

# Competitive Solutions for Online Financial Problems

RAN EL-YANIV

*Institute of Computer Science, The Hebrew University, email: <ranni@cs.huji.ac.il>*

This article surveys results concerning online algorithms for solving problems related to the management of money and other assets. In particular, the survey focuses on search, replacement, and portfolio selection problems.

Categories and Subject Descriptors: F.1.2 [**Computation by Abstract Devices**]: Modes of Computation—*online computation, probabilistic computation*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*sorting and searching*; G.2.M [**Discrete Mathematics**]: Miscellaneous; G.3 [**Mathematics of Computation**]: Probability and Statistics; J.1 [**Computer Applications**]: Administrative Data Processing—*financial*

General Terms: Algorithms, Measurement, Performance, Theory

## 1. INTRODUCTION

This article<sup>1</sup> surveys work related to online financial problems, that is, online problems (i.e., a single player decision problem under uncertainty) related to the management of money and other assets. More precisely, we are concerned with online tasks whose natural modeling is in terms of monetary cost or profit functions. The family of such problems (and their related applications) is extremely broad, and the related literature is vast and originates from a number of disciplines including economics, finance, operations research, and decision theory.

In all these fields much of the related work is *Bayesian*. This means that all uncertain events are modeled by means of stochastic processes that are typically assumed to be known to the decision maker. Then the goal is to optimize the average-case performance (often with

respect to a utility function). Indeed, for financial problems this approach has been dominant over the last several decades and has led to the development of rich mathematical theory.

In contrast, in this survey we are concerned with non-Bayesian analyses of online financial problems while focusing on those using the *competitive-ratio* (worst-case) optimality criterion (defined in Section 2). This criterion provides a complementary type of analysis for online problems that can lead to different types of algorithms and a different perspective.

There are vastly fewer papers related to worst-case competitive analysis of financial problems than papers related to Bayesian analyses of these problems. Still, however, due to space limitations we focus here only on a subset of the work done in this field of study.

Many of the financial problems stud-

---

<sup>1</sup> Parts of this article are based on Ch. 14 of *Online Computation and Competitive Analysis* by A. Borodin and R. El-Yaniv, Cambridge University Press, 1998.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 1998 ACM 0360-0300/98/0300-0028 \$5.00

## CONTENTS

1. INTRODUCTION
    - 1.1 General Modeling Issues
    - 1.2 Optimality Criteria
    - 1.3 Descriptive Versus Prescriptive Theories
    - 1.4 Survey Organization
  2. PRELIMINARIES
    - 2.1 Online Problems and Competitive Analysis
    - 2.2 Viewing Online Problems as Two-Player Games
    - 2.3 Competitive Analysis
  3. SEARCH PROBLEMS
    - 3.1 The Elementary Search Problem
    - 3.2 Applications of Search
    - 3.3 Basic Features of Bayesian Results
    - 3.4 Competitive Search Algorithms
  4. REPLACEMENT PROBLEMS
    - 4.1 A General Replacement Problem
    - 4.2 Applications of the Replacement Problem
    - 4.3 On Some Bayesian Solutions for the Replacement Problem
    - 4.4 Competitive Replacement Algorithms for the Elementary Continuous-Time Finite-Horizon Problem
    - 4.5 Discrete-Time Replacement with Multiple Permanent Replacement Options
  5. TWO-WAY TRADING AND PORTFOLIO SELECTION
    - 5.1 The Elementary Portfolio Selection Problem
    - 5.2 Buy-and-Hold Versus Market Timing
    - 5.3 Online Portfolio Selection
    - 5.4 Other Performance Measures
    - 5.5 Results for the Two-Way Trading Problem
    - 5.6 Other Statistical Adversaries and “Money-Making” Algorithms
    - 5.7 The Fixed Fluctuation Model
    - 5.8 Online Portfolio Selection, Results for Arbitrary  $m$
  6. COMPETITIVE RISK MANAGEMENT
    - 6.1 One-Way Trading Revisited: Analysis with Risk Management
  7. CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE RESEARCH
- ACKNOWLEDGMENTS  
REFERENCES

ied in the literature and all of those discussed in this survey are a variant or an application of one of the following elementary problems, which are all described in terms of a one-player game against an *adversary* (sometimes called *nature*). The player is called the *online player*. Each one of these games takes place during some time horizon that may be continuous or discrete, divided into time periods. In all the games the online player’s general objective is to minimize (resp., maximize) some monetary cost (resp., profit) function.

—*Search problems.* The online player is searching for the maximum (resp., minimum) price in a sequence of prices that unfolds sequentially. At the beginning of each time period the player can pay some sampling cost to obtain a price quotation  $p$ , after which the player has to decide whether to accept  $p$  or continue sampling more prices. The game ends when the player accepts some price. The player’s total return is the accepted price  $p$  minus the sum of all sampling costs incurred. Typical applications are: searches for jobs and employees, search for the best price of some product or asset, and the like.

—*Replacement problems.* At each time period the player is engaged in one activity. Associated with each activity is a pair of numbers called the changeover cost and the flow rate. During the periods in which the player is engaged in some activity his budget is depleted at a rate corresponding to this activity’s flow rate. From time to time new activities are offered as possible replacements for the current one. If the player decides to change to a new activity he must pay the associated changeover cost. The question is when to switch from one activity to another so that the total cost, consisting of the sum of all flows and changeover costs, is minimized. Typical applications are: equipment replacement, job replacement, supplier replacement, mortgage refinancing, and the like.

—*Portfolio selection.* In the basic setup of this problem the online player has some initial wealth and at the start of each period must decide how to reallocate her current wealth among the available investment opportunities, which may be commodities, securities, and their derivatives. The value of each investment opportunity changes from period to period in an uncertain manner and the goal of the player is to maximize her total wealth (determined by the final value of her hold-

ings). In a more realistic version of this problem, each reallocation incurs a commission that is typically proportional to the money flow required to execute the reallocation.

—*Leasing problems.* The player needs to use some equipment during a number of time periods. This number is not known in advance but is made known online. Specifically, at the start of each period it becomes known whether the equipment will be needed in the current period and the player must choose whether to buy the equipment for a price  $b$  or to rent it for a price  $r$ , with  $r < b$ . The game is over when the player purchases the equipment or no longer needs the equipment. Note that leasing problems can be presented as rudimentary forms of replacement problems. Due to lack of space the leasing problem is not discussed here. The reader is referred to the papers by El-Yaniv et al. [1993] and Irani and Ramanathan [1994] that study this problem using two different approaches.

### 1.1 General Modeling Issues

Each of the preceding problems attempts to model a particular economic situation in which an individual or an institution may be repeatedly involved within the modern market. Clearly, each of these problems abstracts away many factors while attempting to capture some of the essence of the decision-making task in question. With respect to each problem, various general or problem-specific features can be considered and incorporated in the model. In particular, the following features may be considered: (i) finite versus infinite time horizons; (ii) discrete versus continuous time; (iii) discounting versus nondiscounting; and (iv) consideration of various utility functions that represent the player's subjective rate of diminishing marginal worth. Clearly, the particular model chosen to formulate the problem determines the degree of its relevance to the corresponding real-life

problem, whereas the addition of more features typically trades off its mathematical tractability.

The analyses discussed in this article typically concern the simpler models of the corresponding problems. Thus they may be used and should be judged as starting points for more elaborate models and results.

### 1.2 Optimality Criteria

This survey is oriented towards results concerning competitive algorithms. That is, our primary concern is the analysis and design of online algorithms using the competitive-ratio decision criterion for uncertainty (definition and discussion follow in Section 2). A broader view encompasses (financial) decision making under uncertainty using other criteria such as the minimax cost (maximin return) and others. Indeed, we also discuss analyses of online algorithms via other optimality criteria including some criteria that relax the strict uncertainty assumptions typical of "pure" competitive analysis. Due to space limitations, Bayesian solutions are not surveyed at all, although occasionally we point out key ideas or relevant features of Bayesian solutions that may bring further insight or motivation.

### 1.3 Descriptive Versus Prescriptive Theories

*Descriptive* decision-making theories attempt to analyze and predict how individuals or institutions act. In contrast, *prescriptive* (or *normative*) theories attempt to describe how rational decision makers should act ideally. Although there may be an overlap between descriptive and prescriptive theories, it is not in general expected that they will coincide. Of course, both kinds of theories have their own independent merits. For example, if one wishes to predict the behavior of a certain market based on the aggregate actions of the individual decision makers, then descriptive theories should be employed. On the

other hand, prescriptive theories attempt to advise decision makers how to optimize their performance, sometimes even against their intuitive understanding. The “competitive” approach discussed in this survey is, for the most part, a prescriptive approach.

### 1.4 Survey Organization

This survey is organized as follows. Section 2 briefly presents the definitions and notation used throughout. This section also briefly discusses the competitive analysis approach, its advantages, and drawbacks. Sections 3 through 5 survey results concerning these basic financial problems in the order previously stated. For each problem (except for the leasing problem) there is a special section that includes: a formal description of the basic problem and some of its variants, a discussion of some basic features of the problem and its notable applications, and a survey of selected known solutions. In Section 6 we present a recent generalization of the competitive analysis framework that allows risk management, a required feature in financial decision making. We conclude with Section 7, pointing out directions for future research.

## 2. PRELIMINARIES

### 2.1 Online Problems and Competitive Analysis

Consider a cost-minimization problem  $\mathcal{P}$  consisting of a set  $\mathcal{I}$  of inputs and a cost function  $C$ . Associated with each input  $I \in \mathcal{I}$  is a set of feasible outputs  $F(I)$ . For each input  $I$  and a feasible output  $O \in F(I)$ , the cost associated with  $I$  and  $O$  is  $C(I, O) \in \mathbb{R}^+$ . Let ALG be any algorithm for  $\mathcal{P}$ . Denote by  $\text{ALG}[I]$  a feasible output produced by ALG given the input  $I$ . The cost incurred by ALG is denoted  $\text{ALG}(I) = C(I, \text{ALG}[I])$ . In the problems considered here each input  $I$  is a finite sequence,  $I = i_1, i_2, \dots, i_n$  and a corresponding feasible output is a finite sequence  $O = o_1, o_2, \dots, o_n$ . An

online algorithm for  $\mathcal{P}$  must produce a feasible output in stages such that at the  $j$ th stage the algorithm is presented with the  $j$ th component of the input and must produce the  $j$ th component of (a feasible) output before the rest of the input is made known. Denote by  $\text{OPT}$  an optimal offline algorithm for  $\mathcal{P}$ . That is, for each input  $I$ ,

$$\text{OPT}(I) = \min_{O \in F(I)} C(I, O).$$

An online algorithm ALG is  $c$ -competitive (or “attains a competitive ratio  $c$ ”) if there exists a constant  $\alpha$  such that for each input  $I$ ,

$$\text{ALG}(I) \leq c \cdot \text{OPT}(I) + \alpha.$$

The smallest  $c$  such that ALG is  $c$ -competitive is called ALG’s *competitive ratio*. Thus a  $c$ -competitive algorithm is guaranteed to incur a cost no larger than  $c$  times the smallest possible cost (in hindsight) for each input sequence up to the additive constant  $\alpha$ . For profit maximization problems the competitive ratio is defined analogously. Specifically, with  $\text{ALG}(I)$  (resp.,  $\text{OPT}(I)$ ) denoting the profit (or return) accrued by ALG (resp., OPT), ALG is  $c$ -competitive if there exists a constant  $\alpha$  such that for all  $I$ ,

$$c \cdot \text{ALG}(I) \geq \text{OPT}(I) + \alpha.$$

For bounded problems where the cost (resp., profit) function is bounded we require in the preceding definitions that the constant  $\alpha$  be zero.

The extension of the competitive-ratio definition to randomized online algorithms is straightforward under the assumption that the adversary generating the input sequences is oblivious to the random choices made by the online player.<sup>2</sup> Specifically, the definition of the competitive ratio with respect to an oblivious adversary is the same as the

<sup>2</sup> Such an adversary is called an oblivious adversary. Other kinds of adversaries are of adaptive type (see Ben-David et al. [1990]). In this survey we consider only oblivious adversaries.

preceding with  $E[\text{ALG}(I)]$  replacing  $\text{ALG}(I)$ , where  $E[\cdot]$  is the expectation with respect to the random choices made by  $\text{ALG}$ .

## 2.2 Viewing Online Problems as Two-Player Games

When we optimize through the competitive ratio measure, it is often convenient to view an online problem as the following two-player game. The first player is the online player (running the online algorithm). The second player is called the adversary or the offline player. The online player chooses an (online) algorithm  $\text{ALG}$  and makes it known to the adversary. Then, based on  $\text{ALG}$ , the adversary chooses an input so as to maximize the competitive ratio. The online player's objective is to minimize the competitive ratio (which means that this game is a zero-sum game). Thus determining competitive-optimal algorithms is equivalent to determining an optimal strategy for the first player and the best possible competitive ratio is the value of this game, which in general is obtained using randomized online algorithms.

## 2.3 Competitive Analysis

The use of the competitive ratio for the evaluation of online algorithms is called *competitive analysis*.<sup>3</sup> Competitive analysis was first used by computer scientists in the '70s in connection with approximation algorithms for NP-complete problems (see, e.g., Graham [1966], Johnson [1973], Johnson et al. [1974], and Yao [1980]) and was more explicitly formulated in the '80s in the seminal work of Sleator and Tarjan [1985] on list accessing and paging algorithms. Since then, competitive analysis has been used extensively to analyze and design online algorithms for many online optimization problems related to computer systems. Within the theoretical computer science community, the compet-

itive ratio has gained much recognition and has become a standard approach for the analysis of online problems.

One common argument against the use of competitive algorithms is that they are inherently risk-averse as they are optimized with respect to worst-case event sequences. This argument is certainly valid in cases where decision makers possess reliable information about stochastic processes that determine the uncertainties. In this case the use of competitive algorithms that blatantly ignore this information may lead to inferior performance relative to Bayesian algorithms. Nevertheless, in many instances decision makers do not possess such information. In some cases, by investing sufficient resources they may acquire such information, whereas in other situations, often due to the complexities of event sequences, not much can be learned about the underlying stochastic process. Whatever the reason for the absence of such information, competitive algorithms offer reasonable initial solutions upon which more elaborate algorithms may be constructed after additional information is determined.

Although the baseline competitive analysis approach is purely worst-case, it can be extended to a framework that can utilize and exploit forecasts while allowing risk control (see Section 6), a desirable feature especially in financial decision making. Nevertheless, this generalized competitive analysis with risk management is still a worst-case approach that dispenses with probabilistic assumptions. This property can be favorable for risk-averse financial decision makers who may prefer somewhat inferior but guaranteed performance to better average performance.

## 3. SEARCH PROBLEMS

### 3.1 The Elementary Search Problem

The online player is searching for the maximum price in a sequence of prices that unfolds sequentially. At the begin-

<sup>3</sup> This term was coined by Karlin et al. [1988].

ning of each time period  $i$  the player can pay some sampling cost  $c_i \geq 0$  to obtain a price quotation  $p_i$ , after which the player has to decide whether to accept  $p_i$  or continue sampling more prices. The game ends when the player accepts some price  $p_j$  and the total return is then

$$p_j - \sum_{1 \leq i \leq j} c_i.$$

Search for the minimum price is similarly defined with respect to a cost function where the total cost of the player is  $p_j + \sum_{1 \leq i \leq j} c_i$ .

This elementary search problem has many extensions and variations. A search problem is termed *with recall* if all or some number  $k$  of the most recent price offers are retained and the player may choose any of the retained offers. We distinguish between search problems with and without recall and with and without discounting. Also, we distinguish between discrete and continuous time problems. In either case we assume a finite time horizon but distinguish between searches of known and unknown duration. In the case of unknown duration we assume that the player is informed just before the last period that the game will end, thus giving the player the opportunity to accept the last price if he or she has not accepted any price earlier.

### 3.2 Applications of Search

Search is a most fundamental feature of economic markets. Let us mention some of its basic applications.

**3.2.1 Job and Employee Search.** These two applications are of major importance to labor markets. In the *job search* application, the player is seeking employment. In each period the job seeker obtains one job offer (which corresponds to the preceding price quotation). The offer can be interpreted as the lifetime earning from the job. The sampling cost corresponds to the cost of generating an offer; it includes all expenditures such

as advertising and transportation and may also include the loss incurred from being currently unemployed. Thus each sampling cost may be modeled as a constant  $c$ .

Just as prospective employees are searching for jobs, employers are searching for employees to fill job vacancies. In the *employee search* application, the player is an employer searching for an employee with certain identifiable characteristics that should correspond to the employee's productivity (modeled via the preceding "price quotation").<sup>4</sup> This modeling has an underlying assumption that there are such characteristics that are correlated to the employee's productivity and that these characteristics can be tested by the employer.

**3.2.2 Search for the Lowest Price of Goods.** Here the player considers the purchase of some goods that are sold at different stores at different prices. The player can elicit quotations from the various sellers by paying the sampling fee that takes into account traveling or phone costs and the time wasted for sampling, and may include a penalty of managing without the goods since the start of the search. Like the job (and employee) search applications, this application is of fundamental value to economics because optimal searching rules determine the demand function that sellers face, which in turn determines (some of) the nature of the markets themselves.

**3.2.3 One-Way Trading.** In this application the player is a trader who is considering the exchange of some initial wealth  $w_0$ , given in some currency (say, dollars), to some other asset or currency (say, yen). Each period starts when a new price quotation is made available

<sup>4</sup> This application is closely related to the well-known secretary problem [Freeman 1983; Ajtai et al. 1995], the difference being that in secretary problems the objective is typically to accept one or more "secretaries" of best ordinal value among an ordered set of all secretaries.

and the trader must decide whether to accept it or wait for a better price. In this application the sampling cost is typically negligible, as prices are widely available in banks, newspapers, and quotation services. Nevertheless, the trader is required to pay transaction fees (e.g., to a financial institution), that are typically some fixed percentage of the return. Note that in this application the trader may partition the initial wealth and exchange  $w_0$  sequentially in parts. One-way trading algorithms can be applied in various economic situations. For instance, consider a fund manager who decides to change the position of a portfolio and enter (or exit) some market (in which case  $w_0$  is the part of her wealth allocated to the new position). Another natural instance related to foreign exchange is when the player, for the purpose of emigrating to a foreign country, sells his local property in order to exchange the local currency received into the foreign one.

### 3.3 Basic Features of Bayesian Results

The Bayesian approach derives search strategies that are dependent on a prior distribution of prices that is usually assumed to be known to the player. Also, it is typically assumed that this distribution is fixed throughout the search and that prices are independent observations drawn from this distribution. The theory developed is very rich and relies on mathematical tools from the theory of optimal stopping [Chow et al. 1971]. The reader is referred to the excellent surveys by Lippman and McCall [1976; 1981] that discuss Bayesian solutions for a wide array of search variants. One notable feature of Bayesian optimal search algorithms (applicable to many problem variants) is that they have the following structure: based on the problem parameters (in particular, the probability distribution assumed) there is a single fixed critical number, the *reservation price*, such that the optimal policy is to reject all prices below the reservation price and to accept any

offer above it. Reservation prices change dynamically in the case of finite horizon and known duration (and they are non-decreasing in problems without recall; the case of recall is more complicated and not fully solved). Another issue of interest when pursuing Bayesian solutions is the expected number of price quotations required until stopping.

The least acceptable assumption of the Bayesian search models is that the probability distribution of prices is fully known to the player. Several models attempt to relax this assumption. For example, Rosenfield and Shapiro [1981] studied cases where the distribution itself is a random variable and the player knows the probability distribution of some of its moments (say, the price distribution is unknown but known to be normal and the distribution of its mean is known). In various cases such relaxed models do not admit optimal solutions with a reservation price (see Lippman and McCall [1976]).

