauffman and E. A. Walden. Economics and electronic commerce: Survey and directions for research. *International Journal of Electronic Commerce*, 5(4):5 –116, 2001.

[41] C. Kenyon. Best-fit bin-packing with random order. In *Proceedings of the seventh annual ACM-SIAM symposium on Discrete algorithms*, SODA '96, pages 359–364, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.

[42] E. Koutsoupias and C.H. Papadimitriou. Beyond competitive analysis. *SIAM Journal on Computing*, 30(1):300–317, 2000.

[43] P. Krokhmal, M. Zabarankin, and S. Uryasev. Modeling and optimization of risk. *Surveys in Operations Research and Management Science*, 16(2):49–66, 2011.

[44] K.-Y. Kwon and R.J. Kish. A comparative study of technical trading strategies and return predictability: An extension of Brock, Lakonishok, and LeBaron (1992) using NYSE and NASDAQ indices. *Quarterly Review of Economics and Finance*, 42(3):611–631, 2002.

[45] S. N. Lahiri. Theoretical comparisons of block bootstrap methods. *The Annals of Statistics*, 27(1):386–404, 1999.

[46] H. Li and G. S. Maddala. Bootstrapping time series models. *Econometric Reviews*, 15(2):115–158, 1996.

[47] J. Lorenz, K. Panagiotou, and A. Steger. Optimal algorithms for k-search with application in option pricing. *Algorithmica*, 55(2):311–328, 2009.

[48] D. G. Luenberger. *Investment Science*. Oxford University Press, 2013.

[49] E. Mohr. *Online Algorithms for Conversion Problems*. PhD thesis, Saarland University and Max Planck Institute for Informatics, July 2011.

[50] E. Mohr, I. Ahmad, and G. Schmidt. Solving uni-directional conversion problems by online algorithms under an extreme value distribution. In *DSOR Contributions to Information Systems, Band 8: Applied Mathematical Optimization and Modelling (APMOD)*, pages 359–470. L. Suhl, G. Mitra, C. Lucas, A. Koberstein, L. Beckmann, 2012.

[51] E. Mohr and G. Schmidt. Empirical analysis of an online algorithm for multiple trading problems. In H.A. Le Thi, P. Bouvry, and T. Pham Dinh, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, volume 14 of *Communications in Computer and Information Science (CCIS)*, pages 293–302. Second International Conference MCO 2008, Metz, France - Luxembourg, September 8-10, 2008, Springer, 2008.

[52] E. Mohr and G. Schmidt. Trading in financial markets with online algorithms. In B. Fleischmann, K.-H. Borgwardt, R. Klein, and A. Tuma, editors, *Operations Research Proceedings 2008*, pages 33–38. Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), University of Augsburg, September 3-5, 2008, Springer, 2008.

[53] J. Ni and C. Zhang. An efficient implementation of the backtesting of trading policies. In *Proceedings of the third international Conference on Parallel and Distributed Processing and Applications*, ISPA'05, pages 126–131, Berlin, Heidelberg, 2005. Springer.

[54] K. Panagiotou and A. Souza. On adequate performance measures for paging. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 487–496, New York, NY, USA, 2006. ACM.

[55] J. Pickands. Statistical inference using extreme order statistics. *Annals of Statistics*, 3(1):119–131, January 1975.

[56] D. N. Politis and J. P. Romano. The stationary bootstrap. *Journal of the American Statistical Association*, 89(428):1303–1313, 1994.

[57] P. Raghavan. A statistical adversary for on-line algorithms. In L.A. McGeoch and D.D. Sleator, editors, *On-line algorithms: Proceedings of a DIMACS Workshop*, volume 7 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 79–83. ACM, February 11-13 1992.

[58] M. Ratner and R.P.C. Leal. Tests of technical trading strategies in the emerging equity markets of Latin America and Asia. *Journal of Banking and Finance*, 23(12):1887–1905, 1999.

[59] G. Schmidt, E. Mohr, and M. Kersch. Experimental analysis of an online trading algorithm. *Electronic Notes in Discrete Mathematics*, 36:519–526, 2010.

[60] P. Shen. Market-timing strategies that worked. *Journal of Portfolio Management*, 29(2):57–68, 2003.

