Sequential Decision Making with Vector Outcomes

Yossi Azar
Tel Aviv University
azar@post.tau.ac.il

Uriel Feige*
Weizmann Institute
uriel.feige@weizmann.ac.il

Michal Feldman
Tel Aviv University
mfeldman@tau.ac.il

Moshe Tennenholtz
Microsoft Research, Israel,
and Technion
moshet@ie.technion.ac.il

ABSTRACT

We study a multi-round optimization setting in which in each round a player may select one of several actions, and each action produces an outcome *vector*, not observable to the player until the round ends. The final payoff for the player is computed by applying some known function f to the sum of all outcome vectors (e.g., the minimum of all coordinates of the sum). We show that standard notions of performance measure (such as comparison to the best single action) used in related expert and bandit settings (in which the payoff in each round is scalar) are not useful in our vector setting. Instead, we propose a different performance measure, and design algorithms that have vanishing regret with respect to our new measure.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems—*games*; F.2 [**Analysis of Algorithms and Problem Complexity**]: Miscellaneous

Keywords

expert; bandit; vector outcome

1. INTRODUCTION

We consider the following multiround optimization setting. There is one player, m *actions* A_i for $1 \leq i \leq m$ and T *rounds* t for $1 \leq t \leq T$. With every action A_i and round t there is some associated *outcome* of the action for that round, denoted by $O_{i,t}$. The outcome is represented by a vector of dimension d. The values $O_{i,t}$ are a-priori unknown to the player. However, for every t, the outcomes $O_{i,t}$ for all A_i are revealed to the player at the end of round t. The multiround optimization proceeds as follows. In every round t, the player who already observed $O_{i,t'}$ for all $t' < t$ chooses one action, say A_j. Thereafter, the outcomes $O_{i,t}$ of all actions for round t are revealed to the player, and the player accumulates $O_{j,t}$ for the action A_j that was selected in round t. This proceeds for T rounds. After round T, the vector of accumulated outcomes is divided by T (thus averaging over all rounds). Let V denote the resulting vector. The goal of the player is to achieve a most favorable vector V. The quality of vectors is measured using a value function f. Namely, the *value* that the player gets is $f(V)$. Our goal is to characterize the maximum value achievable by the player in such settings, and to design strategies for the player that achieve this value.

To make the setting less abstract, consider the following motivating example. The player is an advertiser (or more accurately, its bidding agent) who wishes to run an advertising campaign on an ad exchange over a month. Ad opportunities from various publishers arrive repeatedly, and the bidding agent needs to decide how to use advertising budget (e.g., on which keywords to bid, and how much), say every hour. This situation can be modeled using our framework, where the bidding agent needs to make real-time decisions over $T = 720$ instances (i.e., the number of hours in a month), and every instance there are m actions available to the agent, where actions here correspond to different bidding strategies. On any given instance t, the agent, using information available to it from previous instances, may choose an action for that instance. The outcome of the action is represented by a vector, where each component of the vector specifies some aspect of the campaign. For example, one component can be the number of impressions (denote it $N(t)$), another component can be the number of clicks (denote it $C(t)$), and yet another component can be the amount of budget spent (denote it $B(t)$). After every instance, the agent can estimate the outcome of each action for that instance. (For example, using statistics released by the ad exchange.) The same set of actions are available on every instance, but the outcomes of the same action on different instances may differ from each other, and are largely unpredictable. The overall value of the campaign is determined only at the end of T time periods (in our example, a month). The outcome vectors are added up (or equivalently, averaged), and then some value function f is applied to the resulting monthly vector so as to measure its quality. A concrete function may be, for example, $f = \min(N/10, C) - |B - 1000|$, where N, C and B denote the sum of $N(t), C(t)$ and $B(t)$, respectively, over the T time periods. This function has two aspects of

*Work carried out at MSR Herzlyia.

ITCS'14, January 12–14, 2014, Princeton, New Jersey, USA.

ACM 978-1-4503-2698-8/14/01 ...$15.00.
http://dx.doi.org/10.1145/2554797.2554817.

non-linearity: The first term is a non-linear aggregation of two linear components, and the second term is a non-linear operation applied to a single component. The goal of the player is to run a campaign (that is, choose actions on each instance) that maximizes the value of f at the end of the month.

It should be noted that while this work has been motivated by challenges in advertising and pricing, our goal here is to present and study a stylized model, extending upon previous theoretical work, and not to provide concrete implementation that can be used "as is" to solving these highly practical problems. Nonetheless, in many cases, our results extend beyond these that are stated in the formal results, and thus are more applicable than might be first perceived. Some of these issues are elucidated in Section 4, which perhaps allows the reader to better understand the scope of applicability of our results.