### 3.4 Competitive Search Algorithms

In this section we describe various competitive solutions to variants of the search problem without recall in which the sampling cost is zero or is a fixed percentage of the return. With respect to the competitive ratio these two variants are readily equivalent. For the rest of this section assume that the sampling cost is zero. In all the variants we consider we assume only that prices are drawn from some finite interval  $[m, M]$ . Call the ratio  $\varphi = M/m$  the global fluctuation ratio. In one of the variants the online player knows the values  $m$  and  $M$  and in another variant the player knows only  $\varphi$ . We seek to determine the optimal competitive performance in terms of these parameters.

Suppose that  $m$  and  $M$  are known to the player. The optimal deterministic solution is the following reservation price policy (denoted RPP): accept the first price greater than or equal to  $p^* = \sqrt{Mm}$ . This strategy is  $\sqrt{\varphi}$ -competitive and its analysis is trivial. Using a bal-

ancing argument,  $p^*$  should be chosen to equate the performance ratio, offline to online, corresponding to the two events: the maximum price encountered,  $p_{\max}$ , will be greater than  $p^*$ ; and  $p_{\max} \leq p^*$ . It follows that  $p^*$  is the solution  $p$  of  $M/p = p/m$ . The preceding reservation policy is optimal for both an infinite and finite time horizon and when the duration is known or unknown. It is not hard to see that if only  $\varphi$  is known to the player, then no competitive ratio smaller than the trivial one,  $\varphi$ , is achievable by a deterministic algorithm.

A considerable improvement is obtained by using randomization.<sup>5</sup> First we show how a simple randomized strategy due to Levin [1994] attains a competitive ratio of  $O(\log \varphi)$ . For simplicity, assume first that  $\varphi = 2^k$  (for some integer  $k$ ). For  $i = 0, 1, \dots, k - 1$  denote by  $S(i)$  the deterministic reservation price policy with reservation price  $m2^i$ . Levin's randomized strategy, denoted  $S$ , is a uniform probability mixture over  $\{S(i)\}_i$ ; that is,  $S$  chooses  $S(i)$  with probability  $1/k$ . It is easy to show that there exists a function  $f(\varphi)$  such that  $S$  is  $[f(\varphi) \log \varphi]$ -competitive with  $f(\varphi)$  greater than but approaching 1 as  $\varphi$  grows. Let  $p_{\max}$  be the (posteriori) maximum price obtained and let  $j$  be an integer such that  $m2^j \leq p_{\max} < m2^{j+1}$ . The optimal offline return is  $p_{\max}$ . For each  $i \leq j$  the strategy  $S(i)$  chooses a price not smaller than  $m2^i$  and for all  $i > j$ ,  $S(i)$  obtains at least  $m$ . It follows that  $S$  will obtain at least  $(2^{j+1} + (k - j - 1))m/k$  on average. The resulting competitive ratio is greater than, but approaching,  $k = \log \varphi$ .

Exactly the same bound holds even if the player does not know the values of  $m$  and  $M$  and knows only  $\varphi$ . Here, however, the strategies  $S(i)$  are set after the first price  $p_1$  is revealed (in which case

$S(i)$  has reservation price  $p_1 2^i$ ). For an arbitrary  $\varphi$  (not a power of two) the same method gives  $O(k)$ -competitive performance where  $b^k = \varphi$  ( $b$  real and  $k$  positive integer). Recall that for this variant (only  $\varphi$  is known) no deterministic strategy can achieve a competitive ratio smaller than  $\varphi$ .

This simple randomized strategy can be modified to work even when the player does not know  $\varphi$ . In this case it can be shown that there exists a function  $c(\varepsilon)$  such that a competitive ratio of  $O(c(\varepsilon) \cdot \log^{1+\varepsilon}(\varphi))$  can be obtained for every positive  $\varepsilon$ , where  $\varphi$  is the posteriori global fluctuation ratio.

For the variants where  $m$  and  $M$  are known or only  $\varphi$  is known to the player (known or unknown duration), the competitive ratio of  $O(\log \varphi)$  is within a multiplicative factor of the best that can be obtained against an oblivious adversary. This is established in El-Yaniv et al. [1993] which presents strictly optimal algorithms for these variants. The presentation of the results of this paper is simplified when described as a one-way trading game. Consider a trader with initial wealth of \$1. The trader is presented with a sequence of exchange rates  $p_1, p_2, \dots$ , where  $p_i$  gives the exchange rate, yen per dollar, for the  $i$ th period (say, day). It is assumed that all rates are drawn from  $[m, M]$ . On each day  $i$  the trader can exchange any amount  $s_i$  of his remaining dollars for yen. Upon completion the total return of the trader is the amount of yen he accumulated. It is assumed that arbitrary fractions of dollars can be traded.

Notice that in the search problem the online player must accept one price and in the one-way trading problem the trader can partition his initial wealth and trade the parts sequentially, each part at different exchange rates. Nevertheless, these problems are closely related and, in fact, equivalent in the following sense. Any deterministic one-way trading algorithm can be interpreted as a randomized search algorithm and vice versa. This follows from the fact that any deterministic one-way

<sup>5</sup> Note that in general randomization does not give the player any advantage if it is assumed that the price distribution is known and an average-case performance is sought (i.e., the Bayesian approach).

trading algorithm that trades the initial wealth in parts is equivalent (in terms of returns) to a randomized trading algorithm and vice versa. Specifically, suppose that a deterministic algorithm trades a fraction  $s_i$  of its initial wealth at the  $i$ th period,  $\sum_i s_i = 1$ . The quantity  $s_i$  can be interpreted as the probability of trading the entire wealth at the  $i$ th period. Clearly, the average return of this randomized algorithm equals the return of the deterministic algorithm. Thus the competitive ratio of the deterministic algorithm is exactly the average competitive ratio of the randomized algorithm against an oblivious adversary. On the other hand, using Kuhn's [1953] theorem of game theory, any randomized one-way trading algorithm is equivalent to a mixed randomized algorithm (i.e., a probability distribution over deterministic algorithms), and by linearity of expectation (and using the fact that the sum of all traded amounts is 1) this mixed algorithm is equivalent to a deterministic algorithm that trades the initial wealth in parts. It follows that an optimal deterministic one-way trading algorithm is an optimal randomized search algorithm. This implies that randomization cannot improve the competitive performance in one-way trading. In contrast, randomization is advantageous for search. When fixed sampling costs are introduced (i.e., the player pays some constant for each price quotation she obtains) there is no longer an equivalence between deterministic one-way trading algorithms and randomized search (and one-way trading) algorithms. The reason is that the randomized algorithm may spend less on sampling on the average.

**3.4.1 Threat-Based Algorithms.** The optimal performance is obtained by algorithms that obey the following threat-based policy. Let  $c$  be any competitive ratio that can be attained by a deterministic trading algorithm. Assume that  $c$  is known to the trader. For each such  $c$  the corresponding threat-based policy consists of the rules:

- Rule 1.** consider trading dollars for yen only when the current rate is the highest seen so far;
- Rule 2.** whenever you convert dollars, convert just enough to ensure that a competitive ratio  $c$  would be obtained if an adversary dropped the exchange rate to the minimum possible rate<sup>6</sup> and kept it there throughout the game.

Thus algorithms prescribed by this policy convert dollars to yen based on the threat that the exchange rate will drop permanently to the minimum possible rate. For each attainable competitive ratio  $c$  the corresponding threat-based algorithm can be shown to be  $c$ -competitive. Intuitively, this statement can be justified as follows. Consider the first trade (exchange rate is  $p_1$ ). Since the current exchange rate is the highest seen so far, the algorithm considers a trade. The competitive ratio  $c$  is attainable (by some deterministic trading algorithm), and therefore there exist some  $s \geq 0$  such that the ratio  $c$  will still be attainable if  $s$  dollars are traded for yen. Furthermore, the chosen amount of dollars  $s$  is such that the ratio  $c$  is so far guaranteed even if there is a permanent drop of the exchange rate and no further trades are conducted (except for one last trade converting the remaining dollars with the minimum possible exchange rate). In particular, there is no need to consider any exchange rate smaller than  $p_1$ . Similar arguments can be used to justify the choice of the amounts chosen for the rest of the trades and may convince the reader that this policy induces a  $c$ -competitive algorithm. A sketch of a more formal analysis follows.

Assume the problem variant with

<sup>6</sup> The "minimum possible rate" is defined with respect to the information known to the trader. That is, it is  $m$  if  $m$  is known, and it is  $p/\varphi$  if only  $\varphi$  is known and  $p$  is the highest price seen so far.

known duration  $n$  and known  $m$  and  $M$ . We now show how the optimal threat-based algorithm (denoted `THREAT`) for this variant can be derived. Any exchange rate that is not a global maximum at the time it is revealed to the trader is ignored (Rule 1). Hence we can assume w.l.o.g. that the exchange rate sequence consists of an initial segment of successive maxima of length  $k \leq n$ . In order to realize a threat the adversary may choose  $k < n$  and choose  $p_{k+1} = p_{k+2} = \dots = p_n = m$ .

Initially, assume that the optimal competitive ratio attainable by `THREAT`  $c^*$ , is known. For each  $i = 0, 1, \dots, n$ , set  $D_i$  and  $Y_i$  to be the number of remaining dollars and the number of accumulated yen, respectively, just after the  $i$ th period. By assumption the trader starts with  $D_0 = 1$  dollar and  $Y_0 = 0$  yen. Let  $s_i = D_{i-1} - D_i$  be the number of dollars traded at the  $i$ th period,  $i = 1, 2, \dots, n$ . Thus  $Y_i = \sum_{j=1}^i s_j p_j$ . The ratio  $c^*$  is attained by algorithm `THREAT` and therefore, by Rule 2, it must be that the amounts  $s_i$  are chosen such that for any  $1 \leq i \leq n$ ,

$$\begin{aligned} & \frac{p_i}{Y_i + m \cdot D_i} \\ &= \frac{p_i}{(Y_{i-1} + s_i p_i) + m \cdot (D_{i-1} - s_i)} \leq c^*. \end{aligned} \quad (1)$$

Here the denominator of the left-hand side represents the return of `THREAT` if an adversary dropped the exchange rate to  $m$ , and the numerator of the left-hand side is the return of `OPT` for such an exchange rate sequence. By Rule 2, `THREAT` must spend the minimal  $s_i$  that satisfies inequality (1). The left-hand side is decreasing with  $s_i$ . Since  $s_i$  is chosen to achieve the target competitive ratio  $c^*$ , and since the trader must spend the minimum possible amount (in order to leave as much as possible for higher rates), we replace the inequality

in (1) with equality. Solving the resulting equality for  $s_i$  we obtain

$$s_i = \frac{p_i - c^* \cdot (Y_{i-1} + m D_{i-1})}{c^* \cdot (p_i - m)}. \quad (2)$$

From (1) (with equality) we also obtain the relation:

$$Y_i + m \cdot D_i = p_i / c^*. \quad (3)$$

Closed-form expressions for the  $s_i$ s can now be obtained. From (2) at  $i = 1$  we get  $s_i = 1/c^* (p_1 - mc^*)/(e_1 - m)$ . From (2) and (3), with  $i - 1$  replacing  $i$ , we obtain  $Y_{i-1} + m D_{i-1} = p_{i-1}/c^*$ . Hence for  $i > 1$  we have  $s_i = 1/c^* (p_i - p_{i-1})/(p_i - m)$ .

It remains of course to determine  $c^*$ , the optimal competitive ratio attainable by algorithm `THREAT`. For any sequence of  $k$  exchange rate maxima it must be that the  $s_i$ s satisfy  $\sum_{1 \leq i \leq k} s_i \leq D_0 = 1$ . If the value of  $k$  is known to the trader, then the optimal choice of  $s_i$ s is such that no dollars will remain after the last transaction. That is, the optimal competitive ratio has the property that  $\sum_{1 \leq i \leq k} s_i = 1$ . Substituting into this equation the expressions determined for the  $s_i$ s, one can obtain

$$\begin{aligned} c^* &= c^*(k, m, p_1, p_2, \dots, p_k) \\ &= 1 + \frac{p_1 - m}{p_1} \cdot \sum_{i=2}^k \frac{p_i - p_{i-1}}{p_i - m}. \end{aligned}$$

Denote by  $c_n^*(m, M)$  the optimal competitive ratio for the  $n$ -day game. Thus,

$$\begin{aligned} c_n^*(m, M) &= \max_{\substack{k \leq n \\ m \leq p_1 < p_2 < \dots < p_k \leq M}} \\ & c^*(k, m, p_1, p_2, \dots, p_k). \end{aligned}$$

We skip this maximization routine and discuss the end result. An explicit expression for  $c_n^*(m, M)$  cannot be obtained but it can be shown that  $c_n^*(m, M)$  is the unique root  $c^*$  of the

**Table I.** Numerical Examples of Competitive Ratios for Some Search and One-Way Trading Algorithms (unknown duration)

Algorithm	Value of $\varphi$					
	1.5	2	4	8	16	32
RPP ( $m, M$ known)	1.22	1.41	2	2.82	4	5.65
LEVIN's (only $\varphi$ known)	1.5	2	2.66	3.42	4.26	5.16
THREAT (only $\varphi$ known)	1.27	1.50	2.11	2.80	3.53	4.28
THREAT ( $m, M$ known)	1.15	1.28	1.60	1.97	2.38	2.83

equation

$$c = n \cdot \left( 1 - \left( \frac{m(c-1)}{M-m} \right)^{1/n} \right). \quad (4)$$

Thus  $c_n^*(m, M)$  is the minimum competitive ratio attainable by algorithm THREAT for an  $n$ -day game. On the other hand, it can be shown that  $c_n^*(m, M)$  is a lower bound on the competitive ratio of any randomized algorithm against an oblivious adversary for an  $n$ -day trading game [El-Yaniv et al. 1992]. Hence  $c_n^*(m, M)$  is the exact competitive ratio for the trading problem of known duration and known bounds ( $m$  and  $M$ ). The fact that randomization cannot help in this problem is perhaps somewhat surprising but clear given that any deterministic trading algorithm that partitions the initial wealth and performs multiple trades is equivalent to a randomized one-way trading (and search) algorithm with the same performance.

The preceding solution implies a solution for one-way trading with unknown duration. The threat-based algorithm corresponding to the ratio  $c_\infty^*(m, M) = \lim_{n \rightarrow \infty} c_n^*(m, M)$  can handle any finite number of days while attaining a competitive ratio of  $c_\infty^*(m, M)$ . However, when the duration is not known (it is made known online on the last day) the adversary can choose an arbitrary large  $n$ , thus forcing a competitive ratio approaching  $c_\infty^*(m, M)$  (note that  $c_n^*(m, M)$  is strictly increasing with  $n$ ).  $c_\infty^*(m, M)$  can be shown to be the unique root  $c^*$  of

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

Clearly,  $c_\infty^*(m, M) = \Theta(\ln \varphi)$ .

The paper by El-Yaniv et al. [1992] studies two other variants of the one-way trading problem in which the trader knows only the global fluctuation  $\varphi = M/m$  but not the actual values of  $m$  and  $M$  (known and unknown duration). The value of  $c_n^*(\varphi)$ , the optimal competitive ratio of this trading game (known duration,  $n$ -day game), is determined and is shown to be

$$c_n^*(\varphi) = \varphi \left( 1 - (\varphi - 1) \left( \frac{\varphi - 1}{\varphi^{n/(n-1)} - 1} \right)^n \right). \quad (5)$$

Here again, this optimal performance is attainable by algorithm THREAT. Not surprisingly,  $c_n^*(\varphi)$  is monotone increasing with  $n$  and  $\varphi$  and the optimal competitive ratio of the unknown duration game  $c_\infty^*(\varphi)$  can be shown to be

$$c_\infty^*(\varphi) = \varphi - \frac{\varphi - 1}{\varphi^{1/(\varphi-1)}} = \Theta(\ln \varphi).$$

In order to get some feel for the actual competitive ratios obtained, it is interesting to observe some numerical examples. Table I summarizes the competitive ratio of the algorithms discussed in this section for some values of  $\varphi$ . We see that the optimal threat-based algorithm is always significantly superior to all other algorithms. Note that the deterministic algorithm RPP is superior to Levin's algorithm for small values of  $\varphi$ . Nevertheless, recall that the growth rate of the competitive ratio of Levin's algorithms is almost the logarithm of

the growth rate of the competitive ratio of the deterministic algorithm.

It is also interesting to consider the rate of increase of the optimal competitive ratio as a function of the number of trading days  $n$ . It is possible to show that this function grows very quickly to its asymptote. Nevertheless, there is still a slight advantage to playing short games. For instance, already at the 20th period  $c_n^*(1, 2)$  almost approaches its asymptote,  $c_\infty^*(1, 2) \approx 1.278$  (which is equivalent to guaranteeing 78.2% of the optimal offline return) at  $n = 10$ ; the ratio achieved is 1.26 (79.3% of OPT) and at  $n = 5$  the ratio is 1.24 (80.6% of OPT).

**3.4.2 Other Results.** El-Yaniv et al. [1992] study two other variants of the trading game: a continuous-time model and a model in which the adversary chooses a probability distribution that determines the exchange rates (this probability distribution is made known to the trader). We note that the continuous-time model is significantly simpler to analyze but its precise formulation is more involved.

## 4. REPLACEMENT PROBLEMS

### 4.1 A General Replacement Problem

At each time the online player must be engaged in a single activity. Associated with each activity  $a$  is its cost  $c(a)$  and its flow rate  $f(a)$  (which may also be a function of time  $f(a) = f(a, t)$ ). Throughout the time period in which the player is engaged in activity  $a$ , she pays money at the rate  $f(a)$ . From time to time new activities are offered as possible replacements to the current activity. If at some time  $t$  the player chooses to replace the current activity  $a$  with a new one  $a'$ , she pays a replacement (or changeover) cost of  $r_t(a, a')$ , where  $r_t$  is a real-valued function parameterized by time (for  $t = 0$  the replacement cost is  $r_t(a')$ , where  $a'$  is the initial activity chosen). For any sequence of activities  $a_0, a_1, \dots, a_k$  chosen at times  $t_0, t_1, \dots, t_k$ , the total cost

incurred by the player up to time  $T$  is

$$r_0(a_0) + \sum_{i=1}^k ((t_{i+1} - t_i)f(a_i) + r_{t_i}(a_{i-1}, a_i)),$$

where  $t_0 = 0$  and  $t_{k+1} = T$ .

We note that this general replacement model generalizes the Metrical Task System model of Borodin et al. [1992], which abstracts many online problems.

**4.1.1 Some Problem Variants.** This section discusses two special instances of the previous general replacement problem using the following notation. In the continuous-time (resp., discrete-time) variant of the problem, at each time  $t$  (resp., at the start of each period  $i$ ) the online player is presented with a finite set  $R_t$  (resp.,  $R_i$ ) containing some number of activities offered as a possible replacement to the current one.  $R_t$  is called the replacement set of time  $t$  (resp.,  $R_i$  is called the  $i$ th replacement set). We first consider a continuous-time finite-horizon model such that at each time there may be only a single replacement alternative to the current one; that is, for all  $t$ ,  $|R_t| = 1$ . Also in this simple variant all replacement costs equal a constant  $C$ ; that is, for all activities  $a, b$ ,  $r_t(a, b) = C$ . This problem variant is referred to as the elementary continuous-time finite-horizon replacement problem.

In the second variant we assume discrete time and that for all activities  $a, b$ ,  $r_t(a, b) = c(b)$ . Also we require that for all  $i$ ,  $R_i \subseteq R_{i+1}$ . That is, an activity that is once offered is always available thereafter (i.e., replacement options are permanent). This variant is referred to as the discrete-time replacement problem with multiple permanent replacement options. In contrast to the first variant, here we allow multiple replacement alternatives and varying replacement costs. We note that the assumption of permanent replacement options

together with the assumption of discrete time simplifies the problem. Indeed, despite the added complication that replacement costs may vary, this variant allows for somewhat easier analysis and much better competitive ratios. Finally, notice that the preceding replacement model also captures the leasing problem as a special case. Specifically, at each time there are optional activities: one with a small replacement cost that corresponds to renting, and another with a higher replacement cost that corresponds to buying. It is also given that the game ends as soon as the player chooses the activity that corresponds to buying.

