A Risk-Reward Framework for the Competitive Analysis of Adaptive Trading Strategies

(Thesis Proposal)

(Technical Report: CUCS-019-97)

Sabah S. al-Binali
albinali@cs.columbia.edu

Department of Computer Science
Columbia University
New York, NY 10027

May 6, 1997

Abstract

Competitive analysis is concerned with minimizing a relative measure of performance. When applied to financial trading strategies, competitive analysis leads to the development of strategies with minimum relative performance risk. This approach is too inflexible. Many investors are interested in managing their risk: they may be willing to increase their risk for some form of reward. They may also have some forecast of the future.

We propose to extend competitive analysis to provide a framework in which investors may develop trading strategies based on their risk tolerance and forecast. We introduce a new, nonstochastic, measure of the risk of an online algorithm, and a reward metric that is in the spirit of competitive analysis. We then show how investors can select a strategy that maximizes their reward should their forecast be correct, whilst still respecting their risk tolerance.

Keywords: Adaptive trading strategies, Competitive analysis, Finacial trading strategies, Forecast, Online algorithms, Reward, Risk.

Contents

Chapter 1

Introduction

The fundamental problem in investing is the lack of knowledge of future events. One method to make up for the lack of this knowledge is to model future events using a stochastic model. Analysis of historical data usually provides estimates for the parameters of the model. Investment decisions are then based on properties that can be calculated from the model [8]. Often the most important goal is maximizing the expected return, but in many situations properties such as the variance of the return or the probability of ruin are also considered. For investors who feel that stochastic models are appropriate, especially using normal or lognormal distributions as the underlying model, useful trading strategies can be developed. However, there are drawbacks to this traditional approach that can limit its applicability:

- Finding a model that approximates the behaviour of future events is hard.
- The best models are usually too complex to analyze for arbitrary probability distributions, and consequently most analysis of these problems assume that the events follow a common probability distribution.

A different method of dealing with the lack of knowledge of future events is to move from an absolute measure of performance to a relative one [1, 2, 3, 4, 6]. Our setting is based on an approach suggested by El-Yaniv *et al.* [4] in which *competitive analysis* is used to study the performance of financial trading strategies without relying on stochastic assumptions. Competitive analysis is a technique that has been developed and used in computer science research in which the performance of a strategy is measured relative to some benchmark strategy. This benchmark strategy is usually a (theoretical) omniscient strategy that has full knowledge of future events.

Competitive analysis seeks to minimize the ratio of the performance of the benchmark strategy versus the performance of the trading strategy. This ratio is called the competitive ratio of the algorithm.

Another key issue for investors is risk [18]. In most cases, investors do not seek to minimize risk, but to manage it. Work has been done on quantifying risk [15] and on benefiting from increasing risk [16]. On the other hand, most analysis in computer science focuses on worst case performance. In effect, most analysis put forth by computer scientists is risk averse. Competitive analysis is no different: it analyzes the worst case performance of an algorithm. The measure is just one of relative performance rather than absolute performance. This leads us to the central contribution of our work: we develop a risk-reward framework in which strategies can be analyzed in terms of their riskiness and the potential benefits (reward) for using them.

The motivation for, and major application of, our work is the development of flexible trading strategies.

- We begin by proposing a new measure of the *risk* of a trading strategy. We define the risk of an online algorithm to be the ratio of the competitive ratio of the algorithm to the optimal competitive ratio. This risk measure breaks with the more prevalent measures that depend on stochastic assumptions, and is more in in keeping with the competitive analysis paradigm.

- We next introduce the notion of a *forecast*: what the investor anticipates the market will do in the future. Again, we depart from making stochastic assumptions, and investigate the type of forecasts that certain investors make. These forecasts only provide partial information about what may happen (such as "the price will increase by $5 at some point in the next 30 days.")

- The last piece of the framework is the specification of a *reward function*. The first step is to define the *restricted ratio* of a strategy: the competitive ratio of the strategy restricted to the class of inputs in which the forecast comes true. Our reward is then the ratio of the optimal competitive ratio to the restricted ratio (a measure of how much better the strategy does should the forecast come true).

Given this framework, the investor specifies a maximum acceptable risk level, a forecast of how he thinks the market will behave and then develops an algorithm that maximizes the reward should the forecast come true, but which does not exceed the investors risk level for any scenario.

The remainder of this proposal is organized as follows. In Chapter 2, we briefly review competitive analysis and describe its application to the analysis of financial trading strategies. In Chapter 3 we present some further analysis of previous work. In Chapter 4 we develop the risk-reward framework that allows the investor some discretion over his risk exposure and use a traditional online problem, the ski-rental problem, to highlight how such a concept can be incorporated into traditional competitive analysis. In Chapter 5, we use the risk-reward framework to analyze an online financial trading strategy. We conclude with Chapter 6, reviewing our contributions, and outlining the future direction of our research.

Chapter 2

Online Algorithms and Competitive Analysis

We begin this chapter with a brief review of competitive analysis. The rest of the chapter is devoted to previous work in competitive analysis that is relevant to our proposal.

2.1 Definitions and Concepts

An *online* algorithm is an algorithm that must make decisions as input becomes available, without the full knowledge of the future input. Trading strategies are online algorithms, since they make trading decisions without knowing future prices.

An *offline* algorithm is an algorithm that makes decisions with full knowledge of future events. A clairvoyant investor can devise an offline algorithm, as she knows all future prices. An offline trading algorithm can always maximize the return of the investor.

Competitive analysis is a method by which the performance of an online algorithm is measured relative to the performance of the optimal offline algorithm [17]. We now summarize competitive analysis, following [10] and [13]. Let A be an algorithm that solves an optimization problem. For each such problem there is an associated performance function which measures the quality of a solution for a given input. Let $\mathbf{perf}_A(\sigma)$ denote the performance of algorithm A on input σ, where σ is an instance of the problem Σ (i.e. $\sigma \in \Sigma$). The performance $\mathbf{perf} > 0$ and higher values indicate "better" performance. If A is an online algorithm, then A must produce part of its output as it receives its input. How the input and output are interleaved is problem dependent. We wish to analyze the performance of A, when A is an online algorithm.

A worst case analysis of the the performance of A (i.e. examining $\inf_\sigma \mathbf{perf}_A(\sigma)$) generally yields little information. This is because for most problems there are input sequences that cause any algorithm, whether online or offline, to perform poorly; sometimes arbitrarily poorly. Since an online algorithm can never perform better than the optimal offline algorithm, in these situations all that we can hope an online algorithm do is not perform too much worse than the optimal offline algorithm.

Let OPT denote the optimal offline algorithm. Define $\mathbf{perf}(\sigma)$ to be OPT's performance on input σ. That is,

$$\mathbf{perf}(\sigma) = \sup_A \mathbf{perf}_A(\sigma)$$

The *competitive ratio*, r_A, of an online algorithm A is defined as

$$r_A = \sup_\sigma \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$$

It follows immediately from these definitions that the competitive ratio is always greater than or equal to 1. The closer the competitive ratio is to 1, the better. We wish to find an algorithm with the smallest possible competitive ratio. This ratio is called the *optimal competitive ratio*, and is defined as

$$r^\star = \inf_A r_A = \inf_A \sup_\sigma \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$$

An online algorithm that achieves the optimal competitive ratio is called an *optimal online algorithm*. We will denote such an algorithm by $A^\star$. There exist problems for which the optimal competitive ratio is not finite, or very large. For such cases, competitive analysis is of little value.

The highest valued benefit that must be sacrificed because an investor chooses an investment strategy is the *opportunity cost* of the choice [7]. Opportunity cost is a well known measure, used to differentiate between economic alternatives. From a financial standpoint, the competitive ratio of an algorithm is a measure of the opportunity cost of the algorithm, since the return of OPT is always the highest possible benefit. Hence, the goal of traditional competitive analysis can be regarded as finding the algorithm with the minimum opportunity cost.

2.2 Refinements and Alternatives to Competitive Analysis

2.2.1 The Cost of an Algorithm

There is a dual definition of competitive analysis were we measure not the performance of an algorithm, but its cost. Let $\mathbf{cost}_A(\sigma)$ be the cost of the solution produced by algorithm A on input σ, where **cost** > 0 and higher values indicate "worse" cost. In this case, we seek to minimize the competitive ratio of the cost of the algorithm to the cost of OPT:

$$r^\star = \inf_A r_A = \inf_A \sup_\sigma \frac{\mathbf{cost}_A(\sigma)}{\mathbf{cost}(\sigma)}$$

Whether we use cost or performance depends on the nature of the problem. For some problems it is natural to use a performance measure (such as the trading problem of Section 2.3.1), whilst for other problems (such as the ski-rental problem of Section 4.3) it is more natural to use a cost measure.

2.2.2 The Hurwicz Criterion

A common criticism of the competitive ratio is that it is too pessimistic. The Hurwicz Criterion [9], as referred to in [11], is a weighted average of the pessimistic case (minimizing the maximum performance ratio, or "nature will always do its worst") and the optimistic case (minimizing the minimum performance ratio, or "nature will always do its best"). Denote by R_A the *best case competitive ratio* of an algorithm A:

$$R_A = \inf_\sigma \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$$

Let α be a fixed number between 0 and 1. Hurwicz defines the α-index of an algorithm A to be

$$\psi_\alpha^A = \alpha r_A + (1-\alpha) R_A$$

We would then like to find the algorithm $A^\star$ with the lowest α-index:

$$\psi_\alpha^{A^\star} = \inf_A \left(\alpha r_A + (1-\alpha) R_A\right)$$

The weight α is a measure of how we believe nature will behave. At one end of the scale, $\alpha = 0$ indicates complete optimisim that nature will do its best, and therefore we wish to minimize the best case competitive ratio. At the other end of the scale, $\alpha = 1$ indicates complete pessimism, and so the problem reduces to the competitve analysis approach of minimizing the worst case competitive ratio. Notice that the weight α is in some sense a general forecast of the future behaviour of the adversary.

2.3 Competitive Analysis of Financial Trading Strategies

In [4] El-Yaniv *et al.* apply competitive analysis to the problem of trading financial instruments. We now briefly summarize their major results. They begin with the *unidirectional conversion* *problem*, in which an online investor wants to convert a fixed amount of dollars to yen over a fixed period of time. Each day, the online investor is offered a new exchange rate, and he must decide how many dollars he wishes to convert to yen at that exchange rate. The measure of the performance of an algorithm is the number of yen obtained by the algorithm at the end of the last trading period. In the *bidirectional problem*, the investor may trade dollars for yen and yen for dollars, and the performance of an algorithm is the return[1] of the algorithm.

2.3.1 Unidirectional Trading

Our formulation of the unidirectional trading problem is slightly more general than in [4], which simplifies the analysis. Consider an (online) investor who starts with d_0 US dollars, all of which he wishes to convert to Japanese yen according to the following rules:

1. There are n trading periods T_i, where $i = 1, \dots, n$.
2. In each trading period T_i, the investor is offered an exchange rate e_i at which he may exchange dollars for yen.
3. In period T_i, the investor may exchange any amount $s_i \in [0, d_0 - \sum_{j=1}^{i-1} s_j]$ of the remaining dollars into $s_i e_i$ yen.