Finally, while the guarantees given by our algorithms are appropriate for multiround optimization settings, but not necessarily so for multiround *games*, our framework is appropriate for 0-sum game settings with vector outcomes. A particular application of interest is a situation in which a *master* player who plays on behalf of several players in a repeated game, wishes to guarantee good performance to all the players under his control (independent of the actions of the other players). A related setting, motivated by [10], has been studied in [3], which considered a player who observes the actions taken by the opponent but not the obtained payoff. We study the complementary problem, where the obtained payoff is observed, but not the action. See Section 5 for a discussion of these issues.

1.1 A scalar setting and restrictions on the value function

To gain intuition to our problem and put it in context of earlier work, it is instructive to first consider the simpler case in which the outcome is one dimensional, namely, $d = 1$. If in addition we impose the condition that f is the identity function, we recover exactly a well known setting often referred to as *expert algorithms*. In the corresponding literature, what we refer to as actions is referred to as *experts*. Informally, the main result associated with expert algorithms is that the player has a strategy that obtains a value essentially as good as the value of the best single expert. This well known result suggests that we should be seeking a similar result in our setting. However, as we shall see, there are substantial differences between our setting and the classical expert setting.

Let us comment first that the aspect that specialized our model to the known expert setting was not really the issue of a one dimensional outcome, but rather a choice of f that is linear. Even if the outcome is multidimensional, if f is linear, then applying f commutes with averaging (applying f to the average of outcome vectors is the same as first applying f to each outcome vector and then averaging), and hence in every round it suffices to consider only a one dimensional value (rather than a multidimensional outcome) and average all values at the end.

Given the above comment, we return to the special case that $d = 1$, but now f is not linear. The first question that we address is whether any function f is reasonable in our context, or whether we must place some restrictions on f. The answer of course depends on what we want to achieve. Let us put forward one goal that seems to be a minimum requirement in our setting.

Conservative goal. If among all actions there is one action (say, action A_1) that on every round gives the same outcome V, then the algorithm should be able to obtain a value of at least $f(V) - o(1)$, where the $o(1)$ term is a term that tends to 0 as T grows.

To achieve an $o(1)$ term as in the conservative goal, we need to have f continuous. Otherwise, choosing the desirable action A_1 on all rounds but one might still not give a value of $f(V) - o(1)$. Continuity is a qualitative property, whereas we will wish to achieve quantitative guarantees (give an explicit bound on the $o(1)$ term). A quantitative version of continuity is a Lipschitz condition. Namely, for some $c > 0$ and every two vectors u and v, we have that $|f(u) - f(v)| \leq c|u - v|$ (where distance between vectors is computed in the ℓ_2 norm). The Lipschitz condition ensures that if the player manages to attain a vector close (in ℓ_2 distance) to a vector of high value, then the player also attains high value. To make the Lipschitz constant c meaningful, we use the convention (that can be attained by some simple scaling) that outcome vectors have ℓ_2 norm at most 1, and that f when applied to such vectors has value in the range $[-1, 1]$.

A Lipschitz condition by itself will not suffice in order to attain the conservative goal.

Example 1. Consider the function $f(x) = |x|$, attaining its maximum at $x = \pm 1$. Let A_1 be an action that on the first $T/2$ rounds gives outcome $+1$, and let A_2 be an action that on the first $T/2$ rounds gives outcome -1. In the last $T/2$ rounds, either both actions give the outcome $+1$, or both give -1, but before completing round $T/2+1$ the player does not know which case holds. Note that in either case, the conservative goal requires that the player achieves a value of $1 - o(1)$, but regardless of what the player does in the first $T/2$ rounds, his average value over these two cases is at most $1/2$.

To overcome such negative examples, we shall require that f is *quasiconcave*. Recall that a function f is *convex* if for every $0 \leq \lambda \leq 1$ and vectors u and v, we have $f(\lambda u + (1-\lambda)v) \leq \lambda f(u) + (1-\lambda)f(v)$, and *concave* if for every $0 \leq \lambda \leq 1$ and vectors u and v, we have $f(\lambda u + (1-\lambda)v) \geq \lambda f(u) + (1-\lambda)f(v)$. A function is *quasiconcave* if for every $0 \leq \lambda \leq 1$ and vectors u and v, we have the weaker condition $f(\lambda u + (1-\lambda)v) \geq \min[f(u), f(v)]$ (and for *quasiconvex* we have $f(\lambda u + (1-\lambda)v) \leq \max[f(u), f(v)]$). Equivalently, for every c, for quasiconvex functions the region of vectors for which $f(v) \leq c$ is convex, and for quasiconcave functions the region of vectors for which $f(v) \geq c$ is convex. An example of a quasiconcave function that is not concave (in fact, it is strictly convex) is the scalar function $f(x) = x^2$ over the domain $x \geq 0$. For the motivating example that we started with, quasiconcavity of f is a natural assumption. It is also natural that a property such as quasiconcavity will be necessary in our setting. It is inherent in our model that the player averages the outcome vectors from different rounds, and quasiconcavity ensures that averaging vectors cannot reduce the value below that of the least valuable vector contributing to the average.

