Infinite Games: Randomization, Computability, and Applications to Online Problems

(Preliminary Version)

Xiaotie Deng and Sanjeev Mahajan
School of Computing Science
Simon Fraser University
Burnaby, BC V5A 1S6

Abstract. *We are interested in understanding the exact power randomization provides to online algorithms. We study this problem in terms of computability, using the formulation of online problems as closed or F_σ infinite games [BBKTW, RaS]. Thus, our discussion will mainly focus on the case when our player follows a computable strategy and the adversary may use any strategy, which formulates the notion of computer against extremely formidable nature. In this context, we say that an infinite game is semicomputably determinate if either the adversary has a winning strategy or our player has a computable winning strategy.*

We show that, whereas all open games are semicomputably determinate, there is a semicomputably indeterminate closed game. Since we want to prove indeterminacy result of closed games and the adversary's strategy set is uncountable and our player's strategy set is countable, our proof for the indeterminacy result requires a new diagonalization technique, which might be useful in other similar cases.

We also show that, there exists an infinite F_σ game such that there is a $1-$competitive randomized computable strategy, but there is no $\alpha-$competitive computable deterministic strategy for any $\alpha > 0$. A slightly weaker result can be obtained if we insist that the game be closed. By defining a smoothly growing cost function, we can associate an artificial online problem with the game constructed above. For this problem, there is a simple competitive randomized algorithm but no computable competitive deterministic algorithms. We also obtain an indeterminate game for which both players have a simple randomized winning strategy against all the deterministic strategies of the other.

Section 1. Introduction.

One of the motivations of our study is the following question: How much more power does randomization provide to solve online problems than deterministic algorithms? Formulating online problems as closed and F_σ games, respectively, [BBKTW, RaS] proved that, when playing against offline adaptive adversary, randomization does not yield more competitive algorithms. Raghavan and Snir also noticed that although the randomized strategy may be computable, the deterministic strategy guaranteed by their theorem may not be computable [RaS]. We confirm this belief by showing that there is an infinite F_σ game for which we have a simple randomized competitive strategy but there is no computable competitive deterministic strategy.

The differences in these results, which manifest a myth of randomization, arise out of the distinction between determinacy and semicomputable determinacy of infinite games. When each player has complete information and each has unlimited power of computation, games formulating online problems are determinate [RaS, Mar]. That is, either there is a strategy for player I which beats all the strategies of player II, or, there is a strategy for player II which beats all the strategies of player I. However, in some cases, it may not be realistic to assume that the two players have the same computational power. One such situation is in the task of designing online algorithms to serve an unknown sequence of future requests, such as server problems [MMS] and metrical task systems [BLS]. Robot navigation in unknown world [PY, BRS], and robot learning [RiS, DP] are similar situations. In all such cases, while we may assume that the adversary may have unlimited power, our player is restricted by the computing machinery that is available to it. [Pa] noticed the asymmetry of players in terms of computational complexity by formulating nature as an amiable indifferent adversary: a randomized adversary with equal probabilities on

choices of its moves.

The study of situations dealing with unknown future events leads to the concept of *competitiveness*: An online algorithm is $\alpha-$competitive if its cost on any sequence of requests is within a factor of α of the optimal cost by an algorithm knowing all the future requests [KMRS]. For one problem of this type, the $k-$server problem, a matched upper bound and lower bound of k is conjectured [MMS]. The conjecture remains unsolved in spite of many efforts, even though it is shown to be true on several particular metric spaces [CL, CKPV,MMS]. The above mentioned result in [BBKTW, RaS] provides another approach to solve the conjecture. Thus, if one can design a $k-$competitive randomized online algorithm for the $k-$server problem, the above conjecture is proven even though one may not be able to design a computable deterministic algorithm. This approach depends very much on the possibility of a k-competitive algorithm against the offline adaptive adversary. However, all known $k-$competitive randomized algorithms for the server problem are against an online adaptive adversary [RaS, CDRS]. The failure in resolving the conjecture for general metric space, however, may not be attributed to the conjecture being false: *The optimal strategy may simply be noncomputable.*

When we require that our player use only computable strategies (but the adversary is not so constrained) and we call strategy sets of the game to be *semicomputable*. If either our player or the adversary player has a winning strategy under this restriction, we say the game is semicomputably determinate. Our new result depends on the construction of a semicomputably indeterminate closed game.

In fact, it would be much easier if we only want to construct a semicomputably indeterminate game. For this purpose, we can simply take the list of all the computable strategies each player can have. Following standard diagonization technique, we can obtain two subsets A, B of infinite paths of plays such that for each computable strategy σ of the adversary, there is a computable strategy τ_σ of our player with the path $< \sigma, \tau_\sigma > \in A$; for each computable strategy τ of our player, there is a computable strategy σ_τ of the adversary with the path $< \sigma_\tau, \tau > \in B$. Let A be the winning set of our player and B be the winning set of the adversary, we have

[0] There exists an indeterminate game when both players are restricted to use only computable strategies.