The optimal offline algorithm is to convert all the dollars at the maximum exchange rate (minimum price). If $U_j = \max_{i \le j} e_i$ then the maximum exchange rate is U_n, and so the online investor wishes to minimize the competitive ratio

$$r = \sup_{e_i} \frac{d_0 U_n}{\sum_{i=1}^{n} s_i e_i}$$

subject to the constraints:

1. $\forall i, s_i \ge 0$
2. $\sum_{i=1}^{n} s_i = d_0$

In [5], El-Yaniv *et al.* show that if the exchange rate and the number of trading periods is unbounded, then the optimal competitive ratio is unbounded. In the following Lemma, we generalize this result.

Lemma 2.1 *If the sequence of exchange rates for the unidirectional trading problem is unconstrained, then* $r^\star \ge n$.

[1] Return is the amount, expressed as a percentage, earned on the investor's original amount of money.

Proof: Consider any online algorithm A. There are two cases.

- There is an exchange rate sequence $\langle e_1, \ldots, e_{n-1}\rangle$ such that $\forall i = 1, \ldots, n-1$ the algorithm A spends $s_i^A \geq \frac{d_0}{n}$ dollars. Then set $e_n > U_{n-1}$. The optimal offline algorithm makes $d_0 e_n$ yen and A makes y_n^A yen where

$$y_n^A \leq \frac{d_0}{n} e_n + \sum_{i=1}^{n-1} s_i^A e_i \leq \frac{d_0}{n}\left(e_n + \sum_{i=1}^{n-1} e_i\right)$$

since $s_n^A \leq \frac{d_0}{n}$. Therefore

$$r_A \geq \frac{n}{1 + \frac{1}{e_n}\sum_{i=1}^{n-1} e_i} \to n \quad \text{as} \quad e_n \to \infty$$

- There is a trading period T_j and an exchange rate sequence $\langle e_1, \ldots, e_j\rangle$ such that $\forall i = 1, \ldots, j-1$ the algorithm A spends $s_i^A \geq \frac{d_0}{n}$ dollars and for any $e_j, s_j^A < \frac{d_0}{n}$. Then set $e_j > \max(1, U_{j-1})$ and $e_i = 1$ for all $i = j+1, \ldots, n$. The optimal offline algorithm makes $d_0 e_j$ yen and A makes y_n^A yen where

$$y_n^A \leq \frac{d_0}{n} e_j + \sum_{\substack{i=1 \\ i \neq j}}^{n} d_0 e_i$$

Therefore

$$r_A \geq \frac{n}{1 + \frac{1}{e_j}\sum_{\substack{i=1 \\ i \neq j}}^{n} e_i} \to n \quad \text{as} \quad e_n \to \infty$$

■

Lemma 2.2 *The competitive ratio of the algorithm which spends $s_i = \frac{d_0}{n}$ dollars in each period T_i is n.*

Proof: From Lemma 2.1, we have $r \geq n$. Since the above algorithm will always spend $\frac{d_0}{n}$ dollars per period, it will make at least $y_n \geq \frac{d_0}{n} U_n$ yen. Therefore, its competitive ratio is $r = \frac{d_0 U_n}{y_n} \leq n$. ■

Corollary 2.3 *The optimal competitive ratio for the unidirectional conversion problem with unconstrained exchange rates is n, and an optimal online algorithm is to convert $s_i = \frac{d_0}{n}$ dollars in each period T_i.*

Notice that the optimal online algorithm is what is called a *dollar cost averaging* strategy.

Lemma 2.1 means that the general unidirectional trading problem is not interesting, because reasonably low competitive ratios are not possible. To continue we need to make some assumptions about the exchange rate. One of the scenarios that El-Yaniv *et al.* [4] consider is that the exchange rate is bounded, $m \leq e_i \leq M$, for $i = 1, \ldots, n$, and that the online investor knows the bounds m, M. Will will adopt this assumption. This gives us the added constraint

3. $\forall i, m \leq e_i \leq M$

2.3.2 An Optimal Online Algorithm

El-Yaniv *et al.* [4] show that when the exchange rate is bounded, there exists a finite optimal competitive ratio, and present an optimal online algorithm. In this section we will present this algorithm. We use the following notation:

- d_i^A is the amount of dollars that a trading strategy A will have at the end of trading period T_i. All strategies start with an amount d_0. For OPT we write $A = 0$.
- $s_i^A = d_i^A - d_{i-1}^A$ is the amount of dollars traded by strategy A in period T_i.
- $y_i^A = \sum_{j=1}^{i} s_j^A e_j$ is the amount of yen that trading strategy A has at the end of trading period T_i.
- $U_i = \max_{j \leq i} e_j$ is the maximum exchange rate offered in the periods $T_1, \ldots, T_i$.

Then supposing that the optimal competitive ratio is $r^\star = r^\star(n, m, M, d_0, y_0)$, (we will show how to calculate $r^\star$ later), the optimal online algorithm is as follows:

Trading Strategy 1
Given m, M, n, d_0, and a new exchange rate e_i in period T_i that is bounded by m and M, then trading by the following rules will achieve a competitive ratio of $r^\star$ versus the optimal offline strategy:

1. Only trade when the exchange rate hits a new high (i.e. when $e_i > U_{i-1}$)
2. When executing a trade, only trade enough to guarantee a competitive ratio of $r^\star$ would be obtained should the exchange rate drop to m and remain there for the remainder of the trading periods. This means that $\frac{U_i \cdot d_0}{y_i^1 + m d_i^1} = r^\star$ which we show in Appendix A.1 implies that $s_i^1 = \frac{d_0}{r^\star} \frac{e_i - U_{i-1}}{e_i - m}$ where $e_0 \triangleq m r^\star$.

This strategy seeks to eliminate any risk of falling too far behind the optimal offline algorithm. El-Yaniv *et al.* call such a strategy a *threat based strategy*. In the risk-reward framework that we will develop, the term *minimum-risk* is more appropriate, and that is what we will use. The minimum-risk strategy will be our benchmark strategy, against which we will compare the performance of other strategies.

2.3.3 The Optimal Competitive Ratio

El-Yaniv *et al.* show that $r^\star$ is the solution to the equation

$$r^\star = n\left(1 - \left(\frac{m(r^\star - 1)}{M - m}\right)^{\frac{1}{n}}\right) \tag{2.1}$$

If the investor does not know the number of trading periods n in advance, he should use the competitive ratio $r_\infty = \lim_{n\to\infty} r^\star(m, M, n, d_0)$. This is the solution to the equation:

$$r^*_\infty = \ln \frac{M - m}{m(r^\star_\infty - 1)} \tag{2.2}$$

In Section 3 we give a further analysis of the optimal competitive ratio $r^\star$, including a table of $r^\star$ for various values of m, M and n.

2.3.4 Worst Case Exchange Rate Sequence

The adversary can guarantee that TS1's competitive ratio is $r^\star$ by forcing the exchange rate to be $\underline{\mathbf{E}}$. The worst case exchange rate $\underline{\mathbf{E}} = \{e_1, e_2, \dots, e_n\}$ as constructed by El-Yaniv *et al.* [5] is as follows:

1. Set $e_0 = mr^\star$
2. Set e_1 to the root of the equation
$$\left(\frac{e_1 - m}{M - m}\right)^{\frac{1}{n-1}} = \frac{m(n-1)}{e_1 + mn - m}$$
3. Iteratively construct the rest of the exchange rate sequence using
$$e_{i+1} = \frac{(e_i - m)^2}{e_{i-1} - m} + m$$

In Appendix A.2 we derive a closed form solution for the worst case exchange rate sequence $\underline{\mathbf{E}}$.

2.3.5 Non-optimal Adversary and an Adaptive Strategy

Our analysis in Appendix A.2 shows that there is only one exchange rate sequence that causes the minimum achievable competitive ratio to be $r^\star$. Therefore, in general, we expect exchange rates will not be worst case. An investor can take advantage if offered an exchange rate $e_i \neq \underline{e}_i$ by starting a new game, recomputing the new minimum achievable competitive ratio, and running TS1 with the new parameters.

2.4 Other Trading Scenarios

We now describe some of the other trading scenarios considered in [3] and [4].

2.4.1 Bidirectional Trading

In the bidirectional trading problem, the investor may convert dollars to yen and yen to dollars. For this problem, the performance of the algorithm is its return (which will be the same, regardless of which currency it is measured in). The investor wants to minimize his competitive ratio. El-Yaniv *et al.* show that if the number of local maxima and minima in the exchange rate sequence is unbounded, then the competitive ratio is not finite. If the number of maxima and minima is bounded by k, then the competitive ratio has an upper bound of $(r^\star)^k$ and a lower bound of $(r^\star)^{k/2}$.

2.4.2 Trading with Options

The paper [4] also considers the case of trading call options. A trader starts with d_0 dollars, and receives a sequence of call option offers of the form (x_i, c_i). For any option offer (x_i, c_i), a trader may buy $s_i \geq 0$ options for a price of $s_i c_i$ dollars givng the trader the right, but not the obligation, to convert s_i dollars at the rate x_i to $s_i x_i$ yen. The securities that the option allows the investor to trade are called the underlying (in this case the yen), and the price at which one may buy these securities are called the strike, or exercise, price (in this case x_i). Trading Strategy 1 is also the optimal online algorithm for this problem.

2.4.3 Trading Against a Statistical Adversary

In [3], Chou *et al.* recognize the deficiency of looking only at the worst case scenario generated by a powerful adversary. They attempt to remedy this situation by introducing Ragahavan's *statistical* adversary [14]. A statistical adversary is one who must generate exchange rates satifying a certain statistical property (e.g. one could pick an adversary who is constrained to generating a sequence that has a given mean). In essence, the statistical restrictions on the adversary amounts to what we call a forecast.

Chou *et al.* study the bidirectional problem against a statistical adversary. The statistical adversary is forced to generate a sequence of exchange rate of length n such that the optimal offline algorithm can make a return that is greater than a known amount Π. They prove that the return of the optimal online algorithm, R is bounded by

$$0 < R \leq \frac{1}{1 - \left(1 - \frac{1}{\Pi}\right)^{n-1}}$$

Chapter 3

Further Analysis of the Unidirectional Problem

In this section we extend the analysis of the unidirectional problem in an attempt to get some additional insight into the results.

3.1 The Worst Case Exchange Rate Sequence

Our analysis of TS1 in Appendix A.2 leads to a closed form solution for the worst case exchange rate sequence $\underline{\mathbf{E}} = \langle \underline{e}_1, \ldots, \underline{e}_n \rangle$:

$$\underline{e}_0 = mr^\star$$

$$\underline{e}_i = m(r^* - 1)\left(\frac{M-m}{m(r^\star - 1)}\right)^{\frac{i}{n}} + m$$

In Figure 3.1 we plot the worst case exchange rate for the bounds $m = 100, M = 120$ and the number of trading periods is $n = 30$.