Summarizing the discussion above, we say that f is *admissible* if the following requirements and conventions hold.

1. Outcome vectors have ℓ_2 norm at most 1

2. The value function f when applied to such vectors has value in the range $[-1, 1]$.

3. The value function f is continuous with a Lipschitz constant $c > 0$.

4. The value function f is quasiconcave. Namely, for every $0 \leq \lambda \leq 1$ and vectors u and v, one has $f(\lambda u + (1-\lambda)v) \geq \min[f(u), f(v)]$.

If f is admissible, can we achieve the same goals as those achieved in the standard expert setting? That is, can the player attain a value at least as high as the value of the best action (up to low order terms)? Interestingly, the answer is negative.

Example 2. Consider the concave function $f(x) = -|x|$, attaining its maximum at $x = 0$. As in example 1, let A_1 be an action that on the first $T/2$ rounds gives outcome $+1$, and let A_2 be an action that on the first $T/2$ rounds gives outcome -1. In the last $T/2$ rounds, either both actions give the outcome $+1$, or both give -1, but before completing round $T/2 + 1$ the player does not know which case holds. Note that in either case, there is one action whose average outcome is 0. However, regardless of what the player does in the first $T/2$ rounds, his average value over these two cases is at most $-1/2$.

Example 2 illustrates that to be able to nearly match the performance of the best action, one may need to place additional restrictions on the value function f. Indeed, earlier work of [9] (when adapted to our setting, see more detailed discussion in Section 1.4) can be viewed as implementing such an approach, in combination with placing restrictions on the outcome vectors $O_{i,t}$. However, in this work we do not wish to place such restrictions, as then the results will not be applicable for our intended applications. Instead we shall consider a different performance measure for our algorithms.

1.2 Maxmin comparison classes

When quantifying the performance of an algorithm for the player, one often specifies a class of algorithms against which to compare the performance of the player's algorithm (or using standard jargon, against which to measure the *regret* of the algorithm). Much of previous work on expert algorithms uses as a comparison class those algorithms that in every round perform the same action, and wishes to nearly match the performance of the best algorithm in this class. As Example 2 shows, in our setting it is impossible to do so. Instead, in this work we follow a different approach. We define a collection $\mathcal{B}$, where each $B \in \mathcal{B}$ is a block of rounds that may serve as an *excuse block*. In addition, we define a comparison class $\mathcal{A}$ of algorithms. Let $f_A(B)$ be the value achieved by algorithm A on block B of rounds. Then the value that we attempt to attain is $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$. One interpretation for this expression is that any block from $\mathcal{B}$ can serve as an excuse of why our algorithm did not attain a high value – no algorithm from the comparison class can do well on the block B, and hence we are excused from doing well on the whole sequence of T rounds. A special case of this maxmin comparison class is when $\mathcal{B}$ contains only a single block, that of all rounds (we call this the *standard excuse block*), and $\mathcal{A}$ contains those algorithms that in every round perform the same action. This special case is the same as the comparison class conventionally used for expert algorithms.

Before presenting our results, we introduce the following terminology.

Definition 1. We say that a strategy *asymptotically matches* a target value g, if the expected difference between g and the value obtained by the strategy is $O(\nu(T))$, where T is the number of rounds and $\nu(T)$ is a nonnegative function that tends to 0 as T grows.

An interesting choice for $\mathcal{B}$ is to take as excuse blocks all suffixes. Namely, for $1 \leq t \leq T$, block B_t contains rounds t up to T. We call this the *suffix blocks*. As above, $\mathcal{A}$ contains those algorithms that in every round perform the same action. We call this, the *pure strategies*. The proof of the next proposition appears in Appendix A.

PROPOSITION 1. *If f is admissible and the outcome is one dimensional, then there is a strategy that asymptotically matches $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$, where $\mathcal{B}$ is the suffix blocks and $\mathcal{A}$ are the pure strategies.*

The requirement that the outcome is one dimensional is essential in Proposition 1. For multidimensional outcomes, suffix blocks do not suffice, and neither does a natural larger class of blocks. This is established in the following theorem, whose proof appears in Appendix A.

