The Competitive Analysis of Risk Taking with Applications to Online Trading

(Technical Report: CUCS-018-97)

Sabah al-Binali
albinali@cs.columbia.edu
Department of Computer Science
Columbia University
New York, NY 10027

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. In this paper, we extend competitive analysis to provide a framework in which investors can develop optimal trading strategies based on their risk tolerance and forecast. We first define notions of risk and reward that are smooth extensions of classical competitive analysis. We then illustrate our ideas using the ski-rental problem, and analyze a financial game using the risk-reward framework.

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

1 Introduction

Traditional worst case analysis yields little information for online algorithms. A number of approaches have been developed in an attempt to remedy this situation. *Competitive analysis* compares the performance of online algorithms to a benchmark (optimal offline) algorithm [10]. *Probabilistic analysis* compares the expected performance of online algorithms under the assumption that the inputs are drawn from a known probability distribution (see, for example, [4]). Competitive analysis of online algorithms has been criticized as being too conservative. Probabilistic analysis has been criticized as making too strong assumptions (although the *statistical adversary* of [7] attempts to remedy this by weakening the assumptions made). Our risk-reward framework blends the two approaches. A conservative measure is used to select a set of "near-optimal" algorithms and then assumptions are made so as to select an optimal algorithm from this reduced set.

In this paper, we specifically study the *unidirectional conversion problem* of El-Yaniv *et al.* [2]. In this problem, an online investor wishes to convert a given amount of dollars to yen over a specified number of days. Each day, the investor is offered an exchange rate at which he may convert dollars to yen. El-Yaniv *et al.* [2] find an optimal competitive online algorithm for this problem. Chou *et al.* [1] recognize that investors will usually make some assumptions about the future exchange rates, and so analyze the problem using the statistical adversary of [7]. However, another key property of trading strategies is risk (see the reference [11] and the RiskMetrics™ system [5]); risk is not considered in [1] or [2]. In most cases, investors do not seek to minimize risk, but to manage it. The classic work on quantifying risk is [8] and on benefiting from increasing risk is [9]. 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.

In this paper, we extend competitive analysis to allow an investor to provide and benefit from a forecast but also allow him to control his risk of performing too poorly with respect to the optimal offline algorithm should his forecast be incorrect. In this risk-reward framework strategies can be analyzed in terms of their riskiness and the potential benefits (reward) for using them. We also study the unidirectional conversion problem and derive an optimal algorithm for our framework.

1.1 The Risk-Reward Framework

We begin by introducing 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 keeping with the competitive analysis paradigm. In fact, it is a smooth extension, since for a risk of 1 (i.e. we do not accept any algorithm that performs worse than the optimal online algorithm) the analysis within the risk-reward framework coincides with the classical competitive analysis. We next introduce the notion of a *forecast*: what the investor anticipates the market will do in the future. We define a forecast to be a subset of the possible inputs. Again, we depart from making stochastic assumptions, and investigate the type of forecasts that certain investors make. These forecasts usually provide only 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, an investor may develop flexible trading strategies. The investor specifies a maximum acceptable risk level, the 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 investor's risk level for any input sequence.

2 Motivation of the Framework: What is Risk?

We introduce the risk-reward framework by presenting a brief introduction to the concepts, perceptions, measurement and management of risk as discussed in [12]. Most definitions of risk share two characteristics:

- Risk involves some form of negative outcome.
- Risk does not 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, while others see this as a measure, and not a definition, of risk. The definition we will follow is that of risk as the "exposure to a chance of loss" [6]. There are three components to this definition:

- **Exposure to 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.
- **Control of 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 [6] 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 extend this framework to allow the investor to select a riskier strategy. Figure 1 gives a schematic view of the risk-reward framework.

2.1 Definitions: Risk, Reward and Forecasts

Let $\mathbf{perf}_A(\sigma)$ be the performance of the online algorithm A on input σ where σ is an instance of the problem Σ. The performance of the optimal offline algorithm, OPT, is $\mathbf{perf}(\sigma) = \inf_A \mathbf{perf}_A(\sigma)$. Then the *competitive ratio* of an algorithm A on a problem Σ is

