Information and Computation **150**, 244–267 (1999)
Article ID inco.1998.2775, available online at http://www.idealibrary.com on IDEAL®

On Randomization in On-Line Computation

Allan Borodin

Department of Computer Science, University of Toronto, Canada
E-mail: bor@cs.toronto.edu

and

Ran El-Yaniv

Department of Computer Science, Technion–Israel Institute of Technology, Israel
E-mail: rani@cs.technion.ac.il

This paper concerns two fundamental but somewhat neglected issues, both related to the design and analysis of randomized on-line algorithms. Motivated by early results in game theory we define several types of randomized on-line algorithms, discuss known conditions for their equivalence, and give a natural example distinguishing between two kinds of randomizations. In particular, we show that *mixed* randomized memoryless paging algorithms can achieve strictly better competitive performance than *behavioral* randomized algorithms. Next we summarize known—and derive new—''Yao principle'' theorems for lower bounding competitive ratios of randomized on-line algorithms. This leads to four different theorems for bounded/unbounded and minimization/maximization problems.

1. INTRODUCTION

This paper studies two issues related to randomized on-line algorithms. In particular, we will consider these issues in the context of competitive analysis but the same issues are relevant for other criteria that measure the performance of on-line algorithms. The first issue concerns models for randomized algorithms. Informally, we conceptually think of randomized algorithms (especially for upper bounds) that toss coins (say integer or 0–1 valued)[1] and then based on these coin tosses and the present "state" of the computation, take various actions (so as to process the present request) and change "state." This informal description can be formalized in terms of computation trees. However, it is often common (especially

[1] The formulation that follows is set in terms of finite games and discrete (finite) distributions. Nevertheless, all the stated definitions and theorems can be formulated in terms of infinite games with continuous distributions (see Remark 2).

0890-5401/99 $30.00

for lower bounds) in the analysis of randomized algorithms to define randomized algorithms as probability distributions on deterministic algorithms. In particular, we often adopt this latter view in order to apply the Yao principle, which allows us to infer a lower bound for randomized algorithms by considering input distributions for deterministic algorithms. Needless to say, matching an upper bound, based on one model, of randomization with a lower bound based on a different model is only meaningful if the lower bound model is at least as powerful as the upper bound model. We shall see that there are natural situations where this is *not* the case. Hence we need a more careful study of the relative power of models for randomized algorithms.

The second issue we consider is the application of the Yao principle in the context of randomized on-line algorithms. Online algorithms can introduce some subtlety in applying the Yao principle for several reasons. First, there is a natural ambiguity in how to define the competitive ratio for a randomized algorithm. Second, unlike traditional off-line optimization problems, we often think of on-line problems as having an infinite sequence of requests and, moreover, there may be an infinite number of actions that an on-line algorithm may take to process a given request. Finally, on-line problems can be either cost minimization or profit maximization problems with bounded or unbounded costs/profits. We shall see that some care is needed to accommodate these differences.

Section 2 begins with a review of some basic game theoretic definitions and results that are relevant to the foundations of competitive analysis. In particular, we formalize the models discussed above as behavioral and mixed strategies. In Section 3 we consider the important case of games of perfect recall for which there is a known equivalence between behavioral and randomized strategies. However, Section 5 demonstrates that there can be a difference between behavioral and mixed algorithms (in particular, that mixed algorithms can sometimes be more powerful) and we illustrate this by considering "memoryless" algorithms for paging. Section 6 provides a variety of ways one can apply the Yao principle in the context of competitive analysis, including one example (due to Kirk Pruhs) on how the principle can be improperly applied.

2. RANDOMIZED STRATEGIES: MIXED, BEHAVIORAL AND GENERAL

In what follows we assume familiarity with the basic game-theoretic definitions of games in extensive and strategic forms. For the reader's convenience we include a review of the basic game theoretic concepts in Appendix B. For simplicity in this section we develop the game theoretic concepts for finite games. The generalization of these concepts to infinite (enumerable) games are straightforward.

Mixed strategies. Let Γ be a finite n-person game (either in extensive or strategic form). A *mixed strategy* x^i for a player i is a probability distribution over the set of his pure strategies S^i. For each pure strategy $s \in S^i$ we denote by $x^i(s)$ the probability to choose s (that is, $\sum_{s \in S^i} x^i(s) = 1$). The interpretation is that the player uses some randomizing device before the start of the game and chooses one pure strategy according to this probability distribution. After that the player's actions are completely deterministic. Given a mixed strategy profile $\mathbf{x} = (x^1, x^2, ..., x^n)$, where x^i is the mixed strategy of player i, the expected payoff vector $H(\mathbf{x})$ for the n players, is given by $H(\mathbf{x}) = \sum_{\mathbf{s}} \mathbf{x}(\mathbf{s})\, H(\mathbf{s})$, where $\mathbf{s} = (s^1, s^2, ..., s^n)$ is any pure strategy profile, $H(\mathbf{s})$ is the vector payoff associated with $\mathbf{s}$ (see Appendix B.2), and $\mathbf{x}(\mathbf{s}) = \prod_{i \in N} x^i(s^i)$.

Behavioral strategies. Let Γ be a finite n-person game in extensive form. An alternative way to randomize is to make an independent random choice at each decision point (i.e., at each information set). Such a randomized strategy is called "behavioral." For complicated games, and also from an algorithmic viewpoint, behavioral strategies often appear to be more "natural." Nevertheless, for analysis one would often like to deal with mixed strategies.

Clearly, when defining a behavioral strategy, we must take into consideration the constraints given by the information sets. Since a player cannot distinguish between any two nodes in the same information set, he must use the same randomizing device (i.e., probability distribution) for all nodes in the same information set. Formally, a behavioral strategy b^i for player i is a k_i-tuple $(b^i_1, b^i_2, ..., b^i_{k_i})$ of probability distributions, where k_i is the number of his information sets, $U^i_1, U^i_2, ..., U^i_{k_i}$. The probability distribution b^i_j is over the set $D(U^i_j) = \{1, 2, ..., d^i_j\}$ of decision alternatives in each node in the jth information set, U^i_j (i.e., $D(U^i_j)$ is the set of labels of edges outgoing from each node in the jth information set of player i). Given a behavioral strategy profile $\mathbf{b} = (b^1, b^2, ..., b^n)$, where b^i is the behavioral strategy chosen by player i, we let $p(\ell \mid \mathbf{b})$ denote the conditional probability that leaf ℓ is reached given $\mathbf{b}$. Then, define the expected payoff vector,

$$H(\mathbf{b}) = \sum_{\text{leaf } \ell} p(\ell \mid \mathbf{b})\, H(\ell).$$

In the context of (off-line) complexity theory it is easy to prove an equivalence, in terms of computational power, between behavioral and mixed algorithms. That is, in the case of worst case analysis of off-line problems whenever there are no memory restrictions on the machine (games of complete information), there is a complete equivalence between two types of randomized algorithms.[2] With regards to games of incomplete information, the situation is similar. Here again, whenever there are no memory restrictions, the two types of randomized are equivalent. However, under memory restrictions the equivalence does not hold anymore and in fact (as we shall see) there is even a third (typically neglected!) type of randomization called *general strategies* (see Section 2.1). Therefore, one cannot simply assume that all representations of randomized algorithms are equivalent for on-line algorithms. Consider the following example.[3]

[2] It is well known that this equivalence need not hold in off-line computational models when there are memory restrictions. See a negative example given for the "WAG" model by Beame *et al.* [BRR+90] and a discussion in [Poo96]

[3] This example is known in the game-theoretic folklore. The particular interpretation here is borrowed from [PR94] (see also [AHP96]).

EXAMPLE 1 (The absent-minded driver). Consider a person sitting late at night in a bar after a long evening of drinking (he is currently sober but a little bit absent-minded). In order to return home (payoff 1) he has to take the highway and then get off using the second exit. Getting off at the first exit or continuing on the highway after the second exit will lead to dangerous areas with a risk of being killed (payoff 0). The driver knows that he is absent-minded and realizes, still in the bar, that he would not be able to distinguish between the first and second exits because they look exactly the same. What would be the optimal strategy for this person?

As depicted in Fig. 1 (Appendix A), this problem is easily formulated as a one-person game in extensive form. Clearly, this is a game of incomplete information as the two decision nodes, a and b, are in the same information set. This means that the player cannot distinguish between them. The set of pure strategies for the player is $\{L, R\}$ (L for "leave" the highway and R for "remain") and any probability mixture of them yields an expected payoff of 0. In contrast, a behavioral strategy that chooses L or R with probability 1/2 independently, at each decision node has an expected payoff of 1/4. Hence, in this example a behavioral strategy is strictly superior to mixed strategy.