THEOREM 1. *Even for admissible f and when $\mathcal{A}$ are the pure strategies, no strategy asymptotically matches $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$ for $\mathcal{B}$ that contains all single rounds, all prefixes and all suffixes.*

Remark. Our proof of Theorem 1 has interesting consequences beyond those stated in the theorem. Consider the following *transparent version* of the multiple round optimization problem: in every round the player is first told what are the outcome vectors associated with each action on that particular round, and only then needs to choose an action. Then if any round can serve as an excuse block, the player now has a strategy that achieves at least $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$. Namely, in each round the player chooses the action that gives the outcome vector of highest value, and by quasiconcavity of the value function f, this gives the desired global guarantee. However, if the excuse blocks are only all prefixes and all suffixes, then the proof of Theorem 1 actually shows that no strategy asymptotically matches $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$, not even in the transparent version.

Given the negative results of Theorem 1, in our Theorem 2 we identify a collection of excuse blocks with respect to which the player does have a good strategy.

1.3 Main theorem

For our positive results, the comparison class $\mathcal{A}$ need not be restricted to that of pure strategies. It can be generalized to be that of *fixed mixed strategies*. A fixed mixed strategy is a probability vector α over actions, and the value resulting from it within a block B is that of the α weighted average of the pure strategies, namely, $f(\frac{1}{|B|}\sum_i \alpha_i \sum_{t\in B} O_{i,t})$. The collection of excuse blocks that we use are described next.

Medium size blocks. Blocks of $\sqrt{T}$ consecutive rounds that end at a round that is a multiple of $\sqrt{T}$. For concreteness, let $\mathcal{B}$ contain $\sqrt{T}$ blocks, where block j contains rounds $(j-1)\sqrt{T}+1$ up to $j\sqrt{T}$.

THEOREM 2. *Let f be admissible with Lipschitz constant c, let $\mathcal{B}$ be the medium size collection of blocks, and let $\mathcal{A}$ be the class of fixed mixed strategies. Then there is a randomized strategy that asymptotically matches* $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$.

The proof of Theorem 2 (which appears in Section 2) combines expert algorithms with Blackwell's approachability theorem [6]. The ability to have fixed mixed strategies rather than only pure strategies as the comparison class is a natural consequence of the use of Blackwell's approachability theorem, and a similar extension of the comparison class appears also in [9].

An important aspect of Theorem 2 is that it uses the medium size collection of blocks rather than the standard one (that only contains all rounds as a single block). As implied by Theorem 1, this is unavoidable in our setting. (Of course, some variations on the medium size collection of blocks would also work for us, but Theorem 1 places limitations on what these variations might be.)

From a motivational point of view, the two versions of excuse blocks are incomparable. In a *stock market* example, actions correspond to buying stocks in the stock market. There it is natural to assume that in the long run (say, in a 15 years period) the stock market goes up, but this assumption is not made for periods of medium length (say, three months). Hence it is desirable to use the standard excuse block, comparing our outcome with that of the best stock that we could have bought and held on to, rather than with the collection of medium size excuse blocks, which might contain excuse blocks in which the value attainable is very poor, and this will reflect badly on our guarantees. In the *tennis player* example, actions correspond to supporting a tennis player, and making profit in times in which the tennis player is ranked among the top players in the world. There is no single tennis player that maintains high rank over a long period of 30 years, but within every medium length period (say one year), there always is some player who is consistently ranked high. In this setting it is desirable to use the medium size excuse blocks rather than the standard excuse block.

From a technical point of view, achieving results with respect to the standard excuse block (when possible) are preferable over results with respect to the collection of medium size excuse blocks, since the former can be used to obtain results also for the latter (apply the former algorithm to each medium size block treating it as a standard block, and concatenate the outcomes – the proof that this works uses quasiconcavity of f), but not vice versa.

The proof of Theorem 2 establishes a regret of $\tilde{O}(T^{3/4})$ (or an average of $\tilde{O}(T^{-1/4})$ per round with respect to medium size excuse blocks, where the $\tilde{O}$ notation hides factors that are either logarithmic or independent of T (such as number of actions, or Lipschitz constant c). We next show that this result is tight; i.e., a regret of $\Omega(T^{3/4})$ is unavoidable.