#### 4.2 Applications of the Replacement Problem

The replacement problem has various interesting applications, in all of which the basic question is when to switch from one activity, investment, or facility, to another more rewarding one, when there is a cost associated with making the switch. Some striking examples of particular applications are the following.

**4.2.1 Equipment and Machine Replacement.** Here the player needs to use some piece of equipment throughout the time horizon. For its regular use, the equipment incurs some operating, production, and/or maintenance cost. From time to time, due to a priori unknown economic events, technological improvements, and/or equipment deterioration, the player can and may wish to switch to different or newer equipment that incurs a lower operating cost (or higher payoff). Some examples of equipment for which this application is relevant are cars, computers, industrial machinery, and the like. The same formulation applies of course to more abstract types of “equipment” such as jobs. In many of these examples the operating cost can be approximated by a fixed-rate-payment flow (e.g., gasoline consumption rate, salary, etc.). Other applications may require more elaborate modeling.

For instance, in order to model deterioration, the maintenance may be required to be some monotone nondecreasing function (rather than a constant).

**4.2.2 Supplier Replacement.** A firm is purchasing goods at a constant rate from one supplier. The cost of purchasing the same goods from other suppliers varies with time. The firm can switch to another supplier but at a certain cost. The cost of this switchover can be approximated by some constant that takes into account the paperwork, the wasted time, and possibly the costs involved in breaking the contract with the first supplier.

**4.2.3 The Menu-Cost Problem.** Many firms are constantly faced with the problem of when to adjust prices of the goods or services they offer. Due to inflationary markets and/or other economic events, the firm may wish to update its price menu to reflect the goods’ or services’ “real” values in order to increase its overall payoff. Each of these price adjustments, which correspond to our (flow) changeovers, incurs some fixed cost to physically update the “menu,” advertise, and so on.

**4.2.4 Mortgage Refinancing.** In this application the flow rate corresponds to the mortgage payment rate, which is based on a fixed interest rate (and the principal). Among the popular mortgages available in North American markets are those where refinancing costs are fixed (sometimes called zero-point mortgages). These fixed costs correspond to the paperwork and time overhead required by the switch to another mortgage and thus are best modeled by a fixed changeover cost. In other kinds of (fixed-rate) mortgages a changeover cost is some fixed percentage of the principal.

#### 4.3 On Some Bayesian Solutions for the Replacement Problem

The literature related to online replacement problems is quite extensive. The

typical assumption is that the flow rate function follows a particular (usually simple) stochastic process that may or may not be known to the online player. Let us describe two examples.

Derman [1963] studies a simplified discrete-time replacement problem where the analogue of our flow rate function is a piecewise constant function in which the next value is determined via a simple one-stage Markov process.

Sheshinski and Weiss [1993] study price-adjustment policies solving the menu-cost problem under the assumption that “real” prices are determined by the following two-state process. During each state the price level is changed at a fixed rate (in particular, they assume that in one state the price is fixed and in the second state the price increases at a fixed rate). The duration of each state is an independent exponentially distributed random variable. Note that in both these examples (and in most other analyses of this kind) the optimal policy is heavily dependent on the stochastic assumptions.

#### 4.4 Competitive Replacement Algorithms for the Elementary Continuous-Time Finite-Horizon Problem

In this section we discuss competitive solutions for the elementary replacement problem with continuous time and finite horizon [El-Yaniv and Karp 1997]. Recall that in this version of the replacement problem all changeover costs equal a constant  $C$ . Also in this variant there is only one possible replacement activity for the current one. Hence we denote the flow rate of the activity offered at time  $t$  by  $f(t)$ . The player starts with the initial activity paying at flow rate  $f(0)$  and may choose any number  $k$  of changeover times,  $0 < t_1 < t_2 < \dots < t_k < T$ . For each such changeover time  $t_i$ , the player pays a changeover cost  $C$  and throughout the interval  $[t_i, t_{i+1})$ ,  $i = 0, 1, \dots, k$ , his payment flow is at the rate  $f(t_i)$ . (By convention, take  $t_0 = 0$  and  $t_{k+1} = T$ .) For each particu-

lar choice of changeover times the total cost incurred by the player, composed of payment flows and changeover costs, is

$$kC + \sum_{i=0}^k (t_{i+1} - t_i) f(t_i).$$

Any choice of  $k$  and  $(k)$  changeover times is called a *replacement policy*. Of course we are interested in replacement policies that minimize the total cost. Given  $f(t)$  and  $C$ , it is straightforward to compute an optimal offline replacement policy and  $\text{OPT}(f)$ , the optimal offline cost, via (continuous) dynamic programming [Bellman 1955].

As in the search problem, we assume that the flow rate function is bounded such that for all  $t$ ,  $m \leq f(t) \leq M$ , where  $m, M \in \mathbb{R}$ , and  $0 \leq m < M$ . Furthermore, in this online variant of the replacement problem the player must determine the changeover times online without knowledge of future values of the flow rate function. Thus we assume that at each time  $t$  the player knows  $f$  only over the interval  $[0, t]$ . Let  $S$  be any online replacement policy and denote its total cost with respect to the flow  $f$  by  $S(f)$ .

*Problem reduction.* By suitably scaling the time and cost axes, we may assume that  $C = T = 1$ . Specifically, given an initial problem setup with parameters  $m', M', T'$ , and  $C'$ , we map each flow rate  $x' \in [m', M']$  to  $x = x'T'/C'$  and each time  $t' \in [0, T']$  to  $t/T'$ . Thus the new problem setup is given by  $M = M'T'/C'$ ,  $m = m'T'/C'$ , and  $T = C = 1$ . It is not hard to see that this scaling preserves the competitive ratio. After scaling, we further assume that  $m + 1 < M$ . For  $M \leq m + 1$  the problem is trivial in the sense that the online player can always achieve a “perfect” competitive ratio of 1. For the rest of this section we consider the reduced replacement problem with  $C = T = 1$  so that the only relevant parameters are  $m$  and  $M$ .

4.4.1 *Some Types of Online Replacement Policies.* Perhaps the most naive online replacement policies that are still interesting are the following class of *time-independent* policies. A policy in this class is a sequence of constant changeover thresholds that are fixed over time independent of the flow rate function. Specifically, a time-independent policy is a decreasing sequence of real numbers,

$$M \geq M_1 > M_2 > \dots > M_k \geq m.$$

The interpretation is that the online player changes over for the  $i$ th time when the flow rate decreases to the level of (or below)  $M_i$ .

A more sophisticated class of policies is the following class of *time-dependent* or *refusal* policies (we use both terms). A refusal policy is defined as a sequence  $\{M_i(t)\}_{i=1}^k$  of functions such that  $M_i(t) : [0, 1] \rightarrow [m, M] \cup \{-1\}$ . Each of these functions is nonincreasing and for all  $i$  and  $t$ ,  $M_i(t) > M_{i+1}(t)$ . Here again the interpretation is that the online player changes over for the  $i$ th time at the first instance  $t$  when  $f(t) \leq M_i(t)$  but refuses to change over as long as  $f(t) > M_i(t)$ . A particular subclass of simple refusal strategies is the one where each  $M_i$  is a constant except for one step at some time  $b_i$  from which the function remains at  $-1$ ; that is,

$$M_i(t) = \begin{cases} M_i & \text{if } t \leq b_i; \\ -1 & \text{otherwise.} \end{cases}$$

Call such a strategy a *constant-threshold* refusal policy. Any such strategy is thus specified by the two sequences  $\{M_i\}$  and  $\{b_i\}$ . Notice that a time-independent policy is a rudimentary form of a constant threshold refusal policy where the  $b_i$ s are all 1. Here we focus on time-independent and constant threshold refusal policies. Nevertheless, note that more sophisticated strategies would make use of the history of flow rates. Somewhat surprisingly, it turns out that in all instances of the preceding replacement problem it is possible to

obtain optimal or approximately optimal online performance using only time-independent and constant threshold refusal policies (without resorting to history-dependent policies).

4.4.2 *A Lower Bound.* A lower bound on the competitive ratio of any deterministic policy for this variant of the replacement problem is obtained in El-Yaniv and Karp [1997]. We now sketch the essential ideas of this bound. Consider an adversary that can choose the flow rate functions from a restricted family consisting of functions that start at time zero at the rate  $M$ , then drop “instantaneously” (i.e., during an infinitesimally short time) and continuously to some rate  $\mu$  chosen by the adversary, and then “jump” back to the maximum possible rate  $M$  and remain there. We may assume that  $\mu \leq M - 1$ , since no sensible strategy will change over for any flow rate larger than  $M - 1$  (the changeover cost is 1). It can be shown that against such functions the optimal online performance can be attained by a time-independent policy. Intuitively this is due to the fact that all replacement opportunities occur during an infinitesimally short time period.

For any choice of  $\mu$  the optimal offline cost is  $\mu + 1$ , since the optimal offline algorithm changes over to the rate  $\mu$  paying 1 for the changeover penalty. Let  $S = \{M_i\}_{i=1}^k$  be a time-independent policy and assume that it is  $r$ -competitive. It can be shown that  $S$  must satisfy the relations:

$$M_i + i = r(M_{i+1} + 1), \\ i = 0, 1, \dots, k \quad (6)$$

(by convention we set  $M_0 = M$ ,  $M_{k+1} = m$ ). Intuitively, the reason is that if an adversary wishes  $\mu$  to lie in some interval  $[M_{i+1}, M_i]$ , it will pick  $\mu = M_{i+1} + \varepsilon$  (for some small  $\varepsilon$ ), in which case the optimal offline cost will be (arbitrarily close) to  $M_{i+1} + 1$  and the online cost,  $M_i + i$  (the online strategy will change over  $i$  times). On the other hand, the strategy defined by the recurrence rela-

tion (6) is  $r$ -competitive provided that  $M_{k+1} = m$ . From properties of this recurrence relation, it can be shown that if  $S$  is  $r$ -competitive, then the number of thresholds in  $S$  is

$$k = \lceil (m + 1)(r - 1) \rceil. \quad (7)$$

A closed-form expression for the recurrence relation (6) is

$$M_i = \left( a + \frac{r^2}{(r-1)^2} \right) r^{-i} + \frac{i}{r-1} - \frac{r^2}{(r-1)^2}. \quad (8)$$

The optimal competitive ratio  $r^*$  can now be determined using (7) and (8). It is the minimum  $r$  that solves the equation  $M_{k+1} = m$ . It can be shown that for a fixed  $m$ ,  $r^* = \Theta(\ln M / \ln \ln M)$ . Since  $r^*$  is the optimal competitive ratio with respect to restricted flow-rate functions, it is a lower bound on the competitive ratio of any replacement policy against unrestricted flow-rate functions.

**4.4.3 A Characterization of Competitive Refusal Policies.** El-Yaniv and Karp [1997] establish a characterization theorem for constant threshold refusal policies that have a decreasing refusal time sequence  $\{b_j\}$  (i.e., if the policy refuses to change over for the first time, it will refuse to change over for the rest of the game). Given any refusal policy  $S$  (of the preceding type) with  $k$  changeover thresholds and a real number  $r > 1$ , the theorem specifies the following conditions, C1 and C2, such that they are both satisfied if and only if  $S$  is  $r$ -competitive.

**C1** for all  $i$  and  $j$  with  $0 \leq i \leq j \leq k$ ,

$$M_i b_{j+1} + M_j(1 - b_{j+1}) + j \leq r \cdot \min$$

$$\left[ \begin{array}{l} M_0, \\ M_{i+1} + 1, \\ M_0 b_{j+1} + m(1 - b_{j+1}) + 1, \\ M_{i+1} b_{j+1} + m(1 - b_{j+1}) + 2 \end{array} \right];$$

**C2** for all  $i$  and  $j$  with  $0 \leq i < j \leq k$ ,

$$M_i b_j + M_j(1 - b_j) + j \leq r \cdot \min$$

$$\left[ \begin{array}{l} M_0, \\ M_{i+1} + 1, \\ M_0 b_j + m(1 - b_j) + 1, \\ M_{i+1} b_j + m(1 - b_j) + 2 \end{array} \right].$$

The proof of this theorem is obtained by bounding from below the optimal offline cost via a linear form of variables chosen by the adversary (these variables determine the flow function). It follows that the set of feasible choices for the adversary is a polytope. By a convexity argument, it is then sufficient to consider only corner points of this polytope. In each corner point most of the coordinates (i.e., variables) are zero. Then with respect to the corner points we obtain simple expressions for the offline costs (as they appear in the right-hand side of C1 and C2); in particular, each possible cost includes at most two changeovers.

**4.4.4 Upper Bounds.** Using the preceding characterization theorem it is possible to obtain various interesting upper bounds for this replacement problem. We now describe a policy that achieves a competitive ratio that for sufficiently large  $M$  is within a constant factor of  $r^*$  for every positive  $m$ .

Consider the constant threshold time-independent policy  $\{M_i\}_{i=1}^k$ , where the sequence of changeover thresholds  $\{M_i\}$  is defined by the following recurrence relation. For each  $\rho > 1$ , set  $k = \lfloor \rho \rfloor$ . Define

$$\begin{cases} M_0 = M; \\ M_{i+1} = \frac{M_i + k}{\rho} - 1, & \text{integer } i \geq 1 \end{cases} \quad (9)$$

It is possible to show that for every positive  $m$  and for sufficiently large  $\rho$ , the sequence defined by (9) decreases below  $m$  within  $k$  steps. Call a  $\rho$  for which  $M_{k+1} \leq m$  and  $M_k > m$  good. For each  $\rho$ , each  $m > 0$ , and each  $M > m + 1$ ,

let  $S_\rho^*(m, M)$  denote the policy  $\{M_i\}$  (as defined by (9)). Now, by considering  $S_\rho^*(m, M)$  as a (degenerate) time-dependent policy, we can apply the previous characterization conditions and prove that  $S_\rho^*(m, M)$  is  $\rho$ -competitive for all good  $\rho$  and almost all values of  $M$ . Specifically, assume that  $M > \lfloor \rho \rfloor / (\rho - 1) = k / (\rho - 1)$ . This implies that  $M_1 + 1 - M \leq 0$ , which means that  $\min\{M_i + 1, M\} = M_i + 1$ . We use this fact later. Let us now specialize the conditions C1 and C2 of the characterization result to the case where  $b_{k+1} = 0$  and  $b_i = 1$ ,  $i = 1, 2, \dots, k$ , that is, when the (degenerate) refusal policy is a time-independent policy. For  $r = \rho$ , C1 and C2 reduce to the condition: for all  $0 \leq i \leq j \leq k$ ,

$$M_i + j \leq \rho \cdot \min\{M_0, M_{i+1} + 1\}.$$

But this condition readily holds by the definition of the  $M_i$ s and the fact that  $M_1 + 1 \leq M$ . For each (sufficiently large)  $M$ , define  $\rho(M)$  to be the minimum (infimum) good  $\rho$ . Then it is possible to show that for a fixed  $m$ ,  $\rho(M) = \Theta(\ln M / \ln \ln M)$ , which has the same asymptotic growth as  $r^*$ , the lower bound for the problem.

El-Yaniv and Karp [1997] also consider constant threshold refusal policy and prove its (strict) optimality whenever  $\sqrt{M/(m+1)} \leq (m+2)/(m+1)$  or  $m = 0$ . The proof relies on the preceding characterization result but is much more involved.

#### 4.5 Discrete-Time Replacement with Multiple Permanent Replacement Options

In this section we only state the results of a recent paper by Azar et al. [1996] that studies the discrete time replacement problem variant with multiple permanent replacement options. Recall that in this problem variant we require that  $|R_i| > 0$  and that for all  $i$ ,  $R_i \subseteq R_{i+1}$  (see the notation of Section 4.1). That is, an activity that is once offered is always available thereafter.

**4.5.1 The Convex Problem Variant.** An instance of this replacement problem is called *convex* if for each  $i$  and  $b_1, b_2 \in R_i$  if  $f(b_1) < f(b_2)$ , then  $r(b_1) \geq r(b_2)$ . For the convex variant Azar et al. [1996] obtain a simple 7-competitive algorithm.

**4.5.2 The Nonconvex Variant.** An instance of this problem, which is not convex in the previous sense, is called nonconvex. Nonconvex instances turn out to be markedly harder. Azar et al. [1996] introduce an algorithm that attains the competitive ratio:

$$O(\min\{\log(cr_{\max}), \log \log(cf_{\max}), \log(cn_{\max})\}),$$

with  $r_{\max}$  being the ratio between the maximum and minimum replacement costs,  $f_{\max}$  the ratio between the maximum and minimum flow rates,  $n_{\max}$  the total number of replacement alternatives presented to the player throughout the game, and  $c$  some constant. Azar et al. also present a nemesis request sequence of replacement sets that forces the following competitive ratio on every online algorithm.

$$\Omega\left(\min\left\{\log(cr_{\max}), \frac{\log \log(cf_{\max})}{\log \log \log(cf_{\max})}, \frac{\log(cn_{\max})}{\log \log(cn_{\max})}\right\}\right),$$

where  $c$  is some constant. Thus, as a function of  $r_{\max}$ , their algorithm attains a competitive ratio that is within a constant factor of the best possible.

## 5. TWO-WAY TRADING AND PORTFOLIO SELECTION

### 5.1 The Elementary Portfolio Selection Problem

Consider a market of  $m$  securities: these can be stocks, bonds, foreign currencies, or commodities. Let  $\vec{p}_i = (p_{i1}, p_{i2}, \dots, p_{im})$  denote a vector of prices where for each  $j = 1, 2, \dots, m$ ,  $p_{ij}$  denotes the

number of units of the  $j$ th security that can be bought for one dollar at the start of the  $i$ th trading period,  $i = 1, 2, \dots$ . The “local” currency, say, dollars, may or may not be one of the  $m$  securities. This local currency is referred to as *cash*. The change in security prices during the  $i$ th period is represented as a vector  $\tilde{x}_i = (x_{i1}, x_{i2}, \dots, x_{im})$  where for each  $i$  and  $j$ ,  $x_{ij} = p_{ij}/p_{(i+1)j}$ . The quantity  $x_{ij}$  is called the *price relative* of security  $j$  (of the  $i$ th period). Thus an investment of  $d$  dollars in the  $j$ th security just before the start of the  $i$ th period yields  $dx_{ij}$  dollars by the end of the  $i$ th trading period.

An investment in the market, or *portfolio*, is specified as the proportion of dollar wealth currently invested in each of the  $m$  securities. Specifically, we represent a portfolio as a probability distribution  $\vec{b} = (b_1, b_2, \dots, b_m)$ , where  $b_i \geq 0$  and  $\sum_i b_i = 1$ . Consider a portfolio  $\vec{b}_1$  invested just before the first period. By the start of the second period this portfolio yields

$$\vec{b}_1^t \cdot \tilde{x}_1 = \sum_{j=1}^m b_{1j} x_{1j}$$

dollars per each initial dollar invested. At this stage the investment can be cashed in and adjusted, say, by reinvesting the entire current wealth in some other proportion  $\vec{b}_2$ , and so on. Assuming an initial wealth of \$1, the compounded return of a sequence of portfolios,  $B = \vec{b}_1, \vec{b}_2, \dots, \vec{b}_n$  with respect to a sequence of market price relatives  $X = \tilde{x}_1, \dots, \tilde{x}_n$  is defined as

$$R(B, X) = \prod_{i=1}^n \vec{b}_i^t \cdot \tilde{x}_i = \prod_{i=1}^n \sum_{j=1}^m b_{ij} x_{ij}.$$

A *portfolio selection algorithm* is any sequence of portfolios specifying how to reinvest the current wealth from period to period. The compounded return of a portfolio selection algorithm ALG with respect to a market sequence  $X$  is denoted by  $\text{ALG}(X) = R(\text{ALG}, X)$ . The port-

folio selection problem revolves around the question of identifying and analyzing portfolio selection algorithms. Of course, here we are mainly concerned with online portfolio selection algorithms.