It is not hard to construct examples that show the converse situation where mixed strategies are superior to behavioral (we shall present one later).

2.1. General Strategies

There is yet another class of randomized strategies. Let Γ be a finite n-person game in extensive form. A *general strategy* for player i is a probability measure on the set of his behavioral strategies B^i. Given a general strategy g^i for player i we denote by $g^i(b^i)$, with $b^i \in B^i$, the probability of b^i. Then, given a general strategy profile $\mathbf{g} = (g^1, g^2, ..., g^n)$, the definition of the payoff vector $H(\mathbf{g})$ associated with $\mathbf{g}$ is as follows. For each behavioral strategy profile $\mathbf{b}$ let $\mathbf{g}(\mathbf{b})$ be the joint probability distribution $\mathbf{g}(\mathbf{b}) = \prod_i g^i(b^i)$. Then,

$$H(\mathbf{g}) = \sum_{\mathbf{b}} \sum_{\text{leaf } \ell} \mathbf{g}(\mathbf{b})\, H(\ell).$$

Remark 1. Note that we can define the (expected) payoff vector associated with any strategy profile, where the various strategies of the players can be pure, mixed, behavioral, and general. The definition is straightforward.

EXAMPLE 2 (Isb57). Consider the two-person zero-sum game in extensive form of Fig. 2 (in Appendix A). This game [Isb57] shows that all three types of randomizations are different. Consider the general strategy that chooses with probability 1/2 the behavioral strategy $b_1 = (3/4{:}\, x_1,\ 0{:}\, x_2,\ 1/4{:}\, x_3,\ 0{:}\ y_1,\ 1{:}\ y_2)$ (i.e., with probability 3/4 choose x_1, with probability 0 choose x_2, and with probability 1/4 choose x_3, etc.) and with probability 1/2 the behavioral strategy $b_2 = (0{:}\, x_1,\ 3/4{:}\, x_2,\ 1/4{:}\, x_3,\ 1{:}\ y_1,\ 0{:}\ y_2)$. It is not hard to see that this general strategy guarantees the first player an expected payoff of 9/16. Now consider mixed strategies. Using an optimal mixed strategy the first player will obtain an expected payoff of

1/2. Finally, it is possible to show that the optimal behavioral strategy for the first player yields an expected payoff of 25/64.

3. TWO KNOWN EQUIVALENCE THEOREMS FOR LINEAR GAMES AND GAMES OF PERFECT RECALL

An n-person game in extensive form is called *linear* if the game tree is such that for each player i no information set of player i is intersected twice by a path from the root to a leaf. It is known that in a linear game every general strategy for any player is equivalent to some mixed strategy. This follows from Lemma 1 below, which states that, in a linear game, behavioral strategies are a special case of mixed strategies. Before we state the lemma it is necessary to define the following equivalence relation between strategies. Let Γ be a game. We say that two strategies of player i, x^i and y^i, are *equivalent* if for any strategy profile $\mathbf{s}^{\{-i\}}$ of the other players,

$$H(x^i, \mathbf{s}^{\{-i\}}) = H(y^i, \mathbf{s}^{\{-i\}}).$$

That is, the resulting payoff vector $H(\cdot)$ (consisting of the payoffs for all players) is the same under x^i and y^i. It is immediate to see that x^i and y^i are equivalent if for each leaf ℓ in the game tree the probability of reaching ℓ, given the strategy profile $(x^i, \mathbf{s}^{\{-i\}})$, equals the probability of reaching ℓ, given the profile $(y^i, \mathbf{s}^{\{-i\}})$.

LEMMA 1 [Isb57]. *Let Γ be a linear game. Then for any behavioral strategy b^i for player i there is an equivalent mixed strategy x^i.*

Hence, in a linear game the set of behavioral strategies B^i for player i is, in effect, a subset of the set of his mixed strategies X^i. In fact, B^i is a small subset of X^i. To see that note that the number of independent probabilities (i.e., degrees of freedom) required to specify a behavioral strategy is $\beta = \sum_{j=1}^{k_i} (|D(U_j^i)| - 1)$. In contrast, for a mixed strategy this number is $\chi = \prod_{j=1}^{k_i} (|D(U_j^i)| - 1)$. Thus, β is linear in the total number of choices (labels) of player i ($\sum_j |D(U_j^i)|$), whereas χ is exponential. In contrast, as indicated by Example 1 there are (nonlinear) games where the linear dimension of the space of behavioral strategies is larger than that of the mixed strategies. The following theorem is a simple conclusion of Lemma 1.

THEOREM 1 [Isb57]. *In a linear game every general strategy g^i of player i is equivalent to a mixed strategy x^i.*

Before stating the second known equivalence theorem we define the known concept of "perfect recall." Intuitively, a player has perfect recall if at all times during a play, he remembers what he knew, as well as what he chose at all previous decision nodes. Formally, we say that an edge e *precedes* a decision node v if there is a path on the game tree, starting with e and leading to v. We say that *player i has perfect recall* if for each edge e, outgoing from an information set U_e^i, and each decision node $v \in U_v^i$, if e precedes v then e precedes any other $w \in U_v^i$. It is not hard to prove the following lemma.

LEMMA 2 [Isb57]. *Any game in which every player has perfect recall is linear.*

The following equivalence theorem is due to Kuhn [Kuhn53].

THEOREM 2 [Kuh53]. *Let Γ be any game in extensive form. If player i has perfect recall then every mixed strategy x^i of player i has an equivalent behavioral strategy b^i.*

Remark 2. Kuhn's theorem was proven only for finite games. Nevertheless, this theorem can be extended, in a straightforward manner for infinite games, whenever the length of the game and the number of choices in each information set are at most countable. Further, Kuhn's theorem was extended by Aumann [Aum64] to the case where the number of choices in some information sets is uncountable. Note that in general, games in which time is continuous do not fit in the model presented here.

COROLLARY 3 [Kun53; Isb57]. *In games of perfect recall all three classes of randomized strategies* (*behavioral, mixed, and general*) *are equivalent.*

Since every game of complete information (see Appendix B) is, in particular, a game of perfect recall, it is clear that in games of complete information all three classes of randomized strategies are equivalent.

4. ONLINE ALGORITHMS AND COMPETITIVENESS

We use the basic formulation from Ben-David *et al.* [BDBK+94] of "request-answer games." Within this formulation we define on-line and off-line algorithms (and problems), and the competitive ratio performance measure.

A (*cost minimization*) *request-answer game* is defined as follows. (It is easy to define the analogous concepts for profit maximization problems.) Let R and W be nonempty sets. R is called the *request set* and W is called the *answer set*. Associated with these sets is a sequence of computable cost functions

$$cost_i: R^i \times W^i \to \mathbb{R}, \qquad i = 1, 2, \ldots.$$

Denote elements of R and W by r and w, respectively. Each such r is called a *request* and each w is called an *answer*. Sequences of requests and answers are denoted by σ and ω, respectively. A *request sequence* is any finite sequence of requests.

A *deterministic online algorithm* ALG is a sequence of functions $a^{(i)}: R^i \to W$, $i = 1, 2, \ldots,$. Given a (deterministic) on-line algorithm $\text{ALG} = \{a^{(i)}\}$ and a request sequence $\sigma = r_1, r_2, \ldots, r_j$, define $a[\sigma]$ as an element of W^j, $a[\sigma] = w_1, w_2, \ldots, w_j$ with $w_i = a^{(i)}(r_1, \ldots, r_i)$, $i = 1, 2, \ldots, j$. Thus, $\text{ALG}[\sigma]$ is the "answer sequence" of ALG with respect to the input sequence σ. For any input sequence σ with $|\sigma| = j$, the cost incurred by ALG with respect to σ, denoted $\text{ALG}(\sigma)$, is defined to be $cost_j(\sigma, \text{ALG}[\sigma])$. An *off-line algorithm* is a function $b: R^* \to W^*$ such that for each input sequence σ with $|\sigma| = j$, $b[\sigma] \in W^j$. An off-line algorithm b is *optimal* if for all σ,

$$b(\sigma) = cost_{|\sigma|}(\sigma, b[\sigma]) = \inf_{x \in W^*} \{cost_{|\sigma|}(\sigma, x)\}.$$

We denote by OPT a generic optimal off-line algorithm.

A deterministic on-line algorithm ALG is *c-competitive* (or "attains a competitive ratio of c") if there exists a constant α such that for each request sequence σ,

$$\mathrm{ALG}(\sigma) - c \cdot \mathrm{OPT}(\sigma) \leqslant \alpha.$$

The smallest c such that ALG is c-competitive is called ALG's *competitive ratio* and is denoted R(ALG). 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 α).