THEOREM 3. *Even with only two actions and scalar outcome, no algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to medium size excuse blocks.*

In Section 3, we establish the robustness of this lower bound. Specifically, we show that a regret of $\Omega(T^{3/4})$ is unavoidable not only with respect to medium size excuse blocks, but also with respect to other natural excuse blocks.

There are some natural directions in which one may want to extend Theorem 2. One of them is to raise the bar and rather than compare the value attained by the player to that attainable by an algorithm from the comparison class in the worst block, compare it to the kth-worst block for some suitably small value k. That is, a single bad block no longer serves as an excuse for bad performance – only multiple bad blocks do. Another is to consider a *bandit* setting in which the player can only observe the outcomes of those actions that he performed (rather than of all actions). With these extensions one can still design strategies for which the expected regret term tends to 0 as T grows, though the rate at which the regret term tends to 0 is slower than that of Theorem 2. These extensions are discussed in Section 4.

1.4 Related work

In the classical *expert* setting (see, e.g., [13]), there are m experts and T rounds. In every round t, every expert j provides some scalar payoff $p_{j,t}$ in the range $[-1,1]$. There is a player that does not know these payoffs a-priori. In every round t the player may select one expert (say, j), receive the payoff from that expert ($p_{j,t}$ in our example), and in addition the values $p_{i,t}$ for all i are revealed. The *bandit* setting (see, e.g., [2]) is the same except that only $p_{j,t}$ for the selected expert j is revealed in round t, but not $p_{i,t}$ for $i \neq j$. The goal of the player in either setting is to maximize the some of payoffs received in al rounds. As the selection of a player may be randomized, one considers the expected total payoff. The basic theorems in these settings relate the expected payoff achievable by the player with that given by the best expert – namely, with payoffs achievable by the best fixed strategy that selects the same expert in every round, with hindsight of knowing which is the expert with highest total payoff. Many bounds have been established over the years for different variants of the problem. Here, we present the best existing bounds (to the best of our knowledge) to the expert and bandit settings described above.

THEOREM 4. *[8] In the above expert setting, there is a randomized strategy for the player that achieves payoff at least* $\max_j \sum_t p_{j,t} - O\left(\sqrt{T\ln(m/\delta)}\right)$ *with probability at least* $1-\delta$ *for every* $\delta \in (0,1)$.

THEOREM 5. *[8] In the above bandit setting, there is a randomized strategy for the player that achieves expected payoff*[1] *at least* $\max_j \sum_t p_{j,t} - O\left(\sqrt{mT\ln(m)}\ln(1/\delta)\right)$ *with probability at least* $1-\delta$ *for every* $\delta \in (0,1)$.

[6] considered vector-valued games (where the payoff of an agent is given by a vector of unexchangeable values), and defined the notion of "approachability" of a convex set in such games. Specifically, a convex set is said to be approachable if the player has a strategy, in the repeated game, that guarantees an average payoff vector (over all rounds) that is approaching the convex set; i.e., whose distance from the convex set goes to zero as the number of rounds goes to infinity. The convex regions that are feasible in this setting were exactly characterized by Blackwell's approachability theorem [6]. Blackwell's approachability theorem applies in a

[1]Note that the $\ln(m)$ factor in the regret expression can be discarded if one is interested in a regret bound that holds in expectation rather than with high probability [1].

special case of the vector setting, when the experts represent pure strategies for the approaching player, the payoff vectors represent payoffs obtained from mixed strategies of the other player, and the payoff matrix of the game is known.

The work closest to ours that we are aware of is that of [9]. A motivating example presented in [9] is as follows. There are m machines and T rounds. In every round t, a load balancing scheduler needs to place a job J_t on a machine. The marginal load $l_{i,t}$ that will be suffered by machine i if job J_t is placed on it is not a-priori known, but is revealed after round t (for all machines, not only for the machine on which J_t was actually placed). After T rounds one has the m-dimensional vector whose jth entry is the total load (over all rounds) placed on machine j. The goal is to achieve a vector that minimizes some function f (e.g., the load of the most loaded machine, or the sum of squares of loads). The problem studied in [9] can readily be seen to be a special case of the problem that we study. It too is concerned with optimizing a function of the sum of outcomes, and the outcome in each round is a vector. As such, much of what we do in our current paper was already done in [9]. This includes placing restrictions on f (it is required to be convex in [9] and quasiconcave in our setting, where the switch between convexity and concavity is explained by a switch between minimization and maximization), using Blackwell's approachability theorem in the proofs, and using a comparison class that allows for fixed mixed strategies. However, there is a major aspect in which our setting is more general than that of [9], which significantly changes the nature of results. In our setting, the outcome of an action can be an arbitrary vector (of norm at most 1). In [9], for every i the outcome of action i is a vector for which all coordinates except for i are 0. This restriction in [9] corresponds to a model in which all load on the machines is a consequence of the jobs that the load balancing scheduler places on them (e.g., nobody else is using the machines except for the scheduler), and hence each round adds load only to the machine on which the corresponding job is placed. This restriction allows [9] to use the standard excuse block (whereas in our more general setting this is impossible, as Example 2 above shows). A major technical ingredient in their proof is to show that in their setting, when f is an ℓ_p norm with $p > 1$, the value of the outcome of the best fixed mixed strategy enjoys a concavity property. The proof uses heavily the fact that the outcome vectors are of special form, and the concavity property is simply incorrect in our more general setting. (To see where the concavity property fails in Example 2 above, consider only the last $T/2$ rounds. Each of the two options given for outcome vectors has poor value, but averaging the two outcome vectors gives good value. This holds regardless of the action chosen, and hence also with respect to the best fixed mixed strategy. Translating this example to the terminology of [9], this contradicts the concavity property.)