It is important to notice that the preceding basic portfolio selection problem is only a crude approximation of the corresponding real-life problem. For practical purposes, perhaps the most important factor missing in this model is transaction costs (or alternatively, bid-ask spreads). (Note that in Section 5.8.7 we add this feature into the portfolio selection model.) Another simplification of this model is to assume that money and units of securities are arbitrarily divisible. Other important missing factors are taxes and interest rates that are typically important issues in investment planning. Finally, this model does not allow many investment opportunities that exist in a modern market, starting with short selling and ending with myriad derivative instruments such as futures, options, and the like. Nevertheless, this model is rich enough in itself and makes a reasonable foundation for studying some of the essential questions related to portfolio selection.

**5.1.1 The Two-Way Trading Problem.** The special problem instance with  $m = 2$ , where one of the two securities is cash, has received special attention and turns out to be sufficiently interesting in itself. We call this particular instance the *two-way trading problem*. Note that in the two-way trading problem we assume that the prices and price relatives of the cash are always 1. In the special case of the two-way trading problem we sometimes refer to portfolio selection algorithms simply as (*two-way*) *trading algorithms*.

## 5.2 Buy-and-Hold Versus Market Timing

Financial agents study and use a large variety of portfolio selection strategies, including the “slow” and almost static

strategies typically used by mutual fund managers that select and buy some portfolio and hold it for quite a long time. Such strategies rely on the natural tendency of securities to increase their value due to natural economic forces. For instance, stocks pay dividends and increase their prices as the underlying firms succeed in their businesses. Such “slow” strategies are generally called *buy-and-hold* (BAH). Thus the essence of devising a BAH strategy is the particular selection of assets to hold and investors attempt to diversify their holdings in order to reduce the risk (variance) of their portfolios.<sup>7</sup>

In contrast, some financial agents use more aggressive strategies that buy and sell securities more frequently, sometimes even many times during one day. Such strategies mainly attempt to take advantage of securities’ price fluctuations and are called *market-timing* strategies. Of course, in the long run every buy-and-hold strategy is also a market-timing strategy. After all, sooner or later owners of securities want to realize the monetary value of the assets they hold. Hence, whether a strategy is buy-and-hold or market timing is relative to the time horizon considered.

The fact that market-timing strategies have the potential for enormous returns is not surprising. Consider the following illustration due to Shilling [1992]. A \$1 portfolio invested in the Dow Jones Industrial Average in January 1946 was worth \$116 at the end of 1991 (including reinvestment of dividends but excluding tax deductions). This is equivalent to 11.2% compound annual gain. A market-timing strategy that was lucky enough to be out of the market during the 50 weakest months in that 552-month period but otherwise was fully invested using the same fixed portfolio would return \$2,541, or a 19% annual gain. Furthermore, an offline

strategy which during the 50 weakest months was in a short position would have returned \$44,967, or a 26.9% annual gain. The reader is referred to the comprehensive survey by Merton [1981] for classical theories and ideas regarding buy-and-hold and market-timing strategies.

**5.2.1 The Constant Rebalanced Algorithm.** An algorithm that is often used in practice and is considered in this survey is the *constant rebalanced* algorithm that invests in a fixed portfolio  $\vec{b} = (b_1, b_2, \dots, b_m)$  at the start of each trading period. This algorithm, denoted  $\text{CBAL}_{\vec{b}}$ , is clearly a market-timing strategy. The performance of the optimal offline CBAL is always at least as good as that of the optimal offline BAH but usually it is significantly better (see, e.g., the examples in Cover [1991]). The reason is that the optimal offline BAH only performs as does the best security in the market but the optimal offline CBAL also takes advantage of fluctuations in the market giving rise to exponential returns. Of course, CBAL makes as many as  $m$  transactions at the start of each trading period to adjust its portfolio, whereas BAH performs two transactions during the entire trading period. Although in the models considered here this large number of transactions is not a consideration, it definitely becomes significant when transaction costs are introduced.

### 5.3 Online Portfolio Selection

Let ALG be an online portfolio-selection algorithm. Recall that  $\text{ALG}(X)$  denotes the compounded return of ALG with respect to the market sequence  $X = \vec{x}_1, \dots, \vec{x}_n$  of price-relative vectors and starting with an initial wealth of \$1. The competitive ratio of ALG is

$$\sup_X \frac{\text{OPT}(X)}{\text{ALG}(X)},$$

where OPT is an optimal offline portfolio selection algorithm. By considering the

<sup>7</sup> The reader is referred to Bodie et al. [1993] for a thorough treatment of this topic, as well as other issues in elementary mathematical finance.

simpler one-way trading problem (see Section 3.4), the paper by El-Yaniv et al. [1992] obtains a lower bound of  $c_2^*(\varphi)^{n/2}$  for portfolio selection ( $m = 2$ ) where the constant  $c_2^*(\varphi) > 1$  is the optimal bound for a one-way two-day trading game in which the player knows the global fluctuation ratio  $\varphi$  (see Equation (5)). This simple lower bound is essentially obtained via a decomposition of the two-way trading problem to a sequence of one-way trading games, using the known lower bound for one-way trading games. Note that El-Yaniv et al. [1992] present an upper bound of  $c_\infty^*(\varphi)^n$  for the case  $m = 2$ , again based on a straightforward decomposition of the two-way trading game into a sequence of one-way games.

Thus, for the online portfolio selection problem, one cannot hope for competitive ratios that are subexponential in  $n$ , the number of trading periods. Also, to obtain bounded ratios one must assume lower and upper bounds on the securities' prices (or other equivalent constraints). Nevertheless, one should keep in mind that for typical market sequences the optimal offline algorithm accrues astronomical returns (which are also typically exponential in  $n$ ), so the preceding lower bound does not exclude possibilities of large returns to the online player. On the other hand, notice that competitive online strategies can lose money (i.e., end the game with fewer dollars than the initial wealth), whereas the optimal offline strategy will never lose.

### 5.4 Other Performance Measures

Worst-case studies of online portfolio selection strategies consider performance measures other than the competitive ratio. These performance measures differ from the standard competitive ratio optimization in (some of) these ways:

—considering a restricted kind of offline benchmark algorithm, for example, some kind of a “static” offline algorithm (such as the optimal offline

CBAL) instead of the standard, unrestricted optimal offline algorithm;

—imposing more constraints on the adversary; for example, considering adversaries that must choose their worst-case market sequence while conforming to some additional (statistical) parameters. Such adversaries are called *statistical adversaries* [Raghavan 1992]<sup>8</sup>;

—using decision criteria other than the competitive ratio, for example, the maximin or the minimax regret (definitions follow);

—instead of measuring the compounded return of the strategy (online or offline), measuring returns via some utility functions. For example, one popular measure is the *exponential growth rate* defined (for an algorithm ALG and a market sequence  $X$ ) as  $1/n \cdot \log(\text{ALG}(X))$  where  $n = |X|$ .

Let us now formally define other measures that are discussed later. As usual, ALG denotes an online algorithm and OPT denotes an offline algorithm. Let  $U$  denote any utility function. With respect to ALG, OPT, and  $U$  (and assuming profit maximization) we now distinguish among three adversaries. First, a *competitive-ratio* adversary chooses a market sequence  $X^*$  such that

$$X^* = \arg \max_X U(\text{OPT}(X))/U(\text{ALG}(X)).$$

The *maximin* adversary chooses a market sequence  $X^*$  such that

$$X^* = \arg \min_X U(\text{ALG}(X)).$$

The *minimax-regret* adversary chooses  $X^*$  with

$$X^* = \arg \max_X \{U(\text{OPT}(X)) - U(\text{ALG}(X))\}.$$

<sup>8</sup> Notice that most of the adversaries considered so far are “minimal” forms of statistical adversaries (e.g., some of the adversaries assumed in the one-way trading problem must conform to lower and upper bounds on prices).

Applying additional restrictions on the adversary simply means that we constrain the preceding maximizations (minimizations) with respect to these additional restrictions that specify what feasible market sequences are. Notice that the use of the exponential growth-rate utility in conjunction with the minimax-regret adversary is equivalent to using the competitive-ratio adversary with the identity utility function.

We note that most of the preceding variations apply in general to any on-line problem. There are various intuitive motivations for the use of any of these variations. For instance, the use of restricted statistical adversaries is motivated by the desire to avoid considering unrealistic adverse market sequences that are very unlikely to occur in practice. The reader is referred to El-Yaniv [1996], where the rationality of the competitive ratio as well as a comparison to some of the other performance measures is studied. Nevertheless, we note that the practical utility of all these worst-case performance measures is not yet well understood and should be supported by means of experimental studies.

## 5.5 Results for the Two-Way Trading Problem

Here we focus on several results for the two-way trading problem, that is, the portfolio-selection problem with  $m = 2$  securities such that one of them is cash. In this special case, since the cash prices and price relatives are always 1, a market sequence is specified by a sequence of the security prices  $p_1, p_2, \dots$  (or price relatives).

**5.5.1 Games Against Mean-Variance Statistical Adversaries.** Raghavan [1992] considers a market in which the security price sequence  $p_1, p_2, \dots$  maintains known mean of 1 and standard deviation  $\sigma \in [0, 1]$ . He also imposes the restrictions that (i) prices are drawn from the interval  $[1 - c, 1 + c]$  where  $c \leq 1$  is some constant and (ii)  $p_1 =$

$p_n = 1$ , where  $n$  is the number of trading periods. With respect to a (statistical) maximin adversary conforming to the preceding restrictions he obtains the following results (proofs of the stated results are not given). First it is stated that any deterministic online algorithm cannot return more than  $1 + \sigma$ . He then considers two strategies: CBAL and the following averaging strategy, denoted AVE. Assuming  $n$  trading periods, AVE invests  $1/n$  dollars in the security at the start of each trading period. The following bounds hold for AVE for each feasible market sequence  $X = p_1, p_2, \dots, p_n$ ,

$$1 + \sigma^2(1 - c) \leq \text{AVE}(X) \leq 1 + \sigma^2.$$

For  $\text{CBAL}_b$  that maintains a portfolio  $\vec{b} = (\frac{1}{2}, \frac{1}{2})$ , the following bounds are stated,

$$1 + \sigma^3/3 \leq \text{CBAL}(X) \leq 1 + \sigma^3/2.$$

It is stated that the  $1 + \sigma^3/2$  upper bound holds for any constant rebalanced strategy (under the same assumptions). The bounds for algorithms AVE and CBAL can be obtained as follows. Consider a price sequence  $X = p_1, p_2, \dots, p_n$ . Any dollar investment of AVE at the  $i$ th trading period returns  $1/p_i$  dollars (since  $p_n = 1$ ). Since AVE invests  $1/n$  dollars in each period, the total return is  $1/n \sum_i 1/p_i$ . Hence, bounds on AVE's total return can be obtained by maximizing this expression over all price vectors satisfying the mean-variance (standard deviation) constraints. Similarly, obtaining the bounds for CBAL can be reduced to such constrained optimization. Note that the return of CBAL for the market sequence  $X$  is

$$\text{CBAL}(X) = \frac{1}{2^{n-1}} \prod_{i=1}^{n-1} \left( \frac{p_i + p_{i+1}}{p_{i+1}} \right).$$

## 5.6 Other Statistical Adversaries and "Money-Making" Algorithms

**5.6.1 Money-Making Algorithms.** Define the *profit* of a trading strategy as its final compounded return minus its

initial wealth (a negative profit is called *loss*). The realization that any competitive trading algorithm may end the trading game with a loss motivated Chou et al. [1995a] to study models with additional constraints that guarantee positive profit for the online player with respect to market sequences for which the optimal offline algorithm accrues positive profits. An online trading algorithm that satisfies this property is called *money-making*. The models considered assume several kinds of statistical adversaries that must conform to prespecified constraints. Not surprisingly, in order to achieve the money-making property the assumptions on market sequences must be quite strong. Nevertheless, considering such models leads to quite interesting results. For the rest of this section we consider finite market sequences prescribed by a sequence of security prices  $p_1, p_2, \dots, p_n$ , or alternatively by a sequence of the price relatives  $x_1, x_2, \dots, x_{n-1}$ , where  $x_i = p_i/p_{i+1}$ ,  $i = 1, 2, \dots, n - 1$ . When referring to a market sequence  $X$  we take the most convenient of these representations.

**5.6.2 The  $(n, \phi)$ -Adversary.** Fix some  $n \geq 2$ . Assume that each feasible price sequence is of length  $n$  and impose the restriction that the optimal offline return associated with a feasible sequence is at least  $\phi$  (clearly  $\phi \geq 1$ ). The underlying assumption here is that true  $(n, \phi)$  pairs exist in relevant real price sequences in the sense that such pairs can be statistically estimated from past markets with a reasonable degree of confidence. An adversary that is constrained to generate only such feasible sequences is called the  $(n, \phi)$ -adversary. Let  $X = x_1, x_2, \dots, x_{n-1}$  be a feasible sequence of price relatives. It is not hard to see that the optimal offline return  $\phi$  is given by

$$\phi = \prod_{1 \leq i \leq n-1} \max\{1, x_i\}. \quad (10)$$

It follows that, for any such sequence of prices (price relatives), an online player

knowing  $\phi$  and  $n$  at the start of the game can determine at the start of the  $(i + 1)$ st period, just after the  $(i + 1)$ st price is revealed ( $i = 1, 2, \dots$ ), what the optimal offline return  $\phi_{n-1}$  would be for a new  $(n - i)$ -period game starting in that period, that is, if the optimal offline algorithm were to start in that period a “new game” (with initial wealth of 1) with respect to the suffix of the original market sequence,  $x_{i+1}, \dots, x_n$ . Specifically, using (10) we have  $\phi_{n-1} = \min\{\phi, \phi/x_1\}$  and

$$\phi_{n-j-1} = \min\{\phi_{n-j}, \phi_{n-j}/x_{j+1}\}.$$

Hence, against the  $(n, \phi)$ -adversary, the online player can track the “state” (i.e., current wealth accrued) of the optimal offline algorithm with a delay of one day. This property allows for a dynamic programming derivation of the optimal online algorithm. Denote by  $R_n(\phi)$  the return of the optimal online algorithm  $S^*$  (which knows the parameters  $n$  and  $\phi$ ). It is not hard to obtain the following recurrence relation for  $R_n$ .

$$R_n(\phi) = \sup_{0 \leq b \leq 1} \inf_{x \leq \phi} \{(bx + 1 - b)R_{n-1}(\phi_{n-1})\}; \quad (11)$$

$$R_2(\phi) = \phi.$$

Notice that algorithm  $S^*$  attempts to choose its best investment  $b$  against the worst possible price relative  $x$ . The wealth obtained from the investment  $bx$  plus the remaining cash  $1 - b$  are then reinvested optimally with respect to an  $(n - 1)$ -period game in which the optimal offline return is  $\phi_{n-1}$ , and so on.

A closed form for  $R_n$  is probably beyond reach, but it is not hard to prove by induction on  $n$  that  $S^*$  is money-making. However, it is not difficult to obtain the following upper bound on  $R_n(\phi)$  for any  $\phi > 1$  and  $n \geq 2$ .

$$R_n(\phi) \leq \frac{1}{1 - (1 - 1/\phi)^{n-1}}. \quad (12)$$

This bound can be obtained by considering the following restricted version of

the  $(n, \phi)$ -adversary. In each period this adversary has two options: either to decrease the price by a factor of  $\phi$  or to increase the price by a very large factor so that the dollar value of the previous investment becomes negligible. Once the adversary chooses the first option, there will be no further downward fluctuations since the optimal offline of  $\phi$  has been realized. Hence, if this is the case, the game is over. The optimal online return  $R'_n(\phi)$  for this restricted game is

$$R'_n(\phi) = \max_{0 \leq b \leq 1} \min\{b\phi + 1 - b, (1 - b)R'_{n-1}(\phi)\},$$

and its closed form is the right-hand side of (12), which clearly bounds above  $R_n(\phi)$ .

Using the approximation  $(1 - 1/\phi)^{n-1} = (1 - 1/\phi)^{\phi^{(n-1)/\phi}} \approx e^{-(n-1)/\phi}$ , for large  $\phi$  we have

$$R'_n(\phi) \approx 1/(1 - e^{-(n-1)/\phi}),$$

and then the relations are obtained:

- if  $\phi = \omega(n)$ , then  $e^{-(n-1)/\phi} \approx 1 - (n - 1)/\phi$  and  $R'_n(\phi) \approx \phi/(n - 1)$ ;
- if  $\phi = \Theta(n)$ , then  $R'_n(\phi) \approx 1/(1 - e^{-c})$ , where  $c$  is some positive constant;
- if  $\phi = o(n)$ , then  $R'_n(\phi)$  approaches 1 as  $n \rightarrow \infty$ .

Hence, although  $S^*$  is “money-making,” the optimal online return  $R_n(\phi)$  against the  $(n, \phi)$ -adversary can be a minuscule fraction of  $\phi$ . In terms of competitiveness, the competitive ratio of  $S^*$  is not smaller than  $\max\{n - 1, \phi\}$  and no upper bound was determined.

**5.6.3 The General “Money-Making” Scheme Against Statistical Adversaries.** The preceding derivation of an optimal online algorithm against the  $(n, \phi)$ -adversary gives rise to a general scheme for obtaining money-making optimal online algorithms with respect to any statistical adversary that is at least as restricted as the  $(n, \phi)$ -adversary. Spe-

cifically, for any collection of constraints  $C$  that subsume the  $(n, \phi)$  constraint, a relation similar to (11) obtains the corresponding optimal online algorithm when the constraints in  $C$  are included appropriately in the recurrence.

**5.6.4 Weaker Adversaries.** Against the  $(n, \phi)$ -adversary the online player is forced to invest very small amounts in most trading periods, since the adversary can depreciate the value of most investments by increasing the price arbitrarily. Theoretically, such threats can be made until the second to last period. Such market sequences are of course unrealistic. By imposing additional constraints it is possible to reduce such threats. The following constraints can substantially reduce these unrealistic threats: (i) upper bounds on price relatives; (ii) minimum and maximum bounds on prices; (ii) upper bounds on the length of runs of monotone increasing (decreasing) prices; and (iii) other statistical parameters such as mean and higher moments of the empirical distribution observed. Note that the  $(n, \phi)$  constraint is “minimal” in the sense that when one of these parameters is relinquished the money-making property cannot be achieved.

## 5.7 The Fixed-Fluctuation Model

In the *fixed-fluctuation model* all price relatives  $x_i$  are in  $\{\alpha, \alpha^{-1}\}$  where  $\alpha > 1$ . Assuming w.l.o.g. that the initial price is 1, it follows that all prices are in  $\{\alpha^j : j \text{ integer}\}$ . We now add the restriction that each feasible sequence of price relatives is of length  $n$  and the number of downward (i.e., profitable)  $\alpha^{-1}$ -fluctuations is exactly  $k$ ,  $0 \leq k \leq n$ . Hence the number of upward  $\alpha$ -fluctuations is  $n - k$ . Thus the “type” of each market sequence is specified by the number  $k$ . Call the adversary that produces such feasible sequences the  $(\alpha, n, p)$ -adversary. Clearly this  $(\alpha, n, p)$ -constraint subsumes the preceding  $(n, \phi)$ -constraint since the implied optimal offline return for each feasible market se-

quence is exactly  $\phi = \alpha^k$ . Hence it is possible to obtain the optimal online trading algorithm (against the  $(\alpha, n, p)$ -adversary) using the preceding “money-making” scheme. Chou et al. [1995a] study this optimal online algorithm and characterize some of its properties.