A *bounded cost* request answer game is a request answer game such that $\sup_\sigma cost_{|\sigma|}(\sigma, \cdot)$ is finite. A *finite* request answer game is a bounded cost request answer game with a finite request set and a finite answer set such that there exists a positive integer i_0 such that for all σ with $|\sigma| > i_0$, $cost_{|\sigma|}(\sigma, \cdot) = cost_{i_0}(\sigma', \cdot)$, where σ' is the proper i_0-prefix of σ (i.e., $|\sigma'| = i_0$). It is not hard to see that any finite request answer game is "isomorphic" to a request answer game where the number of possible deterministic (on-line) algorithms and the number of different request sequences is finite. When dealing with a finite request answer game we require, in the definition of c-competitiveness, that the additive constant α is zero for, otherwise, a sufficiently large additive constant (which may be dependent on the problem description) can allow for arbitrarily small competitive ratios.

We now need to introduce the concept of a randomized algorithm (for a request answer game) and its competitive ratio. As we have seen in Section 2 and Appendix B, there are two prominent approaches for defining the concept of a randomized algorithm: one based on mixed strategies and one on behavioral strategies. (As previously mentioned there is also the combined approach of general strategies.) We shall now briefly consider each of these two approaches in the context of the request-answer formulation. In the mixed strategy approach, a randomized on-line algorithm for a request answer game is a distribution on deterministic algorithms; that is, it is a distribution on sequences $\langle a_1, a_2, \ldots \rangle$ of answer functions. Of course this distribution is independent of any request sequence. From the behavioral approach, for each time step i, a randomized algorithm provides a distribution on a set of answer functions. Of course, the distribution used at any time step can depend on the request sequence seen thus far. However, if there is "no memory," then the distribution at each time step is independent of previous events. Whichever approach we used (i.e., mixed or behavioral) for any input sequence σ, we have a well-defined notion of the expected cost $\mathbf{E}[\mathrm{ALG}(\sigma)]$ of a randomized algorithm ALG on this input. An *oblivious* adversary for a randomized algorithm, must choose a nemesis request sequence σ, based only on the definition of the algorithm, without observing the choices of the algorithm (e.g., without observing the random coin tosses of the algorithm). For any request sequence σ, the optimal off-line cost is still well defined and now we can say that a randomized algorithm is c-competitive if there exists a constant α such that for each request sequence σ,

$$\mathbf{E}[\mathrm{ALG}(\sigma)] - c \cdot \mathrm{OPT}(\sigma) \leqslant \alpha.$$

The smallest c such that randomized ALG is c-competitive against an oblivious adversary is denoted $\bar{\mathscr{R}}_{\mathrm{OBL}}(\mathrm{ALG})$.

For on-line algorithms, in addition to oblvious adversaries, we can also consider the more powerful adaptive adversaries. Informally, an adaptive adversary[4] can observe the behavior of a randomized algorithm and selects the next request as a function of the present configuration of the on-line algorithm (but not knowing the outcome of any future random coin tosses). In this paper we are only concerned with oblivious adversaries but we note that the lower bound for demand paging in Theorem 6 is proven by first considering adaptive adversaries in Theorem 5.

5. MIXED AND BEHAVIORAL MEMORYLESS PAGING ALGORITHMS

Theorems 1 and 2 (and Examples 1 and 2) are of fundamental importance for competitive analysis. To understand the relevancy of these results it is sufficient to observe the structural properties (in terms of linearity and imperfect/perfect recall) of extensive form representations of randomized on-line algorithms. Clearly, when the available on-line algorithms have no memory restrictions, the on-line player essentially has perfect recall because during the game he can remember at each time everything he observed in the past. Hence, by Corollary 3 the three kinds of randomized strategies are equivalent.

Nevertheless, whenever there are memory restrictions, for example when the on-line player is restricted to use bounded memory, the equivalence theorems do not hold. It is not hard to see that the extensive form representation of on-line games, where the on-line player has bounded memory will in general exhibit nonlinearity and imperfect recall. Hence, for such games there is no a-priori equivalence between the three types of randomized strategies and in light of Isbell's example it can be anticipated that no such equivalence is possible. In this section we support this assertion by giving a natural example that distinguishes between memoryless mixed and behavioral on-line "demand paging" algorithms.

On the one hand, we use a known lower bound for the competitive ratio of any memoryless cat-and-rat behavioral algorithm to obtain a lower bound of k on the competitive ratio of any memoryless behavioral paging algorithm whenever $N = k + 1$ is the number of slow memory pages. On the other hand, we describe a new memoryless mixed paging algorithm which is $((k + 1)/2)$-competitive.

5.1. *Memoryless Behavioral Demand Paging Algorithms*

Consider a paging problem with a fast memory of size k and a slow memory of size $N > k$. A deterministic paging algorithm ALG can be defined as

$$\mathrm{ALG}: \mathscr{S} \times \mathscr{C} \times \mathscr{R} \to \mathscr{S} \times \mathscr{C},$$

[4] See [BDBK+94] or [BEY98] for the defintion of on-line and off-line adaptive adversaries and the corresponding defintions of the competitive ratio.

where $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{R}$ are the state, configuration and request sets, respectively. (Informally, the configuration of a paging algorithm is simply the contents of its fast memory.) That is, based on its current state and configuration of the algorithm and the present request, ALG changes state and (if the request is not in the current configuration) changes the configuration by evicting exactly one node and then including the present request in the new configuration. In fact, we have defined a *demand paging* algorithm which never evicts a page unless required (by a request for a page not presently in the cache) to do so. All the common paging algorithms are demand paging algorithms and hereafter we will identify paging with demand paging.[5] A paging algorithm is called *bounded memory* if $\mathcal{S}$ is finite (i.e., independent of the length of the request sequence) and it is called *memoryless* if $|\mathcal{S}| = 1$; i.e., there is only one state, in which case

$$\text{ALG}: \mathcal{C} \times \mathcal{R} \rightarrow \mathcal{C}.$$

The same definitions can be applied to randomized algorithms in which case the function ALG is a probabilistic function. Consider a deterministic or randomized memoryless on-line algorithm ALG. Since we assume demand paging, a request for a node that is already in the current configuration is ignored. Therefore in the case of a deterministic algorithm or a randomized algorithm against an adaptive adversary, we can assume that the adversary is "cruel" which means that it only requests nodes not in the current cache of ALG. When $N = k + 1$ each paging algorithm exposes a single hole at all times and we can summarize the situation as follows: wherever a randomized memoryless algorithm ALG has its hole (say at node u), the adversary requests u, causing ALG to move the hole to some node v (and incur one page fault) according to some distribution that depends only on u.

Coppersmith, Doyle, Raghavan, and Snir [CDRS90] have modeled this situation as a game between an on-line "blind"[6] cat and an adversarial rat (or mouse) on a complete $(k+1)$-node graph $G = (V, E, d)$, where the distance function d is uniform.[7] The cat is situated on some node of the graph and the rat requests ("threatens") that node, causing the cat to move. The cat's moves are defined by a stochastic matrix $P = (p_{ij})$, where p_{ij} is the probability that the cat will move from i to j when i is requested. Note that in general $p_{ij} \neq p_{ji}$.

Initially (at the start of a stage), both the cat and the rat are at the same node. Then, after generating a request for that node, the rat moves to some other node. In every step, the adversary generates a request for the node on which the cat is located. Then the cat moves to another node, according to the probabilities on the outgoing edges, from its current node. That is, the cat begins a *random walk* (determined by P) on the graph. When the cat finally catches the rat (i.e., moves to the node occupied by the rat) the stage ends and a new stage begins.

[5] In the case of unbounded memory, any paging algorithm can be converted to a demand paging algorithm having the same competitive ratio. This is not necessarily the case for the bounded memory algorithms that follow but, again, one usually assumes demand paging for all paging algorithms.

[6] We say "blind" since the cat will never know when the cat and rat are occupying the same node.

[7] For the abstract cat and rat game that is now being defined, we will only assume that d is symmetric. For the application to the paging problem, $d_{uv} = 1$ for all u and v.