3.2 The Optimal Competitive Ratio $r^\star$

In this section we study the form of the optimal competitive ratio which El-Yaniv *et al.* show to be the solution to Equation 2.1. In particular, we are interested in how "good" the optimal competitive ratio really is. We begin by presenting in Table 3.1 some numerical values of $r^\star$ for typical values of m, M and n. The bounds m and M are reasonable assuming a current level of 110

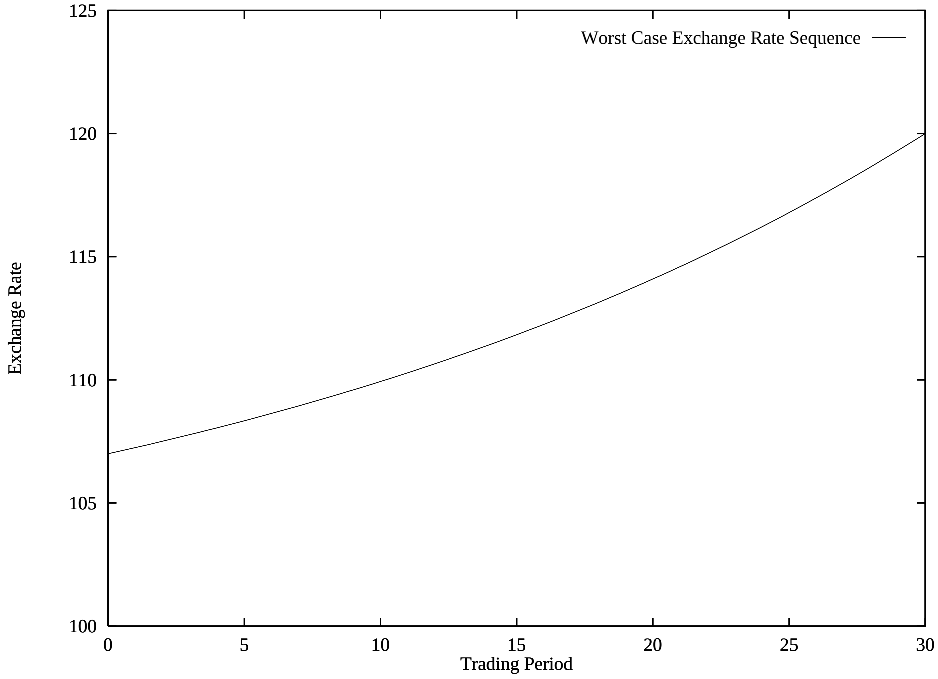

Figure 3.1: A plot of the worst case exchange rate for the $n = 30$ period unidirectional trading problem, with bounds $m = 100, M = 120$. In this case, the optimal competitive ratio is $r^\star = 1.07$.

yen per dollar. The last column is the amount of yen that TS1 makes expressed as a percentage of the amount of yen that the optimal offline algorithm achieves.

n	**m**	**M**	$\mathbf{r}^\star$	$\mathbf{100/r}^\star$
10	105	115	1.03	97%
10	100	120	1.06	94%
30	100	120	1.07	94%
30	90	135	1.15	87%
30	50	175	1.52	66%
∞	50	175	1.54	65%

Table 3.1: Some values of the optimal competitive ratio $r^\star$ computed by numerically solving Equations 2.1 and 2.2. The last column gives the minimum amount of yen that TS1 makes as a percentage of the amount of yen that OPT makes.

The simulation results indicate that TS1 does quite well in the unidirectional problem, but is this because the problem is easy? We now study this question in detail. In particular, we compare $r^\star$ with the worst competitve ratio that an algorithm can have, which we denote

$$r' = \sup_A \sup_{\sigma \in \Sigma} \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$$

It is easy to show that $r' = \frac{M}{m}$:

Lemma 3.1 *If the number of trading periods is $n \geq 2$, then the worst achievable competitive ratio for the unidirectional trading problem of Section 2.3.1 with the exchange rate bounded by m, M, is $r' = \frac{M}{m}$.*

Proof: $r' \leq \frac{M}{m}$ since any online algorithm must make at least md_0 yen and no offline algorithm can ever make more than Md_0 yen. One algorithm that achieves a competitive ratio of $\frac{M}{m}$ is to convert all the dollars to yen on the first day. In this case, the adversary can force a competitive ratio of $\frac{M}{m}$ by setting $e_1 = m$ and $e_i = M$ for $2 \leq i \leq n$. ■

We now wish to compare $r^\star$ and r'. Since the goal competitive ratio is 1, we consider $\frac{r^*-1}{r'-1}$. This expression will always lie in the range $[0,1]$, and the smaller it is, the better the relative performance of the optimal algorithm compared to the performance of the worst algorithm. We then have:

Theorem 3.2 *If the number of trading periods is $n \geq 2$ then for the unidirectional trading problem of Section 2.3.1 with the exchange rate bounded by m and M with $m < M$, we have :*

$$\left(1 - \frac{r_2^\star}{2}\right)^2 \leq \frac{r^\star - 1}{r' - 1} < \frac{1}{e}$$

where

$$r_2^\star = r^\star(m, M, 2, d_0) = \frac{2\left(M - \sqrt{Mm}\right)}{M - m}$$

is $r^\star$ evaluated when $n = 2$.

Proof: The optimal competitive ratio $r^\star$ is nondecreasing in n since increasing n allows the adversary to choose from a larger number of exchange rates. The worst competitive ratio r' is independent of n. Therefore, we only need to consider the two cases $n = 2$ and $n \to \infty$. When $n = 2$, Equation 2.1 is a quadratic polynomial in $r^\star$, and since $r' - 1 = \frac{M-m}{m}$ it simplifies to

$$\frac{r_2^\star - 1}{r' - 1} = \left(1 - \frac{r_2^\star}{2}\right)^2$$

which proves the left hand inequality. Setting $r' = \frac{M}{m}$ and solving the quadratic yields

$$r_2^\star = \frac{2\left(M - \sqrt{Mm}\right)}{M - m}$$

For arbitrary n, equation 2.1 is a polynomial of degree n:

$$\frac{r^* - 1}{r' - 1} = \left(1 - \frac{r^\star}{n}\right)^n$$

In the limit we have

$$\lim_{n\to\infty} \frac{r_n^\star - 1}{r' - 1} = e^{-r^\star}$$

If $m \neq M$ then $r_n^\star > 1$ and so

$$\frac{r_n^\star - 1}{r' - 1} < \frac{1}{e}$$

which completes the proof. ■

There are a number of interesting consequences of this theorem. We will look at the case $n \leq 60$ which we belive is a conservative bound on the number of periods that an investor will want to execute a strategy in. Given $n \leq 60$, it is also reasonable to assume that $\frac{M}{m} < 2$; indeed, many financial securities are not allowed to change in price by more than a set amount each day because of linits enforced by some governing bodies and exchanges.

- The first consequence is that $r_2^\star = \frac{2(M-\sqrt{Mm})}{M-m}$ is a monotonically increasing function of $\frac{M}{m}$ and as $\frac{M}{m} \to 0$, $r_2^\star \to 1$ and as $\frac{M}{m} \to \infty$, $r_2^\star \to 2$. It follows that $\left(1 - \frac{r_2^\star}{2}\right)^2$ is a montonically decreasing function of $\frac{M}{m}$ and as $\frac{M}{m} \to 0$ then $(1 - \frac{r_2^\star}{2})^2 \to 0.25$ and as $\frac{M}{m} \to \infty$ then $\left(1 - \frac{r_2^\star}{2}\right)^2 \to 0$. Therefore the left hand inequality of Theorem 3.2 lies in the range $[0, 0.25]$ and is a monotonically decreasing function of $\frac{M}{m}$.
- For $\frac{M}{m} < 2$ we have $\left(1 - \frac{r_2^\star}{2}\right)^2 > 0.17$ and so:

$$0.17 < \frac{r^\star - 1}{r' - 1} < 0.38 \tag{3.1}$$

  This means that, using this measure the optimal algorithm outperforms the worst algorithm only by a factor in the range 3-6, for reasonable values of $\frac{M}{m}$. This is surprisingly little.

n	**m**	**M**	$\mathbf{r'}$	$\mathbf{(r^\star - 1)/(r' - 1)}$	$\mathbf{100/r^\star}$	$\mathbf{100/r'}$
10	105	115	1.10	0.30	97%	91%
10	100	120	1.20	0.30	94%	83%
30	100	120	1.20	0.35	94%	83%
30	90	135	1.50	0.30	87%	67%
30	50	175	3.50	0.21	66%	28%
∞	50	175	3.50	0.22	65%	26%

Table 3.2: Some values for the ratio $\frac{r^\star - 1}{r' - 1}$ computed numerically. The last column gives the minimum amount of yen that an algorithm with the worst competitive ratio makes as a percentage of the amount of yen that OPT makes.

In Table 3.2 we give the ratio $\frac{r^\star - 1}{r' - 1}$ for the same values of m, M and n as in Table 3.1. We also provide the minimum amount of yen that both the optimal online algorithm and the worst possible algorithm make as a percentage of the amount of yen that OPT makes. The consequences of Theorem 3.2 are clearly shown in Table 3.2. Comparing the results for the scenarios that fall within our previously discussed sensible bounds of $\frac{M}{m} < 2, n \leq 60$, we find that the optimal competitive ratio is not much better than the worst competitive ratio. We will show in the next two chapters that if the investor is willing to accept some risk, they can do better than this in certain scenarios.

Chapter 4

The Risk-Reward Framework

In this chapter we introduce our central contribution. As motivation, we start with a discussion of general risk concepts, to motivate the risk-reward paradigm which we base our work on. We then present our risk-reward framework for competitive analysis, defining the notions of risk, reward and forecasts. We end the chapter with an analysis of a simple example online problem, giving both the traditonal analysis as well as one in our risk-reward framework.

4.1 What is Risk?

In this section we give a brief introduction to the concepts, perceptions, measurement and management of risk as discussed in [19]. *The Concise Oxford Dictionary* (1982) defines the noun risk as "hazard, chance of bad consequences, loss, etc., exposure to mischance..." and the *Webster's Dictionary* (1981) definition is "possibility of loss or injury". These two definitions share two characteristics:

- risk involves some form of negative outcome.
- risk does not necessarily involve the probability of outcomes.

The first characteristic is not shared by all people: for some, risk is a property of the uncertainty of an outcome, regardless of whether it is negative or positive. The second characteristic is also not universally agreed upon: some definitions tie risk to the probability of the outcomes, whilst others see this as a measure, and not a definition of, risk.