**5.7.1 On the Fixed-Fluctuation Model and Time Scaling.** Before we continue sketching the results of Chou et al. [1995a], let us consider the practical relevancy of the fixed fluctuation model. Clearly “daily” price relatives are variable. Hence, to approximate the fixed-fluctuation model one can use a *time-scaling* approach where each trading period is of variable length such that the  $(i + 1)$ st price “tick” occurs at the first time instance after the  $i$ th price tick when a price that approximates  $p_i\alpha$  or  $p_i/\alpha$  is encountered (with  $p_i$  denoting the price at the  $i$ th tick).<sup>9</sup> One advantage of this fixed-fluctuation model is that the player may choose a suitable  $\alpha$  to filter out “noisy” fluctuations (i.e., very small fluctuations that should be avoided when transaction costs are taken into consideration). Of course, the choice of the fluctuation parameter  $\alpha$  should be correlated to the choice of the market type parameter  $k$ , which in general should be based on a rough prediction of the market trend within the horizon  $n$ .

**5.7.2 The Optimal Strategy Against the  $(\alpha, n, p)$ -Adversary.** Denote by  $R(n, k) = R(\alpha, n, k)$  the return of the optimal online algorithm  $S^{**}$  against the  $(\alpha, n, k)$ -adversary. Using the preceding scheme, a recurrence relation for  $R(n, k)$  is easily obtained:

$$R(n, k) = \max_{0 \leq b \leq 1} \min \left\{ \begin{array}{l} (\alpha^b + 1 - b)R(n - 1, k - 1), \\ (b\alpha^{-1} + 1 - b)R(n - 1, k) \end{array} \right\}$$

<sup>9</sup> Of course, to allow such time scaling  $\alpha$  must be chosen in accordance with the special properties of the market in question.

$$\begin{aligned} R(n, 0) &= 1 \\ R(n, n) &= \alpha^n. \end{aligned} \quad (13)$$

Here the  $b$  that minimizes (13) is the first investment made by algorithm  $S^{**}$ . Since the upper operand in the min operator in (13) is decreasing with  $b$  and the lower one is increasing with  $b$ , by a balancing argument the optimal  $b$ , denoted  $b^*$ , equates both operands. Therefore,

$$b^* = \frac{R(n - 1, k) - R(n - 1, k - 1)}{(\alpha - 1)R(n - 1, k - 1) - (\alpha^{-1} - 1)R(n - 1, k)}.$$

Substituting  $b^*$  for  $b$  in either operand of the min in (13) and rearranging, we obtain the expression for the reciprocal  $R^{-1}$ :

$$\begin{aligned} R^{-1}(n, k) &= \frac{1}{\alpha + 1} R^{-1}(n - 1, k - 1) \\ &\quad + \frac{\alpha}{\alpha + 1} R^{-1}(n - 1, k). \end{aligned} \quad (14)$$

It is possible to solve  $R^{-1}$  in a closed form (in terms of partial binomial sums) as follows. Artificially extend the domain of  $R^{-1}(n, k)$  such that

$$R^{-1}(n, j) = \begin{cases} 1 & j \leq 0; \\ \alpha^{n-2k} & j \geq n. \end{cases} \quad (15)$$

(It is possible to prove by induction on  $n$  that the extended recurrence is well defined.) Now consider the following graph (similar to the “Pascal triangle”) called a *binomial tree*,<sup>10</sup> corresponding to an expansion of the extended recurrence  $R^{-1}(n, k)$ . In this graph, each node is labeled by some pair  $(x, y)$  that corresponds to the value  $R^{-1}(x, y)$ . The root of this graph is  $(n, k)$  and it has two outgoing edges, one to its left child  $(x - 1, y - 1)$  and the other to its right

<sup>10</sup> The name *binomial tree* is typically used in finance [Cox and Rubinstein 1985]. Note that the resulting graph is not a tree in the graph-theoretic sense, but a lattice.

child  $(x - 1, y)$ . All the leaves are of the form  $(1, k - (n - i))$ ,  $i = 1, \dots, n$  and their values are obtained by the extension (15). By (14), the value for each node  $(x, y)$  is computed from the values of its children  $(x - 1, y - 1)$  and  $(x - 1, y)$ . Notice that in each node  $(x, y)$ ,  $x$  corresponds to the height of the node (with the leaves at level 1 and the root at level  $n$ ) and  $y$  corresponds to the edge distance from the top-left to bottom-right diagonal (i.e., the sequence of edges connecting  $(n, k)$  to  $(1, k)$ ). Notice that all the paths to the same leaf have the same number of left-right (and right-left) moves. In particular, the path to the leaf  $(1, k - (n - i))$  has  $n - i$  top-left edges and  $(n - 1) - (n - i) = i - 1$  top-right edges (there is a total of  $n - 1$  edges in each path). Using (14), when we calculate the value  $R^{-1}(n, k)$  each left move contributes a factor  $q = 1/(1 + \alpha)$  and each right move contributes a factor  $p = \alpha/(1 + \alpha)$  (notice that  $q = 1 - p$ ). Hence the “weight” of the path to the leaf  $(1, k - (n - i))$  is  $p^{i-1}q^{n-i}$ . Define

$$B_p \left\{ \begin{matrix} n \\ k \end{matrix} \right\} = \sum_{0 \leq i \leq k} \binom{n}{i} p^i (1-p)^{n-i},$$

the partial binomial sum. Abbreviating  $z_i = k - (n - i)$ , we thus have

$$\begin{aligned} R^{-1}(n, k) &= \sum_{\text{leaf } (1, z_i)} R^{-1}(1, z_i) \\ &\quad \cdot [\# \text{ of paths to } (1, z_i)] \\ &\quad \cdot [\text{weight of a path to } (1, z_i)] \\ &= \sum_{1 \leq i \leq n} R^{-1}(1, z_i) \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &= \sum_{z_i \leq 0} R^{-1}(1, z_i) \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &\quad + \sum_{z_i \geq 1} R^{-1}(1, z_i) \binom{n-1}{i-1} p^{i-1} q^{n-i} \end{aligned}$$

$$\begin{aligned} &= \sum_{z_i \leq 0} 1 \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &\quad + \sum_{z_i \geq 1} \alpha^{n-2k} \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &= \sum_{i=1}^{n-k} \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &\quad + \alpha^{n-2k} \sum_{i=n-k+1}^n \binom{n-1}{i-1} p^{i-1} q^{n-i} \\ &= \sum_{i=0}^{n-k-1} \binom{n-1}{i} p^i q^{n-i-1} \\ &\quad + \alpha^{n-2k} \cdot \sum_{i=0}^{n-k} \binom{n-1}{i} q^i p^{n-i-1} \\ &= B_p \left\{ \begin{matrix} n-1 \\ n-k-1 \end{matrix} \right\} + \alpha^{n-2k} \cdot B_q \left\{ \begin{matrix} n-1 \\ n-k \end{matrix} \right\} \\ &= B_p \left\{ \begin{matrix} n-1 \\ n-k-1 \end{matrix} \right\} + \alpha^{n-2k} \cdot B_p \left\{ \begin{matrix} n-1 \\ k-1 \end{matrix} \right\}. \end{aligned}$$

Using the preceding expression and tail estimates of the binomial distribution, Chou et al. [1995b] obtained the following asymptotic behavior of  $R(n, k)$ . Let  $c \in [0, 1]$  and define

$$\gamma_x = \frac{x^x(1-x)^{(1-x)}(1+\alpha)}{\alpha^{(1-x)}}. \quad (16)$$

Then the following bounds hold (cruder but similar bounds were obtained earlier in Chou et al. [1995a]).

- (1) if  $0 \leq c < p$ , then  $R(n, cn) \rightarrow 1$ ;
- (2) if  $c = p$ , then  $R(n, cn) \rightarrow 2$ ;
- (3) if  $p < c < q$ , then  $R(n, cn) = \Theta(\sqrt{n} \gamma_c^n)$ ;
- (4) if  $c = q$ , then  $R(n, cn) \rightarrow 2\gamma_c^n = 2\alpha^{(2c-1)n}$ ;
- (5) if  $q < c \leq 1$ , then  $R(n, cn) \rightarrow \alpha^{(2c-1)n}$  ( $R(n, cn) \geq \alpha^{(2c-1)n}$ ).

This result entails an interesting corollary. Consider the (optimal offline) BAH. If  $c > \frac{1}{2}$ , BAH invests its entire wealth on the first period and cashes it at the end of the game. Otherwise, BAH keeps its wealth in cash. The return of BAH,  $R(\text{BAH})$ , is thus

$$R(\text{BAH}) = \begin{cases} 1 & \text{if } c \leq \frac{1}{2}; \\ \alpha^{n(2c-1)} & \text{if } c \geq \frac{1}{2}. \end{cases} \quad (17)$$

It follows that algorithm  $S^{**}$  always outperforms BAH. Moreover, if  $q < c < p$ ,  $S^{**}$  performs exponentially better than BAH. Whenever  $c \in [0, \frac{1}{2}]$  or  $c \in [p, 1]$  this result readily follows from the bounds obtained for  $S^{**}$  and the return of BAH. Whenever  $c \in (\frac{1}{2}, p)$ , it is not hard to see that  $\gamma_c > \alpha^{2c-1}$  (the function  $f(c) = \gamma_c/\alpha^{2c-1} = (c/\alpha)^c(1-c)^{1-c}(1+\alpha)$  is strictly decreasing in  $(\frac{1}{2}, p)$  and  $f(p) = 1$ ).

Thus, whenever the market is “stable,” in the sense that it does not exhibit a “major” trend but is nevertheless “active” (i.e., there are fluctuations), algorithm  $S^{**}$  performs remarkably well (in particular, relative to BAH). Moreover, even if the market exhibits a slight unfavorable trend (i.e.,  $c \in (q, \frac{1}{2})$ ), algorithm  $S^{**}$  still yields exponential returns. Notice that the size of this “profitable” interval  $(q, p)$  increases with  $\alpha$  and thus can be controlled by the online player. Nevertheless, for larger values of  $\alpha$ , because of the time scaling required to approximate the fixed-fluctuation model, algorithm  $S^{**}$  may ignore many prices and therefore miss profitable transactions.

Chou et al. [1995b] also compare these bounds on the performance of  $S^{**}$  to two more algorithms. First, they study the performance of the optimal constant rebalanced algorithm, denoted CBAL\*, against the  $(\alpha, n, k)$ -adversary. Second, they study the following distributional model corresponding to the constraint that feasible market se-

quences must be of type  $(n, k)$ . Specifically, in this model a random price-relative sequence is chosen uniformly among all the feasible sequences (of length  $n$  and with exactly  $k$  downward fluctuations). With respect to this model they determine the optimal trading algorithm (i.e., optimal with respect to average return). Consider first the constant rebalanced algorithm. Let  $b$  be the constant fraction of wealth invested by CBAL in the security. With respect to any feasible market sequence the return of CBAL is

$$(1 - b + b\alpha)^k(1 - b + b\alpha^{-1})^{n-k}.$$

The optimal value of  $b$ , denoted  $b^*$ , is easily determined to be  $b^* = (\alpha c + c - 1)/(\alpha - 1)$ , where  $c = n/k$ .

The distributional variant described can be presented equivalently as follows. Define  $l = n - k$ . Given the  $(n, k)$  restriction, the probability of an  $\alpha^{-1}$ -fluctuation in the first period is  $k/n$  and the probability of an  $\alpha$ -fluctuation is  $l/n$ . After the first fluctuation is (randomly) obtained, the remaining sequence must be of type  $(n - 1, k - 1)$  or  $(n, k - 1)$  depending on whether the first fluctuation was  $\alpha^{-1}$  or  $\alpha$ , respectively, and so on. Denote by  $R(n, l)$  the expected return of the optimal online algorithm when feasible markets are of type  $(n, k)$ . Thus

$$R(n, l) = \max_{0 \leq b \leq 1} \left\{ \begin{aligned} & \frac{n}{n} (1 - b + b\alpha^{-1})R(n - 1, l) \\ & + \frac{l}{n} (1 - b + b\alpha)R(n, l - 1) \end{aligned} \right\}.$$

Since the return is a maximum of a linear function of  $b$ , it follows that  $b$  is either 0 or 1. Note that  $b$  is only the investment for the first period. Nevertheless, a nontrivial analysis presented in Chou et al. [1995b] shows that the optimal online algorithm is constant rebalanced. Therefore, given that the first

**Table II.** Asymptotic Performance of CBAL\*, S\*\*, DIST, and BAH for “Trendy” and “Nontrendy” Markets

Region of $c = k/n$	CBAL*	S**	DIST	BAH
$(0, q)$	1	1	$\Theta(1)$	1
$c = q$	1	$\frac{2}{\alpha}$	$\Theta(\sqrt{n})$	1
$(q, \frac{1}{2})$	$\gamma_c$	$\Theta(\sqrt{n} \cdot \gamma_c)$	$\Theta(\sqrt{n} \cdot \gamma_c)$	1
$(\frac{1}{2}, p)$	$\gamma_c$	$\Theta(\sqrt{n} \cdot \gamma_c)$	$\Theta(\sqrt{n} \cdot \gamma_c)$	$\alpha^{(2c-1)n}$
$p$	$\alpha^{(2c-1)n}$	$2\alpha^{(2c-1)n}$	$\Theta(\sqrt{n} \cdot \alpha^{(2c-1)n})$	$\alpha^{(2c-1)n}$
$(p, 1)$	$\alpha^{(2c-1)n}$	$\alpha^{(2c-1)n}$	$\Theta(\alpha^{(2c-1)n})$	$\alpha^{(2c-1)n}$

investment is “all or nothing,” the optimal online strategy is BAH which invests in the security iff  $k \geq l$ . We note that, with respect to a sequence of price relatives generated randomly by a sequence of i.i.d. random variables, it is known that the optimal algorithm is constant rebalanced [Breiman 1961].

The performances of the optimal constant rebalanced algorithm CBAL\* and the optimal algorithm of the distributional problem variant (denoted DIST), plus the bounds for algorithm S\*\*, are summarized in Table II (which also includes, for comparison, the return of the optimal offline BAH).

A striking fact immediately observed in this table is that the asymptotic returns of all three optimal online algorithms are similar. Denote the four regions of  $c$ ,  $(0, q)$ ,  $(q, \frac{1}{2})$ ,  $(\frac{1}{2}, p)$ , and  $(p, 1)$  by R1, R2, R3, and R4, respectively. For markets that exhibit significant trend (regions R1 and R4), the returns of all three algorithms are within a constant factor of BAH. In fact, in these regions the optimal constant rebalanced algorithm degenerates to BAH (and DIST acts in a similar way to BAH across all regions). For markets with no trend (regions R2 and R3), all three algorithms perform exponentially better than BAH.

These results also indicate that information about the “type” of the market can be almost as valuable as information about the distribution (note that in region R3, DIST outperforms S\*\* and CBAL by a constant factor).

These results, together with the fact that both CBAL\* and S\*\* make transactions in almost every trading period, may indicate that the following is true whenever transaction costs are intro-

duced. When the market is trendy, even if the trader has perfect prediction of the trend’s slope it may be beneficial to avoid market timing and follow the trend using BAH. On the other hand, the exponential advantage of market-timing strategies in nontrendy markets may very well compensate for the losses incurred by transaction costs.

Finally, we note that Chou et al. [1995b] show how to derive the return of S\*\* using a standard method called *binomial risk-neutral pricing* (see Hull [1993]) that is usually used to price options and other derivative instruments. Their results imply that the binomial risk-neutral pricing method is equivalent to worst-case analysis.

**5.7.3 On the Empirical Performance of S\*\*.** Chou [1994] reported on preliminary experimental results testing the performance of S\*\*. He used two data sets, both of which included intraday prices for both US dollars versus Japanese yen and US dollars versus German marks. Data set A spanned one month and included all price quotations<sup>11</sup> and data set B included prices at six-minute intervals during one year. In both cases  $\alpha$  was set to  $1 + 5/(\text{initial exchange rate})$ , since almost all changes in these data were of 5 points<sup>12</sup>; since such changes were small relative to the exchange rates, the additive changes could be reasonably well approximated by multiplicative changes of  $\alpha$ . The value of  $n$  varied between 100 and

<sup>11</sup> In such data a new price “tick” occurs every 10–120 seconds.

<sup>12</sup> A *point* is the smallest unit used to measure prices.

1,000, thus breaking the trading period into a sequence of short games with reinvestment. Finally,  $k$  was naively chosen to be  $n/2$  for all games. The returns of  $S^{**}$  with respect to data set A were remarkably high (223% and 167% for USD/DM and USD/JY, respectively). The results with respect to data set B were marginal (104% and 111%). When transaction costs of 0.02% were introduced, the returns against data set A remained very high (133% and 134%) but against data set B,  $S^{**}$  suffered severe losses (73% and 89%). Examinations of these data sets revealed that in data set A the assumption of  $k = n/2$  was quite closely satisfied, whereas in data set B it was severely violated. One not surprising drawback of  $S^{**}$  that was found in these experiments is that  $S^{**}$  reacts to and trades with every price tick.

These results indicate that algorithm  $S^{**}$  is quite robust with respect to the assumed parameters. Nevertheless, no theoretical results on the algorithm's sensitivity to noise were presented. Note that sensitivity analyses are desired in all problems where one makes use of statistical adversaries.

Finally, it is important to emphasize that empirical results such as the preceding usually cannot be conclusive. In general, in order to exhibit informative empirical performance of financial algorithms one must quantify the risk associated with the exhibited levels of return.<sup>13</sup>

Chou et al. [1995b] reported on extensive statistical tests measuring the trends exhibited in price sequences as a function of  $\alpha$  and  $n$ . In particular, their goal was to measure the empirical distribution of trend types in terms of the

preceding four regions (R1, R2, R3, and R4) with varying values of  $\alpha$  and  $n$ . The data used included 486 of the 500 stocks comprising the Standard and Poor 500 index (S&P 500) spanning a time period of about 30 years. The tests reveal the following relations. Call regions R2 and R3 the “nontrendy” regions and regions R1 and R4 the “trendy” regions. As  $\alpha$  or  $n$  grow, the fraction of sequences (of length  $n$ ) in the nontrendy regions grows (and the fraction of sequences in region R3 is always larger than in R2). For small values of  $\alpha$  (e.g.,  $1.005 \leq \alpha \leq 1.05$ ) the majority of sequences are in the trendy regions; for larger values of  $\alpha$  the majority of sequences are in the nontrendy regions. From these results it follows that with respect to the stock market, the practical advantage of market-timing strategies like  $S^{**}$  or constant rebalanced, using a fixed fluctuation model, can be obtained only when using large values of  $\alpha$  and  $n$  (while using time scaling). On the other hand, if, in a particular market, such large values of  $\alpha$  are not feasible, it may be wiser to remain “static” and use BAH. Note that for large values of  $\alpha$  (say,  $\alpha \geq 1.01$ ), transaction points cannot occur too often (which may be considered an advantage by some traders).

## 5.8 Online Portfolio Selection, Results for Arbitrary $m$

5.8.1 *The Family of  $\mu$ -Weighted Portfolio-Selection Algorithms.* Cover and Ordentlich [1996] define the following general class of  $\mu$ -weighted online portfolio-selection algorithms. Each algorithm in this class is specified by a probability measure  $\mu$  over the set of all portfolios. The algorithm starts by hedging uniformly on all possible portfolios. Then it rebalances the weight it gives to various securities according to the past performance of constant rebalanced portfolios while putting more weight on the better performing ones. The probability measure  $\mu$  provides an additional weighting mechanism that can favor particular portfolios.

<sup>13</sup> There are various common measures of risk, for example, Sharpe's ratio, given by  $(E(R(\text{ALG})) - R(\text{risk-free}))/\sigma(\text{ALG})$ , where  $E[R(\text{ALG})]$  and  $\sigma(\text{ALG})$  are the mean and standard deviation of ALG's return and  $R(\text{risk-free})$  is the risk-free rate of return [Von Neumann and Morgenstern 1944]. The reader is referred to any elementary finance book such as Bodie et al. [1993] that discusses measures of risk.