Let $e=(i, j)$ be an edge in the graph. The *stretch* factor of e is the ratio h_{ij}/d_{ij}, where h_{ij} is the *hitting time* of nodes i and j, where the hitting time is defined to be the expected distance of a random walk (according to P) starting at node i until first landing on node j. The *edge stretch* factor of the graph G is the maximum stretch factor of any edge in the graph. The *stretch factor* of a cycle $\sigma=u_1, u_2, ..., u_\ell, u_1$ is the ratio $\mathrm{CAT}(\sigma)/\mathrm{RAT}(\sigma)$ (i.e., the ratio between the cost paid by the cat to the cost paid by the rat). The (*cycle*) stretch factor of the graph G is the maximum stretch factor of any cycle in G. The following theorems, due to Coppersmith, Doyle, Raghavan, and Snir [CDRS90] provide a lower bound on the competitive ratio of any randomized algorithm for the cat.

THEOREM 4 [CDRS90]. *For every symmetric weighted N-node graph and for every stochastic matrix P, the cycle stretch factor is at least* $N-1$.

THEOREM 5 [CDRS90]. *Let G be any symmetric weighted graph and let* CAT *be any online algorithm* (*i.e., let P be any stochastic matrix*). *If the cycle stretch factor of a graph with respect to P is c, then there exists an adaptive adversary* (*and also an* oblivious *adversary*) *that extracts a competitive ratio of c from the online player.*

Coppersmith *et al.* [CDRS90] also prove a converse for Theorem 5, showing that the stretch factor is also an upper bound on the competitive ratio attainable by the cat.

For $N=k+1$, Coppersmith *et al.* [CDRS90] apply Theorems 4 and 5 to obtain a lower bound of k for any memoryless paging algorithm against adaptive adversaries. The argument is as follows. The adversary chooses a cycle $v_0, v_1, ..., v_{m-}$ having stretch factor $c=k$. We can assume that initially the adversary (i.e. the rat) and on-line algorithm (the cat) have the same configuration (i.e. occupy the same hole) and that this initial hole is one of the v_i. Otherwise, the adversary moves its hole to (say) v_0 and continues to hit nodes other than v_0 until the on-line algorithm moves its hole to v_0. Whenever, the rat adversary and the cat on-line algorithm occupy the same hole (say v_i) on this cycle, the adversary requests v_i and moves its hole to $v_{(i+1) \bmod m}$. The adversary then continues to hit the hole of the on-line algorithm until it moves its hole to $v_{(i+1) \bmod m}$. In this way, the adversary rat is forcing the cat to take a random walk on the cycle. By definition of the cycle stretch factor, this random walk will have expectation k times the weight of the cycle cost. This result can be extended to oblivious adversaries. The problem with constructing an oblivious adversary is that the adversary does not know where the cat will move in any step, or how long the cat will take to find the rat. Instead, when the rat is in some node v, the adversary can hit every node in the graph *except* v many times, before hitting v and moving the rat. The number of times each node will be hit should be large enough to guarantee that the on-line cat will pay at least twice the stretch factor unless it reaches v. This way, the expected cost of the on-line player is at least the expected cost conditioned on the on-line player going across that cycle, which (as for the adaptive adversary) is k times the cost of the adversary. We thus have

THEOREM 6. *For* $N=k+1$ *the optimal competitive ratio of a memoryless behavioral paging algorithm against an oblivious adversary is k.*

5.2. *Memoryless Mixed Paging Algorithms*

Here again we consider the paging problem with $N=k+1$. Consider the following "permutation" paging algorithm due to Chrobak, Karloff, Payne, and Vishwanathan [CKPV91].[8]

Let π be a cyclic permutation of $\{1, 2, ..., N\}$ (i.e., the set of slow memory pages). Let $\pi^{(m)}$ denote the m-fold composition of the (cyclic) permutation π. Consider the following deterministic "permutation" algorithm.

ALGORITHM $\mathbf{PERM}_\pi$. Upon a page fault on page i evict page $\pi^{(m^*)}(i)$ such that m^* is the minimum m with $\pi^{(m)}(i)$ currently in the cache.

We first note that $\mathbf{PERM}_\pi$ is memoryless (as it does not store any information between successive requests). Note also that when $N=k+1$ Algorithm PERM_π always evicts page $\pi(i)$ on a fault on page i. Further, when $N=k+1$, $\pi^{(m)}(i)$ is the configuration of PERM_π after m faults, starting in the configuration i.

It is known [CKPV91] that Algorithm PERM_π is k-competitive for all $N>k$. Clearly, there are $(N-1)!$ different permutation algorithms. Define **MIX2PERM** to be the uniform mixture of two permutation algorithms, $\mathbf{PERM}_{\pi_1}$ and $\mathbf{PERM}_{\pi_2}$, where $\pi_1=(123\cdots N)$ and $\pi_2=(NN-1\cdots 1)$.

THEOREM 7. *For* $N=k+1$ **MIX2PERM** *is* $(k+1)/2$-*competitive against an oblivious adversary.*

The proof of Theorem 7 proceeds as follows. Since $N=k+1$ we can denote any configuration of a permutation algorithm by an integer in $\{1, ..., N\}$. Specifically, when $N=k+1$ the integer that represents a configuration is the index of the page that is currently *not* in the fast memory. For all i and j define

$$d_\pi(i, j)=\arg\min_m \pi^{(m)}(i)=j.$$

For example, if $k=4$ then $d_{12345}(2, 1)=4$ and $d_{54321}(2, 1)=1$.

Clearly, the range of the function $d_\pi(i, j)$ is $\{1, ..., k\}$ and for i and j with $i\neq j$, we immediately have $\sum_{\pi_1} d_\pi(i, j)+\sum_{\pi_2} d_\pi(i, j)=k+1$.

Consider any request sequence. We apply a standard k-phase partition technique.[9] Since $N=k+1$, during each k-phase the optimal off-line algorithm incurs exactly one fault. It follows that the adversary can benefit by always prolonging each k-phase thus forcing a particular configuration on the on-line algorithm. We thus seek to characterize the behavior of **MIX2PERM** with respect to such request sequences. Denote by $\phi_k(j)$ the set of all k-phases that do not include a request to page j (that is, a sequence is in $\phi_k(j)$ iff it includes references to all pages $i\neq j$). We need the following lemma.

[8] Chrobak *et al.* called this algorithm ROTATE.

[9] The k-phase partition of a request sequence σ is its partition into segments as follows: phase 0 is the empty sequence. For every $i-1$, phase i is the maximal sequence following phase $i-1$ that contains at most k distinct page requests (see [FKL+91])

LEMMA 3. *Let* $\phi \in \phi_k(j)$. *Then,* $\mathbf{PERM}_\pi(\phi)$, *the cost of* $\mathbf{PERM}_\pi$ *to process* ϕ, *is less than or equal to* $d_\pi(i, j)$ *if it starts at configuration i.*

Proof. We prove the lemma by induction on $n = |\phi|$ (where ϕ is any suffix of a k-phase in $\phi_k(j)$). The claim clearly holds for the empty suffix. Now, consider the first request $\phi(1)$ of ϕ. If $\phi(1) \neq i$, PERM_π does not change its configuration and by the induction hypothesis, $Z_i = \mathrm{PERM}_\pi(\phi(2), ..., \phi(n)) \leqslant d_\pi(i, j)$, with Z_i denoting the cost of PERM_π for processing the suffix $\phi(2), ..., \phi(n)$, starting at configuration i. Otherwise, $\phi(1) = i$ and by the induction hypothesis $Z_{\pi(i)} \leqslant d_\pi(\pi(i), j) = d_\pi(i, j) - 1$ so $\mathrm{PERM}_\pi(\phi) = 1 + Z_{\pi(i)} \leqslant d_\pi(i, j)$.

Note that in the worst case, the inequality in Lemma 14 is replaced with equality. Now, suppose that both permutation algorithms (that is, for permutations π_1 and π_2) work concurrently and start at configuration i. Consider any k-phase $\phi \in \phi_k(j)$. It is clear that upon processing ϕ both permutation algorithms will end in configuration j. By Lemma 14 the total cost of both permutation algorithms to process ϕ is $k + 1$. Hence the expected cost of **MIX2PERM** for this k-phase is $(k + 1)/2$ and Theorem 13 follows.

This completes the proof of the Theorem 13. The algorithm **MIXPERM** that uniformly mixes all $(N - 1)!$ permutation algorithms is also $(k + 1)/2$-competitive for $N = k + 1$ (see [BEY98]). Recently, Chi-Lok Chan (personal communication) has shown that $(k + 1)/2$ is a lower bound for mixed (and general) strategies of memoryless demand paging and has exhibited counterexamples showing that neither **MIX2PERM** nor **MIXPERM** can achieve $(k + 1)/2$ for $N > k + 1$.