Put all the other paths in the winning set of our player, we see that this becomes a semicomputably indeterminate game.

[0′] There exists a semicomputably indeterminate game.

With possible applications to online problems in mind, however, we want to study the semicomputable determinacy of F_σ games and closed games. In particular, we want to know whether F_σ games (that is, games in which the winning set is a countable union of closed sets) are semicomputably determinate; or there is a closed semicomputably indeterminate game. Our main results are:

[1.] There is a semicomputably indeterminate closed game.

In contrast, we have:

[2.] All open games (open with respect to our player) with a finite adversary choice space are semicomputably determinate.

To study randomized algorithms for infinite games, we also need to specify the proper probability distribution. Raghavan et al. formally defined it through a probability distribution over the strategy space of our player [RaS]. When the adversary's strategy is specified, it induces a probability distribution over the pruned tree according to the adversary strategy. Halpern and Tuttle had a similar idea for distributed systems [HT]. Thus, a statement about a randomized algorithm is true iff it is true for all the pruned trees. We prove

[3.] There exists a semicomputably indeterminate F_σ game on which there is a randomized computable strategy which wins almost surely.

We can also prove a slightly weak result for a closed game:

[3′.] For any $\epsilon > 0$, there is a closed semicomputably indeterminate game on which there is a randomized computable strategy which wins with probability $1 - \epsilon$.

When none of the players is restricted in their computational power, we have

[4] There exists an indeterminate game for which both players have a randomized strategy which wins against all the deterministic strategies of the other player.

Section 2 will introduce the necessary notations and definitions. We discuss semicomputable determinacy of infinite games in Section 3. In Section 4, we look into possible implications to online problems of our

main result. Section 5 will discuss randomized strategies for indeterminate games. Section 6 concludes the paper with remarks and many open problems.

Section 2. Definitions and Notations.

An infinite game is described as an infinite tree on which the two players make their moves in turn, starting at the root. We distinguish the players by naming Player I *the adversary* and Player II *our player*. The set of all the infinite paths is partitioned into two subset A and B, where A is the winning set for our player and B is the winning set of the adversary. A strategy τ (σ) for the adversary (our player) corresponds to a pruned tree T_τ (T_σ) from the original game tree on which each branching at even (odd) levels is pruned to allow at most one possible child. The resulting play (σ, τ) for a given τ and a given σ is specified by a path in the game tree. We call σ a *winning strategy* for our player if for every τ, (σ, τ) belongs to A. The winning strategies for the adversary are defined similarly. Finally the game is *determinate* if either player has a winning strategy.

We now define a topology on the set of all plays. A subset S of all the paths is defined to be *open* iff for every path $(a_0, a_1, ...) \in S$, there exists a number n such that, for all $b_{n+1}, b_{n+2}, ...$,

$$(a_0, a_1, ..., a_n, b_{n+1}, b_{n+2}, ...) \in S.$$

A set of paths is *closed* if it is complement of an open set.

A game is *open* (*closed*) if the winning set of our player is open (closed). In their classical paper on infinite games [GS], Gale and Stewart showed that all open and closed games are determinate, and that there exists a game that is indeterminate. Martin [Mar] then showed that all Borel games are determinate (a game is *Borel* if the winning set of Player I (or II) is Borel under the topology defined above). Observe that, in classical infinite game theory, there is no restriction on the strategies in terms of computability.

In [BBKTW], online problems are formulated as finite games, and in [RaS], an infinite game formulation is given. Depending on the criteria of competitiveness, one may gets different winning sets for an online problem. Let us denote by OC, the cost of our player and by AC, the adversary's cost. A simple $\alpha-$competitiveness requirement is defined by $OC \leq \alpha \cdot AC$. We will call it the strong *competitive* condition. When the cost function is accumulative, e.g., in the case of the server problem, strong competitiveness will will define a closed game: The adversary wins iff the play reach a node the condition is violated. Raghavan and Snir use a formulation which allows an arbitrary additive constant, which gives rise to a game with a winning set in F_σ (countable union of closed sets):

$$\cup_i \cup_j \cap_{k \geq j} \{(x_0, x_1, \cdots) : OC(x_0, \cdots, x_k) - \alpha \cdot AC(x_0, \cdots, x_k) \leq i\}.$$

Thus, infinite paths of constant cost are in the winning set. We call it the weak *competitive* condition. Even though the weak competitive condition is more general, it suffers in the situation when the cost of the online problem is bounded: It can be $0-$competitive in that case! Therefore, we may want to impose a requirement that an online algorithm be at most $1-$competitive.