The introduction of the  $\mu$ -weighted algorithms requires some notation. Fix some positive integer  $m$ . Let  $\mathcal{B}$  be the set of all possible portfolios over  $m$  securities (i.e.,  $\mathcal{B}$  is the  $(m - 1)$ -dimensional simplex). With respect to a market sequence  $X = \vec{x}_1, \dots, \vec{x}_n$ , define  $X_i = \vec{x}_1, \dots, \vec{x}_i$ ,  $i = 1, 2, \dots, n$ , the prefix of  $X$  consisting of the first  $i$  market vectors. With respect to  $X_i$  define

$$R_i(\vec{b}) = R(\vec{b}, X_i) = \prod_{j=1}^i \vec{b}^t \cdot \vec{x}_j.$$

That is,  $R_i(\vec{b})$  is the compounded return of a fixed portfolio  $\vec{b}$  after  $i$  trading periods. By convention, set  $R_0(\vec{b}) = 1$ , the initial wealth. (Alternatively,  $R_i(\vec{b}) = \text{CBAL}_{\vec{b}}(X_i)$ .)

Fix some market sequence  $X$  and let  $\mu$  be a probability measure over  $\mathcal{B}$ . An algorithm is a  $\mu$ -weighted portfolio selection algorithm if its  $i$ th period portfolio  $\vec{b}_i$  is specified by

$$\vec{b}_i = \frac{\int_{\mathcal{B}} \vec{b} R_{i-1}(\vec{b}) d\mu(\vec{b})}{\int_{\mathcal{B}} R_{i-1}(\vec{b}) d\mu(\vec{b})}. \quad (18)$$

Here the integral is the Lebesgue integral. Note that when the probability measure  $\mu$  has a density function the Lebesgue integral can be expressed in terms of ordinary (multiple) integrals. Clearly, any  $\mu$ -weighted algorithm operates online.

Let  $X$  be any market sequence of length  $n$  and let  $\vec{b}_1, \vec{b}_2, \dots, \vec{b}_n$  be the portfolios obtained by a  $\mu$ -weighted algorithm  $\text{ALG}_{\mu}$ . The expression for the compounded return  $\text{ALG}_{\mu}(X)$ , of the  $\mu$ -weighted algorithm  $\text{ALG}_{\mu}$ , can be easily simplified as:

$$\begin{aligned} \text{ALG}_{\mu}(X) &= \prod_{1 \leq i \leq n} \vec{b}_i^t \cdot \vec{x}_i \\ &= \prod_{1 \leq i \leq n} \frac{(\int_{\mathcal{B}} \vec{b}^t R_{i-1}(\vec{b}) d\mu(\vec{b})) \cdot \vec{x}_i}{\int_{\mathcal{B}} R_{i-1}(\vec{b}) d\mu(\vec{b})} \end{aligned}$$

$$\begin{aligned} &= \prod_{1 \leq i \leq n} \frac{(\int_{\mathcal{B}} \vec{b}^t \cdot \vec{x}_i R_{i-1}(\vec{b}) d\mu(\vec{b}))}{\int_{\mathcal{B}} R_{i-1}(\vec{b}) d\mu(\vec{b})} \\ &= \prod_{1 \leq i \leq n} \frac{\int_{\mathcal{B}} R_i(\vec{b}) d\mu(\vec{b})}{\int_{\mathcal{B}} R_{i-1}(\vec{b}) d\mu(\vec{b})}. \quad (19) \end{aligned}$$

Hence since the last product telescopes we have

$$\begin{aligned} \text{ALG}_{\mu}(X) &= \int_{\mathcal{B}} R_n(\vec{b}) d\mu(\vec{b}) \\ &= \int_{\mathcal{B}} \text{CBAL}_{\vec{b}}(X) \mu(\vec{b}). \quad (20) \end{aligned}$$

Notice that although the  $\mu$ -weighted algorithm adapts its portfolio online and gives more weight to the better performing (constant rebalanced) portfolios (see Equation (18)), its final return is simply a  $\mu$ -average of the returns of all constant rebalanced portfolios in  $\mathcal{B}$ .

**5.8.2 The Uniform-Weighted Algorithm.** Denote by UNI the  $\mu$ -weighted algorithm corresponding to the uniform distribution over  $\mathcal{B}$ . Denote by CBAL-OPT the optimal offline constant rebalanced algorithm. That is,  $\text{CBAL-OPT} = \text{CBAL}_{\vec{b}^*}$  where for any market sequence  $X = \vec{x}_1, \dots, \vec{x}_n$ , the fixed portfolio  $\vec{b}^*$  used by CBAL-OPT is

$$\vec{b}^* = \arg \max_{\vec{b}} \prod_{1 \leq i \leq n} \vec{b} \cdot \vec{x}_i.$$

Cover and Ordentlich show that with respect to the optimal offline constant rebalanced algorithm CBAL-OPT, and any market sequence  $X$  of length  $n$ , the following holds.

$$\begin{aligned} \frac{\text{CBAL-OPT}(X)}{\text{UNI}(X)} &\leq \left( \frac{n + m - 1}{m - 1} \right) \\ &\leq (n + 1)^{m-1}. \quad (21) \end{aligned}$$

Thus, in a restricted sense of competitiveness, algorithm UNI is  $(n + 1)^{m-1}$

competitive (with respect to CBAL-OPT). Stated equivalently (but perhaps more astoundingly), using the preceding exponential growth-rate utility function and the minimax regret criterion, this result states that the minimax regret (of the exponential growth rates) of algorithm UNI is bounded above by  $(m - 1)\log(n + 1)/n$ . This means that the regret is diminishing when  $n$  grows.

The proof of the bound (21) is interesting and is based on profound ideas. Somewhat unexpectedly, it is possible to transform the analysis into information-theoretic terms. Indeed, in proving this theorem we rely on some elementary results from information theory.

We now sketch the proof of this bound. Let  $[m] = \{1, 2, \dots, m\}$ . Denote by  $J_n = (j_1, j_2, \dots, j_n)$  any sequence of indices from  $[m]$  (i.e.,  $J_n \in [m]^n$ ). Let  $X = \tilde{x}_1, \dots, \tilde{x}_n$  and let  $\vec{b}$  be any portfolio. The analysis relies on the following trick that gives a different representation for the compounded return of the constant rebalanced algorithm  $\text{CBAL}_{\vec{b}}$ .

$$\begin{aligned} \text{CBAL}_{\vec{b}}(X) &= \prod_{1 \leq i \leq n} \vec{b}^t \cdot \tilde{x}_i \\ &= \prod_{1 \leq i \leq n} \sum_{1 \leq j \leq m} b_j x_{ij} \\ &= \sum_{J_n \in [m]^n} \prod_{1 \leq i \leq n} b_{j_i} x_{ij_i}. \end{aligned} \quad (22)$$

Fix any market sequence  $X$  of length  $n$ . Let  $\vec{b}^*$  be the portfolio used by CBAL-OPT. Using the representation (22) for CBAL-OPT and UNI we have

$$\begin{aligned} \text{CBAL-OPT}(X) &= \sum_{J_n \in [m]^n} \prod_{i=1}^n b_{j_i}^* x_{ij_i}; \\ \text{UNI}(X) &= \sum_{J_n \in [m]^n} \int \prod_{i=1}^n b_{j_i} x_{ij_i} d\mu(\vec{b}), \end{aligned}$$

where  $\mu$  is a uniform probability measure over  $\mathcal{B}$ . The ratio of compound

returns,  $\text{CBAL-OPT}(X)/\text{UNI}(X)$ , is now a ratio of two finite summations each involving  $N = m^n$  nonnegative terms. It is not hard to see that if  $\alpha_1, \dots, \alpha_N \geq 0$ ,  $\beta_1, \dots, \beta_N \geq 0$ , then

$$\frac{\sum_{1 \leq i \leq N} \alpha_i}{\sum_{1 \leq i \leq N} \alpha_i \beta_i} \leq \max_j \frac{\alpha_j}{\beta_j}.$$

Using this upper bound on the ratio of sums we have

$$\begin{aligned} &\frac{\text{CBAL-OPT}(X)}{\text{UNI}(X)} \\ &\leq \max_{J_n \in [m]^n: \prod_{i=1}^n x_{ij_i} > 0} \frac{\prod_{i=1}^n b_{j_i}^* x_{ij_i}}{\int_{\mathcal{B}} \prod_{i=1}^n b_{j_i} x_{ij_i} d\mu(\vec{b})} \\ &= \max_{J_n \in [m]^n: \prod_{i=1}^n x_{ij_i} > 0} \frac{\prod_{i=1}^n b_{j_i}^*}{\int_{\mathcal{B}} \prod_{i=1}^n b_{j_i} d\mu(\vec{b})} \\ &= \max_{J_n \in [m]^n} \frac{\prod_{i=1}^n b_{j_i}^*}{\int_{\mathcal{B}} \prod_{i=1}^n b_{j_i} d\mu(\vec{b})}. \end{aligned} \quad (23)$$

The right-hand side of (23) can now be upper bounded as follows using the “method of types” of information theory (see Cover and Thomas [1991], Chapter 12). For each vector  $J_n$  and  $r \in [m]$ , denote by  $\nu_r(J_n)$  the proportion of the number of occurrences  $n_i(J_n)$  of  $r$  in  $J_n$  (i.e., the number of occurrences divided by  $n$ ). In this way, the distribution  $D(J_n) = (\nu_1(J_n), \nu_2(J_n), \dots, \nu_m(J_n))$  specifies the “type” of the sequence  $J_n$ . Theorem 12.1.2 in Cover and Thomas [1991] then states that for any distribution (portfolio)  $\vec{b}$  and any such  $J_n$ .

$$\prod_{1 \leq i \leq n} b_{j_i} \leq \frac{1}{2^{nH(D(J_n))}},$$

where  $H(D(J_n))$  is the Shannon entropy of  $D(J_n)$  (i.e.,  $H(D(J_n)) = -\sum_i \nu_i(J_n) \log \nu_i(J_n)$ ). Equality is obtained at  $\vec{b} = D(J_n)$ . This result upper bounds the numerator of (23). The integral in the denominator can be expressed in a closed

form as

$$\int_{\mathbb{R}} \prod_{1 \leq i \leq n} b_i d\mu(\vec{b}) = 1 / \left( \binom{n+m-1}{m-1} T(J_n) \right),$$

where  $T(J_n)$  is the number of sequences of the type  $D(J_n)$ . Note that

$$\binom{n+m-1}{m-1}$$

is the number of  $D(J_n)$ -types. Theorem 12.1.3 in Cover and Thomas [1991] states that  $T(J_n) \leq 2^{nH(D(J_n))}$ . From these bounds it readily follows that

$$\frac{\text{BOPT}(X)}{\text{UNI}(X)} \leq \binom{n+m-1}{m-1},$$

and the upper bound  $(n+1)^{m-1}$  on the right-hand side follows from Theorem 12.1.1 in Cover and Thomas [1991].

**5.8.3 Implementation of the Uniform-Weighted Algorithm.** Cover and Ordentlich [1996] obtain recursive procedures to compute the preceding algorithm but the time and space requirements of this procedure grow as  $n^{m-1}$  so it can be applied only to a small number of securities. Blum and Kalai [1997] offer the following randomized approximation to the uniform-weighted algorithm. As we see in (20), the algorithm UNI's return is simply a uniform-weighted average of all constant rebalanced algorithms. This fact suggests the following approximation: among all portfolios choose  $K$  uniformly at random. Then divide and invest the initial wealth equally between all  $K$  chosen portfolios. Using standard tail bounds (e.g., Chebyshev's inequality), it is not hard to show that using  $K = (R-1)/\varepsilon\delta^2$ , where  $R$  is the (empirical) "competitive ratio" of UNI (with respect to the optimum constant rebalanced portfolio), we can guarantee with probability at least  $1 - \delta$  a return that is at least as

large as  $1 - \varepsilon$  times the return of algorithm UNI. Although in the worst case  $R$  grows as  $n^{m-1}$ , experiments on historical stock-market data [Cover 1991; Helmbold et al. 1996] suggest that  $R$  is typically much lower. Therefore, in using this approximation when  $m$  is large, one should estimate  $R$  empirically and use this algorithm when  $R$  is sufficiently small.

#### 5.8.4 Dirichlet-Weighted Algorithms.

We now consider  $\mu$ -weighted algorithms where  $\mu$  is a Dirichlet distribution. A random vector  $\vec{b} = (b_1, b_2, \dots, b_m)$  has a Dirichlet distribution (sometimes called the *multinomial beta distribution*), denoted Dirichlet ( $\vec{\alpha}$ ), with parametric vector  $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ ,  $\alpha_i > 0$ , if the probability density function of  $\vec{b}$  with respect to  $\vec{\alpha}$ ,  $f_{\vec{\alpha}}(\vec{b})$  satisfies the following conditions. At any point  $\vec{b} \in \mathbb{R}_m$  with  $b_i \geq 0$ ,  $\sum_i b_i = 1$ ,

$$f_{\vec{\alpha}}(\vec{b}) = \frac{\Gamma(\alpha_1 + \dots + \alpha_m)}{\Gamma(\alpha_1) \dots \Gamma(\alpha_m)} b_1^{\alpha_1-1} \dots b_m^{\alpha_m-1},$$

where  $\Gamma$  is the gamma function.<sup>14</sup> At any other point  $\vec{b}' \in \mathbb{R}_m$ ,  $f_{\vec{\alpha}}(\vec{b}') = 0$ . Thus  $f_{\vec{\alpha}}(\vec{b})$  is positive only when the vector  $\vec{b}$  is a distribution function,  $\sum_i b_i = 1$ . Hence, despite its appearance,  $f_{\vec{\alpha}}(\vec{b})$  is not an  $m$ -dimensional p.d.f. but rather gives the joint p.d.f. of any  $(m-1)$ -subset of the  $m$  random variables  $b_1, b_2, \dots, b_m$  (and one of them is set by the relation  $\sum_i b_i = 1$ ). Notice that the Dirichlet  $(1, 1, \dots, 1)$  density is the uniform density over the  $(m-1)$ -dimensional simplex. Also, we note that the symmetric Dirichlet  $(a, a, \dots, a)$  density with  $a < 1$  is strictly convex (with poles at the unit vectors). This means that if it is used by a  $\mu$ -weighted algorithm it gives more weight to "extremal" portfolios that invest in a small number of securities.

Denote by  $\text{DIR}_{\vec{\alpha}}$  the  $\mu$ -weighted portfolio selection algorithm, with  $\mu$  being the

<sup>14</sup> The gamma function is  $\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} dt$ . It can be shown that  $\Gamma(1) = 1$  and that  $\Gamma(x+1) = x\Gamma(x)$ . Thus if  $n$  is an integer,  $n \geq 1$ ,  $\Gamma(n+1) = n!$ .

symmetric Dirichlet  $(\bar{\alpha})$ . Thus,  $\text{DIR}_{(1,1,\dots,1)}$  is exactly algorithm UNI. For the Dirichlet  $(\frac{1}{2}, \dots, \frac{1}{2})$ -weighted algorithm, Cover and Ordentlich [1996] show that for any market sequence  $X$ ,

$$\frac{\text{CBAL-OPT}(X)}{\text{DIR}_{(1/2,1/2,\dots,1/2)}(X)} \leq \frac{\Gamma(1/2)\Gamma(n+m/2)}{\Gamma(m/2)\Gamma(n+1/2)} \leq 2(n+1)^{(m-1)/2}.$$

Thus  $\text{DIR}_{(1/2,1/2,\dots,1/2)}$  is  $(2(n+1)^{(m-1)/2})$ -competitive with respect to CBAL-OPT. Here again, stated in terms of the exponential growth-rate utility function and the minimax regret criterion, this bound implies that the minimax regret (of the exponential growth rates) of algorithm DIR is bounded above by  $((m-1)\log(2n+2))/2n$ .

This result for algorithm DIR can be proved following the same line as the proof of the bound for algorithm UNI but it is somewhat more technical (since the integrals involved are more complex).

**5.8.5 Extremal Mixture Algorithms.** Equation (22) suggests the following interpretation. Fix some  $J_n = (j_1, j_2, \dots, j_n) \in [m]^n$  and consider the “extremal” algorithm, denoted  $\text{EXT}_{J_n}$ , that at time  $i$  invests the entire wealth in the  $j_i$ th security. Let  $\vec{b} = (b_1, \dots, b_m)$  and consider the constant rebalanced algorithm  $\text{CBAL}_{\vec{b}}$ . It is possible to represent  $\text{CBAL}_{\vec{b}}$  in terms of extremal algorithms as follows. Partition  $\text{CBAL}_{\vec{b}}$ ’s initial wealth into  $m^n$  portions, one for each of the  $m^n$  sequences  $J_n$  in  $[m]^n$ . The number of dollars in the part corresponding to  $J_n$  is  $w(J_n) = \prod_{i=1}^n b_{j_i}$ . Notice that

$$\sum_{J_n \in [m]^n} w(J_n) = 1.$$

That is,  $\{w(J_n)\}$  is a probability distribution (and a proper partition of the initial wealth). For each of the extremal algorithms  $\text{EXT}_{J_n}$ , maintain a separate investment “account” starting with an initial wealth of  $w(J_n)$ . For a market sequence  $X$ , the wealth accrued by  $\text{EXT}_{J_n}$

is thus

$$\text{EXT}_{J_n}(X) = w(J_n) \prod_{i=1}^n x_{j_i} = \prod_{i=1}^n b_{j_i} x_{j_i}.$$

Hence, by (22) we see that the sum of wealth accrued by all the extremal algorithms is exactly  $\text{CBAL}_{\vec{b}}(X)$ .

Exactly as for the one-way (and search) algorithms (see Section 3), we can interpret (and apply) this algorithm as a (mixed) randomized algorithm. Simply choose the extremal algorithm  $\text{EXT}_{J_n}$  with probability  $w(J_n)$ . In this case, of course,  $\text{CBAL}_{\vec{b}}(X)$  will be the expected return of this algorithm. From a computational point of view, operating the extremal mixture algorithm randomly saves most of the large memory required for operating the deterministic algorithm.

Using Equation (20) we can interpret algorithms UNI and DIR (and any  $\mu$ -weighted algorithm) analogously. All that is needed is to partition the initial wealth so that algorithm  $\text{EXT}_{J_n}$  is assigned

$$w(J_n) = \int_{\mathfrak{B}} \prod_{i=1}^n b_{j_i} d\mu(\vec{b})$$

(with  $\mu$  being the corresponding probability measure).

The preceding representation (interpretation) of the constant rebalanced (and  $\mu$ -weighted) algorithms gives rise to the following general class of “extremal mixture” algorithms. For each  $n$  the extremal mixture strategy is specified by a probability distribution  $w$  over the set  $\{J_n\}$  and invests in the extremal algorithm  $\text{EXT}_{J_n}$  a fraction  $w(J_n)$  of its initial wealth. From the previous discussion it follows that for each  $n$ , an optimal online extremal mixture algorithm performs at least as well as the best  $\mu$ -weighted algorithm.

For each  $n$  let  $\text{oEXT}_n$  be the extremal mixture algorithm specified by the following probability distribution over the extremal algorithms. For each  $J_n \in$

$[m]^n$ , set  $q(\mathcal{J}_n) = \prod_{i=1}^n v_i(\mathcal{J}_n)n_i(\mathcal{J}_n)$ . The distribution  $w$  is then

$$w(\mathcal{J}_n) = \frac{q(\mathcal{J}_n)}{\sum_{\mathcal{J}_n \in [m]^n} q(\mathcal{J}_n)}.$$

Ordentlich and Cover [1996] prove that algorithm  $\text{OEXT}_n$  is an optimal extremal mixture algorithm for each market sequence  $X$  of length  $n$ . Furthermore, it is shown that for each such  $X$ ,

$$\frac{\text{CBAL-OPT}(X)}{\text{OEXT}_n(X)} \leq \sum_{\mathcal{J}_n \in [m]^n} \frac{n!}{\prod_{1 \leq i \leq m} n_i(\mathcal{J}_n)!} \prod_{1 \leq i \leq m} \left( \frac{n_i(\mathcal{J}_n)}{n} \right)^{n_i(\mathcal{J}_n)}, \tag{24}$$

and in the worst case this bound is tight.