It is clear that the definition of risk as the uncertainty of an outcome is not the most used definition, however it is important to keep this variant in mind in subsequent sections. The definition we will follow is that of risk as the "exposure to a chance of loss" [12]. There are three components to this definition:

- **Magnitude of Loss**: There has to be a potential loss of some amount. Notice that this loss can be absolute, or it could be an opportunity loss (i.e. choosing a positive outcome at the expense of a better outcome).
- **Chance of Loss**: There has to be some uncertainty of the loss being incurred, a sure loss is not a risk.
- **Exposure to Loss**: The decision maker must be able to have some control over either the magnitude of or the exposure to the loss.

MacCrimmon and Wehrung [12] introduce a basic risk paradigm as the basis for studying risk. In their model there are two actions: a riskless action that leads to a certain outcome, and a risky action that leads to one of two outcomes; one is a gain, the other is a loss. The outcome that occurs for the risky action is uncertain. For our purposes, the action chosen is the strategy selected when trading. The outcome will be the competitive ratio achieved. Conventional competitive analysis does not give the investor a choice; it simply selects the riskless outcome and achieves the optimal competitive ratio. We propose to extend this framework to allow the investor to select a riskier strategy. Figure 4.1 gives a schematic view of the risk-reward framework.

4.2 The Risk-Reward Framework

In Section 2.1 we pointed out that the optimal online algorithm has minimum opportunity cost. As we discussed in Section 4.1, investors require more flexibility when selecting a strategy than simply minimizing their risk. To add the flexibility of risk management, we first introduce a new measure of risk based on the competitive ratio. In return for accepting an increased risk, investors demand a reward of some form. Many investors also want to incorporate predictions about the market into their strategy (i.e. they provide a forecast). In the next section we present our risk-reward framework which may be used by investors to select a strategy within their risk tolerance which maximizes their reward over the set of scenarios that comprises their forecast.

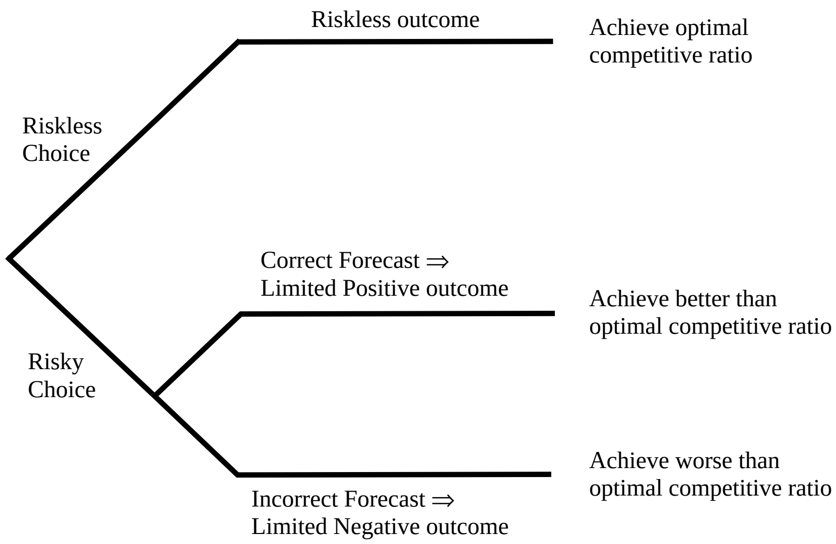

Figure 4.1: A schematic view of our risk-reward paradigm displayed in the manner of [12]. If the investor chooses not to accept any risk, they use the optimal online algorithm and achieve the optimal competitive ratio. If the investor wishes to take on a specified amount of risk, in some situations they will do better than the optimal competitive ratio, and in other situations they will do worse. The key points are (i) that the investor can specify (through a forecast) in which situations they will beat the optimal competitive ratio, and (ii) that they can limit how badly they perform when the forecast is not correct.

4.2.1 Definitions: Risk, Reward and Forecasts

We define the risk of an algorithm A to be

$$\frac{r_A}{r^\star}$$

where r_A is the competitive ratio of A and $r^\star$ is the optimal competitive ratio. From the investor's viewpoint, this measure of risk is the maximum opportunity cost that algorithm A may incur over the optimal online algorithm. If t is the risk tolerance of the investor (where $t \geq 1$ and higher values of t denote a higher risk tolerance) then denote by

$$I_t = \{A \,|r_A \leq tr^\star\}$$

the set of all algorithms that respect the investors risk tolerance.

A forecast is assumed to be a subset of the problem instances. Denote the forecast by $F \subset \Sigma$. We now need to define a measure of the reward for taking on any incremental risk over the optimal trading strategy. First define $\hat{r}_A$ to be the competitive ratio of A restricted to cases when the forecast is correct (otherwise it is nonsensical to talk about a reward):

$$\hat{r}_A = \sup_{\sigma \in F} \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$$

Now, we need to measure the reward of A as an improvement over the optimal online algorithm. We do this by defining the reward, f_A, of A to be

$$f_A = \frac{r^\star}{\hat{r}_A}$$

Then, given a problem Σ, a forecast $F \subset \Sigma$ and a risk tolerance t, the optimal risk-tolerant algorithm is $A^\star \in I_t$ such that:

$$f_{A^\star} = \sup_{A \in I_t} f_A$$

The Range of the Reward

The maximum possible reward is bounded by 1 and the optimal competitive ratio $r^\star$, as shown in the following Lemma.

Lemma 4.1 *Given the online problem Σ, the optimal risk-tolerant online algorithm $A^\star$, with forecast $F \subset \Sigma$ has a reward $f_{A^\star} \in [1, r^\star]$, where $r^\star$ is the optimal competitive ratio for the problem Σ.*

Proof: For the lower bound, consider the optimal online algorithm which is risk-tolerant for all $t \geq 1$. In this case, $\hat{r}_{A^\star} = r^\star \Rightarrow f_{A^\star} = 1$. For the upper bound, notice that no algorithm can perform better than the optimal offline algorithm, which implies that $\hat{r} \geq 1 \Rightarrow f_A \leq r^\star$. ■

4.3 The Ski Rental Problem

In this section we will demonstrate the ideas that we have just introduced using a simple problem which is often used to demonstrate concepts in the analysis of online algorithms. In the ski rental problem, we wish to acquire equipment for skiing. However, since we do not know how many times we will use this equipment, we do not know if it is cheaper to rent or to buy. Let a be the rental price, and b the buying price. For simplicity, assume that $a|b$ and $a, b \geq 1$. Our measure of the performance of an algorithm will be the reciprocal of the amount spent (this is equivalent to using the amount spent as the cost of the algorithm in the dual formulation of competitive analysis in Section 2.2.1).

4.3.1 Conventional Analysis

The optimal offline algorithm knows exactly how many times that we will ski, and so will choose to either rent or buy depending on which is cheaper. If j is the actual number of times that we ski,

OPT spends $\min(ja, b)$, and so $\mathbf{perf} = \frac{1}{\min(ja,b)}$. The adversary can force any online algorithm to spend $ja + b$ by waiting until the online algorithm buys skis, and then deciding to not ski again. Since $\forall j$, $\frac{ja+b}{\min(ja,b)} \geq 2$, the optimal competitive ratio is at least 2. An algorithm that achieves the lower bound of 2 is to rent for the first $j = \frac{b}{a}$ times, and then buy the equipment.

4.3.2 Analysis within the Risk-Reward Framework

Consider the situation where we believe that we know whether we will ski more or less times than $\frac{b}{a}$. In the risk-reward framework, this translates into two possible forecasts. One is that we will ski less than $\frac{b}{a}$ times. The other is if we guess that we ski more than $\frac{b}{a}$ times. We wish to use our forecast in an attempt to improve our performance (decrease our cost). The analysis of these two scenarios follows.

Forecast 1: $j \leq \frac{b}{a}$

In this case, the original optimal online algorithm is the reward maximizing strategy for all tolerance levels:

Theorem 4.2 *The optimal risk-tolerant (tolerance t) algorithm for the forecast of $j \leq \frac{b}{a}$ is to buy after renting for $j = \frac{b}{a}$ times.*

Proof: This algorithm is the optimal online algorithm, and so is in I_t for all $t \geq 1$. The restricted ratio is $\hat{r} = 1$, since if the forecast comes true, we never buy which is also what OPT does. The reward in this case is $f = r^\star = 2$. By Lemma 4.1, this is the maximum reward. ■

Corollary 4.3 *For the forecast $j \leq \frac{b}{a}$, the optimal restricted ratio is $\hat{r} = 1$ and the optimal reward is $f = 2$.*

Forecast 2: $j > \frac{b}{a}$

In this case the correct thing to do is to buy after $j = \left\lceil \frac{b}{a(2t-1)} \right\rceil$ times[1] as is shown in the following theorem:

Theorem 4.4 *The optimal risk-tolerant (tolerance t) algorithm for the forecast of $j > \frac{b}{a}$ is to buy after renting for $j = \left\lceil \frac{b}{a(2t-1)} \right\rceil$ times.*

[1] Notice that there is nothing gained by tolerance levels that do not make $\frac{b}{a(2t-1)}$ an integer.

Proof For an algorithm to be in I_t we require that $\frac{ja+b}{\min(ja,b)} \leq 2t$. The optimal offline algorithm when our forecast comes true is to buy immediately. Therefore, for the online algorithm which buys after j times, $\hat{r} = \frac{ja+b}{b}$. Since we want to minimize $\hat{r}$, we want j as small as possible subject to $\frac{ja+b}{\min(ja,b)} \leq 2t$. Since we know that $j = \frac{b}{a}$ (the minimum-risk algorithm) satisfies this, we know $j \leq \frac{b}{a}$ and so $\min(ja, b) = ja$. Therefore we want the smallest j such that $\frac{ja+b}{ja} \leq 2t$. Hence $j = \left\lceil \frac{b}{a(2t-1)} \right\rceil$. ∎

Corollary 4.5 *For the forecast $j > \frac{b}{a}$, the optimal restricted ratio is*

$$\hat{r} = \frac{ja+b}{b} = 1 + \frac{a}{b}\left\lceil \frac{b}{a(2t-1)} \right\rceil \sim 1 + \frac{1}{2t-1} < 2 = r^{\star}$$

and the optimal reward is $f = 2/\hat{r} \sim 2 - \frac{1}{t} > 1$.

Notice that as $t \to \infty$, the reward f_A approaches the maximum reward of $r^{\star} = 2$, and for $t = 1$ the restricted ratio reduces to the optimal competitive ratio and so the reward is 1.

Chapter 5

Unidirectional Trading in a Risk-Reward Framework

We now analyze the unidirectional trading problem of Section 2.3.1 in our risk-reward framework. We will consider a number of forecasts which may typically be made by investors. For each forecast, our aim is to develop a risk-tolerant algorithm which maximizes the reward. We conclude the chapter with a section on how we propose to complete the analysis of the problem.