Our vector setting can be also seen as a generalization of some nonlinear bandit or expert settings that have previously appeared in the literature. An example of such a setting is that of dynamic pricing with unknown demand [4, 5]. In this setting, a seller has a set of identical items, and is facing n potential buyers, who arrive sequentially. The seller's goal is to maximize profit; to do so, the seller can make a "take it or leave it" offer to each arriving buyer. If supply were unlimited, then this situation corresponds to a standard bandit setting [7, 12]. If, however, the number of items is smaller than n (as is the case in [4, 5]), then the payoff becomes nonlinear and hence a blackbox application of bandit algorithms is inappropriate[2].

To avoid possible confusion, let us emphasize that our framework is different from that of Online Convex Optimization in the sense of [14]. We average the outcome vectors over all rounds, and then compute the payoff of the average. Zinkevich allows a choice of an action vector (which is different from having an outcome vector), and the loss function in a round is convex function of this vector (and revealed only at the end of the round). The main difference is that the loss is computed per round, and then averaged. Our setting should also not be confused with that of [11], where the *strategy* taken by the decision maker can be represented as a vector, while the obtained payoff is a scalar.

2. MEDIUM SIZE EXCUSE BLOCKS

We restate and prove Theorem 2 and Theorem 3.

Theorem 2 *Let f be admissible with Lipschitz constant c, let $\mathcal{B}$ be the medium size collection of blocks, and let $\mathcal{A}$ be the class of fixed mixed strategies. Then there is a randomized strategy that asymptotically matches* $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$.

We break the proof of the theorem into two parts. In the first part, we assume that the algorithm knows which value it needs to approach. The result of this part is summarized in the following theorem.

THEOREM 6. *Let f be admissible with Lipschitz constant c, let $\mathcal{B}$ be the medium size collection of blocks, and let $\mathcal{A}$ be the class of fixed mixed strategies. Let $g = \min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$ be the minimum over all blocks of the value achievable by the fixed mixed strategy of highest value in that block. Then there is a randomized strategy that given g achieves value $g - O(c\sqrt{\log(mT/\delta)}/T^{1/4})$ with probability at least $1-\delta$.*

PROOF. Let K denote the set of vectors u in R^d that satisfy $f(u) \geq g$. By quasiconcavity of f we have that K is a convex set. We are required to design a strategy that over the T rounds obtains an average outcome vector that is either in K or very close to K. This requirement is similar to the requirement in Blackwell's approachability theorem. However, the premises of Blackwell's theorem do not hold, and hence we cannot apply Blackwell's theorem directly. Instead, we shall use algorithms from the (scalar) expert setting on each block separately and use this in order to obtain an approximate version of Blackwell's setting with T/B rounds, where $B = \sqrt{T}$ is the number of rounds in a block. In our proof of Theorem 6 we shall use Theorem 4 (regarding the expert setting) as a blackbox, in combination with the proof technique that is used in the proof of Blackwell's approachability theorem.