6. THE YAO PRINCIPLE FOR ON-LINE ALGORITHMS

We refer the reader to Appendix C, where we review the game-theoretic definitions and concepts underlying the Yao principle, as applied to finite games in strategic form, viewing randomized algorithms as mixed strategies. In particular, we now consider unbounded memory on-line randomized algorithms which can (as games of perfect recall) be viewed as mixed strategies. The Yao principle can be used for proving lower bounds on the competitive ratio of randomized on-line algorithms against an oblivious adversary (and thus against all adversary types). For a profit maximization problem the method is essentially the following: to obtain a lower bound of c on the competitive ratio of the best randomized algorithms it is sufficient to choose any probability distribution y over inputs (request sequences) and to bound from below, by c, the ratio of average optimal off-line profit to average on-line profit of any *deterministic* on-line algorithm (where the expectations are taken with respect to y). In this case, c is a lower bound on the competitive ratio of the best randomized on-line algorithm. As we shall see, there are several variations of this method depending on whether the on-line problem is finite (i.e., the sets of deterministic algorithms and possible request sequences are finite), or not, and whether the payoff function is bounded, or not.

As it turns out, the technique applied for profit maximization problems differs somewhat from its counterpart for cost minimization problems. We first develop the technique for profit maximization problems and then develop it for cost minimization problems.

6.1. The Yao Principle for Profit Maximization Problems

A profit maximization request-answer game is called *finite* if both the set of deterministic on-line algorithms and the set of request sequences are finite. Denote the finite set of deterministic algorithms by $\mathscr{A} = \{\text{ALG}_1, \text{ALG}_2, ..., \text{ALG}_m\}$. Similarly, let the finite set of possible request sequences be $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$. We now define the following two-person zero-sum game G. Player 1 is the on-line player and $\mathscr{A}$ is the set of his pure strategies. Player 2 is the adversary and Σ is the set of his pure strategies. It remains to define the payoff function. Here we have several possibilities. Perhaps the most natural choice for the payoff function is $h_1(i, j) = \text{OPT}(\sigma_j)/\text{ALG}_i(\sigma_j)$. At the outset, this choice for the payoff function seems to be natural since it specifies the competitive ratio explicitly. Nevertheless, this is not quite the case. In particular, let us consider the right-hand side of Yao's inequality. (See the inequality in displayed line (8) in Appendix C.) When applied to the matrix H specified by h_1, we have

$$\max_{x(i)} \min_j \mathbf{E}_{x(i)} \left[\frac{\text{OPT}(\sigma_j)}{\text{ALG}_i(\sigma_j)} \right], \tag{1}$$

where $x = x(i)$ denotes a mixed strategy for player 1 and $\mathbf{E}_{x(i)}[\,\cdot\,]$ denotes the expectation with respect to x. Similarly we use the same notation for $y = y(j)$ and $\mathbf{E}_{y(j)}[\,\cdot\,]$. Notice that this expression that is supposed to specify the optimal randomized competitive ratio is not quite compatible with the standard definition of the competitive ratio. In particular, according to the "standard" definition, the optimal randomized competitive ratio is

$$\max_{x(i)} \min_j \frac{\text{OPT}(\sigma_j)}{\mathbf{E}_{x(i)}[\text{ALG}_i(\sigma_j)]}.$$

But in general for a random variable X, $\mathbf{E}[1/X] \neq 1/\mathbf{E}[X]$. Hence, one cannot prove lower bounds on the standard randomized competitive ratio using this definition.[10] As seen in the following example,[11] this confusion may lead to erroneous conclusions.

EXAMPLE 3 (Wrong use of the Yao Principle). Consider the following on-line problem given by

[10] But of course, it is certainly possible to *define* the randomized competitive ratio for a profit maximization problem by using the form (1).

[11] This example was given to us by K. Pruhs (personal communication).

	σ_1	σ_2	$\cdots$	σ_i	$\cdots$	σ_k	σ_{k+1}
ALG_1	$\frac{k}{1}$	$\frac{k}{2}$	$\cdots$	$\frac{k}{i}$	$\cdots$	$\frac{k}{k}$	$\frac{2k}{2k}$
ALG_2	$\frac{k}{k}$	$\frac{k}{k}$	$\cdots$	$\frac{k}{k}$	$\cdots$	$\frac{k}{k}$	$\frac{2k}{k}$

The (i, j) entry of this matrix is a fraction of the form $\mathrm{OPT}(\sigma_j)/\mathrm{ALG}_i(\sigma_j)$ which corresponds to the payoff function h_1. Consider the mixed strategy for the adversary,

$$y=\left(\frac{1}{kH_k}, ..., \frac{1}{kH_k}, 1-\frac{1}{H_k}\right),$$

where H_k is the kth harmonic number, $H_k=1+1/2+\cdots+1/k$. It is not hard to check that for $i=1, 2$, $\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)/\mathrm{ALG}_i(\sigma_j)] \geqslant 2-o(1)$. Hence, we have by the Yao Principle that $\max_{x(i)} \min_j \mathbf{E}_{x(i)}[\mathrm{OPT}(\sigma_j)/\mathrm{ALG}_i(\sigma_j)] \geqslant 2-o(1)$. Nevertheless, it is easy to verify that mixing ALG_1 and ALG_2 with probabilities 1/3 and 2/3 is 1.5-competitive.

A slight modification of the payoff function (i.e., taking $1/h_1$ instead of h_1 and letting Player 1 be the maximizer) yields a correct "Yao's principle." Moreover, we can formulate the Yao Principle using a different game. This is summarized in the following theorem whose proof is given in Appendix D.

THEOREM 8 (Yao principle: Finite profit problems). *Let G be any finite*[12] *request-answer game. Let* ALG *be any on-line randomized algorithm for G and let* $\bar{\mathscr{R}}_{\mathrm{OBL}}(\mathrm{ALG})$ *be the competitive ratio of* ALG *against an oblivious adversary. Let* $y(j)$ *be any probability distribution over request sequences. Then,*

$$\mathscr{R}_{\mathrm{OBL}}(\mathrm{ALG}) \geqslant \max\left\{\min_i \frac{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)]}{\mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j)]}, \min_i \frac{1}{\mathbf{E}_{y(j)}\left[\dfrac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)}\right]}\right\}$$

EXAMPLE 4. In continuation to Example 3 we now show how to use Theorem 8 to obtain a correct (and tight) lower bound to the problem of Example 3. Consider the following mixed strategy y for the adversary with y_i, the ith component of y being

$$y_i=\begin{cases} 1/2^{i+1}, & i=1, 2, ..., k-1, \\ 1/2^k, & i=k, \\ 1/2, & i=k+1. \end{cases}$$

[12] The finiteness requirement ensures that the von Neumann minimax theorem holds. It can be relaxed whenever the request-answer game can be appropriately formulated as a two-person zero-sum game with a payoff function that guarantees minimax (e.g., when this payoff function is continuous; see [Rag94] for other classes of such payoff functions).

Clearly, $\sum_i y_i = 1$ and it is not hard to see that a lower bound of $3/2 - o(1)$ can be proved, using this distribution, via Theorem 8 (using the ratio of expectations).

Theorem 8 can be applied only when the von Neumann minimax theorem applies. The minimax theorem always holds with respect to finite games. For infinite games the minimax theorem is known to hold for some particular types of games (e.g., when the payoff function is continuous) but does not hold in general.[13] Hence, one cannot apply Theorem 8 without verifying that the minimax theorem applies. Luckily, for unbounded (and therefore infinite) games we have the following version of the Yao principle that is somewhat more complicated to state but does not rely at all on the minimax theorem. This theorem is an adaptation of a cost minimization variant proved by Borodin, Linial, and Saks [BLS92] for metrical task systems. The proof of the theorem is given in Appendix E.

THEOREM 9 (Yao principle: Unbounded Profit Problems). *Let G be any unbounded payoff maximization request-answer game. Let* ALG *be any randomized online algorithm in G. Let $y(j)$ be any probability distribution over the set of all finite request sequences, $\{\sigma_j\}$. For each positive integer n, let $y^n(j)$ be the marginal distribution over the set of all request sequences of length n $\{\sigma_j^n\}$.*[14] *Suppose that two conditions are satisfied*:

$$\limsup_{n\to\infty} \frac{\mathbf{E}_{y^n(j)}[\mathrm{OPT}_x(\sigma_j^n)]}{\sup_i \mathbf{E}_{y^n(j)}[\mathrm{ALG}_i(\sigma_j^n)]} \geqslant c, \tag{2}$$

$$\limsup_{n\to\infty} \mathbf{E}_{y^n(j)}[\mathrm{OPT}(\sigma_j^n)] = \infty. \tag{3}$$

Then, $\bar{\mathscr{R}}_{\mathbf{OBL}}(\mathrm{ALG}) \geqslant c$.