5.1 Trading Strategy 2

The first forecast that we consider will be that the exchange rate will increase to at least $m + \Delta$; that is, there exists an i such that $1 \leq i \leq n$ and $e_i \geq m + \Delta$. The forecast is then $F_2 = \{\mathbf{E} = \langle e_1, \ldots, e_n \rangle \mid \exists i \text{ such that } e_i \geq m + \Delta\}$ ($F_1 = \Sigma$ is reserved to denote the null forecast and for which TS1 is the optimal risk-tolerant online algorithm). The investor will also have a risk tolerance of $t \geq 1$. Finally, the investor will want to maximize his reward should his guess be correct. The optimal risk-tolerant algorithm for F_2 trades in two stages. In the first stage, the algorithm trades under the threat that the forecast is incorrect, and converts enough dollars to ensure a competitive ratio of $tr^\star$. The second stage begins when the forecast comes true. The algorithm first computes the new minimum achievable competitive ratio $\hat{r}^\star$ (which is also the minimum achievable restricted ratio, and which we show how to compute in Section 5.2.4). The algorithm then trades so as to ensure that a competitive ratio of $\hat{r}^\star$ is achieved under the threat that the exchange rates drops to m and remains there for the rest of the trading periods. Notice that $\hat{r}^\star$ can be computed in advance.

5.1.1 The Algorithm

Trading Strategy 2

Given $m, M, n, r^\star, \hat{r}^\star, d_0$, a new exchange rate e_i in period T_i, a tolerance t, and a forecast $F_2 = \{\mathbf{E} = \langle e_1, \ldots, e_n \rangle \mid \exists i$ such that $e_i \geq m + \Delta\}$, trade according to the following rules:

1. Only trade when the exchange rate hits a new high (i.e. when $e_i > U_{i-1}$).

2. Whilst $e_i < m + \Delta$, convert $s_i^2 = \frac{1}{t} s_i^1$ dollars to yen (i.e. whilst our forecast is not true, only trade as little as our tolerance will allow, which is to keep a competitive ratio of at most t with the trading strategy TS1). This means that $\frac{U_i \cdot d_0}{y_i^2 + m d_i^2} = t r^\star$ which we show in Section 5.2 implies that $s_i^2 = \frac{d_0}{t r^\star} \frac{e_i - U_{i-1}}{e_i - m}$ where $e_0 \triangleq m t r^\star$.

3. When the forecast first comes true start a new game: when executing a trade, only trade enough to guarantee a competitive ratio of $\hat{r}^\star$ would be obtained should the exchange rate drop to m and remain there for the remainder of the trading periods. This means that $\frac{U_i \cdot d_0}{y_i^2 + m d_i^2} = \hat{r}^\star$ which we show in Section 5.2 implies that $s_\lambda^2 = \frac{d_0}{e_\lambda - m} \left(\frac{e_\lambda}{\hat{r}^\star} - \frac{e_{\lambda-1}}{t r^\star} \right)$ and for $i > \lambda$, $s_i^2 = \frac{d_0}{\hat{r}^\star} \frac{e_i - U_{i-1}}{e_i - m}$

5.2 Analysis of Trading Strategy 2

Our analysis of TS2 to is similar to that of TS1 in Appendix A. We begin by showing that TS2 is the optimal risk-tolerant online algorithm. In the later sections, we compute s_i^2, the amount spent by TS2 in each period. We then conclude by showing how to compute $\hat{r}^\star$, the optimal restricted ratio for step 2 of TS2.

Remark 5.1 *By similar arguments made in Lemmas A.1 and A.2, we know that when analyzing TS2, we need only consider exchange rate sequences of the form* $m t r^\star < e_1 < e_2 < \cdots < e_k < M$ *and* $e_{k+1} = e_{k+2} = \cdots = m$, *for some* k *in* $[1..n]$. *For all such exchange rates for which the forecast* F_2 *is correct, TS2 achieves the optimal competitive ratio* $\hat{r}^\star$ *as long as it does not run out of dollars.*

5.2.1 Proof of Optimality and Risk-Tolerance

We first show that TS2 is risk-tolerant. We begin by proving that the optimal restricted ratio is less than the optimal competitive ratio. We do this by constructing a risk-tolerant algorithm that acquires more yen than TS1, but achieves a restricted ratio less than the optimal competitive ratio.

Lemma 5.2 *The optimal restricted ratio for the unidirectional trading problem with a correct forecast F_2 is strictly less than the optimal competitive ratio: $\hat{r}^\star < r^\star$.*

Proof: Consider the algorithm A that trades as follows:

- $s_i^A = s_i^2 = \frac{1}{t}s_i^1$ for $1 \leq i < \lambda$.
- $s_\lambda^A = s_\lambda^1 + d_{\lambda-1}^2 - d_{\lambda-1}^1$.
- $s_i^A = s_i^1$ for $\lambda < i \leq n$.

Algorithm A is clearly risk-tolerant. Notice that

$$\begin{aligned}
s_\lambda^A &= s_\lambda^1 + (d_0 - \sum_{i=1}^{\lambda-1} s_i^2) - (d_0 - s_i^2 + \sum_{i=1}^{\lambda-1} s_i^1) \\
&= s_\lambda^1 + \sum_{i=1}^{\lambda-1} s_i^1 - \sum_{i=1}^{\lambda-1} \frac{1}{t} s_i^1 \\
&= s_\lambda^1 + \left(1 - \frac{1}{t}\right) \sum_{i=1}^{\lambda-1} s_i^1
\end{aligned}$$

Now

$$\begin{aligned}
y_n^A - y_n^1 &= \sum_{i=1}^{n} e_i s_i^A - \sum_{i=1}^{n} e_i s_i^1 \\
&= \sum_{i=1}^{\lambda} e_i s_i^A - \sum_{i=1}^{\lambda} e_i s_i^1 \qquad \text{(since } s_i^A = s_i^1 \text{ for } \lambda < i \leq n\text{)} \\
&= e_\lambda s_\lambda^A + \sum_{i=1}^{\lambda-1} e_i s_i^A - e_\lambda s_\lambda^1 - \sum_{i=1}^{\lambda-1} e_i s_i^1 \\
&= \left[e_\lambda \left(1 - \frac{1}{t}\right) \sum_{i=1}^{\lambda-1} s_i^1 + \sum_{i=1}^{\lambda-1} e_i s_i^A \right] - \sum_{i=1}^{\lambda-1} e_i s_i^1 \qquad \text{(susbtituting for } s_\lambda^A\text{)} \\
&= \left[e_\lambda \left(1 - \frac{1}{t}\right) \sum_{i=1}^{\lambda-1} s_i^1 + \frac{1}{t} \sum_{i=1}^{\lambda-1} e_i s_i^1 \right] - \sum_{i=1}^{\lambda-1} e_i s_i^1 \qquad \text{(since } s_i^A = \frac{1}{t} s_i^1 \text{ for } 1 \leq i < \lambda\text{)} \\
&> 0 \qquad \text{since } (e_\lambda > e_i \text{ for } 1 \leq i \leq \lambda - 1)
\end{aligned}$$

Therefore, $y_n^A > y_n^1 \Rightarrow r_A < r_1 = r^\star$. Therefore, $\hat{r}^\star \leq r_A < r^\star$. ■

Corollary 5.3 *The reward of TS2 for the unidirectional trading problem with forecast F_2 is $f_2 > 1$.*

Proof: $f_2 = \frac{r^\star}{\hat{r}^\star} > 1$. ■

Corollary 5.4 *TS2 is a risk-tolerant algorithm for the unidirectional trading problem with forecast F_2.*

Proof: There are two cases

- If the exchange rate is $\mathbf{E} \in \Sigma \setminus F_2$ then by rule 2 of TS2, $r_2(\mathbf{E}) \leq tr^\star$.
- If the exchange rate is $\mathbf{E} \in F_2$ then by rule 3 of TS2 and Lemma 5.2 we have that $r_2(\mathbf{E}) = \hat{r}^\star < r^\star < tr^\star$. ■

We next turn our attention to the optimality of TS2. We begin with two technical lemmas that we will use to show that for any algorithm A that deviates from TS2, there exists an exchange rate sequence that forces A to achieve a competitive ratio worse than $\hat{r}^\star$.

Lemma 5.5 *Consider the family of sequences $x_1, \ldots, x_n$ where $m < x_1 < \cdots < x_p < M$, $0 < m < M$ and $x_{p+1} = \cdots = x_n = m$. Let $f \neq 0$ be any function of the x_i such that*

- $\sum_{i=1}^n f(x_i) = 0$
- $k = \min\{i | f(x_i) \neq 0\}$ *and* $f(x_k) < 0$

Then there exists some sequence $\underline{x}_i$ such that $\sum_{i=1}^n \underline{x}_i \cdot f(\underline{x}_i) < 0$.

Proof: Let $p = k + 1$.

$$\sum_{i=1}^n x_i \cdot f(x_i) < 0 \Leftrightarrow \sum_{i=k}^n x_i \cdot f(x_i) < 0 \Leftrightarrow x_k \cdot f(x_k) < \sum_{i=k+1}^n x_i \cdot (-f(x_i))$$

Since $\sum_{i=k}^n f(x_i) = 0$ then $-f(x_k) = \sum_{i=k+1}^n f(x_i)$. Finally, because $x_k > x_{i+1} = m$ for $i > k$, we have that

$$x_k \cdot (-f(x_k)) > m \sum_{i=k+1}^n f(x_i)$$

■

Lemma 5.6 *Consider the family of sequences $x_1, \ldots, x_n$ where $m < x_1 < \cdots < x_p < M$, $0 < m < M$ and $x_{p+1} = \cdots = x_n = m$. Let $f \neq 0$ be any function of the x_i such that*

- $\sum_{i=1}^n f(x_i) = 0$

- $k = \min\{i|f(x_i) \neq 0\}$ *and* $f(x_k) > 0$

Then there exists some sequence $\underline{x}_i$ *such that* $\sum_{i=1}^{n} \underline{x}_i \cdot f(\underline{x}_i) < 0$.

Proof: We prove this using induction on the number of elements x_i for which $f(x_i) > 0$.