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

The *optimal competitive ratio* for a problem is

$$r^\star = \inf_A r_A$$

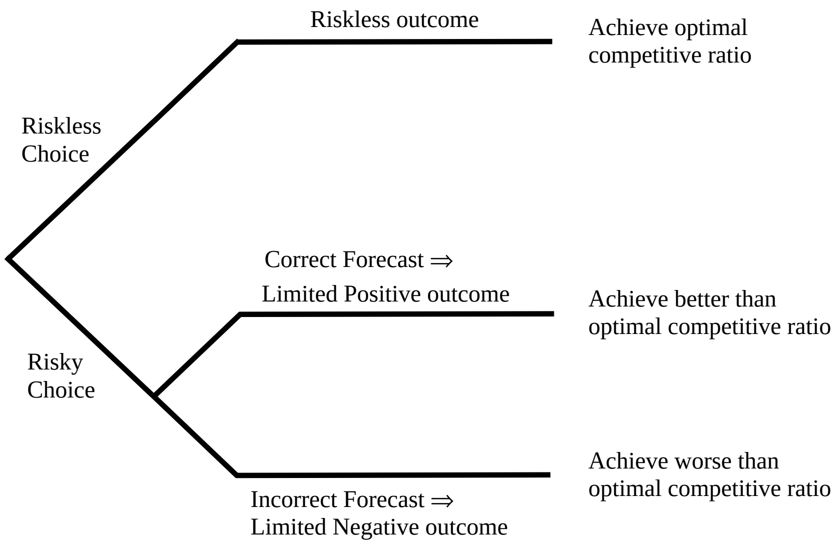

Figure 1: A schematic view of our risk-reward paradigm displayed in the manner of [6]. If the investor chooses not to accept any risk, they use the optimal online algorithm and achieve the optimal competitive ratio. If the investor wishes to take on a specified amount of risk, in some situations they 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.

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

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

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):

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

We denote by $\hat{r}^\star = \inf_A \hat{r}_A$ the *optimal restricted ratio*.

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, an optimal risk-tolerant algorithm is $A^\star \in I_t$ such that:

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

The maximum possible reward for an optimal risk-tolerant is bounded below by 1 and above by the optimal competitive ratio $r^\star$.

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.

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.

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 that we guess we will 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 of renting at most $\frac{b}{a}$ times is the reward maximizing strategy for all tolerance levels. This algorithm is the optimal online algorithm, and so is in I_t for all $t \geq 1$. The restricted ratio of this algorithm is $\hat{r} = 1$, since if the forecast comes true we never buy which is also what OPT does. Hence, the reward in this case is $f = r^\star = 2$, which is the maximum possible reward.

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

In this case, the optimal risk tolerant algorithm is to buy after $j = \left\lceil \frac{b}{a(2t-1)} \right\rceil$ times[1]. 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 the restricted ratio of the online algorithm which buys after j times is $\hat{r} = \frac{ja+b}{b}$. 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

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

$j = \left\lceil \frac{b}{a(2t-1)} \right\rceil$. For this algorithm, 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.

4 Unidirectional Trading in a Risk-Reward Framework

In this section we use the risk-reward framework to analyze the *unidirectional conversion problem* of [2]. We will use the following notation:

T_i Trading period i.

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.

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

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

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

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.

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.

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.

4.1 Unidirectional Conversion Problem

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 $i = 1, \ldots, n$.
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 [3], El-Yaniv *et al.* show that if the exchange rate and the number of trading periods is unbounded, then the optimal competitive ratio is also unbounded. This 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.* [2] consider is that the exchange rate is bounded, $m \le e_i \le M$, for $i = 1, \dots, n$, and that the online investor knows the bounds m, M. Will will adopt this assumption. This gives us the added constraint

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

El-Yaniv *et al.* give an optimal online algorithm, which we will refer to as Trading Strategy 1 (TS1), and which they call a *threat-based* algorithm. That is, TS1 converts the minimum number of dollars to yen to achieve the optimal competitive ratio $r_1^\star$ under the threat that the adversary will drop the exchange rate to m and keep it there for the remaining trading periods. In the next section, we introduce a forecast and an optimal risk-tolerant trading strategy (TS2) for it. The algorithm TS2 is a two stage threat-based algorithm, the first stage is when the forecast has not come true yet, and the second stage is after the forecast has come true.