W.l.o.g., we assume that each player has two choices at each turn of their play. A randomized strategy for our player is a function that makes an assignment of probability to the choices $a0, a1$ depending on the position of the node on the game tree. We say a randomized strategy is *computable* if the probability distribution on the choice space is a computable function. As noted in [RaS, HT], a statement about a randomized strategy is true if it is true for all adversary strategies. Thus, given a randomized strategy, we consider each adversary deterministic strategy τ, and the induced probability distribution for the pruned tree T_τ. We specify a topology and a probability measure on the smallest $\sigma-$algebra generated by the topology, by specifying them on all the basic open sets: A basic open set U is specified by a node x on T_τ such that it contains all the paths passing through x and its probability measure is the probability the randomized strategy reaches x. The measure is extended to all the Borel sets in the topology by the standard method [CT]. When we specify an adversary strategy τ, similar method is applied to define the conditional distribution on the pruned tree T_τ. Again, a randomized strategy is $\alpha-$competitive almost surely, iff for all the pruned tree T_τ, it is $\alpha-$competitive almost surely with respect to this probability distribution.

Section 3. Semicomputable Determinacy of Infinite Games.

While all Borel games are determinate [Mar], we would like to know, under what topological conditions, a game is semicomputably determinate. First, we have

Theorem 1. There is a semicomputably indeterminate closed game.

We give both players two choices of actions: $r0, r1$ for the adversary and $a0, a1$ for our player. We first give some intuitions on the proof of the theorem. We need to partition the set of all the paths into two sets A (the winning set for our player which is closed) and B (the winning set for the adversary) such that for each computable strategy σ of our player, there exists an adversary strategy τ such that (σ, τ) is in B (call it condition C1), and for each adversary strategy τ of the adversary, there exists a computable strategy of our player σ such that (σ, τ) is in A (call it condition C2). C1 and C2 force certain plays to be put in A and B respectively, and we should make sure that (A, B) is a partition. Moreover, we want a construction which makes A a closed set.

Observe that the indeterminacy proof given in [GS] cannot be translated into this case. Our result is obtained via a new method which may be useful in other similar situations.

We construct A and B in stages. Initially both are empty. Let the computable strategies of our player be ordered as $\sigma_i, i = 0, 1, 2, \cdots$. Say that a strategy σ is *killed* in stage j, if we put $(\sigma, \tau') \in B$ for some τ' of the adversary in B in stage j. Similarly for an adversary strategy. At each stage, we kill at least one σ and perhaps an uncountable number of τs, so that A and B remain disjoint and make sure that each σ and each τ is killed in some finite stage without destroying the disjointness criterion. An indeterminate game is thus constructed. The construction will guarantee that A so constructed is closed. Now we give the technical details of the result.

Proof of Theorem 1. For simplicity, we assume each player has two choices at each step of their plays: The adversary has move $r0, r1$ and our player has move $a0, a1$. We list all (computable) strategies our player in set

$$\Sigma = \{\sigma_0, \sigma_1, \cdots, \sigma_n, \cdots\}$$

such that σ_0 is the strategy that choose move $a0$ all the time. Informally, we need to construct a game with winning sets A for σ's and B for τ's such that

[C1] for each $\sigma \in \Sigma$ there is $\tau \in \mathcal{T}$ and $(\sigma, \tau) \in B$;
[C2] for each $\tau \in \mathcal{T}$ there is $\sigma \in \Sigma$ and $(\sigma, \tau) \in A$.

Construction of A, B. Initially, we set

$$A = (r0, a0, r0, a0, \cdots, r0, a0, \cdots)$$

and $B = \emptyset$. Let us denote the root of the game tree to be Level 0. Incrementally assign level number to the tree. We will prune the tree in levels. First Level 1 is processed, and then we show inductively how to process Level n for each $n = 2, 3, \cdots$.

Level 1. Denote by $\mathcal{T}_1$ all the strategies making first request as $r1$. We assign the paths $\{(\sigma_0, \tau) : \tau \in \mathcal{T}_1\}$ to the set B_1 and all the other paths starting with $r1$ are assigned to A_1. Update $A \leftarrow A \cup A_1$ and $B \leftarrow B \cup B_1$. Thus, Condition [C1] holds for σ_0 and Condition [C2] holds for all $\tau \in \mathcal{T}_1$. At Level 2, according to the choice of our player, the strategy set for our player is partitioned into two subset Σ_0, Σ_1, where the lists for Σ_0, Σ_1 keep the same order as the list in Σ.

Level $2i$. Inductive assumption for the pruning process at the end of Level $2i$. Each remaining node at Level $2i$ is a descendent of the adversary playing $r0$ at all the past i requests. Thus, each node can be denoted by an i–bit binary number corresponding plays made by our player from the root to the node. At node j, Σ_j represents all our player strategies which are consistent with j up to this node. Σ_j's, $j = 0^i, \cdots, 1^i$, form a partition of the remaining members in Σ which does not satisfy Condition [C1] yet. All the adversary strategies remaining at node j are those which makes i consecutive requests of $r0$'s, when played against our player which answers j correspondingly. We will denote them by $\mathcal{T}_j$.