Base case: Assume that $|\{i|f(x_i) > 0\}| = 1$ and let $p = n$. Then

$$\sum_{i=1}^{n} x_i \cdot f(x_i) < 0 \Leftrightarrow \sum_{i=k}^{n} x_i \cdot f(x_i) < 0 \Leftrightarrow x_k \cdot f(x_k) < \sum_{i=k+1}^{n} x_i \cdot (-f(x_i))$$

Since $\sum_{i=k}^{n} f(x_i) = 0$ then $f(x_k) = \sum_{i=k+1}^{n}(-f(x_i))$. Finally, because $x_i < x_{i+1}$ for $1 \leq i < n$, we have that

$$x_k \cdot f(x_k) < x_{k+1} \sum_{i=k+1}^{n} (-f(x_i)) < \sum_{i=k+1}^{n} x_i \cdot (-f(x_i))$$

Induction: Assume that $|\{i|f(x_i) > 0\}| > 1$ and let $l = \min\{i|i > k \text{ and } f(x_i) \neq 0\}$. Consider the function g such that

$$g(x_i) = \begin{cases} 0 & \text{for } 1 \leq i < l, \\ \sum_{i=k}^{l} f(x_j) & \text{for } i = l, \\ f(x_i) & \text{for } i > l. \end{cases}$$

Then $\sum_{i=1}^{n} g(x_i) = \sum_{i=1}^{n} f(x_i) = 0$ and, by the same arguement as for the base case, $\sum_{i=1}^{n} x_i \cdot g(x_i) > \sum_{i=1}^{n} x_i \cdot f(x_i)$. If $g(x_l) \geq 0$ then we are done by induction. If $g(x_l) < 0$ then we are done by Lemma 5.5. ∎

Theorem 5.7 *No online risk-tolerant algorithm for the unidirectional trading problem with forecast* F_2 *can achieve a restricted ratio less than that of TS2.*

Proof: Let A be an online risk-tolerant algorithm with forecast F_2 that deviates from TS2. Consider an exchange rate sequence $\mathbf{E} = \langle e_1, \ldots, e_n \rangle$ of the form of Remark 5.1. Then $s_i^A = s_i^2 + f(e_i)$ where $\sum_{i=1}^{n} f(e_i) = 0$. Then, by Lemmas 5.5 and 5.6 we have that $\sum_{i=1}^{n} e_i f(e_i) < 0 \Rightarrow \sum_{i=1}^{n} e_i s_i^A < \sum_{i=1}^{n} e_i s_i^2$.

Corollary 5.8 *For the unidirectional trading problem with forecast* F_2*, TS2 is an optimal risk-tolerant online algorithm.*

5.2.2 Computing s_i^2

Lemma 5.9 *Whilst the forecast has not come true ($i < \lambda$), TS2 spends*

$$s_i^2 = \frac{d_0}{tr^\star}\frac{e_i - e_{i-1}}{e_i - m} \qquad (1 \le i \le \lambda - 1)$$

Proof: Since TS2 needs to ensure that it achieves a competitive ratio of not more than $tr^\star$, it needs to spend enough to ensure that ratio under the threat that the adversary drops the exchange rate to m and keeps it there. Therefore, TS2 must guarantee that

$$\frac{d_0 e_i}{y_i^2 + m d_i^2} = tr^\star$$

Solving for s_i^2 (in the same way as Lemma A.4 of Appendix A.1) gives us the above. ■

Lemma 5.10 *After the forecast has come true ($i > \lambda$), TS2 spends*

$$s_i^2 = \frac{d_0}{\hat{r}^\star}\frac{e_i - e_{i-1}}{e_i - m} \qquad (\lambda + 1 \le i \le n)$$

Proof: TS2 needs to ensure that it achieves the optimal restricted ratio of $\hat{r}^\star$, and therefore it must guarantee that

$$\frac{d_0 e_i}{y_i^2 + m d_i^2} = \hat{r}^\star$$

Solving for s_i^2 (in the same way as Lemma A.4 of Appendix A.1) gives us the above. ■

Lemma 5.11 *The first period that the forecast comes true ($i = \lambda$), TS2 spends*

$$s_\lambda^2 = \frac{d_0}{e_\lambda - m}\left(\frac{e_\lambda}{\hat{r}^\star} - \frac{e_{\lambda-1}}{tr^\star}\right)$$

Proof: TS2 needs to ensure that

$$\frac{d_0 e_\lambda}{y_\lambda^2 + m d_\lambda^2} = \hat{r}^\star$$

which implies that

$$\begin{aligned} \frac{d_0 e_\lambda}{\hat{r}^\star} &= y_\lambda^2 + m d_\lambda^2 \\ &= (y_{\lambda-1}^2 + m d_{\lambda-1}^2) + s_\lambda(e_\lambda - m) \qquad (5.1) \end{aligned}$$

However, from Lemma 5.9, we know that

$$\frac{d_0 e_{\lambda-1}}{y_{\lambda-1}^2 + m d_{\lambda-1}^2} = tr^\star$$

and so

$$y_{\lambda-1}^2 + m d_{\lambda-1}^2 = \frac{d_0 e_{\lambda-1}}{tr^\star}$$

Substituting into Equation 5.1 and rearranging completes the proof. ■

Corollary 5.12 *For TS2 we have*

$$\sum_{i=1}^{n} s_i^2 = \sum_{i=1}^{\lambda-1} \frac{d_0}{tr^\star}\frac{e_i - e_{i-1}}{e_i - m} + \sum_{i=\lambda+1}^{n} \frac{d_0}{\hat{r}^\star}\frac{e_i - e_{i-1}}{e_i - m} + \frac{d_0}{e_\lambda - m}\left(\frac{e_\lambda}{\hat{r}^\star} - \frac{e_{\lambda-1}}{tr^\star}\right)$$

5.2.3 Maximizing $\sum_{i=1}^{n} s_i^2$

By a similar analysis to Appendix A.2, we have that

$$(e_i - m)^2 = (e_{i-1} - m)(e_{i+1} - m)$$

for $1 \le i \le \lambda - 2$, and $\lambda + 1 \le i \le n - 1$, where $e_0 = mtr^\star$. For $i = \lambda - 1$ we have

$$\begin{aligned}
\frac{\partial}{\partial e_{\lambda-1}} \sum_{i=1}^{n} s_i^2 &= \frac{\partial}{\partial e_{\lambda-1}} \left[\frac{d_0}{tr^\star}\frac{e_{\lambda-1} - e_{\lambda-2}}{e_{\lambda-1} - m} - \frac{d_0}{tr^\star}\frac{e_{\lambda-1}}{e_\lambda - m}\right] \\
&= \frac{d_0}{tr^\star}\left[\frac{(e_{\lambda-1} - m) - (e_{\lambda-1} - e_{\lambda-2})}{(e_{\lambda-1} - m)^2} - \frac{1}{e_\lambda - m}\right] \\
&= \frac{d_0}{tr^\star}\left[\frac{(e_{\lambda-2} - m)(e_\lambda - m) - (e_{\lambda-1} - m)^2}{(e_{\lambda-1} - m)^2(e_\lambda - m)}\right] \qquad (5.2)
\end{aligned}$$

Setting $\frac{\partial}{\partial e_{\lambda-1}} \sum_{i=1}^{n} s_i^2 = 0$, we find that

$$(e_{\lambda-1} - m)^2 = (e_{\lambda-2} - m)(e_\lambda - m)$$

For $i = \lambda$ we have

$$\begin{aligned}
\frac{\partial}{\partial e_\lambda} \sum_{i=1}^{n} s_i^2 &= \frac{\partial}{\partial e_\lambda}\left[\frac{d_0}{\hat{r}^\star}\frac{e_{\lambda+1} - e_\lambda}{e_{\lambda+1} - m} + \frac{d_0}{\hat{r}^\star}\frac{e_\lambda}{e_\lambda - m}\right] \\
&= \frac{d_0}{\hat{r}^\star}\left[\frac{-(e_{\lambda+1} - m)}{(e_{\lambda+1} - m)^2} + \frac{(e_\lambda - m) - e_\lambda}{(e_\lambda - m)^2}\right] \\
&= -\frac{d_0}{\hat{r}^\star}\left[\frac{(e_\lambda - m)^2 + m(e_{\lambda+1} - m)}{(e_{\lambda+1} - m)(e_\lambda - m)^2}\right] \\
&< 0
\end{aligned}$$

The derivative $\frac{\partial}{\partial e_\lambda} \sum_{i=1}^{n} s_i^2$ is negative, and so $\sum_{i=1}^{n} s_i^2$ is monotonically decreasing in e_λ. Therefore, the sum $\sum_{i=1}^{n} s_i^2$ is maximized when e_λ is the lowest it can be, i.e. $e_\lambda = m + \Delta$.

We may now solve for e_i. We have two cases:

Case 1: $1 \le i \le \lambda$

$$\frac{e_i - m}{e_{i+1} - m} = \frac{e_{i-1} - m}{e_i - m} = \frac{1}{\alpha_1}$$

and so

$$\frac{1}{\alpha_1} = \left(\frac{e_0 - m}{e_\lambda - m}\right)^{\frac{1}{\lambda}} = \left(\frac{m(tr^\star - 1)}{\Delta}\right)^{\frac{1}{\lambda}}$$

therefore, for $1 \leq i \leq \lambda$

$$e_i = m + m(tr^\star - 1)\alpha_1^i$$

Case 2: $\lambda < i \leq n$

$$\frac{e_i - m}{e_{i+1} - m} = \frac{e_{i-1} - m}{e_i - m} = \frac{1}{\alpha_2}$$

and so

$$\frac{1}{\alpha_2} = \left(\frac{e_\lambda - m}{e_n - m}\right)^{\frac{1}{n-\lambda}} = \left(\frac{\Delta}{M - m}\right)^{\frac{1}{n-\lambda}}$$

therefore, for $\lambda \leq i \leq n$

$$e_i = m + \Delta\alpha_2^{i-\lambda}$$

■

5.2.4 Computing $\hat{r}^\star$

Lemma 5.13 *For the k period unidirectional trading game, with the exchange rate bounded by m and M, initial dollar and yen amounts of d_0 and y_0 respectively, and given the first exchange rate e_1, the optimal competitive ratio $r^\star(y_0, d_0, e_1)$ is*

$$r^\star(y_0, d_0, e_1) = \frac{d_0 e_1}{d_0 e_1 + y_0} + \frac{d_0(e_1 - m)}{d_0 e_1 + y_0}(k-1)\left(1 - \left(\frac{e_1 - m}{M - m}\right)^{\frac{1}{k-1}}\right)$$

Proof: We follow the same reasoning as for TS1 in Appendix A.2. For ease of exposition, we set $r^\star = r^\star(y_0, d_0, e_1)$. We wish to maximize the amount spent by TS2,

$$\sup_{e_1 < \ldots < e_k} (d_0 - d_k) = \sup_{e_1 < \ldots < e_k} \left(\frac{d_0}{r^\star} \sum_{i=1}^{k} s_i\right)$$

We first calculate s_1 using the constraint

$$\frac{d_0 e_1}{y_0 + s_1 e_1 + m(d_0 - s_1)} = r^\star$$

Rearranging, we get

$$s_1 = \frac{d_0 e_1 - r^\star(y_0 + m d_0)}{r^\star(e_1 - m)}$$

So

$$\sup_{e_1 < \ldots < e_k} (d_0 - d_k) \quad = \quad \sup_{e_1 < \ldots < e_k} \left(\frac{d_0 e_1 - r^\star(y_0 + m d_0)}{r^\star(e_1 - m)} + \frac{d_0}{r^\star} \sum_{i=2}^{k} \frac{e_i - e_{i-1}}{e_i - m}\right)$$

Setting $\sup(d_0 - d_k) = d_0$ and rearranging, we have

$$r^\star = \frac{d_0 e_1}{d_0 e_1 + y_0} + \frac{d_0(e_1 - m)}{d_0 e_1 + y_0} \sum_{i=2}^{k} \frac{e_i - e_{i-1}}{e_i - m}$$

Taking the partial derivatives, we find that $r^\star$ is maximized when

$$\frac{(e_i - m)}{(e_{i+1} - m)} = \frac{(e_{i-1} - m)}{(e_i - m)}$$

and so

$$r^\star = \frac{d_0 e_1}{d_0 e_1 + y_0} + \frac{d_0(e_1 - m)}{d_0 e_1 + y_0}(k-1)\left(\frac{e_1 - m}{e_k - m}\right)^{\frac{1}{k-1}}$$

This is maximized when $e_k = M$. ■

The optimal restricted ratio $\hat{r}^\star = \inf_{\mathbf{E}} r^\star(y_{\lambda-1}, d_{\lambda-1}, m + \Delta)$.