4.2 Trading Strategy 2

We now analyze the unidirectional trading problem in our risk-reward framework. We will consider a forecast which is typical of many forecasts made by traders. The forecast will be that the exchange rate will increase to at least $m + \Delta$ in one of the trading periods; that is, there exists an i such that $1 \le i \le n$ and $e_i \ge m + \Delta$. The forecast is then $F_2 = \{\mathbf{E} = \langle e_1, \dots, e_n \rangle \mid \exists i \text{ such that } e_i \ge m + \Delta\}$. ($F_1 = \Sigma$ is reserved to denote the null forecast and for which TS1 is trivially the optimal risk-tolerant online algorithm.) Suppose the investor has a risk tolerance of $t \ge 1$. 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 $tr_1^\star$. This is the conservative stage, guaranteeing the investor a competitive ratio of at most $tr_1^\star$, but still "saving" dollars for when the forecast comes true. The second stage begins when the forecast comes true. The algorithm first computes the new minimum achievable competitive ratio $\hat{r}_2^\star$ (which is also the minimum achievable restricted ratio, and which we show how to compute in a later section). The algorithm then trades so as to ensure that a competitive ratio of $\hat{r}_2^\star$ is achieved under the threat that the exchange rate drops to m and remains there for the rest of the trading periods. In this stage, TS2 has more dollars to spend than TS1, and will be able to convert these dollars at higher exchange rates (lower prices) than TS1 did. Thus, TS2 will achieve more yen than TS1 should the forecast come true.

In the rest of this paper, we restrict ourselves to a certain subset of exchange rates, that does not invalidate the results, but allows for a simpler analysis. By similar arguments to those made in [2], we can show that we need only consider exchange rate sequences of the form $mtr_1^\star < e_1 < e_2 < \cdots < e_k < M$ 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 restricted ratio $\hat{r}_2^\star$. We will denote by e_λ the first exchange rate that makes the forecast come true ($\lambda = \min\{i | e_i \geq m + \Delta\}$).

4.2.1 The Algorithm

Trading Strategy 2
Given $m, M, n, r_1^\star, d_0$, a new exchange rate e_i in period T_i, a tolerance t, and a forecast $F_2 = \{\mathbf{E} = \langle e_1, \ldots, e_n \rangle \mid \exists i \text{ such that } e_i \geq m + \Delta\}$, 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. While $e_i < m + \Delta$, convert $s_i^2 = \frac{1}{t} s_i^1$ dollars to yen (i.e. while 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 $\frac{e_i \cdot d_0}{y_i^2 + md_i^2} = tr_1^\star$ which we show in section 4.3.2 implies that $s_i^2 = \frac{d_0}{tr_1^\star} \frac{e_i - e_{i-1}}{e_i - m}$ where $e_0 \triangleq mtr_1^\star$.
3. When the forecast first comes true, compute the optimal restricted ratio $\hat{r}_2^\star$ and start a new game: when executing a trade, only trade enough to guarantee a competitive ratio of $\hat{r}_2^\star$ would be obtained should the exchange rate drop to m and remain there for the remainder of the trading periods. This means that $\frac{e_i \cdot d_0}{y_i^2 + md_i^2} = \hat{r}_2^\star$ which we show in section 4.3.2 implies that $s_\lambda^2 = \frac{d_0}{e_\lambda - m} \left(\frac{e_\lambda}{\hat{r}_2^\star} - \frac{e_{\lambda-1}}{tr_1^\star} \right)$ and for $i > \lambda$, $s_i^2 = \frac{d_0}{\hat{r}_2^\star} \frac{e_i - e_{i-1}}{e_i - m}$

4.3 Analysis of Trading Strategy 2

Trading Strategy 2 is a two stage threat-based algorithm. In the first stage, when the forecast has not yet come true, it attempts to achieve a competitive ratio of $tr_1^\star$ under the threat that the exchange rate falls to m. In the second stage, it attempts to achieve a competitive ratio of $\hat{r}_2^\star$ under the threat that the exchange rate drops to m.

4.3.1 Optimality and Risk-Tolerance

It is easy to see that TS2 is risk-tolerant: If the forecast is not true, then by rule 2 of TS2, the algorithm will achieve a competitive ratio of $tr_1^\star$, and is risk-tolerant for this case. If the forecast does come true, then TS2 achieves a competitive ratio of $\hat{r}_2^\star < r_1^\star$. This is because TS2, by saving dollars in the first stage of the game, converts more dollars at a higher exchange rate in the second stage of the game. A full proof is in the appendix.