6.2. The Yao Principle for Cost Minimization Problems

For cost minimization problems we have two theorems analogous to Theorems 8 and 9 (i.e., for finite and unbounded request-answer games). The theorems for cost minimization and their proofs are almost analogous to the profit maximization theorems. The main difference is with the finite case where, in the case of cost minimization, we do not resort to the inversion of the "natural" payoff function (h_1 in the case of profit maximization).

THEOREM 10 (Yao principle: Finite Cost Problems). *Let G be any finite*[15] *cost minimization request-answer game. Let* ALG *be any online randomized algorithm for G. Let $y(j)$ be any probability distribution over request sequences. Then,*

$$\bar{\mathscr{R}}_{\mathbf{OBL}}(\mathrm{ALG}) \geqslant \max\left\{\min_i \frac{\mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j)]}{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)]}, \min_i \mathbf{E}_{y(j)}\left[\frac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)}\right]\right\}.$$

[13] See [Rag94] for a survey that lists all known types of games for which the minimax theorem holds.

[14] The marginal distribution $y^n(j)$ is obtained by normalizing probabilities of sequences of length n with respect to the total sum of these probabilities.

[15] Here again the finiteness requirement ensures that the von Neumann minimax theorem holds and can be relaxed whenever the request-answer game can be formulated as a two-person zero-sum game with a payoff function that guarantees minimax.

The following theorem is due to Borodin, Linial, and Saks [BLS92].

THEOREM 11 (Yao principle: Unbounded Cost Problems). *Let G be any unbounded cost minimization request-answer game. Let* ALG *be any randomized online algorithm for G. Let y(j) be a probability distribution over the set of all finite request sequences,* $\{\sigma_j\}$. *For each positive integer n, let* $y^n(j)$ *be the marginal distribution over the set of all request sequences of length n* $\{\sigma_j^n\}$. *Suppose that two conditions hold*:

$$\liminf_{n\to\infty} \frac{\inf_i \mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j^n)]}{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j^n)]} \geqslant c \tag{4}$$

$$\limsup_{n\to\infty} \mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j^n)] = \infty. \tag{5}$$

Then, $\bar{\mathscr{R}}_{\mathrm{OBL}}(\mathrm{ALG}) \geqslant c$.

7. CONCLUDING REMARKS

The use of the Yao principle for on-line algorithms can be a source for some confusion. We hope that by formulating this principle rigorously in the context of competitive on-line analysis we have helped to clarify the applicability of the principle.

The issue of bounded recall, non-linear games and in general, of games involving bounded memory algorithms is not well studied in game theory.[16] Nevertheless, it is of fundamental importance for the theory of computation. Clearly, bounded memory (randomized) algorithms play an important role in algorithm design and analysis. For example, most real-time on-line algorithms are of bounded memory. Thus, this paper only scratches the surface of what we believe to be a wide and important area of research, where we seek to understand the computational power of the various kinds of (bounded memory) randomized algorithms.

To conclude this paper we would like to suggest several directions for future research:

- Games of extensive form are far from satisfactory for representations of bounded memory randomized algorithms. (For example. the reader may want to try to model a few stages of the "competitive game" of a *k*-server algorithm with three bits of memory against an adversary using extensive form with the appropriate information sets.) It would be of great interest to devise a more suitable model for games involving bounded memory algorithms.

- Find other examples of natural on-line games where optimal (memory bounded) behavioral, mixed (and general) algorithms attain different competitive ratios. In particular, do the results for demand paging extend to non demand paging algorithms?

- Characterize two-person zero-sum on-line games of imperfect recall (games involving bounded algorithms) for which Kuhn's theorem is still valid.

[16] Two recent papers that do relate to these issues are [PR94; AHP96].

APPENDIX A: SOME FIGURES

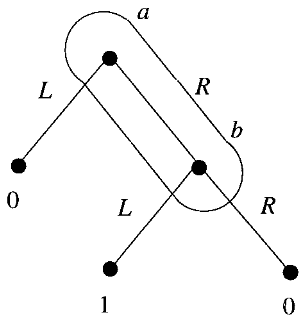

FIG. 1. One person game with incomplete information and imperfect recall.

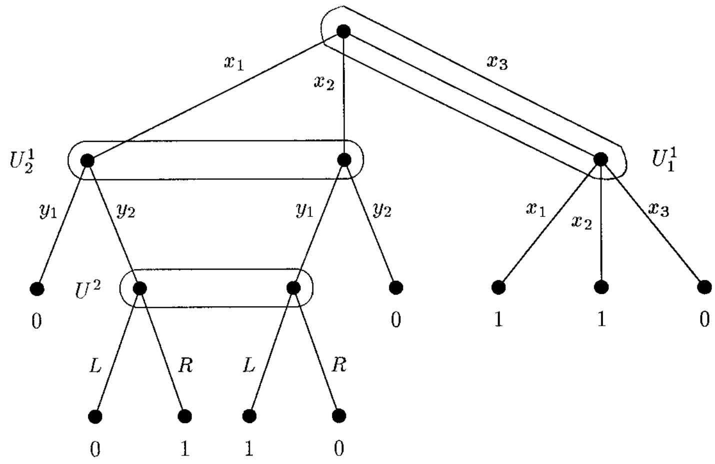

FIG. 2. Two-person zero-sum game distinguishing between general, mixed and behavioral strategies.

APPENDIX B: GAMES IN EXTENSIVE AND STRATEGIC FORMS

There are two primary mathematical abstractions of games, called "extensive" and "strategic." The extensive form representation of a game is what is commonly referred to as a "game tree." It explicitly specifies the rules of the game via the order of the moves for the various players, the information and choices available to a player whenever it is one's turn to move and the payoff obtained by all players in any possible play. Indeed, one possible way to depict or imagine a game in extensive form is via a tree.

A game in strategic form (also called "normal form") is what is commonly referred to as a "matrix game." As the name suggests, this representation abstracts away the individual moves and focuses only on strategies. As a more abstract representation of a game it provides an important theoretical tool.

For simplicity, the formulations that follow deal only with *finite games* that have a finite set of players and finite total number of decision alternatives for each player. Note that the modeling of (competitive analysis of) on-line problems requires in general infinite games (and only two players, the on-line player and the adversary). All the formulations that follow generalize naturally to infinite enumerable games. Later we shall discuss generalizations for infinite games with larger cardinality.

B.1. Games in Extensive Form

An *n-person game* in extensive form consists of a set $N=\{1, 2, ..., n\}$ of *players* and a rooted tree T called the *game tree*. The game tree has the following structure:

(1) The set of internal nodes of T is partitioned into $n+1$ subsets P^i, $i=0, 1, ..., n$, where for each $i \geqslant 1$ the members of P^i are called "the (decision) nodes of player i." A node in P^i is called a *chance node* (or a *nature node*) and for each such node there is an associated probability distribution over its outgoing edges;

(2) All the outgoing edges to descendants of each internal node associated with a player are distinctly labeled by *action labels*;

(3) For each $i=1, 2, ..., n$, the set P^i of decision nodes is partitioned into k_i *information sets*, $U^i_1, U^i_2, ..., U^i_{k_i}$, such that for each information set U^i_j all the decision nodes in U^i_j are "isomorphic" in the sense that they have the same number of outgoing edges, and these outgoing edges have the same labels.

(4) Each leaf of T is labeled by an n-tuple $(h^1, h^2, ..., h^n)$ of *payoffs*.

Under the following interpretation the above model[17] gives a complete description of each "play" of the game. We define a *play* of the game Γ as any path from the root to a leaf with the understanding that this path was obtained by the players (and chance) as follows. First note that the role of chance moves is straightforward. If, during a play of the game, a chance node v is reached, the probability distribution associated with v is "invoked" to choose one outgoing edge of v. The role of players' decision nodes and payoffs is self-explanatory. Now, to better understand the role of information sets, consider the following interpretation. Imagine that each player i is in command of k_i "agents" that play the game for him. For $j=1, 2, ..., k_i$, the jth agent (of player i) is in charge of all decision nodes in the jth information set U^i_j. Before the start of the game player i instructs the agents and gives each of them a strategy. Then after the game starts the agents cannot communicate with each other (and with their boss). Now suppose that the play of the game has progressed to a certain node $v \in U^i_j \subseteq P^i$; that is, the next move should be played by the jth agent of player i, who must choose one outgoing edge of node v. At this time the only information available to this agent is the description of the game Γ and instructions obtained from his "boss" (player i) before the start of the game. In particular, all nodes in U^i_j appear identical to this agent since he is not told which path led to his information set. Since all nodes in the same information set are "isomorphic" the agent specifies one choice that is valid for any node in the information set.