5.2.5 Completing the Analysis

There are two steps left for a complete analysis of TS2:

1. Find the value of λ that maximizes the optimal restricted ratio $\hat{r}^\star$.

2. Solve for $\hat{r}^\star$ in a similar manner to the derivation of $r^\star$ in Appendix A

Chapter 6

Conclusion

In this chapter we first review our contributions. We then detail the proposed future work, including a timetable.

6.1 Contributions

In Chapter 3, we extended the analysis of the unidirectional problem: we found a closed-form solution for the worst case exchange rate and we also showed that the unidirectional trading problem is "easy;" the optimal online algorithm does not achieve a competitive ratio that is much better than the trivial algorithm. In Chapter 4, we presented our main contribution: a risk-reward framework for the competitive analysis of online algorithms. We defined measures of risk and reward, as well as the concept of a forecast, and applied this framework to the ski-rental problem. In Chapter 5, we analyzed the unidirectional trading problem within this framework. We suggested a possible forecast, and find an optimal risk-tolerant algorithm for it.

6.2 Proposed Future Work

6.2.1 Alternative Forecasts

We propose to further analyze the unidirectional problem, using forecasts that we feel are typical of those made by some investors. Some possible forecasts are described below.

Tighter Bounds In the unidirectional problem, we make the assumption that the exchange rate is bounded $m \leq e_i \leq M$. Two related forecasts are that the true upper bound is $M' < M$

or that the true lower bound is $m' > m$.

Low Volatility The maximum daily fluctuation is bounded by some constant v: $\forall i = 2, \ldots, n, |e_i - e_{i-1}| \leq v$ that is known to the investor.

Support Level The exchange rate κ ($m < \kappa < M$) is a "support" level; an exchange rate that, once reached, will be a lower bound for all subsequent exchange rates. In other words, if $\exists i$ such that $e_i \geq \kappa$ then $\forall j > i, e_j \geq \kappa$.

Piercing Resistance The exchange rate η_1 is a resistance level, and η_2 is a goal level ($m < \eta_1 < \eta_2 < M$): if $\exists i$ such that $e_i \geq \eta_1$ then $\exists j > i$ such that $e_j \geq \kappa_2$.

6.2.2 Empirical Study

We have already provided some numerical solutions for the performance of various algorithms under different scenarios for the bounds on the exchange rates. We propose to expand this empirical work to also include different exchange rate scenarios, as well as backtesting the algorithms on historical data. One of the issues involved here is which strategies currently used by traders can and should be included for comparison.

6.2.3 An Analysis of Hedging using Options

El-Yaniv *et al.* [4] analyze the problem of trading an option by simply regarding an option as an isolated financial security (see Section 2.4.2). They do not fully consider the interaction between the options market and the associated cash market. This leads to the same analysis for options as for the original problem of trading dollar-yen, with the added feature that options allow the investor to leverage their position. Leverage refers to the investor being able to control more securities than he would normally be able to buy with his initial capital. With options, this is accomplished by the fact that the price of an option is usually a fraction of the price of the underlying security. The optimal competitive ratio is then the optimal competitive ratio of TS1 multiplied by some appropriate leverage factor.

We propose a different approach to the use of options in the unidirectional trading problem. As indicated, the key to conside is how the options market interacts with the cash market. An option can be viewed as a hedging instrument. A hedging instrument is a security that investors can trade to insure themselves against adverse price movments that might reduce the return of their trading activities. The price of the option can be regarded as an insurance premium.

Consider two investors: one who can only trade dollar-yen, the other who can trade dollar-yen as well as options on dollar-yen. We will then investigate the circumstances under which the second trader can perform (in a competitive sense) better than the first trader. This should yield some insights into the value of an option to a trader. The key is to analyze both the exchange rate (the cash market) and the option price.

We also propose to investigate how the sale of another standard option, the put option, can be integrated into the unidirectional problem. Consider a trader who is also presented with a sequence of put option offers of the form (x_i, p_i). For any option offer (x_i, p_i), a trader may sell $s_i \geq 0$ options for a price of $s_i p_i$ dollars giving the trader the obligation to convert s_i dollars to $x_i s_i$ yen at the rate x_i if the final exchange rate $e_n > x_i$.

6.2.4 Bidirectional Trading

In Section 3.2, we showed that the unidirectional trading problem was "easy." El-Yaniv *et al.* [4] show that the bidirectional problem is hard (see Section 2.4.1): the competitive ratio is exponential in the number of local maxima and minima exchange rates. We propose to study this problem to see if the risk-reward framework allows us to derive meaningful results. In particular, we want to characterize the set of forecasts that allow us to differentiate between different online algorithms.

6.3 Research Plan

1. Complete of the analysis of TS2:

   - Obtain an expression for the minimum achievable restricted ratio $\hat{r}^{\star}$.

   - Provide numerical solutions to the performance of TS2 versus TS1 for different values of the exchange rate bounds m and M, the number of traing periods n and the forecast parameter Δ.

2. Analyze at least one other forecast for the unidirectional trading problem.

3. Perform an empirical study of the algorithms developed.

4. Analyze the use of call options for hedging.

Appendix A

Analysis of Trading Strategy 1

In this appendix, we present our analysis of TS1, by first recasting the unidirectional conversion problem as an optimization problem. In particular, we show how to obtain a closed form solution for the worst case exchange rate, $\mathbf{E}$. We first argue why we may assume that the exchange rate is monotonically increasing, and if it drops, it drops to m and stays there.

Lemma A.1 *Let $\mathbf{E} = \langle e_1, \ldots, e_n \rangle$ where $m \leq e_i \leq M$ and denote by $r(\mathbf{E})$ the competitive ratio of TS1 for the exchange rate sequence $\mathbf{E}$. Suppose that e_j is the first exchange rate in the sequence $\mathbf{E}$ that is greater than $mr^\star$. Let $e'_1 = e_j$ and $\langle e'_1, \ldots, e'_k \rangle$ be the longest monotonically increasing subsequence of $\mathbf{E}$ starting with e_j. If $e'_{k+1} = e'_{k+2} = \cdots = e'_n = m$ and $\mathbf{E}' = \langle e'_1, \ldots, e'_n \rangle$ then $r(\mathbf{E}') = r(\mathbf{E})$*

Proof: The offline algorithm makes the same amount of yen for both exchange rate sequence $\mathbf{E}$ and $\mathbf{E}'$ since the maximum exchange rate in both cases is e_k. TS1 also makes the same amount of yen for the exchange rate sequences $\mathbf{E}$ and $\mathbf{E}'$ since it trades in both cases at the exchange rates $e'_1, \ldots, e'_k$ (in that order). ■

Lemma A.2 *For all exchange rate sequences $\mathbf{E} = \langle e_1, \ldots, e_n \rangle$ in which there exists an exchange rate $e_i \geq mr^\star$, TS1 achieves a competitive ratio of $r(\mathbf{E}) = r^\star$, so long as it does not run out of dollars. Furthermore, for all other exchange rate sequences (i.e. $\forall i, e_i < mr^\star$), TS1 does not convert any dollars, and $r(\mathbf{E}) < r^\star$.*

Proof: Immediate. ■

Remark A.3 *From Lemmas A.1 and A.2, we know that when analyzing TS1, we need only consider exchange rate sequences of the form $mr^\star < e_1 < e_2 < \cdots < e_k < M$ and $e_{k+1} = e_{k+2} =$*

$\cdots = m$, for some k in $[1..n]$. Also note that the only constraint on TS1 achieving a competitive ratio of $r^\star$ is the possibility of it running out of dollars.

A.1 Computing s_i

Given an exchange rate e_i in period T_i, the minimum amount of dollars that TS1 needs to convert to guarantee a competitive ratio of $r^\star$ under the threat of all subsequent exchange rates dropping to m is characterised in the following Lemma.

Lemma A.4 *The amount of dollars that TS1 converts in period T_i is*

$$s_i^1 = \frac{d_0}{r^\star}\frac{e_i - e_{i-1}}{e_i - m} \tag{A.1}$$

where $e_0 \triangleq mr^\star$.

Proof: From rule 2 of TS1, we know that if TS1 trades in period i, then it must trade just enough to ensure that it achieves the optimal competitive ratio, should the exchange rate drop to m and stay there. This gives us the following constraint

$$\frac{d_0 e_i}{y_i^1 + m d_i^1} = r^\star \tag{A.2}$$

Rearranging, we have

$$\frac{d_0}{r^\star} e_i = y_i^1 + m d_i^1 \tag{A.3}$$

Notice that the amount of yen y_i is simply the amount of yen we had at the end of the previous period plut the amount of yen we make in this period, so $y_i = y_{i-1} + e_i s_i$. Also, the amount of dollars $d_i = d_{i-1} - s_i$. This gives us

$$\frac{d_0}{r^\star} e_i = y_{i-1}^1 + e_i s_i + m(d_{i-1}^1 - s_i) = (y_{i-1}^1 + m d_{i-1}^1) + s_i(e_i - m) \tag{A.4}$$

Substitiuting for $(y_{i-1}^1 + m d_{i-1}^1)$ using Equation A.3, we get

$$\frac{d_0}{r^\star} e_i = \frac{d_0}{r^\star} e_{i-1} + s_i(e_i - m)$$

Solving for s_i we get

$$s_i^1 = \frac{d_0}{r^\star}\frac{e_i - e_{i-1}}{e_i - m} \qquad \text{(for } i > 1\text{)}$$

We compute s_1^1 by noticing that $y_0^1 = 0$. Substituting into (A.4) we find that

$$s_1^1 = \frac{d_0}{r^\star}\frac{e_1 - mr^\star}{e_1 - m}$$

which is equivalent to (A.1) if we consider $e_0 \triangleq mr^\star$. ∎

A.2 Computing $r^\star$

Notice that, by rule 2 of TS1 (see Section 2.3.2), TS1's competitive ratio will always be the worst case competitive ratio as long as the exchange rate sequence is of the form in Remark A.3 and TS1 does not spend more dollars than it has (i.e. as long as $\sum_{i=1}^{k} s_i \leq d_0$). For the adversary to cause TS1 to achieve a competitive ratio greater than $r^\star$, she must make TS1 spend more dollars than it has. So, as a first step to computing $r^\star$, we need to find the exchange rate sequence that maximizes the amount of dollars spent by TS1.