We can prove that TS2 is optimal by considering the two cases that an online algorithm A can deviate from TS2. If the first time that A deviates from TS2 is by spending too little, the adversary drops the exchange rate to m, and so A does worse than TS2. If the first time that A deviates from TS2 is by spending too much, then the adversary continues as before. The algorithm A can never make up for spending too many dollars at the lower exchange rate by spending less dollars at the higher exchange rate. A full proof is in the appendix. We therefore have:

Theorem 1 *For the unidirectional trading problem with forecast F_2, TS2 is the optimal risk-tolerant online algorithm. Furthermore, the reward of TS2 is $f_2 > 1$.*

4.3.2 Computing s_i^2

We can compute how many dollars TS2 must convert given an exchange rate e_i in period T_i ($i \neq \lambda$) by considering the threat of the exchange rate dropping to m. If TS2 wishes to achieve a competitive ratio of r, then it must ensure that

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

Notice that $y_i^2 + m d_i^2 = y_{i-1}^2 + s_i^2 e_i + m d_{i-1}^2 - m s_i^2$. Substituting into the above, and solving for s_i^2 gives us

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

In the first stage of the game, we set $r = tr_1^\star$ (assuming ofcourse that $e_i > mtr_1^\star$), and in the second stage of the game, we set $r = \hat{r}_2^\star$. For the case $i = \lambda$, we use the same principle to derive

$$s_\lambda^2 = \frac{d_0}{e_\lambda - m} \left(\frac{e_\lambda}{\hat{r}_2^\star} - \frac{e_{\lambda-1}}{tr_1^\star} \right)$$

4.3.3 The Worst Case Exchange Rate Sequence

The worst case exchange rate sequence can be computed by taking the partial derivatives of $\sum s_i^2$ with respect to e_i, setting them to zero, and solving for e_i (see the appendix).

Theorem 2 *The worst case exchange rate for TS2 is:*

$$e_i = \begin{cases} m + m(tr_1^\star - 1)\alpha_1^i & \text{for } 1 \leq i \leq \lambda, \\ m + \Delta \alpha_2^{i-\lambda} & \text{for } \lambda < i \leq n. \end{cases}$$

where

$$\alpha_1 = \left(\frac{\Delta}{m(tr_1^\star - 1)} \right)^{\frac{1}{\lambda}} \quad \text{and} \quad \alpha_2 = \left(\frac{M - m}{\Delta} \right)^{\frac{1}{n-\lambda}}$$

4.3.4 The Optimal Restricted Ratio

We now turn our attention to computing the optimal restricted ratio. By substituting the worst case exchange rate into our formulation of $\sum s_i^2$, setting this sum equal to 1 we can solve for $\hat{r}_2^\star$:

$$\hat{r}_2^\star = \max_\lambda \frac{\alpha_1 tr_1^\star(\Delta(\alpha_2\lambda + n - \alpha_2 - \lambda - \alpha_2 n) - \alpha_2 m)}{\alpha_2(\Delta(1 - \alpha_1 - \lambda + \alpha_1\lambda - \alpha_1 tr_1^\star) - \alpha_1 m + \alpha_1^\lambda m - \alpha_1^\lambda mtr_1^\star)}$$

There two ways to compute the value of λ that maximizes $\hat{r}_2^\star$. The computationally faster method is to simply evaluate the above expression at all possible values of λ, and choose that value that results in the maximum $\hat{r}_2^*$ (which is what we do). The other method is to take the derivative of the above expression with respect to λ, set it to 0 and solve for λ. This yields an implicit equation in λ that is too long to include.

4.3.5 Practical Matters

There are only n worst case exchange rate sequences. Nature will usually deviate from these sequences, and we can easily take advantage of this. In practice, we would adaptively compute the values of $r_1^\star$ and $\hat{r}_2^\star$ in each trading period. El-Yaniv *et al.* discuss how to do this for TS1. In the appendix, we prove the following for TS2:

Theorem 3 *The new optimal competitive ratio for an n period game in period T_i, given initial dollar and yen amounts of d_0 and y_0 respectively is:*

$$r_1^\star(i) = \frac{d_0 U_i + y_0}{d_{i-1}e_i + y_{i-1}} + \frac{d_{i-1}(e_i - m)}{d_{i-1}e_i + y_{i-1}}(n-i)\left(1 - \left(\frac{e_i - m}{M - m}\right)^{\frac{1}{n-i}}\right)$$

Corollary 4 *The optimal competitive ratio is $r_1^\star = r_1^\star(n)$. The minimum achievable restricted ratio when the forecast F_2 first comes true is $\hat{r}_2^\star = r_1^\star(n - \lambda + 1)$.*

We can thus recompute the optimal competitive ratio and the optimal restricted ratio at each point in the game, so as to take advantage of any deviation in the exchange rate sequence from the worst case sequence.