[17] In fact, as part of the definition one must add the requirement that the extensive form description is "common knowledge" among the n players. This means that all players know it, each player knows that all other players know it, each one knows that everyone else knows it and so on (*ad infinitum*). For a discussion why this requirement is necessary see Myerson [Mye91, Section 2.7].

The introduction of information sets allows for descriptions of complex games in which the players do not have complete information on the actions of the other players and even their own previous actions. Formally we say that a game is of *complete information* if and only if each information set is a singleton. Otherwise we say that the game is of *incomplete information.* Examples of games of complete information are chess and tic-tac-toe. Examples of games of incomplete information are bridge and poker. Also, any off-line request answer game is a game of complete information and any on-line request answer game is a game of incomplete information.

B.2. Games in Strategic Form

An *n-person game* in strategic form consists of a set $N=\{1, 2, ..., n\}$ of players and for each player i, $i=1, 2, ..., n$, there is a set S^i of *pure strategies.* Also there is a function $H: S^1 \times S^2 \times \cdots \times S^n \to \mathbb{R}^n$, called the *payoff function.* Each vector $\mathbf{s} \in S^1 \times S^2 \times \cdots \times S^n$ is called a pure strategy *profile* and for each strategy profile $\mathbf{s}$ the ith coordinate $H^*(s)$ of $H(\mathbf{s})$ specifies the payoff for the ith player for a play where the players choose their respective strategies according to $\mathbf{s}$.

As may be expected, the transformation from an extensive form to strategic form is many-to-one. Indeed, going from extensive form to strategic form abstracts away some information given in the extensive form (e.g., the order of the moves) so that some information is lost.[18] However, this transformation always preserves the sets of pure and mixed strategy sets and their (expected) payoffs.

A key step in transforming an extensive form into strategic form is defining the "pure strategies" in the extensive form game. Intuitively, in an extensive form game a pure strategy for a player is a list of edge labels, one for each of his information sets. Formally, let Γ be an n-person game in extensive form. Consider player i and let $U^i=\{U^i_1, U^i_2, ..., U^i_{k_i}\}$ be the set of his k_i information sets. Each information set U^i_j contains decision nodes that have the same number of outgoing edges and we denote the set of choices (labels of outgoing edges) of each node in U^i_j by $D(U^i_j)=\{1, 2, ..., d^i_j\}$. Each vector $(s_1, s_2, ..., s_{k_i})$, where $s_j \in D(U^i_j)$ is a *pure strategy* of player i. That is, player i has $\prod_{j=1}^{k_i} d^i_j$ pure strategies. Denote by S^i the set of pure strategies of player i. It remains to define the payoff associated with a pure strategy profile $\mathbf{s}=(s^1, s^2, ..., s^n)$ of the n players. For each pure strategy profile $\mathbf{s}$, and for each leaf ℓ denote by $p(\ell \mid \mathbf{s})$ the probability that the leaf ℓ is reached, given the profile $\mathbf{s}$. For each leaf ℓ denote by $H(\ell)$ the vector of expected payoffs associated with ℓ (as defined by the extensive form). Then, the vector payoff is

$$H(\mathbf{s}) = \sum_{\text{leaf } \ell} p(\ell \mid \mathbf{s})\, H(\ell).$$

[18] In the case where the extensive form has chance nodes, the payoffs are random variables and therefore a transformation to strategic form results in a payoff matrix whose entries are *expected* payoffs (i.e., the first moment) so we do lose probabilistic information (i.e., the rest of the moments). (Of course, we could put a probability distribution in each entry.)

APPENDIX C: MINIMAX THEOREM AND THE YAO PRINCIPLE

The competitive analysis of an on-line problem can be seen as a solution of a two-person zero-sum game. In particular, given any finite request-answer game (say of profit maximization) one possible way to view it as a two-player zero-sum game is the following. The first player who is the on-line player seeks a strategy that maximizes the payoff defined to be the reciprocal of its competitive performance; that is, the payoff is the ratio of on-line profit to optimal off-line profit.[19] The adversary seeks a strategy that minimizes the on-line player's payoff. Thus, the two players are in strict opposition and the game is zero-sum.

Unless otherwise is stated, we shall be concerned from now on with games of perfect recall where we have an equivalence between general, mixed, and behavioral strategies (see Theorem 2). In general, the theorems stated in this section, such as the minimax theorem do not apply, with respect to behavioral strategies under games of imperfect recall (and nonlinear games). However, under perfect recall we can assume, without loss of generality, that all games are given in their strategic form.

Let $G=(h_{ij})$ be a finite two-person zero-sum game. In what follows the row (resp. column) index i (resp. j), enumerates the pure strategies of the first (resp. second) player. For mixed strategies $x=(x_1, x_2, ..., x_m)$ and $y=(y_1, y_2, ..., y_n)$ for players 1 and 2, respectively, we define

$$H(x, y)=H_G(x, y)=\sum_{i=1}^{m}\sum_{j=1}^{n} h(i, j)\, x_i y_j,$$

the expected payoff for player 1 under x and y. The game with players 1 and 2 using $H(x, y)$ as payoff function is called the *mixed extension of G.*

THEOREM 12 (von Neumann). *Every finite two-person zero-sum game has a value. That is,*

$$\max_x \min_y H(x, y)=\min_y \max_x H(x, y). \tag{6}$$

One interpretation of the minimax theorem is that in any two-person zero-sum game any player does not lose anything by revealing his best mixed strategy before the start of the game. Further, a simple observation is that if player 1 (resp. player 2) knows that player 2 (resp. player 1) uses (an optimal) mixed strategy, one of his optimal (mixed) strategies is deterministic. This is established in the following simple lemma (which is a corollary of the minimax theorem).

LEMMA 4 (Loomis' lemma). *Let x^* and y^* be optimal mixed strategies for players 1 and 2. Then,*

$$\max_i H(i, y^*)=\min_j H(x^*, j)=H(x^*, y^*).$$

[19] There is a reason for this particular definition. In particular, one cannot take the payoff function to be the optimal off-line profit over the on-line profit (so that player 1 is the minimizer). See Section 6 for details.

Alternatively,

$$\min_y \max_i H(i, y) = \max_x \min_j H(x, j) = H(x^*, y^*). \tag{7}$$

Proof. Set $v = H(x^*, y^*)$, the value of the game. By Theorem 12, v is well defined. Clearly, for all j, $v \leqslant H(x^*, j)$. Hence $v \leqslant \min_j H(x^*, j)$. Assume, by contradiction, that $v < \min_j H(x^*, j)$. Then for all j, $v < H(x^*, j)$. Setting $y^* = (y_1^*, y_2^*, ..., y_n^*)$ we thus have

$$v = \sum_{j=1}^{n} v y_j^* < \sum_{j=1}^{n} H(x^*, j)\, y_j^* = H(x^*, y^*).$$

But this clearly contradicts the minimax theorem (Theorem 12). Hence, $v = \min_j H(x^*, j)$. An analogous arguments proves that $v = \min_i H(i, y^*)$.

Lemma 4 entails a simple but very useful conclusion—the "Yao principle." In particular, from the equality (7) it follows that for any mixed strategy y for player 2,

$$\max_i H(i, y) \geqslant \max_x \min_j H(x, j). \tag{8}$$

This means that we can obtain a bound[20] on player 1's best (randomized) payoff by calculating his best deterministic payoff with respect to any mixed strategy for player 2. We refer to this inequality (8) as *Yao's inequality*. The application of Yao's inequality (as first applied by Yao in [Yao77]) as a tool for obtaining bounds is now routine in standard complexity theory. Nevertheless, for the purpose of competitive analysis it is somewhat delicate.

APPENDIX D: PROOF OF THEOREM 8

We first show that $\bar{\mathcal{R}}_{\mathbf{OBL}}(\mathrm{ALG})$ is at least

$$\frac{1}{\max_i \mathbf{E}_{y(j)} \left[\dfrac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)} \right]}.$$

Formulate the request-answer game as a two-person zero-sum game using the payoff function

$$h_2(i, j) = 1/h_1(i, j) = \frac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)}$$

[20] Since in this game player 1 is a (profit) maximizer this bound is an upper bound (i.e., a negative result concerning player 1's performance). When player 1 is a (cost) minimizer Yao's inequality would be $\min_i H(i, y) \leqslant \min_x \max_j H(x, j)$ for any mixed strategy y.