Theorem A.5 *The exchange rate sequence* $\mathbf{E} = \langle e_1, \ldots, e_n \rangle$ *of the form* $mr^* < e_1 < e_2 < \cdots < e_k$ *and* $e_{k+1} = e_{k+2} = \ldots = e_n = m$ *which maximizes the amount spent by TS1 is:*

$$e_i = m + m(r^* - 1)\alpha^i \qquad \text{for } i = 1, \ldots, k$$

where $\alpha = \left(\frac{M-m}{m(r^\star-1)}\right)^{\frac{1}{k}}$. *The total spent in this scenario is:*

$$\frac{d_0}{r^*} k \left(1 - \frac{1}{\alpha}\right)$$

Proof: The amount of dollars that TS1 spends in k periods is

$$d_0 - d_k = \sum_{i=1}^{k} s_i \tag{A.5}$$

Substituting for s_i using equation (A.1), we have

$$d_0 - d_k = \sum_{i=1}^{k} \frac{d_0}{r^\star} \frac{e_i - e_{i-1}}{e_i - m}$$

Taking the partial derivatives with respect to the e_i we get

$$\frac{\partial(d_0 - d_k)}{\partial e_i} = \begin{cases} \frac{d_0}{r^\star} \frac{(e_{i-1}-m)(e_{i+1}-m)-(e_i-m)^2}{(e_i-m)^2(e_{i+1}-m)} & \text{if } 1 \leq i < k \\ \frac{d_0}{r^\star} \frac{e_{i-1}+m}{(e_i+m)^2} & \text{if } i = k \end{cases} \tag{A.6}$$

Solving for the e_i in

$$\frac{\partial(d_0 - d_k)}{\partial e_i} = 0$$

we find that

$$(e_i - m)^2 = (e_{i-1} - m)(e_{i+1} - m)$$

This implies that

$$\frac{e_i - m}{e_{i+1} - m} = \frac{e_{i-1} - m}{e_i - m} = \frac{1}{\alpha} \tag{A.7}$$

for $1 \leq i < k$ and some constant α. It is immediately clear that

$$\left(\frac{1}{\alpha}\right)^k = \frac{e_0 - m}{e_k - m} = \frac{mr^\star - m}{e_k - m}$$

We therefore have that

$$\begin{aligned}
\sup_{e_1<\ldots<e_k} (d_0 - d_k) &= \sup_{e_1<\ldots<e_k} \left(\frac{d_0}{r^\star}\sum_{i=1}^{k} s_i^1\right) \\
&= \sup_{e_1<\ldots<e_k} \left(\frac{d_0}{r^\star}\sum_{i=1}^{k} \frac{e_i - e_{i-1}}{e_i - m}\right) \\
&= \sup_{e_1<\ldots<e_k} \left(\frac{d_0}{r^\star}\sum_{i=1}^{k} \frac{(e_i - m) - (e_{i-1} - m)}{e_i - m}\right) \\
&= \sup_{e_1<\ldots<e_k} \left(\frac{d_0}{r^\star}\sum_{i=1}^{k} \frac{e_i - m}{e_i - m} - \frac{e_{i-1} - m}{e_i - m}\right)
\end{aligned}$$

Substituting using Equation A.7, we get

$$\begin{aligned}
\sup_{e_1<\ldots<e_k} (d_0 - d_k) &= \frac{d_0}{r^\star}\sum_{i=1}^{k} 1 - \frac{1}{\alpha} \\
&= \frac{d_0}{r^\star} k \left(1 - \frac{1}{\alpha}\right)
\end{aligned}$$

This is maximized by maximizing α, and so

$$\alpha = \left(\frac{M - m}{m(r^\star - 1)}\right)^{\frac{1}{k}}$$

Finally, substituting for α in Equation (A.7), and rearranging terms gives us

$$e_i = m + m(r^\star - 1)\alpha^i \tag{A.8}$$

This completes the proof. ■

Corollary A.6 $\sup_{e_1<\ldots<e_k}(d_0 - d_k)$ *is maximized when* $k = n$.

Proof: Let $c = \frac{m(r^\star - 1)}{M - m} < 1$. The derivative of the logarithm of

$$\frac{d_0}{r^\star} k \left(1 - \left(\frac{m(r^\star - 1)}{M - m}\right)^{\frac{1}{k}}\right)$$

is $\frac{1}{k} - \frac{1}{k^2}\ln(1 - c) > 0$. Therefore,

$$\frac{d_0}{r^\star} k \left(1 - \left(\frac{m(r^\star - 1)}{M - m}\right)^{\frac{1}{k}}\right)$$

is a monotonically increasing function of k, and is maximized when $k = n$. ■

Corollary A.7 *The optimal competitive ratio for the unidirectional problem with bounds m and M is the solution to the implicit equation*

$$r^\star = n\left(1 - \left(\frac{m(r^\star - 1)}{M - m}\right)^{\frac{1}{n}}\right)$$

Proof: TS1 achieves the competitive ratio of $r^\star$ as long as $\sup_{e_1 < \ldots < e_n}(d_0 - d_n) \leq d_0$. The best that TS1 can do is choose $r^\star$ to be such that $\sup_{e_1 < \ldots < e_n}(d_0 - d_n) = d_0$ (i.e. the maximum amount spent by TS1 is d_0):

$$d_0 = \frac{d_0}{r^\star} k \left(1 - \frac{1}{\alpha}\right)$$

Rearranging gives the above equation. ■

Appendix B

Glossary

λ The first time that a forecast has come true for forecasts of the form $\exists i$ such that some condition comes true.

σ An instance of an online problem Σ.

Σ An online problem.

A An online algorithm.

$A^\star$ An optimal online algorithm.

d_i^A The amount of dollars that a trading strategy A will have at the end of trading period T_i. All strategies start with an amount d_0. For OPT we write $A = 0$. We omit A when it is clear which algorithm is being discussed.

$\mathbf{E}$ $= \langle e_1, \ldots, e_n \rangle$. An exchange rate sequence.

$\underline{\mathbf{E}}$ $= \langle \underline{e}_1, \ldots, \underline{e}_n \rangle$. A worst case exchange rate sequence.

e_i The exchange rate offered in period T_i. This is the amount of yen per dollar. The higher the the numerical value of the exchange rate, the cheaper the yen.

F A forecast, i.e. a subset of all the possible problem instances Σ.

f_A $= \frac{r^\star}{\hat{r}_A}$. The reward of the risk tolerant algorithm A, should the forecast come true. We omit A when it is clear which algorithm is being discussed.

I_t $= \{A \,|r_A \leq tr^\star\}$. The set of risk-tolerant algorithms.

$r_A(\mathbf{E})$ The competitive ratio of algorithm A for the exchange rate sequence $\mathbf{E}$. We omit A when it is clear which algorithm is being discussed.

$r^\star$ The optimal competitive ratio.

$\hat{r}_A$ $= \sup_{\sigma \in F} \frac{\mathbf{perf}(\sigma)}{\mathbf{perf}_A(\sigma)}$. The restricted ratio of the algorithm A, which is the competitive ratio of A restricted to the cases when the forecast is true. We omit A when it is clear which algorithm is being discussed.

$\hat{r}^\star$ The optimal restricted ratio.

s_i^A $= d_i^A - d_{i-1}^A$. The amount of dollars traded by strategy A in period T_i. For OPT we write $A = 0$. We omit A when it is clear which algorithm is being discussed.

T_i Trading period i.

U_j $= \max_{i \leq j} e_i$. The maximum exchange rate seen up to and including period T_j.

y_i^A $= \sum_{j=1}^{i} s_j^A e_j$. The amount of yen that trading strategy A has at the end of trading period T_i. For OPT we write $A = 0$. We omit A when it is clear which algorithm is being discussed.

Bibliography

[1] B. Awerbuch, Y. Azar, A. Fiat, and T. Leighton. Making commitments in the face of uncertainty: How to pick a winner almost every time. In *Proceedings of The Twenty-Eighth Annual ACM Symposion On The Theory Of Computing*, pages 519–530, may 1996.

[2] Y. Azar, Y. Bartal, E. Feuerstein, A. Fiat, S. Leonardi, and A. Rosén. On capital investment. In *Proceedings of ICALP 96*, 1996.

[3] A. Chou, J. Cooperstock, R. El-Yaniv, M. Klugerman, and F. T. Leighton. The statistical adversary allows optimal money-making trading strategies. In *Proc. 6th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 467–476, 1995.

[4] R. El-Yaniv, A. Fiat, R. Karp, and G. Turpin. Competitive analysis of financial games. In *Proc. 33th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 327–333, 1992.

[5] R. El-Yaniv, A. Fiat, R. Karp, and G. Turpin. Competitive analysis of financial games. Personal communication, September 1994.

[6] R. El-Yaniv and R. M. Karp. The mortgage problem. In *Proceedings of the 2nd Israeli Symposium on Theory of Computing and Systems*, pages 304 – 312, Jume 1993.

[7] J. D. Gwartney and R. L. Stroup. *Economics: Private and Public Choice.* The Dryden Press, New York, 6 edition, 1992.

[8] Markowitz H. Portfolio selection. *Journal of Finance*, 7(1), March 1952.

[9] L. Hurwicz. Optimality criteria for decision making under ignorance. In *Cowles Commission Discussion Paper*, number 370 in Statistics. 1951.

[10] S. Irani. Lecture notes on online algorithms, 1994.

[11] R. D. Luce and H. Raiffa. *Games and Decisions: Introduction and Critical Survey.* John Wiley and Sons, 1957.

[12] K. R. MacCrimmon and D. A. Wehrung. *Taking Risks: The Management of Uncertainty.* The Free Press, 1986.

[13] M. S. Manasse, L. A. McGeoch, and D. D. Sleator. Competitive algorithms for on-line problems. In *Proc. 20th ACM Symposium on Theory of Computing*, 1988.

[14] P. Raghavan. A statistical adversary for online algorithms. In *DIMACS Series in Discrete Mathematics*, pages 79–83. Association for Computing Machinery, 1992.

[15] M. Rothschild and J. E. Stiglitz. Increasing risk I: A definition. *Journal of Economic Theory*, 2:225 – 243, 1970.

[16] M. Rothschild and J. E. Stiglitz. Increasing risk II: Its economic consequences. *Journal of Economic Theory*, 3:66 – 84, 1971.

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

[18] C. W. Smithson and C. W. Smith Jr. with D. S. Wilford. *Managing Financial Risk: A Guide to Derivative Products, Financial Engineering and Value Maximization.* Irwin, 1995.

[19] F. Wharton. Risk management: Basic concepts and general principles. In J. Ansell and F. Wharton, editors, *Risk: Analysis, Assessment and Managment*, pages 1–14. John Wiley & Sons, 1992.