We now describe the randomized algorithm. Recall that the outcome of each action is a d-dimensional vector. Partition the T rounds into T/B blocks $B_1, B_2, \ldots$. We describe what our algorithm does in block B_j given the outcomes in previous blocks. Let u^j denote the average of outcome vectors achieved by our algorithm in all rounds up to the beginning of block B^j, with $u^1 = 0$. Observe that u^j is a

[2]A major difference between their setting and ours is that their setting is stochastic while we employ a worst-case approach.

random variable (because our algorithm is randomized), but the algorithm knows its value when block B^j is about to begin. If $u^j \in K$ we refer to the block as a *dormant block*, and in all rounds of block B^j our algorithm performs arbitrary actions. (For out theoretical bounds, it does not matter which actions are performed in dormant blocks, though in practice it may matter. See Section 4.) If $u^j \notin K$ we refer to the block as an *active block*, and then let K^j denote the point in K closest to u^j (in Euclidean distance). At an intuitive level, our algorithm attempts to move towards K^j. Formally, let d^j be a unit vector pointing in the desirable direction of movement (namely, $d^j = K^j - u^j$ normalized to be a unit vector). Within the block B^j our algorithm would like to obtain an outcome vector O^j that maximizes the movement in direction d^j. This is the vector that maximizes the inner product $\langle (O^j - u^j), d^j \rangle$. Observe that this inner product is a scalar and not a vector. Hence within block B^j our algorithm shall employ the expert algorithm from Theorem 4, where in each round the value of each outcome O is considered to be $\langle (O - u_j), d^j \rangle$. This completes the description of the strategy of our algorithm in block B^j, and by induction, on the whole sequence of T rounds.

We now analyze the performance of our algorithm. Let B^j be an active block. Let O_i^j denote the average outcome vector of action A_i on the rounds of block B^j. Within block B^j the premises of Theorem 6 imply that there is a convex combination $\sum_i \lambda_i O_i^j$ (where $\lambda_i \geq 0$ and $\sum_i \lambda_i = 1$) that lies within K. For this particular convex combination it must hold $\langle (\sum_i \lambda_i O_i^j - u^j), d^j \rangle \geq \langle (K^j - u^j), d^j \rangle$. In particular, there must be some action (say A_i) for which $\langle (O_i^j - u^j), d^j \rangle \geq \langle (K^j - u^j), d^j \rangle$. As a consequence of Theorem 4, there is probability at least $1 - \delta$ that the outcome vector O^j of our algorithm in block B^j satisfies

$$\langle (O_i^j - u^j), d^j \rangle \geq \langle (K^j - u^j), d^j \rangle - O(\frac{1}{\sqrt{B}} \sqrt{\log(m/\delta)}) \quad (1)$$

For a vector u, let $|u - K|$ denote the ℓ_2 distance between u and the point closest to u in the convex body K. We consider the sequence of average vectors $u^1, u^2, \ldots$ obtained by our algorithm at the beginning of blocks and upper bound the distance of these vectors from K. It will be more convenient for us to scale the distance by the number of rounds that passed up to that point, e.g., replace $|u^j - K|$ by $D^j = (j-1)B|u^j - K|$. In our analysis we will assume that inequality (1) always holds. This can be enforced on all blocks simultaneously with probability $(1 - \delta)$ by taking $\delta < 1/T^2$ and applying the union bound.

We now show that if inequality (1) always holds then for all $1 \leq j \leq T/B$ we have $D^j \leq O(B^{3/2}\sqrt{\log(m/\delta)})$. The proof is by induction on j. Let j_0 be the largest index satisfying $D^{j_0} < B^{3/2}$. Then $D^{j_0+1} \leq D^{j_0} + 2B$ (because in each round the worst that can happen is that the outcome and K are two antipodal points on the unit sphere). Thereafter, for every $j > j_0$ we claim that $D^{j+1} \leq D^j + O(\sqrt{B \log(m/\delta)})$. This follows from the fact that in block $j+1$ the total backward movement along the direction d^{j+1} is no worse than $O(\sqrt{B \log(m/\delta)})$ (by inequality (1)), and in the perpendicular direction it is no worse than B (because this is the number rounds in a block). Hence $(D^{j+1})^2 \leq (D^j + O(\sqrt{B \log(m/\delta)}))^2 + B^2 \leq (D^j + O(\sqrt{B \log(m/\delta)}))^2$, where the last inequality uses the fact that $D^j > B^{3/2}$ and assumes a change of constant in the O notation. As there are B blocks we have that the final value of D is at most $B^{3/2} + B + O(B^{3/2}\sqrt{\log(m/\delta)}) = O(B^{3/2}\sqrt{\log(m/\delta)})$.

It follows that the final average outcome vector u of the algorithm satisfies $|u - K| \leq O(B^{3/2}\sqrt{\log(m/\delta)}/T) = O(\sqrt{\log(m/\delta)}/T^{1/4})$. By the Lipschitz condition on f the final value is no worse than $g - O(c\sqrt{\log(m/\delta)}/T^{1/4})$. Finally, the requirement that $\delta \leq 1/T^2$ gives the bound claimed in the theorem.