Level $2i+1$. Consider each node independently. W.l.o.g., let's look at node 0^i. Let

$$\Sigma_{0^i} = \{\sigma_{0^i 0}, \sigma_{0^i 1}, \cdots, \sigma_{0^i n}, \cdots\}.$$

Denote by $\mathcal{T}_{0^i 1}$ all the strategies in $\mathcal{T}_{0^i}$ which makes the $(i+1)$–st request as $r1$. We assign the paths $\{(\sigma_{0^i 0}, \tau) : \tau \in \mathcal{T}_{0^i 1}\}$ to the set $B_{0^i 1}$ and all the other paths starting from 0^i and continuing with $r1$ are assigned to $A_{0^i 1}$. At Level $2i+2$, according to the choice of our player, the strategy set for our player is partitioned into two subset $\Sigma_{0^i 0}, \Sigma_{0^i 1}$, where the lists for $\Sigma_{0^i 0}, \Sigma_{0^i 1}$ keep the same order as the list in Σ. Thus, Condition [C1] holds for $\sigma_{0^i 0}$ and Condition [C2]

holds for all $\tau \in \mathcal{T}_{0^\frown 1}$. We also do the similar operations on all the nodes j of i bits. For all j of i bits, Condition [C1] holds for σ_{j_0} and Condition [C2] holds for all $\tau \in \mathcal{T}_{j1}$. Update the set A and B by assigning $A \leftarrow A \cup_{j=0}^{1^i} A_{j1}$ and $B \leftarrow B \cup_{j=0}^{1^i} B_{j1}$.

Correctness Proof. Now we prove that Conditions [C1] and [C2] are true for all adversary strategies and our player strategies. Notice that our player's strategies are first enumerated in the set Σ and the ordering is kept when it is partitioned at each level. For the first strategy σ in Σ_j, there is an adversary strategy τ such that $(\sigma, \tau) \in B$, according to our pruning process. Therefore, for each $i = 1, 2, \cdots$, σ_i satisfies Condition [C1] no later than Level $2i$ in our construction. To prove that Condition [C2] holds for all adversary strategies, we consider two cases: one is the case the adversary plays $r0$ all the time; another is the case the adversary plays an $r1$ at least once for some strategy. The first case is done by the initial assignment of the set A. For the second case, we notice that, for any other strategy τ of the adversary, it will play an $r1$ at least once for a strategy of our player at a finite level. If the strategy of our player is not a computable strategy, we can simply truncate the infinite strategy at that finite level and append it by always playing $a0$. This will be a computable strategy $\sigma(\tau)$. Suppose j is the node for the first step the adversary plays an $r1$, then the adversary strategy τ will lose to $\sigma(\tau)$ at one path in $\mathcal{T}_{j1}$.■

In contrast, all open games with a finite choice space for the adversary have enough mathematical structure to make them semicomputably determinate.

Theorem 2. All open games are semicomputably determinate, if the choice space for the adversary is finite.

Since the strategy of our player has to be computable, classical game theory results about determinacy and indeterminacy cannot trivially carry over in this situation. However if we are careful about computability issues, the classical game theory proofs can be modified to obtain the result The proof is thus omitted. We should, however, introduce a simpler proof for the case when the choice space of our player is also restricted to be finite.

Lemma 3. For every open game, either there is an adversary strategy which wins over all the strategies of our player, or there is a computable strategy of our player which wins over all the strategies of the adversary.

Proof: Suppose no adversary strategy wins over all computable strategies of our player. Since open games are determinate[GS,Mar], then there is a strategy σ of our player which wins against all the adversary strategies(though σ may not be computable.) Consider the pruned tree T_σ. Since A is open, for each infinite path in T_σ, there is a node x on the path such that all the paths passing through x are in A. We can thus remove all the children of x, and all the siblings of x as well as their children, from the tree T_σ without changing the win/lose situation of the tree. The game tree thus pruned has no infinite path. Since both players have only a finite number of choices at each node, the pruned tree is finite. Thus, if there is no adversary winning strategy, our player can simply code the structure of the pruned tree and choose its moves accordingly. We have thus reduced the strategy σ to a computable strategy. ■

Theorem 2 follows immediately from the above lemma, if our player's choice space is also finite. Notice that the above lemma also holds even if we restrict our player to use finite state machines. We thus have the following corollary.

Corollary 4. For closed games, either there is a finite state adversary strategy which wins against all strategies of our player or there is a strategy of our player which wins against all adversary strategies, if the choice spaces for both the adversary and our player are finite.

For the notion of strong competitiveness, winning set of our player is closed. If there is no deterministic winning strategy of our player, then the winning strategy of the adversary will enable us to prune the tree to a finite tree, according to the above corollary. We will thus easily conclude that there is no competitive randomized strategy for our player. The result of [BBKTW, RaS] for infinite games follows immediately. The above corollary also implies if we allow our player to use unlimited power, we only need to look for lower bound by adversaries with a simple computational power: finite state machines.