Given the equivalence of deterministic and (mixed) randomized extremal mixture algorithms, the ratio (24) is a lower bound on the competitive ratio of any randomized extremal mixture algorithm (and  $\mu$ -weighted algorithm) against an oblivious adversary (with respect to  $\text{CBAL-OPT}$ ).

For the case  $m = 2$  this competitive ratio (24) of  $\text{OEXT}_n$  (with respect to  $\text{CBAL}$ ) reduces to

$$\sum_{0 \leq k \leq n} \binom{n}{k} \cdot \left( \frac{k}{n} \right)^k \cdot \left( \frac{n-k}{n} \right)^{n-k},$$

and using Stirling approximation can be shown to be approximately  $\sqrt{n\pi/2} \approx 1.253 \sqrt{n}$ . Note that the competitive ratio of the  $\text{DIR}_{(1/2, \dots, 1/2)}$  algorithm for  $m = 2$  is no larger than  $2\sqrt{n+1}$ , so the Dirichlet algorithm attains a competitive ratio that is within a factor of (approximately, for large  $n$ ) 1.6 of algorithm  $\text{OEXT}_n$ .

**5.8.6 Portfolio Selection with Side Information.** Cover and Ordentlich [1996] extended the preceding portfolio-selection model to incorporate possible “side information” that the trader may have.

Typically such “side information” may be based on various kinds of predictions of future values of market vectors. To model side information, we consider an “oracle” that announces a number  $\text{info}_i \in \mathcal{I} = \{1, 2, \dots, k\}$  at the start of the  $i$ th trading period. The number announced represents an abstract state of some prediction apparatus. For example, in the best case  $\text{info}_i$  identifies the best security in the  $i$ th period. Nevertheless, we assume that the online player does not know in advance the quality of the side information provided to her and, moreover, that the quality of the side information may vary during the trading period. Thus in order to benefit from such side information the trader must learn its quality during the trading period. Formally, with respect to an  $n$ -period trading game we define *side information* as any sequence  $\text{info}_1, \text{info}_2, \dots, \text{info}_n$  with  $\text{info}_i \in \mathcal{I}$ .

Let  $X = \tilde{x}_1, \dots, \tilde{x}_n$  be any market sequence and  $I = \text{info}_1, \text{info}_2, \dots, \text{info}_n$  any side information. As usual, denote by  $I_i$  the prefix of  $I$  consisting of the first  $i$  elements. We denote by  $\text{ALG}(X|I)$  the return of the algorithm  $\text{ALG}$  with respect to  $X$  given the side information  $I$ . Let  $\vec{b}$  be any portfolio and  $\ell \in \mathcal{I}$ . With respect to the side information  $I_i$  and market sequence  $X_i$ , define

$$R_i(\vec{b}|\ell) = \prod_{j \leq i: \text{info}_j = \ell} \vec{b}' \tilde{x}_j.$$

That is,  $R_i(\vec{b}|\ell)$  is the compounded return of a constant rebalanced algorithm that is out of the market at all periods where the side information does not equal  $\ell$  and is fully invested, using the portfolio  $\vec{b}$ , at all other periods.

Let  $\mu$  be any probability measure. Using  $R_i(\vec{b}|\cdot)$ , we now define the  $\mu$ -weighted algorithm with side information as the algorithm that at the  $i$ th period uses the portfolio

$$\vec{b}_i(\text{info}_i) = \frac{\int_{\mathcal{B}} \vec{b} R_{i-1}(\vec{b}|\text{info}_i) d\mu(\vec{b})}{\int_{\mathcal{B}} R_{i-1}(\vec{b}|\text{info}_i) d\mu(\vec{b})}.$$

This is a generalization of the definition of the  $\mu$ -weighted algorithm. Clearly, when  $k = |\mathcal{I}| = 1$  (i.e., no side information) this definition reduces to (18).

In a way analogous to the simplification obtained by formula (19), we now simplify the expression for the compounded return  $\text{ALG}_\mu(X|I)$ :

$$\begin{aligned} \text{ALG}_\mu(X|I) &= \prod_{i=1}^n \tilde{b}_i^t(\text{info}_i) \tilde{x}_i \\ &= \prod_{j=1}^k \prod_{i \in n: \text{info}_i=j} \tilde{b}_i^t(j) \tilde{x}_i \\ &= \prod_{j=1}^k \int_{\mathcal{B}} R_n(\tilde{b}|j) d\mu(\tilde{b}) \quad (25) \end{aligned}$$

Although it is probably impossible to determine the performance ratio of the  $\mu$ -weighted algorithm with side information with respect to the optimal offline constant rebalanced algorithm (it is heavily dependent on the side information), it is possible to determine the performance ratio with respect to a more powerful offline algorithm that is also dependent on the side information. The advantage of this approach is that the dependence on the side information will “factor out” and we shall obtain a performance ratio that is dependent only on the cardinality of  $\mathcal{I}$ .

Fix  $\mathcal{I}$ . Let  $B : \mathcal{I} \rightarrow \mathcal{B}$  be any mapping from side information to portfolios. Define the *state-constant rebalanced algorithm (with mapping  $B$ )*, denoted  $\text{SCBAL}_B$ , as the algorithm that at each period uses one of the portfolios  $B(1), B(2), \dots, B(k)$ . Specifically, the algorithm invests according to the portfolio  $B(\text{info}_i)$  during the  $i$ th trading period (for which the side information is  $\text{info}_i$ ). The return of algorithm  $\text{SCBAL}_B$  for a market sequence  $X$  and side information sequence  $I$  is thus given by

$$\text{SCBAL}_B(X|I) = \prod_{i=1}^n B^t(\text{info}_i) \cdot \tilde{x}_i.$$

The optimal offline state-constant rebalanced algorithm, denoted  $\text{SCBAL-OPT}$ , is a state-constant rebalanced algorithm that optimizes its choice of the mapping  $B$  based on advance knowledge of the market sequence  $X$  and the side information  $I$ . That is, this algorithm uses the mapping  $B^*$ , where

$$B^* = \arg \max_B \prod_{i=1}^n B^t(\text{info}_i) \cdot \tilde{x}_i.$$

Denote by  $\text{UNI}(X|I)$  and  $\text{SCBAL-OPT}(X|I)$  the compounded returns of the online uniform-weighted algorithm and the optimal offline state-constant rebalanced algorithm, respectively, for a market sequence  $X$  given side information  $I$ . For each  $j = 1, 2, \dots, k$ , set  $n_j(I)$  to be the number of  $j$ s in  $I$ . Now for each market sequence  $X$  and side information  $I$ ,

$$\begin{aligned} \frac{\text{UNI}(X|I)}{\text{SCBAL-OPT}(X|I)} &\leq \prod_{j=1}^k (n_j(I) + 1)^{(m-1)} \\ &\leq (n + 1)^{k(m-1)}. \end{aligned}$$

To prove this bound, for each  $j = 1, 2, \dots, k$ , denote by  $X(j)$  the subsequence of  $X$ ,  $\tilde{x}_{j_1}, \tilde{x}_{j_2}, \dots, \tilde{x}_{j_l}$ , where  $\text{info}_{j_r} = j$  for all  $1 \leq r \leq l$ . By the definition of  $\text{SCBAL-OPT}$  and Equation (25) we have

$$\frac{\text{UNI}(X|I)}{\text{SCBAL-OPT}(X|I)} = \prod_{j=1}^k \frac{\text{UNI}(X(j))}{\text{SCBAL}_{B^*}(X(j))}.$$

Hence, by the known bound for algorithm  $\text{UNI}$  (21), applied with the sequences  $X(j)$ , we have

$$\begin{aligned} \prod_{j=1}^k \frac{\text{UNI}(X(j))}{\text{SCBAL}_{B^*}(X(j))} &\leq \prod_{j=1}^k (n_j(I) + 1)^{(m-1)} \\ &\leq \prod_{j=1}^k (n + 1)^{(m-1)}. \end{aligned}$$

With respect to a market sequence  $X$  and side information sequence  $I$ , denote by  $\text{DIR}_{(1/2, \dots, 1/2)}(X|I)$  the compounded return of the Dirichlet  $(\frac{1}{2}, \dots, \frac{1}{2})$  algorithm with side information. A derivation similar to the preceding (for algorithm UNI) proves the following result. For each market sequence  $X$  and side information  $I$ ,

$$\begin{aligned} \frac{\text{SCBAL-OPT}(X|I)}{\text{DIR}_{(1/2, \dots, 1/2)}(X|I)} &\leq \prod_{j=1}^k (n_j(I) + 1)^{(m-1)/2} \\ &\leq 2^k (n + 1)^{(k(m-1))/2}. \end{aligned}$$

**5.8.7 Portfolio Selection with Transaction Costs.** All the analyses presented so far completely ignore transaction costs. We now briefly discuss a recent result of Blum and Kalai [1997] giving the competitiveness of the uniform weighted algorithm (algorithm UNI of Section 5.8.2) for a model with transaction costs.

We focus on the following, more common, *proportional transaction-cost* model.<sup>15</sup> In this model we charge a fixed percentage  $\gamma$  of each amount transacted. We call  $\gamma$  the *commission rate*.

Consider the behavior of a constant rebalanced algorithm  $\text{CBAL}_{\bar{b}}$  in a market with commission rate  $\gamma$ . As before,  $\text{CBAL}_{\bar{b}}$  rebalances its holdings so that its portfolio is  $\bar{b}$ -balanced at the start of each trading period. Let  $\bar{b} = (b_1, \dots, b_m) \neq \bar{b}' = (b'_1, \dots, b'_m)$ . If at the end of some period the portfolio of  $\text{CBAL}_{\bar{b}}$  is  $\bar{b}'$ -balanced, then, in order to rebalance, for all  $j = 1, 2, \dots, m$ ,  $\text{CBAL}_{\bar{b}}$  moves a fraction  $|b'_j - b_j|$  of its wealth in or out of the  $j$ th security. It follows that the total (per dollar) transaction fee to rebalance is  $\gamma \sum_{j=1}^m |b'_j - b_j|$ .

Let  $\text{UNI}_{\gamma}$  be the uniform-weighted algorithm that pays commissions at rate  $\gamma$ . Blum and Kalai proved the surprising result that in a market with commission rate  $\gamma$ , for any market sequence  $X$  of

length  $n$ ,

$$\begin{aligned} \frac{\text{CBAL-OPT}(X)}{\text{UNI}_{\gamma}(X)} &\leq \left( \frac{(1 + 2\gamma)n + m - 1}{m - 1} \right) \\ &\leq ((1 + 2\gamma)n + 1)^{m-1}. \end{aligned} \tag{26}$$

Starting with the representation (20), the proof of this result, given by Blum and Kalai, is different and somewhat more direct than Cover and Ordentlich's proof (presented in Section 5.8.2).

**5.8.8 Other Results.** There are quite a few other works concerning online portfolio selection that are not discussed in this survey. Here we briefly mention several other known results. Cover and Gluss [1986] consider a model where the set of possible market vectors of price relatives is finite, say, with cardinality  $k$ . Using the game-theoretic approachability-excludability theorem of Blackwell [1956], they obtain an online portfolio-selection algorithm whose exponential growth rate approaches that of the optimal offline constant rebalanced algorithm at (convergence) rate  $(L\sqrt{k} + 1)\sqrt{2L^2 + 1}/\sqrt{n}$ , where  $L$  is a bound on the logarithm of the maximum price relative and  $n$  is the length of the game.<sup>16</sup> Thus in the long run it is possible in this model to almost track the performance of the optimal offline constant rebalanced algorithm and in particular to outperform the optimal offline BAH strategy.

Cover and Ordentlich's results regarding the  $\mu$ -weighted algorithms are a generalization of a result by Cover [1991] that the exponential growth rate of the uniform-weighted portfolio approaches, as  $n$  grows, the exponential growth rate of the optimal offline constant rebalanced portfolio. This result assumes that all price relatives are bounded away from zero. In the same paper Cover presents a more refined

<sup>15</sup> The alternative model of fixed transaction costs has not been studied much.

<sup>16</sup> Similar results can be obtained using the results of the recent paper by Hart and Mass-Collel [1997].

analysis and bounds the competitive ratio of the uniform-weighted algorithm (with respect to CBAL) in terms of a sensitivity matrix  $A$  measuring the “empirical volatility” (defined in the paper) of prices exhibited in the market sequence presented to the algorithm. In particular, the upper bound on the competitive ratio is  $\sqrt{|A|(n/(2\pi))^{(m-1)/2}/(m-1)!}$ . The value of the determinant  $|A|$  can be bounded by a constant if the prices of all securities are sufficiently volatile so that CBAL gives positive weights to all securities. Cover also provides some interesting experimental results demonstrating the performance of this algorithm. Jamshidian [1992] analyzed the performance of the uniform-weighted algorithm in a continuous-time model. The results obtained are similar to those obtained by Cover [1991].

Following Cover and Ordentlich, Helmbold et al. [1996] presented an online portfolio selection algorithm based on techniques from statistical inference. This algorithm chooses the next  $(i+1)$ st portfolio,  $\tilde{b}_{i+1}$ , to maximize  $\eta \log(\tilde{b}_{i+1}^t \cdot \tilde{x}_i) - D(\tilde{b}_{i+1} \| \tilde{b}_i)$ , where  $\eta$  is some constant that determines the adaptation rate and  $D(\cdot \| \cdot)$  is the Kullback–Leibler dissimilarity measure.<sup>17</sup> The algorithm thus adapts to the recent history while keeping its next portfolio similar to the previous portfolio. Helmbold et al. present several computationally efficient approximation variants of this algorithm for the cases where  $n$  is known or unknown and for a model with side information. All their bounds are comparable but inferior to the respective bounds obtained by Cover and Ordentlich. Nevertheless, it is shown that this algorithm empirically outperforms the uniform-weighted and Dirichlet-weighted algorithms on data sets consisting of several NYSE stock price sequences.<sup>18</sup> Given this empirical evi-

dence and the fact that the time and space requirements of an approximation of this algorithm grow linearly in the number of stocks, this algorithm seems very attractive, despite its inferior theoretical guarantees.

Blum and Kalai [1997] report on experiments with the data sets of Cover (and Helmbold et al.) to test the performance of the uniform-weighted algorithm with transaction costs. On all data sets checked, the uniform-weighted algorithm gave inferior returns to the optimal offline constant rebalanced algorithm but still beat the market (i.e., the average stock) when the commission factor was less than 2%. On these experiments the uniform-weighted algorithm rebalanced monthly (thus ignoring most price ticks that occurred daily). Not surprisingly, it was found that rebalancing less frequently is beneficial when commission factors are high.

Awerbuch et al. [1996] consider the following setting. The online player wishes to invest his entire wealth in one of the  $m$  securities, hoping that the chosen security will be a “winner” that yields high dividends. The decision is irreversible and after the player has chosen one security the game is essentially over. If  $D$  is the posteriori dividend return of the best security, then the optimal expected return of the player is trivially  $\Theta(D/m)$ . Now consider a game such that in each period, each of the securities may issue a dividend of exactly \$1. Under the assumptions that the yield of the best security is  $D \geq 3 \log m$  and that  $D$  is known to the online player, Awerbuch et al. provide a selection strategy obtaining a return of at least  $D/3 \log m$  with probability at least  $1 - (3 \log m)/D - 2/m$ . The strategy is the following. For each security  $j$ ,  $j = 1, 2, \dots, m$ , set  $d(j, i)$  to be the cumulative number of dividends issued by the  $j$ th security within the first  $i$  periods. At the  $(i+1)$ st period choose the  $j$ th security with probability  $m^{3d(j,i)/D-2}$ . This basic result is extended for multiple choices of securities. Finally, in the case where  $D$  is not known in advance,

<sup>17</sup> For two  $m$ -ary probability vectors  $\tilde{u}$  and  $\tilde{v}$   $D(\tilde{u} \| \tilde{v}) = \sum_{i=1}^m u_i \log u_i/v_i$ .

<sup>18</sup> These data sets were first used by Cover [1991] to evaluate the uniform-weighted algorithm.

the player can retain the same yield but the probability of success is decreased.

Auer et al. [1995] consider the following adversarial variant of the *multi-armed bandit* problem. The online player must repeatedly choose one among  $m$  slot machines. The player's goal is to maximize the total reward in a sequence of  $n$  trials (no sampling costs are assumed). Using the minimax regret adversary, who before each round selects an  $m$ -ary vector of the current rewards, and assuming that all individual rewards are in  $[a, b]$ , Auer et al. prove that the expected regret (obtained by a randomized algorithm) is at most  $O((b - a)n^{2/3}(m \log m)^{1/3})$ . They also prove a lower bound of  $\Omega(\sqrt{nm})$  on the regret of any algorithm.

## 6. COMPETITIVE RISK MANAGEMENT

One aspect that distinguishes financial decision making under uncertainty is the need of financial players to understand and control the risk inherent in their activities. This requirement to manage risk is common to all financial applications regardless of the markets involved, the instruments traded, and strategies employed. To manage the risk associated with decision making under uncertainty means to control the inherent tradeoff between risk and reward. The premise here is that investors require more potential reward for higher levels of exposure to risk.

The definitions of risk and reward and the interplay between them depend, first of all, upon the particular optimization model used. When we fix our optimization model, which typically fixes the definition of reward, there are quite a few reasonable definitions for risk. In general, it is widely accepted that risk should measure something like the "exposure" to (chance of) loss (see, e.g., MacCrimmon and Wehrung [1986]). Consider, for example, the classical Rothschild–Stiglitz [1970a, 1971] risk-reward model. In this model, the return of a strategy is given by a random variable  $R$  and the associated re-

ward is given by its expected value  $E[R]$  under some probabilistic assumptions. Now consider two returns (random variables)  $R_1$  and  $R_2$  with equal expectations. Let their distribution functions be  $F_1$  and  $F_2$ , respectively. In the Rothschild–Stiglitz model, the return  $R_1$  is less risky than  $R_2$  if and only if

$$\int_{-\infty}^t (F_2(x) - F_1(x))dx \geq 0, \quad \text{for all } t. \quad (27)$$

Intuitively this means that  $F_2$  has more weight in its tails.<sup>19</sup> This simple model is often approximated by measuring the risk associated with a given return via its variance (or standard deviation) (see Bodie et al. [1993]). There are of course other, more elaborate, measures of risk (see, e.g., Merton [1981]).

An underlying ingredient of any optimization model that supports a measure of risk is the notion of a *forecast*. For example, in the preceding Rothschild–Stiglitz model, the forecast is of course the probability distribution of the return. The forecast ingredient is by definition completely lacking in the pure competitive analysis framework, which gives no flexibility to control risk and reward. The resulting optimal competitive algorithms are extremely risk-averse since they avoid making assumptions about future inputs. It is not surprising then that the empirical rewards of purely competitive algorithms can sometimes be low (see, e.g., Bachrach and El-Yaniv [1997]). In the statistical adversary approach (see Section 5.5) we do take risks by assuming that markets will obey some constraints but even this approach offers no mechanism to quantify and control the tradeoff between risk and reward.

Recognizing the need for risk manage-

<sup>19</sup> This measure of risk is directly tied to von Neumann–Morgenstern [1994] expected utility theory. Specifically, under the assumption that  $E[R_1] = E[R_2]$ , it can be shown that if  $U$  is a concave (utility) function then condition (27) holds if and only if  $E[U(R_1)] \geq E[U(R_2)]$ .

ment in financial applications, al-Binali [1997] offers the following framework that generalizes competitive analysis and allows for flexible risk management. The framework extends pure competitive analysis by introducing two ingredients: risk and forecast. Fix an online problem. For any algorithm ALG for this problem, denote by  $\mathcal{C}(\text{ALG})$  the competitive ratio of ALG. Let  $c^* = \inf_{\text{ALG}} \mathcal{C}(\text{ALG})$  be the optimal competitive ratio for this problem. Given any (online) algorithm ALG, define the risk  $\mathcal{R}(\text{ALG})$  of ALG to be  $\mathcal{R}(\text{ALG}) = \mathcal{C}(\text{ALG})/c^*$ . Thus the risk of any algorithm is  $\geq 1$  and the lower the risk, the better its performance guarantee. Next, define a forecast as any subset  $F$  of the allowable input sequences. For example, the set of market sequences allowable for the  $(\alpha, n, p)$ -adversary of Section 5.7.2 is a forecast.