⟨NEED A DEEPER DISCUSSION HERE, WITH REF. TO IGNORING HISTORY PAPER.⟩ ◁

There is the question of whether or not we should consider the amount of yen acquired so far in computing the restricted ratio for the rest of the game. This is a philosophical question that we will not address here, except to note that if the investor does not wish to consider the yen achieved so far, then he just computes the optimal restricted ratio to be $\hat{r}_2^\star = r_1^\star(n - \lambda + 1, 0, d_{\lambda-1}, e_\lambda)$.

4.3.6 Numerical Results

In this section, we provide numerical results exploring the relationship between the risk-tolerance, the forecast and the optimal restricted ratio. In tables 1 and 2, we set m, M and n, and compute $\hat{r}_2^\star$ for various values of t and Δ. Notice that since TS2 does not trade if $e_i \leq mtr_1\star$, it does not make sense to set $t > \Delta/(mr_1\star)$ since then we would be taking on more risk than necessary.

t	Δ	λ	$\hat{\mathbf{r}}_2^\star$
1.01	10	8	1.06423
1.01	15	21	1.05785
1.05	15	13	1.03964
1.01	17	25	1.05529
1.05	17	20	1.03002

Table 1: Values of $\hat{r}_2^\star$ for different combinations of t and Δ for the game when $m = 100, M = 120, n = 30, r_1\star = 1.0674$.

t	Δ	λ	$\hat{\mathbf{r}}_2^\star$
1.01	25	14	1.14770
1.05	25	9	1.13202
1.10	25	1	1.12649
1.01	35	23	1.14144
1.05	35	21	1.10609
1.10	35	17	1.08302
1.01	40	27	1.13826
1.05	40	26	1.09324
1.10	40	24	1.06200

Table 2: Values of $\hat{r}_2^\star$ for different combinations of t and Δ for the game when $m = 90, M = 135, n = 30, r_1\star = 1.1541$.

It is important to realize that $\hat{r}_2^\star$ and $r_1\star$ are bounded below by 1, and so the maximum improvement is $r_1\star - 1$. Therefore, to compare the results, we should consider the relationship of

$\hat{r}_2^\star - 1$ to $r_1\star - 1$. For the scenario depicted by table 1, if $t = 1.05$ and $\Delta = 0.75(M - m) = 15$ then the ratio $(\hat{r}_2^\star - 1)/(r_1\star - 1) \approx 0.6$. This means that if the investor forecasts that the exchange rate will reach at least the three quarters of the range $[m, M] = [100, 120]$ and is willing to take the risk of achieving a competitive ratio that is 5% larger than optimal, then he can get a performance "improvement" of about $1 - 0.6 = 40\%$ (if his forecast is correct). In table 2, we see that if we more than double the range to $[90, 135]$, again forecast that the exchange rate will reach at least three quarters of the range and take a risk of 10% worse than optimal, then we can get a performance "improvement" of about 46%. These results highlight how, for reasonable values of n, t, Δ and M/m, an investor can improve his performance significantly.

4.4 M and m as Forecasts

In this section we analyze the unidirectional conversion problem but consider the bounds on the exchange rates to be forecasts.

4.4.1 Known n

We first generalize the result of El-Yaniv *et al.* that if the exchange rate and the number of trading periods is unbounded, then the optimal competitive ratio is unbounded:

Lemma 5 *If the sequence of exchange rates for the unidirectional trading problem is unconstrained, and the number of trading periods is known to be n, then $r^\star \geq n$.*

Proof (Sketch): Consider any algorithm A. If there exists an exchange rate sequence that forces A to spend more than $d_0(n-1)/n$ dollars in the first $n-1$ periods, then by making the last exchange rate e_n arbitrarily large, the adversary can force a competitive ratio greater than or equal to n. If there does not exist such an exchange rate sequence, then there must exist a period $T_j (j < n)$ such that no matter how high e_j is, A spends less than d_0/n dollars. ■

Lemma 6 *The competitive ratio of the algorithm which spends $s_i = d_0/n$ dollars in each period T_i is n.*

Proof: From Lemma 5, we have $r \geq n$. Since the above algorithm will always spend d_0/n dollars per period, it will make at least $y_n \geq (d_0 U_n)/n$ yen. Therefore, its competitive ratio is $r = (d_0 U_n)/y_n \leq n$. ■

Corollary 7 *The optimal competitive ratio for the unidirectional conversion problem with unconstrained exchange rates is n, and an optimal online algorithm is to convert $s_i = d_0/n$ dollars in each period T_i.*

The optimal online algorithm of Lemma 6 is called a *dollar cost averaging* strategy.