Section 4. Applicatoins to Online Algorithms.

While online problems are formulated as closed and F_σ games, we would also like to formulate closed and F_σ games as online problems such that there is a

winning strategy for our player in a given game iff there is an $\alpha-$competitive online algorithm for the corresponding online problem. This may not be true in general. However, for games constructed in this paper, we want make sure that the above condition is satisfied.

First, we construct a game similar to the one given in the last section for this goal.

Theorem 5. There exists a semicomputably indeterminate F_σ game such that there is a computable randomized strategy for our player which wins almost surely.

Before go into the proof, we notice that the following two corollaries derived from the theorem give us the desired results for the strong competitiveness and the weak competitiveness, respectively.

Corollary 6. There exists an online problem for which there is no computable deterministic strong competitive strategy but there is a computable randomized strong $1-$competitive strategy(a.s.).

Corollary 7. There exists an online problem of accumulative cost such that there is a computable randomized weak $1-$competitive strategy (a.s.), but there is no computable deterministic weak competitive strategy.

Proof of Theorem 5. We follow a similar construction to the game in the last section. The change is that in forming the sets A and B, we put all the paths irrelevant to the indeterminacy in A instead of B. Initially, we will put all the paths (σ_i, τ_0) into A, where τ_0 is the strategy that always requests $r0$. In level one, we will choose one strategy $\tau^1 \in \mathcal{T}_1$ and put (σ_0, τ^1) in B_1 and all the other paths starting with $r1$ are put into A_1. Similarly, at node j, we will choose one strategy $\tau^j \in \mathcal{T}_j$ and put (σ_{j_0}, τ^j) in B_{j1} and all the other paths starting at node j and continuing with $r1$ are put into A_{j1}. All other constructions follow the same pattern. Similar to the proof in Theorem 1, the game can be shown to be semicomputably indeterminate.

Consider the randomized algorithm which always chooses $a0, a1$ with probability $0.5 : 0.5$. We claim that this simple (computable) randomized algorithm wins almost surely. Consider a pruned tree T_τ corresponding to an arbitrary adversary strategy τ. From the construction of the winning set A, when the adversary first chooses $r1$, the branch of T_τ starting from that node will contain exact one winning path for the adversary and our player wins almost surely starting from that node. With this observation, we further prune T_τ as follows: Start from the root until a request $r1$ is encountered and delete the branch after that node. Thus, the only infinite paths of the newly pruned tree will contain request $r0$ only. Since those paths are all in the winning set of our player, the randomized strategy wins almost surely in T_τ. Because the game is semicomputably indeterminate, any given computable strategy is doomed to lose to some adversary. ■

We may have different formulations of infinite games as online problems but we shall use the two formulations defined below for our discussion.

If we adopt the concept of strong competitiveness, there is no need to require the adversary's cost grow as the game being played. For the strong condition, we can simply assign cost *zero* to each infinite path in the winning set of the player, and cost *one* to each path in the adversary winning set. Corollary 6 follows immediately by assigning such cost function since the randomized strategy has cost zero almost surely.

However, one may want to have an accumulative cost function such that it increases unboundedly as plays proceed. For each request sequence, we eliminate $r0$ at the head of the sequence until the request $r1$ is at the beginning. We call the remaining request sequence the suffix. If all the requests leading to a node are $r0$, the cost of reaching this node will be 0. For other nodes, the cost will be the number of the requests on *the suffix* before the node which are coincident to an infinite path in the winning set of the adversary. The cost of our player will be the cost of the node it is on. The cost of the adversary will be the minimum cost over all the nodes with the same request sequence (i.e., we consider an offline adaptive adversary.) The cost of an infinite path for our player is defined as the limit of the cost of its intermediate nodes. The cost of the adversary is again defined as the minimum cost over all the infinite paths with the same request sequence. The competitive ratio of an algorithm σ is the supremum over the limits of the ratios of the two costs along all the infinite paths on T_σ.

To prove Corollary 7, we want to use the accumulative cost function defined above. First, let us consider an adversary which makes its first request on $r1$. In the construction of the game, we notice that

there is only one winning path for the adversary from this node on. The adversary's cost will be one since it can avoid that path by serving the first request with an answer which is not on that path. Since all the path except the one in the adversary's winning set in this pruned tree has bounded cost, the randomized algorithm is 1−competitive almost surely. For the pruned tree corresponding to each adversary strategy, we can take those nodes for which all but the last request are $r0$ and the last request is $r\mathbb{1}$. From the construction of the game, the conditional distributions from those branches on are the same as the above case. Since paths with all requests being $r0$ are in the winning set of our player anyway, the result follows. We also notice that similar statements hold when we use the notion of *expected value* for competitiveness instead of *almost surely*. In fact, for the case when the first request is $r1$, the randomized algorithm of our player will incur a cost of i with probability $\frac{1}{2^i}$ for all $i \geq 1$. The expected ratio of cost the randomized algorithm over the cost of any offline adaptive adversary will be two.