[61] K. Singh and M. Xie. Bootstrap: A statistical method, 2008.

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

[63] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *J. ACM*, 51:385–463, May 2004.

[64] R. Sullivan, A. Timmermann, and H. White. Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 54(5):1647–1691, October 1999.

[65] G. Szegö. Measures of risk. *European Journal of Operational Research*, 163(1):5–19, 2005.

[66] B.M. Tabak and E.J.A. Lima. Market efficiency of Brazilian exchange rate: Evidence from variance ratio statistics and technical trading rules. *European Journal of Operational Research*, 194(3):814–820, 2009.

[67] E. Torng. A unified analysis of paging and caching. *Algorithmica*, 20:194–203, 1998.

[68] C.F.J. Wu. Jackknife, bootstrap and other resampling methods in regression analysis. *The Annals of Statistics*, 14(4):1261–1295, 1986.

[69] Y. Xu, W. Zhang, and F. Zheng. Optimal algorithms for the online time series search problem. *Theoretical Computer Science*, 412(3):192 – 197, 2011.

[70] W. Zhang, Y. Xu, F. Zheng, and Y. Dong. Optimal algorithms for online time series search and one-way trading with interrelated prices. *Journal of Combinatorial Optimization*, 23(2):159–166, February 2012.

[71] W. Zhang, Y. Xu, F. Zheng, and M. Liu. Online algorithms for the general k-search problem. *Information Processing Letters*, 111(14):678–682, July 2011.

A
Copy Right Permission

A.1 Copy Right Permission for Ahmad and Schmidt 2012

In the following, copy right permission from Springer is attached for the inclusion of the following research paper in this thesis.

I. Ahmad and G. Schmidt, *An Experimental Analysis of Online Unidirectional Conversion Problem,* In: C. Huemer and P. Lops (Eds.): EC-Web 2012, Lecture Notes in Business Information Processing, Vol. 123, pp. 176-187, 2012. Springer-Verlag Berlin Heidelberg 2012.

SPRINGER LICENSE
TERMS AND CONDITIONS

Jul 18, 2013

This is a License Agreement between Iftikhar Ahmad ("You") and Springer ("Springer") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Springer, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

License Number	3192010381549
License date	Jul 18, 2013
Licensed content publisher	Springer
Licensed content publication	Springer eBook
Licensed content title	An Experimental Analysis of Online Unidirectional Conversion Problem
Licensed content author	Iftikhar Ahmad
Licensed content date	Jan 1, 2012
Type of Use	Thesis/Dissertation
Portion	Full text
Number of copies	100
Author of this Springer article	Yes and you are the sole author of the new work
Order reference number	
Title of your thesis / dissertation	Analysis of Algorithms for Online Uni-directional Conversion Problems
Expected completion date	Jul 2013
Estimated size(pages)	150
Total	0.00 EUR
Terms and Conditions	

Introduction
The publisher for this copyrighted material is Springer Science + Business Media. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at http://myaccount.copyright.com).

Limited License
With reference to your request to reprint in your thesis material on which Springer Science and

established by CCC on its behalf in its Billing and Payment terms and conditions for this licensing transaction.

Indemnity
You hereby indemnify and agree to hold harmless Springer Science + Business Media and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

No Transfer of License
This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without Springer Science + Business Media's written permission.

No Amendment Except in Writing
This license may not be amended except in a writing signed by both parties (or, in the case of Springer Science + Business Media, by CCC on Springer Science + Business Media's behalf).

Objection to Contrary Terms
Springer Science + Business Media hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and Springer Science + Business Media (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

Jurisdiction
All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in The Netherlands, in accordance with Dutch law, and to be conducted under the Rules of the 'Netherlands Arbitrage Instituut' (Netherlands Institute of Arbitration).***OR:***

All disputes that may arise in connection with this present License, or the breach thereof, shall be settled exclusively by arbitration, to be held in the Federal Republic of Germany, in accordance with German law.

Other terms and conditions:

v1.3

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number RLNK501069277. Once you receive your invoice for this order, you may pay your invoice by credit card. Please follow instructions provided at that time.

Make Payment To:

B
Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In-noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Saarbrücken, Wednesday 2$^{\text{nd}}$ October, 2013

(Unterschrift)