4.4.2 Forecast of M or m

Lemma 8 *For the unidirectional trading problem with the sequence of exchange rates unconstrained, known number of trading periods n, forecast $F = \{\langle e_1, \ldots, e_n\rangle | e_i \leq M\}$, where $M > 0$ is a known constant, and tolerance t, then the restricted ratio is $\hat{r}^\star = r^\star = n$, and the reward is 1.*

Proof (Sketch): El-Yaniv *et al.* show that the optimal competitive ratio $r_1^\star$ for the unidirectional problem where the exchange rate is known to be bounded by m and M is

$$r_1^\star = n\left(1 - \left(\frac{m(r_1^\star - 1)}{M - m}\right)^{\frac{1}{n}}\right)$$

Clearly $\lim_{M\to\infty} r_1^\star = n$. ∎

Similarly, we have:

Lemma 9 *For the unidirectional trading problem with the sequence of exchange rates unconstrained, known number of trading periods n, forecast $F = \{\langle e_1, \ldots, e_n\rangle | e_i \geq m\}$ where $m > 0$ is a known constant, and tolerance t, then the restricted ratio is $\hat{r}^\star = r^\star = n$, and the reward is 1.*

5 Further Work

Many questions remain, some of which are:

- Analyzing the unidirectional conversion problem using more complex forecasts, such as $F = \{\langle e_1, \ldots, e_n\rangle | m \leq e_i \leq M\}$, or those used by technical analysts.
- The *bidirectional conversion problem* of [2], where an investor may convert dollars to yen and yen to dollars, has an optimal competitive ratio that is exponential. Can the risk-reward framework be used to allow a more interesting analysis?
- We have shown how nonprobabilistic assumptions about the input can be utilized in developing online algorithms. However, the forecast can be of any form. Can our risk-reward framework be easily integrated with probabilistic analysis or the statistical adversary to yield useful information?

Acknowledgements

I would like to thank Simon Baker, Zvi Galil, Sandy Irani, Marti Subrahmanyam and Moti Yung for helpful discussions and comments.

References

[1] 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.

[2] 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.

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

[4] P. A. Franaszek and T. J. Wagner. Some distribution-free aspects of paging algorithm performance. *Journal of the ACM*, 21(1):31–39, 1974.

[5] J.P. Morgan. *RiskMetrics^{TM} – Technical Document*, fourth edition, 1996. Available at: http://www.jpmorgan.com/RiskManagement/RiskMetrics/RiskMetrics.html.

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

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

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

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

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

[11] 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.

[12] 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.

A Analysis of Trading Strategy 2

Remark 10 *From rules* 1 *and* 2 *of TS2, we know that when analyzing TS2, we need only consider exchange rate sequences of the form* $mtr^\star < e_1 < e_2 < \cdots < e_k < M$ *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* $\hat{r}^\star$.

A.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 11 *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 $1 \le i < \lambda$.
- $s_\lambda^A = s_\lambda^1 + d_{\lambda-1}^2 - d_{\lambda-1}^1$.
- $s_i^A = s_i^1$ for $\lambda < i \le n$.

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

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

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

Corollary 12 *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 13 *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 11 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 14 *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 15 *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 14. ■

Theorem 16 *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 10. Then $s_i^A = s_i^2 + f(e_i)$ where $\sum_{i=1}^{n} f(e_i) = 0$. Then, by Lemmas 14 and 15 we have that $\sum_{i=1}^{n} e_i f(e_i) < 0 \Rightarrow \sum_{i=1}^{n} e_i s_i^A < \sum_{i=1}^{n} e_i s_i^2$. ■

A.2 Computing s_i^2

We know that

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

By a similar analysis to [2], we have that

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

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

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

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: $1 \leq i \leq \lambda$

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

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 $\lambda \leq i \leq n$

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

■

A.3 Proof of Theorem 3

We follow the same reasoning as for TS1 in [2]. For ease of exposition, we set $r^\star = r^\star(k, y_0, d_0, e_1)$. 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 $e_k = M$. ■

A.4 Proof of Lemma 5

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

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

■