If a further restriction that the winning set be closed is imposed, we can choose an open set of measure ϵ which contains (σ_0, τ^1) to put in B_1 and do similar thing to all the B'_{j1}s.

Theorem 5′. For any $\epsilon > 0$, there exists a semicomputably indeterminate closed game such that there is a computable randomized strategy for our player which wins with probability $1 - \epsilon$.

Section 5. Indeterminacy and Randomization

In the section, we discuss indeterminate infinite games and the power of randomization in this case.

The result of [BBKTW,RaS] basically says that, for a determinate game, whenever there is a randomized strategy for Player I, which wins with probability 1, there is a deterministic winning strategy for Player I. If this result is extendible to the indeterminate games, it means that, for every indeterminate game, there is no randomized winning strategy for any of the players. The following theorem gives a negative answer to this question. Moreover, this artificially constructed game has another counter-intuitive implication: Even though Player I has a randomized strategy winning almost surely against all the deterministic strategies of Player II, that randomized strategy does not win almost surely against all the randomized strategies of Player II.

Theorem 8. Assuming Axiom of Choice and Continuum hypothesis, there is an indeterminate game for which Player I has a randomized winning strategy which wins almost surely against any deterministic strategy of Player II, and vice-versa.

The theorem shows that the [BBKTW, RaS] result is the best possible in the sense that there is an indeterminate game which has randomized winning strategies for both players, when the other player is only allowed to use deterministic strategies. This result assumes Axiom of Choice and Continuum Hypothesis. Axiom of Choice seems necessary here because it is not even known if indeterminate games exist in absence of Axiom of Choice.

Proof: We assume that each player has two choices at any point in the game. By Axiom of Choice, we can well-order Player I's deterministic strategies as σ_α for $\alpha < 2^{\aleph_0}$, and Player II's deterministic strategies as τ_β for $\beta < 2^{\aleph_0}$.

Our randomized strategy for either the Player I or II (δ and γ respectively) assigns a probability of 0.5 to each of the two possible moves. We now construct the winning set for Player I and that for Player II so that both these strategies are winning strategies if the other player uses only deterministic strategies.

We need to satisfy the following conditions:

[1.] For each deterministic strategy σ of Player I, only countably many paths in the pruned tree corresponding to σ belong to the winning set of Player I and the rest belong to the winning set of Player II.

[2.] For each deterministic strategy τ of Player II, only countably many paths in the pruned tree corresponding to τ can belong to the winning set of Player II.

It is clear that if we can satisfy these conditions, then δ wins with probability 1 against any deterministic strategy of Player II, and γ wins with probability 1 against any deterministic strategy of Player I (Each path has probability measure 0, and by the countable additivity of probability measure, countably many such paths will have measure 0).

We say a deterministic strategy σ is ***killed*** if we can satisfy Condition 1 for this σ (define killing of τ symmetrically). We kill σ's and τ's in stages. At stage $\alpha < 2^{\aleph_0}$ we kill σ_α and then τ_α making sure that the winning sets of the two players are disjoint. We denote the winning set of Player I by A and the winning set of Player II by B. Initially, they are

empty. They are updated in each stage by transfinite induction on the stages.

At stage α, we put in B all paths of the pruned tree T_{σ_α} corresponding to σ_α, that have not already been put in A. Then we put in A all paths of the pruned tree T_{τ_α} corresponding to τ_α, that have not already been put in B.

This completes the construction. It is easy to see that the disjointness condition of A and B is automatically satisfied when these sets were constructed. We prove Conditions 1 and 2 by transfinite induction on stages. To verify Condition 1, consider stage α. In the pruned tree T_{σ_α} for Player I's strategy σ_α, the paths already put in A are $T_{\sigma_\alpha} \cap A$. Since $A \subseteq \cup_{\beta<\alpha} T_{\tau_\beta}$, $T_{\sigma_\alpha} \cap A \subseteq \cup_{\beta<\alpha} T_{\sigma_\alpha} \cap T_{\tau_\beta}$. By Continuum Hypothesis, each $\alpha < 2^{\aleph_0}$ is either finite or countable. $T_{\sigma_\alpha} \cap T_{\tau_\beta}$ is a single path. Therefore, $T_{\sigma_\alpha} \cap A \subseteq \cup_{\beta<\alpha} T_{\sigma_\alpha} \cap T_{\tau_\beta}$ contains only a countable number of paths. This proves Condition 1. Condition 2 can be proven similarly. ∎

We notice that the above proof still works with minor modifications even if the Continuum Hypothesis is replaced by a strictly weaker axiom, the Martin's Axiom. One of the consequences of Martin's Axiom is that for any cardinal κ strictly between $\aleph_0$ and $2^{\aleph_0}$, union of κ sets (as subsets of R) of Lebesgue Measure 0 has Lebesgue measure 0. The topology that we use for the game tree is similar to the real line, and the probability measure on the paths induced by γ or δ is similar to the Lebesgue Measure. So this consequence applies to our case.