The online player specifies a risk tolerance  $T$ . This means that the player is willing to use only algorithms from the set  $\mathcal{T} = \{\text{ALG} : \mathcal{R}(\text{ALG}) \leq T\}$ . Each one of the algorithms in  $\mathcal{T}$  thus has a competitive ratio  $\leq Tc^*$ . Fix any forecast  $F$ . An optimal algorithm, according to this framework, is an algorithm from  $\mathcal{T}$  that minimizes the competitive ratio, restricted to input sequences from  $F$ . Formally, we extend the function  $\mathcal{C}(\cdot)$  to be parameterized by any subset  $F \subseteq \mathcal{I}^*$  of input sequences such that (say, with respect to a profit maximization problem)

$$\mathcal{C}_F(\text{ALG}^*) = \sup_{\sigma \in F} \frac{\text{OPT}(\sigma)}{\text{ALG}(\sigma)}.$$

Thus an optimal  $T$ -tolerant algorithm  $\text{ALG}^*$  (for a player with a risk tolerance  $T$ ) with respect to a forecast  $F$  satisfies

$$\mathcal{C}_F(\text{ALG}^*) = \inf_{\text{ALG}} \{\mathcal{C}_F(\text{ALG}) : \text{ALG} \in \mathcal{T}\}.$$

Clearly this simple risk-reward framework provides a smooth generalization of the pure competitive analysis framework, which is the special case obtained with risk tolerance 1. It is

hoped that by posing reasonable forecasts the player can boost performance significantly as long as input sequences conform to the forecast. Nevertheless, regardless of forecast, the player always keeps the risk within the desired tolerance. In the following section we present a concrete example.

### 6.1 One-Way Trading Revisited: Analysis with Risk Management

To demonstrate the utility of the preceding risk management framework, al-Binali [1997] analyzes the one-way trading problem of Section 3.2.3 within the preceding risk-management framework. Here we briefly describe some of results.

Assume the one-way trading problem variant with known duration  $n$ , and known  $m$  and  $M$  (see Section 3.4.1). Fix any  $\Delta \in [0, M - m]$  and consider the forecast  $F$ : the exchange rate will increase to at least  $m + \Delta$  within the trading period. That is,

$$F = \{(p_1, p_2, \dots, p_n) : \exists i \text{ s.t. } p_i \geq m + \Delta\}.$$

Now fix a risk tolerance  $T$ . It can be shown that the optimal  $T$ -tolerant algorithm is the following two-stage threat-based algorithm  $\text{THREAT}_T$ . In the first, conservative stage,  $\text{THREAT}_T$  operates under the threat that the prediction is incorrect and attempts to achieve a competitive ratio of  $Tc^*$ . In particular, during this stage,  $\text{THREAT}_T$  converts just enough dollars to guarantee this (non-optimal) competitive ratio while saving resources, as much as possible, for the case where the forecast will come true. Then, if and when the forecast comes true (i.e., the exchange rate swings above  $m + \Delta$ ), the algorithm begins its second stage. In this stage, the algorithm first computes the optimal restricted ratio  $\inf_{\text{ALG}} \mathcal{C}_F(\text{ALG})$  and now trades, using this restricted ratio, under the threat that the exchange rate drops to  $m$  and remains there for the rest of the game.

**Table III.** Restricted Competitive Ratio  $c_F^*$ <sup>†</sup>

$T$	$j$	$c_F^*$
1.01	10	1.140
1.01	20	1.143
1.05	10	1.114
1.05	20	1.115
1.1	10	1.0979
1.1	20	1.0975

<sup>†</sup> Obtained for the forecast  $F$  with  $\Delta = 0.35$ ,  $m = 1$ ,  $M = 1.5$ , and  $n = 30$ , for tolerances  $T = 1.01$ ,  $1.05$ ,  $1.1$ , and  $j = 10, 20$  (note that for this game the (pure) competitive ratio is 1.154).

Based on the analysis described in Section 3.4.1 it can be shown that during its first stage (the forecast  $F$  has not been realized yet) algorithm  $\text{THREAT}_T$  trades only when the exchange rate hits a new high  $p_i$  at which time it converts  $s_i/T$  dollars to yen, where  $s_i$  is given by (2). Then immediately when for some  $j \leq n$ ,  $p_j \geq m + \Delta$ , the algorithm enters its second stage. Now the algorithm computes the optimal competitive ratio  $c_T^*$  restricted to the forecast  $F$  which is given by

$$c_F^* = \frac{\alpha_1 T c^* (\Delta(\alpha_2 j + n - \alpha_2 - j - \alpha_2 n) - \alpha_2 m)}{\alpha_2 (\Delta(1 - \alpha_1 - j + \alpha_1 j - \alpha_1 T c^*) - \alpha_1 m + \alpha_1^j m (1 - T c^*))},$$

where  $c^* = c_n^*(m, M)$  is given by (4) and

$$\alpha_1 = \left( \frac{\Delta}{m(Tc^* - 1)} \right)^{1/j}$$

$$\alpha_2 = \left( \frac{M - m}{\Delta} \right)^{1/(n-j)}.$$

Using  $c_T^*$ ,  $\text{THREAT}_T$  trades  $s_j = 1/(p_j - m)$  ( $p_j/c_T^*$ ) - ( $(p_{j-1})/Tc_T^*$ ) dollars in the  $j$ th trading period, and  $s_i = (p_i - p_{i-1})/c_T^*(p_i - m)$  dollars in subsequent periods  $i > j$ .

Consider Table III where we give some numerical examples of the restricted competitive ratio obtained for a thirty-day game with  $m = 1$ ,  $M = 1.5$ . Note that in this case the pure competitive ratio is 1.154. For a forecast  $F$  with

$\Delta = 0.35$  that is realized in the  $j$ th period,  $j = 10, 20$ , the table gives the restricted competitive ratio obtained for tolerances  $T = 1.01, 1.05, 1.1$ .

## 7. CONCLUDING REMARKS AND DIRECTIONS FOR FUTURE RESEARCH

The competitive approach is productive for a variety of financial problems. Although the full extent of its theoretical value and its empirical relevance are yet to be determined, it is already quite clear, from the examples in this survey, that this approach gives rise to interesting and nonobvious algorithms and analyses.

At the time of this writing, the work presented here is the state of the art in this area. Therefore, almost any improvement on the stated bounds and any extensions of the models considered will be desirable advances. Due to the nature of financial applications, one somewhat urgent issue that requires further attention is the power of the risk-reward framework of al-Binali (see Section 6). In particular, it would be important to consider some of the financial problems presented here in this framework while identifying realistic classes of forecasts.

Several other attractive financial problems can and should be studied within the competitive analysis framework. We conclude this survey with formulations of three online financial problems that to the best of our knowledge have not yet been studied using the competitive approach (variants of all these problems have been extensively studied within the Bayesian approach; see the following references).

—*Inventory management.* The theory of online inventory management is concerned with decision problems of when and how much to buy of certain commodities whose future demand, for production or trade, is uncertain. For concreteness, consider the following problem formulation. The online player runs a store that sells a certain

commodity. The player maintains an inventory in which he stores some items of this commodity. During each period a demand for some number of items of this commodity is placed. For each item sold the player earns some price  $c$  per item. At the end of each period the player must decide on a number of items to order so he can replenish his inventory. For each item ordered he pays some wholesale price  $w$ . The items ordered arrive after a delay of  $d$  periods. For each item stored in the inventory the player pays, per period, some holding cost  $h$ . The player's objective is to issue orders so that the total profit is maximized.

The literature on Bayesian inventory management is extensive. The reader is referred to Scarf [1963] and Stokey and Lucas [1989]. We note that preliminary competitive results for this problem were obtained by Karp et al. [1993].

—*Insurance problems.* In insurance problems the online player must decide whether to purchase an insurance contract guaranteeing that the insurer will pay the player a certain amount of money provided that a stipulated event occurs. For instance, consider the following elementary one-stage two-state insurance problem. There are two possible states of nature, one of which will prevail. In the first state the player is endowed with wealth  $w_1$  whereas in the second, disaster state, the player is endowed with wealth  $w_2$ ,  $w_2 < w_1$ . Before the state of nature is made known, the player can guarantee a compensation of  $c$  dollars, in the case where the second state of nature prevails, by paying the insurance company an amount of  $\beta_c$ . The amounts  $w_i$  and  $\beta$  are fixed and known to the online player before the decision is made. The player's goal is to maximize her wealth. This elementary one-stage two-state formulation can be generalized to a multi-

stage multiple-state insurance game. Insurance problems are most fundamental to the economics of uncertainty. For Bayesian solutions to this problem the reader is referred to Lippman and McCall [1981] and references therein.

—*Consumption problems.* In consumption problems the online player must repeatedly decide how much of her current wealth to consume and how much to save for future consumption. Her goal is to maximize her total lifetime consumption. The uncertainties in this problem may arise in one or more of the following ways. First, the player's future income may vary in some unknown manner. Second, future investment appreciation (depreciation) rates may fluctuate in some uncertain way. Finally, the player's lifespan is unknown. Consider the following basic consumption problem. The player starts with some initial wealth  $w_0$ . At the start of the  $j$ th period,  $j = 1, 2, \dots$ , the player receives an income  $s_j$  and must choose some amount  $c$  from her current wealth  $s_j + w_{j-1}$  that she immediately consumes. The rest of the money is allocated for investment and grows at rate  $1 + i_j$  throughout the  $j$ th period, so that the player's wealth at the end of the  $j$ th period is  $w_j = (1 + i_j)(s_j + w_{j-1} - c)$ . At the end of some period  $T$ , unknown in advance, the game ends. The player's goal is to maximize her total consumption throughout the game.

For some Bayesian results concerning variants of this problem, the reader is referred to Levhari and Mirman [1977], Mirman [1971], and Lippman and McCall [1981].

#### ACKNOWLEDGMENTS

My thanks to Thomas Cover and Erik Ordentlich for helping to obtain the bibliographic material. I thank Allan Borodin, Andrew Chou, Vincent Felkamp, and Erik Ordentlich for many discussions and comments that helped me understand the material and greatly improved the presentation. I also thank Susanne Albers, Robert Aumann, Dean

Foster, Sergiu Hart, Sageev Oore, Motty Perry, Steve Ponzio, Adi Rosén, and Asher Wolinsky for their useful remarks. Finally, I thank Brenda Brown, who proofread several versions of this article.

## REFERENCES

- AJTAL, M., MEGIDDO, N., AND WAARTS, O. 1995. Improved algorithms and analysis for secretary problems and generalizations. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 473–482.
- AL BINALI, S. 1997. The competitive analysis of risk taking with application to online trading. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*.
- AUER, P., CESA-BIANCHI, N., FREUND, Y., AND SCHAPIRE, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, 322–331.
- AWERBUCH, B., AZAR, Y., FIAT, A., AND LEIGHTON, T. 1996. Making commitments in the face of uncertainty: How to pick a winner almost every time. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*.
- AZAR, Y., BARTAL, Y., FEUERSTEIN, E., FIAT, A., LEONARDI, S., AND ROSEN, A. 1996. On capital investment. In *Proceedings of the 23rd ICALP*, 429–441.
- BACHRACH, R. AND EL-YANIV, R. 1997. Online list accessing algorithms and their applications: Recent empirical evidence. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, 53–62.
- BELLMAN, R. 1955. Equipment replacement policy. *J. Soc. Ind. Appl. Math.* 3, 133–136.
- BEN-DAVID, S., BORODIN, A., KARP, R., TARDOS, G., AND WIDGERSON, A. 1990. On the power of randomization in on-line algorithms. In *Proceedings of the 22nd Symposium on Theory of Algorithms*, 379–386. Also in *Algorithmica* 11 (1994), 2–14.
- BLACKWELL, D. 1956. An analog of the minimax theorem for vector payoffs. *Pacific J. Math.* 6, 1–8.
- BLUM, A. AND KALAI, A. 1997. Universal portfolios with and without transaction costs. In *Tenth Annual Conference on Computational Learning Theory*.
- BODIE, Z., KANE, A., AND MARCUS, A. J. 1993. *Investments*. Irwin.
- BORODIN, A., LINIAL, N., AND SAKS, M. 1992. An optimal online algorithm for metrical task systems. *J. ACM* 39, 745–763.
- BREMAN, L. 1961. Optimal gambling systems for favorable games. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, Vol. 1. University of California Press, Berkeley, 65–78.
- CHOU, A. 1994. Optimal trading strategies vs. a statistical adversary. Master's Thesis, Massachusetts Institute of Technology.
- CHOU, A., COOPERSTOCK, J., EL-YANIV, R., KLUGERMAN, M., AND LEIGHTON, T. 1995a. The statistical adversary allows optimal money-making trading strategies. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*.
- CHOU, A., SHRIVASTAVA, A., AND SIDNEY, R. 1995b. On the power of magnitude. (To be published).
- CHOW, Y. S., ROBBINS, H., AND SIEGMUND, D. 1971. *The Theory of Optimal Stopping*. Dover Publications.
- COVER, T. M. 1991. Universal portfolios. *Math. Finance* 1, 1 (Jan.), 1–29.
- COVER, T. M. AND GLUSS, D. H. 1986. Empirical Bayes stock market portfolios. *Adv. Appl. Math.* 7, 170–181.
- COVER, T. M. AND ORDENTLICH, E. 1996. Universal portfolios with side information. *IEEE Trans. Inf. Theor.* (March).
- COVER, T. M. AND THOMAS, J. A. 1991. *Elements of Information Theory*. John Wiley, New York.
- COX, J. AND RUBINSTEIN, M. 1985. *Options Markets*. Prentice-Hall, Englewood Cliffs, NJ.
- DERMAN, C. 1963. On optimal replacement rules when changes of the state are Markovian. In *Mathematical Optimization Techniques*, R. Bellman, Ed.
- EL-YANIV, R. 1996. On the decision theoretic foundations of the competitive ratio. (To be published).
- EL-YANIV, R., FIAT, A., KARP, R., AND TURPIN, G. 1992. Competitive analysis of financial games. In *Proceedings of the 33rd Symposium on Foundations of Computer Science*, 327–333.
- EL-YANIV, R. AND KARP, R. M. 1997. Nearly optimal competitive online replacement policies. *Math. Oper. Res.* 22, 4, 814–839.
- EL-YANIV, R., KANIEL, R., AND LINIAL, N. 1993. On the equipment rental problem. (To be published).
- FREEMAN, P. R. 1983. The secretary problem and its extensions. *Int. Stat. Rev.* 51, 189–206.
- GRAHAM, R. L. 1966. Bounds for certain multiprocessor anomalies. *Bell Syst. Tech. J.* 45, 1563–1581.
- HART, S. AND MAS-COLELL, A. 1997. A simple adaptive procedure leading to correlated equilibrium. The Hebrew University of Jerusalem. Discussion paper 120. Jan.
- HELMBOLD, D. P., SCHAPIRE, R. E., SINGER, Y., AND WARMUTH, M. K. 1996. On-line portfolio selection using multiplicative updates. In *Ma-*

- chine Learning: Proceedings of the 13th International Conference, 243–251.
- HULL, J. 1993. *Options, Futures, and Other Derivative Securities*. Prentice-Hall, Englewood Cliffs, NJ.
- IRANI, S. AND RAMANATHAN, D. 1994. The problem of renting versus buying. (Unpublished manuscript).
- JAMSHIDIAN, F. 1992. Asymptotically optimal portfolios. *Math. Fin.* 2, 2, 131–150.
- JOHNSON, D. S. 1973. *Near-Optimal Bin Packing Algorithms*. Ph.D. Thesis, Massachusetts Institute of Technology.
- JOHNSON, D. S., DEMERS, A., ULLMAN, J. D., GAREY, M. R., AND GRAHAM, R. L. 1974. Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM J. Comput.* 3, 299–325.
- KARLIN, A. R., MANASSE, M. S., RUDOLPH, L., AND SLEATOR, D. D. 1988. Competitive snoopy caching. *Algorithmica* 3, 1, 70–119.
- KARP, R. M., OSTROVSKI, R., AND RABANI, Y. 1993. Personal communication.
- KUHN, H. W. 1953. Extensive games and the problem of information. In *Contribution to the Theory of Games*, Vol. II, H. W. Kuhn and A. W. Tucker, Eds., Princeton University Press, Princeton, NJ, 193–216.
- LEVHARI, D. AND MIRMAN, L. J. 1977. Saving and uncertainty with an uncertain horizon. *J. Political Econ.* 85, 265–281.
- LEVIN, L. 1994. Personal communication, July.
- LIPPMAN, S. A. AND MCCALL, J. J. 1976. The economics of job search: A survey. *Econ. Inquiry* (June), 155–189.
- LIPPMAN, S. A. AND MCCALL, J. J. 1981. The economics of uncertainty: Selected topics and probabilistic methods. In *Handbook of Mathematical Economics*, Vol. 1, K. J. Arrow and M. D. Intriligator, Eds. Chapter 6, North-Holland, 211–284.
- MACCRIMMON, K. R. AND WEHRUNG, D. A. 1986. *Taking Risks: The Management of Uncertainty*. The Free Press.
- MERTON, R. C. 1981. On the microeconomic theory of investment under uncertainty. In *Handbook of Mathematical Economics*, Vol. 2, K. J. Arrow and M. D. Intriligator, Eds., Chapter 13, North-Holland, 601–669.
- MIRMAN, L. J. 1971. Uncertainty and optimal consumption decisions. *Econometrica* 39, 179–185.
- ORDENTLICH, E. AND COVER, T. M. 1996. On-line portfolio selection. In *Proceedings of COLT 96*. (To appear in *Math. Oper. Res.*).
- RAGHAVAN, P. 1992. A statistical adversary for on-line algorithms. *DIMACS Series in Discrete Math. Theor. Comput. Sci.* 7, 79–83.
- ROSENFELD, D. B. AND SHAPIRO, R. D. 1981. Optimal price search with Bayesian extensions. *J. Econ. Theor.*
- ROTHSCHILD, M. AND STIGLITZ, J. E. 1970. Increasing risk I: A definition. *J. Econ. Theor.* 82, 225–243.
- ROTHSCHILD, M. AND STIGLITZ, J. E. 1971. Increasing risk II: Its economic consequences. *J. Econ. Theor.* 3, 66–84.
- SCARF, H. E. 1963. A survey of analytic techniques in inventory theory. In *Multistage Inventory Models and Techniques*. H. E. Scarf, D. M. Gilford, and M. W. Shelly, Eds., Chapter 7, Stanford University Press, Stanford, CA, 185–225.
- SHARPE, W. F. 1966. Mutual fund performance. *J. Business* (Jan.).
- SHESHINSKI, E. AND WEISS, Y. 1993. Optimum pricing policy under stochastic inflation. In *Optimal Pricing, Inflation and the Cost of Price Adjustment*, E. Sheshinski and Y. Weiss, eds., The MIT Press, Cambridge, MA.
- SHILLING, A. G. 1992. Market timing: Better than a buy-and-hold strategy. *Finan. Anal. J.* (March–April), 46–50.
- SLEATOR, D. AND TARJAN, R. E. 1985. Amortized efficiency of list update and paging rules. *Commun. ACM* 28, 202–208.
- STOKEY, N. L. AND LUCAS, R. E., JR. 1989. *Recursive Methods in Economic Dynamics*, Chapter 13, Harvard University Press, Cambridge, MA, 389–413.
- VON NEUMANN, J. AND MORGENSTERN, O. 1944. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ.
- YAO, A. C. 1980. New algorithms for bin packing. *J. ACM* 27, 207–227.

Received November 1996; revised August 1997; accepted November 1997