(i.e., using h_2 the on-line player is the maximizer). Suppose that for some mixed strategy $y(j)$ for the adversary we have

$$\frac{1}{c} \geqslant \max_i \mathbf{E}_{y(j)} \left[\frac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)} \right].$$

By Yao's inequality (8) we thus have

$$\begin{aligned} \frac{1}{c} &\geqslant \max_{x(i)} \min_j \mathbf{E}_{x(i)} \left[\frac{\mathrm{ALG}_i(\sigma_j)}{\mathrm{OPT}(\sigma_j)} \right] \\ &= \max_{x(i)} \min_j \frac{\mathbf{E}_{x(i)}[\mathrm{ALG}_i(\sigma_j)]}{\mathrm{OPT}(\sigma_j)} \\ &\geqslant \frac{1}{\overline{\mathscr{R}}_{\mathrm{OBL}}(\mathrm{ALG})}. \end{aligned}$$

We now prove that

$$\min_i \frac{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)]}{\mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j)]}$$

is also a lower bound on the (best) randomized competitive ratio. For each constant c consider a two-person zero-sum game $G(c)$ between the on-line player (player 1) against the adversary (player 2). For each pure strategy pair i and j the payoff to the on-line player is

$$h_3(i, j) = c \cdot \mathrm{ALG}_i(\sigma_j) - \mathrm{OPT}(\sigma_j).$$

Again, the on-line player is the maximizer. By the minimax theorem the game $G(c)$ has a value $V(c)$ and by Loomis' lemma (Lemma 4),

$$V(c) = \max_{x(i)} \min_j \mathbf{E}_{x(i)} h_3(i, j).$$

Clearly $V(c) \geqslant 0$ if and only if the best randomized algorithm for the on-line player is c-competitive. Notice that for any mixed strategy $y(j)$ for the adversary,

$$0 > \max_i \mathbf{E}_{y(j)}[h_3(i, j)]$$

if and only if

$$\min_i \frac{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)]}{\mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j)]} > c. \tag{9}$$

Suppose that the inequality (9) holds. Then by Yao's inequality (8)

$$0 > \max_i \mathbf{E}_{y(j)}[h_3(i, j)] \geqslant \frac{\mathbf{E}_{y(j)}[\mathrm{OPT}(\sigma_j)]}{\mathbf{E}_{y(j)}[\mathrm{ALG}_i(\sigma_j)]} = V(c)$$

and, therefore, the best randomized algorithm for the on-line player is not c-competitive.

APPENDIX E: PROOF OF THEOREM 9

Suppose that (2) and (3) hold with respect to some probability distribution $y(j)$. Assume, by way of contradiction, that the competitive ratio of the (randomized) algorithm ALG (represented by the distribution $x(i)$) is $c' < c$. By definition, there exists a constant α such that

$$c' \cdot \mathbf{E}_{x(i)}[\mathrm{ALG}_i(\sigma_j^n)] \geqslant \mathrm{OPT}(\sigma_j^n) + \alpha$$

for each of the sequences in $\{\sigma_j^n\}$. Taking the expectation of both sides with respect $y^n(j)$ we obtain

$$c' \cdot \mathbf{E}_{y^n(j)} \mathbf{E}_{x(i)}[\mathrm{ALG}_i(\sigma_j^n)] \geqslant \mathrm{OPT}(\sigma_j^n) + \alpha.$$

Since the profits ALG$(\cdot)$ are nonnegative we can exchange the order of the expectations (whether they are defined by sums or integrals) in the left-hand side,

$$c' \cdot \mathbf{E}_{x(i)} \mathbf{E}_{y^n(j)}[\mathrm{ALG}_i(\sigma_j^n)] \geqslant \mathrm{OPT}(\sigma_j^n) + \alpha.$$

Set

$$A_n^* = \sup_i \mathbf{E}_{y^n(j)}[\mathrm{ALG}_i(\sigma_j^n)].$$

Clearly,

$$c' \cdot A_n^* \geqslant c' \cdot \mathbf{E}_{x(i)} \mathbf{E}_{y^n(j)}[\mathrm{ALG}_i(\sigma_j^n)].$$

Hence, for all sufficiently large n (for which $A_n^*) > 0$)

$$\frac{\mathbf{E}_{y^n(j)}[\mathrm{OPT}(\sigma_j^n)]}{A_n^*} + \frac{\alpha}{A_n^*} \leqslant c' < c.$$

By assumption (3) and since ALG is c'-competitive it must be that $\alpha / A_n^* \to 0$. But this contradicts assumption (2). It follows that $\bar{\mathscr{R}}_{\mathrm{OBL}}(\mathrm{ALG}) \geqslant c$.

ACKNOWLEDGMENTS

We thank Kirk Pruhs for Example 3 which inspired us to think of the Yao principle in this context. Also, we thank Robert Aumann and Adi Rosén for helpful discussions. Finally, we are grateful to Chi-Lok Chan and Micheal Loui for many suggestions and corrections to an earlier verion of this paper.

Received April 21, 1997; final manuscript received November 3, 1998

REFERENCES

[AHP96] Aumann, R. J., Hart, S., and Perry, M. (1996), "On the Absent-Minded Paradox, 1996," discussion paper, Center for Rationality, The Hebrew University.

[Aum64] Aumann, R. J. (1994), Mixed and behavior strategies in infinite extensive games, *in* "Advances in Game Theory" (M. Dresher, L. S. Shapley, and A. W. Tucker, Eds.), Vol. 1, pp. 627–650, Princeton Univ. Press, Princeton, NJ.

[BBR+90] Beame, P., Borodin, A., Raghavan, P., Ruzzo, W. L., and Tompa, M. (1990), Time-space trade-offs for undirected graph connectivity, *in* "Proceedings of the 31st Annual Symposium on Foundations of Computer Science," pp. 429–438.

[BDBK+94] Ben-David, S., Borodin, A., Karp, R. M., Tardos, G., and Wigderson, A. (1994), On the power of randomization in on-line algorithms, *Algorithmica* **11**, 2–14.

[BEY98] Borodin, A., and El-Yaniv, R. (1998), "On-line Computation and Competitive Analysis," Cambridge Univ. Press, Cambridge.

[BLS92] Borodin, A., Linial, N., and Saks, M. (1992), An optimal on-line algorithm for metrical task systems, *J. Assoc. Comput. Mach.* **39**, 745–763.

[CDRS90] Coppersmith, D., Doyle, P., Raghavan, P., and Snir, M. (1990), Random walks on weighted graphs, and application to on-line algorithms, *in* "Proceedings of the 22nd Annual ACM Symposium on Theory of Computing," pp. 369–378.

[CKPV91] Chrobak, M., Karloff, H. J., Payne, T., and Vishwanathan, S. (1991), New results on server problems, *SIAM J. Discrete Math.* **4**(2), 172–181.

[FKL+91] Fiat, A., Karp, R. M., Luby, M., McGeoch, L. A., Sleator, D. D., and Young, N. E. (1991), On competitive algorithms paging algorithms, *J. Algorithms* **12**, 685–699.

[Isb57] Isbell, J. R. (1957), Finitary games, *in* "Contributions to the Theory of Games" (M. Dresher, A. W. Tucker, and P. Wolfe, Eds.), Vol. III, pp. 79–96, Princeton Univ. Press, Princeton, NJ.

[Kuh53] Kuhn, H. W. (1953), Extensive games and the problem of information, *in* "Contributions to the Theory of Games" (H. W. Kuhn and A. W. Tucker, Eds.), Vol. II, pp. 193–216, Princeton Univ. Press, Princeton, NJ.

[Mye91] Myerson, R. B. (1991), "Game Theory: Analysis of Conflict," Harvard Univ. Press, Cambridge, MA.

[Poo96] Poon, C. K. (1996), "On the Complexity of the st-Connectivity Problem," Ph.D. thesis, University of Toronto.

[PR94] Piccione, M. and Rubinstein, A. (1994), "On the Interpretation of Decision Problems with Imperfect Recall," Working Paper No. 24–94, The Sackler Institute for Economic Studies, Tel-Aviv University.

[Rag94] Raghavan, T. E. S. (1994), Zero-sum two-person games, *in* "Handbook of Game Theory" (R. J. Aumann and S. Hart, Eds.), Vol. 2, chapter 20, pp. 735–759, Elsevier Science Publishers B.V.

[Yao77] Yao, A. C. (1977), Probabilistic computations: Towards a unified measure of complexity, *in* "Proceedings of the 18th Annual Symposium on Foundations of Computer Science."