Although the randomized strategy δ for player I (γ for player II) wins against all deterministic strategies of player II (player I), it does not win against all randomized strategies of player II (player I). In particular, δ does not win against γ (and vice versa). Perhaps the power of randomization in this case results from its easy access to all deterministic strategies at once.

We have addressed the situation when one player uses randomized strategy and the other uses deterministic strategy. What happens when both use randomized strategies? Is it possible to obtain an equilibrium solution? That is, is there a pair of randomized strategies of the players such that none can gain by deviating from this randomized strategy? The problem has been long open when the choice space is continuous[Mc]. We conjecture that this is true for games with universally measurable winning set. A set is universally measurable if it is measurable under any probability measure. Even if the conjecture is confirmed, we still need to know if there is an indeterminate game which is also universally measurable. For a complete understanding of the exact power randomization provides to infinite games, we need to resolve these problems.

Section 6. Remarks and Open Problems.

While the [BBKTW, RaS] result is a first step in understanding the relationship of randomized strategies and deterministic strategies in online problems, our study attempts to get a more refined understanding of this relationship in terms of computability. The answer to our main question is not very satisfactory since it is done by an artificially constructed problem. It would be much more interesting if this can be done on natural problems.

Though online problems can be easily formulated as infinite games [BBKTW, RaS], there is no immediate transformation from the latter to the former. Even though we tried to construct a game to emulate the behavior of online problems, one may notice that the construction of the specific game for our main result needs the power of enumerating all computable strategies, which makes the game non-computable. Thus, there is still a gap to be filled between our result and the result of [BBKTW,RaS]. A more legitimate candidate for infinite games as online problems is closed computable games. A closed game is computable, if there is a Turing machine which can test for membership of basic open sets of the winning set of the adversary. We thus have an immediate question.

[1.] Does randomization provide more power to computable closed games?

Another question is whether or not the result of [BBKTW,RaS] can be strengthened to apply to semicomputably determinate games.

[2.] Is there an $\alpha-$competitive *computable* deterministic strategy if there is an $\alpha-$competitive randomized strategy against an off-line adaptive adversary and the game is semicomputably determinate.

[BBKTW] also show that competitive ratio of a strategy versus offline adaptive adversaries is related to that versus online adaptive adversaries by a quadratic function. But the exact relative power of these two types of adversaries is still unknown. In particular, Does there exist an online problem which separates online adaptive adversaries from offline adaptive adversaries?

There are two notions of separability that one can talk about. In the first notion, we ask if there is a randomized algorithm for a problem, which is $\alpha-$competitive (for some $\alpha \geq 1$) against any on-line adversary, but it is not $\alpha-$competitive against some off-line adversary. This question was answered in the affirmative by Raghavan and Snir [RaS]. However there is another notion of separability. In this context, we ask if there is an on-line problem for which there exists an $\alpha-$competitive randomized algorithm (for some $\alpha \geq 1$) against any on-line adversary, but there does not exist any $\alpha-$competitive randomized algorithm against an off-line adversary for the problem. A negative answer to this question would give a positive answer the k-server conjecture for resistive metric spaces [CDRS].

Again, we can separate these two types of adversaries for an artificially constructed finite game but the separation for a natural problem seems to be a real challenge in this field.

Theorem 9. There is a game for which offline adaptive adversary is strictly more powerful than online adaptive adversary.

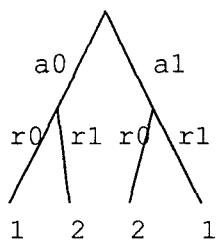

Figure 1: A Game Tree.

Proof. Consider the two stage game tree in Figure 1, where there is only one choice for the request on the first stage and the answers for the second stage. The costs of paths are given at the leaves. The optimal offline cost will always be one no matter what is the request sequence. On the other hand, any online strategy can get no better than two against the worst offline adversary. So by the [BBKTW] result no randomized strategy can perform better than a competitive ratio of 2 against an offline adversary. However it is easy to see that the randomized strategy that answers $a0$ with probability 0.5 and $a1$ with probability 0.5 gives a competitive ratio of 1.5 against the worst adaptive online adversary. Recall that an adaptive online adversary strategy is a pair (α, β) where α is an adversary strategy and β is an algorithm strategy. And (α, β) gives requests according to α and serves its own requests using β. The expected cost is thus calculated according to the probability distribution function of the request sequences induced by our randomized algorithm and the request strategy α, and the cost of serving these request sequences by the algorithm β. ■

Still the question on natural online problems remains open though we believe that they should be separable, at least for some natural online problems.

[3.] Can offline adaptive adversaries be separated from online adaptive adversaries for a natural online problem?

We know that the k-server game whose winning set is defined by set of all those paths which achieve a ratio of less than c for any $c < k$ is semicomputably determinate from the lower bound results of [MMS]. We also know from Fiat et. al.'s result [FRR] that when $c > e^{O(klogk)}$, the k server game is semicomputably determinate for every metric space. (Observe that the 2-server game is semicomputably determinate for any c and any metric space, as we have a computable algorithm [MMS] whose competitive ratio is 2).

[4.] Can we show that the k-server game is semicomputably determinate when c is in neither of these ranges?

Acknowledgment: We are very thankful to Professor Tiko Kameda for many discussions and encouragement, as well as support from his research grants from the Natural Sciences and Engineering Research Council of Canada(NSERC) and the Advanced Systems Institute of British Columbia. The first author also want to acknowledge the support of an International Postdoctoral Fellowship from the NSERC. Thanks are also due to Christos Papadimitriou, Prabhakar Raghavan, and Mike Saks, for their many helpful suggestions and comments on the early versions of this paper, and to Randall Dougherty and Gerald A. Edgar for suggesting some of the references.

References.

[BBKTW] S. Ben-David, A. Borodin, R. Karp, G. Tardos, and A. Wigderson, "On the Power of Randomization in Online Algorithms," *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pp.379-386, 1990.

[BLS] A. Borodin, N. Linial, and M. Saks, "An optimal online algorithm for metrical task systems," *Pro-*

ceedings of the 19th Annual ACM Symposium on Theory of Computing, pp.373-382, 1987.

[BRS] A. Blum, P. Raghavan, and B. Schieber, "Navigating in Unfamiliar Geometric Terrain," *Proceedings of the 23th Annual ACM Symposium on Theory of Computing*, 1991.

[CDRS] D. Coppersmith, P. Doyle, P. Raghavan, and M. Snir, "Random Walks on Weighted Graphs, and Application to On-line Algorithms" *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pp.369-378, 1990.

[CKPV] M. Chrobak, H.J. Karloff, T. Payne, and S. Viswanathan, "New results on server problems," *Proceedings of the First ACM-SIAM Symposium on Discreted Algorithms*, pp.291-300, 1987.

[CL] M. Chrobak, and L.L. Larmore, "An optimal online algorithm for k servers on trees," submitted.

[CT] Y.S. Chow, and H. Teicher, *Probability Theory*, Springer-Verlag, New York, 1978.

[DP] X. Deng, and C. Papadimitriou, "Exploring an unknown graph," *Proceedings of the 31st Symposium on Foundations of Computer Science*, pp. 355-361, 1990.

[FRR] A. Fiat, Y. Rabani, and Y. Ravid, "Competitive k-server Algorithms, *Proceedings of the 31st Symposium on Foundations of Computer Science*, pp. 454-463, 1990.

[GS] D. Gale, and F.M. Stewart, "Infinite games with perfect information," In W.H. Kuhn and A.W. Tucker, editors, *Contributions to the Theory of Games Vol. II, Annals Mathematical Studies, 28*, pp. 245-266. Princeton University Press, Princeton, New Jersey, 1953.

[HT] J.Y. Halpern, and M.R. Tuttle, "Knowledge, Probability, and Adversaries," *Proceedings of the 8th Annual ACM Symposium on Principles of Distributed Computing*, pp.103-118, 1989.

[KMRS] A.R. Karlin, M.S. Manasse, L. Rudolph, and D.D. Sleator, "Competitive Snoopy Caching," to appear in *Algorithmica*.

[Mar] D.A. Martin, "Borel determinacy," *Annals of Math.*, 102: pp.363-371,1975.

[Mc] J. McKinsey, *Introduction to the Theory of Games*, McGraw-Hill Book Company, Inc., New York, 1952.

[MMS] M.S. Manasse, L.A. McGeoch, and D.D. Sleator, "Competitive algorithms for on-line problems," *Twentieth ACM Annual Symposium on Theory of Computing*, pp.322-333, 1988.

[Mos] Y.N. Moschovakis, *Descriptive Set Theory*, North-Holland, New York, 1980.

[Pa] C. Papadimitriou, "Game against Nature," *Proceedings of the 24st Symposium on Foundations of Computer Science*, pp.446-450, 1983.

[PY] C. Papadimitriou and M. Yanakakis, "Shortest paths without a map," *Proc. 16th ICALP*, 1989, pp.610-620.

[RaS] P. Raghavan, and M. Snir, "Memory vs. randomization in online algorithms," ICALP, Italy, 1989.

[RiS] R. L. Rivest and R. E. Schapire, "Inference of Finite Automata Using Homing Sequences," *Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing*, 1989, pp.411-420.

[ST] D.D. Sleator, and R.E. Tarjan, "Amortized efficiency of list update and paging rules," *Communications of the ACM*, **28**(2), pp.202-208, 1985.