□

Remark. If excuse blocks are of size $\sqrt{T \log(mT/\delta)}$ rather than $\sqrt{T}$, the proof in Theorem 6 gives an improved bound of $g - O(c(\log(mT/\delta)/T)^{1/4})$. However, for simplicity, in this manuscript we fix the size of excuse blocks to be $\sqrt{T}$ and do not attempt to optimize the logarithmic terms.

With Theorem 6 at hand, we are now able to prove Theorem 2.

PROOF. Consider a nested sequence of convex bodies $K_1 \subset K_2 \subset K_3 \ldots$, where for every $1 \leq i < T^{1/4}$ we have that K_i is the set of vectors whose value under f is above $1 - 2i/T^{1/4}$ (recall that f takes on values in the range $[-1, 1]$). Our strategy is similar to that of Theorem 6 with the difference that there is no one fixed convex body K that we attempt to approach, but rather the convex body may change from block to block. Namely, suppose that in block B_j the algorithm K_i played the role of K. Then before block B_{j+1} begins, the algorithm determines whether there was a fixed mixed strategy whose average outcome vector for block B_j lies within K_i. If there was such a strategy, then also in block B_{j+1} the algorithm attempts to approach K_i. If there was no such strategy, then in block B_{j+1} the algorithm attempts to approach K_{i+1}. (In a different version of our algorithm, rather than approaching K_{i+1} the algorithm may attempt to approach $K_{i'}$ for the largest i' that is consistent with all previous blocks. However, doing the conservative change to $i+1$ rather than an aggressive change to i' has certain advantages that will become apparent in Section 4.)

We now analyze the strategy. Let $g = \min_{B \in \mathcal{B}}[\max_{A \in \mathcal{A}} f_A(B)]$ and let K be the convex body containing all vectors whose value according to f is at least g. Our goal is to approach K, though K is not known to the algorithm. Let i be the least index for which $K \subset K_i$. In every block there is a fixed mixed strategy whose average outcome vector lies within K_i. Hence in every block B_j it must be the case that our algorithm attempts to approach some convex body $K_{i'}$ with $i' \leq i$, where i' may depend on j. In some blocks it may happen to be the case that there is no fixed mixed strategy whose average outcome vector lies within the respective $K_{i'}$ (since $K_{i'} \subset K_i$ rather than $K_{i'} = K_i$). On these blocks the expert algorithm that is employed on the block offers no guarantees. Luckily, there can be at most $T^{1/4}$ such blocks, because each such block causes the value of i' to increase. Hence altogether they can contribute at most $2BT^{1/4} = O(T^{3/4})$ to the loss in D (where D is as in the proof of Theorem 6), which translates to a loss of $O(1/T^{1/4})$ in the final value. Another source of loss compared to the proof of Theorem 6 is due to the quantization in specifying the convex body, namely, due to the difference between K_i and K. This incurs an additional loss of at most $2/T^{1/4}$ in the final value. Other than the above two issues, the bounds are precisely like in

the proof of Theorem 6, and hence with probability $1-\delta$ the strategy achieves a value of $g - O(c\sqrt{\log(mT/\delta)}/T^{1/4})$. This completes the proof of Theorem 2.

□

Finally, we show that an average regret of $\Omega(T^{-1/4})$ per round with respect to medium size excuse blocks is unavoidable. The following result, as well as the results in the next section, refer to the total regret. An average regret of $\Omega(T^{-1/4})$ is equivalent to a total regret of $\Omega(T^{3/4})$ as the total regret is simply T times the average regret.

Theorem 3 *Even with only two actions and scalar outcome, no algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to medium size excuse blocks.*

PROOF. In each block, for each of the two actions, the outcome of the first $\sqrt{T} - T^{1/4}\log T$ rounds is set independently at random to be either $+1$ or -1. In each block B_j, for each of the two actions, the outcome of each of the last $T^{1/4}\log T$ rounds is set to be a_j, where a_j is chosen such that the better of the two actions in the block averages to 0 in that block. Hence the value for every excuse block is 0.

Observe that the expectation of $a_j T^{1/4}\log T$ is in the order of $-\theta(T^{1/4})$, because the maximum of two random ± 1 walks of length roughly $\sqrt{T}$ is in the order of $T^{1/4}$. Hence any algorithm is expected to average 0 on the first $\sqrt{T} - T^{1/4}\log T$ rounds and then lose $\Omega(T^{1/4})$ on the last rounds of the block. Hence altogether the algorithm has regret $\Omega(T^{3/4})$, or average regret $\Omega(T^{-1/4})$ per round. □

3. LOWER BOUNDS

In Section 2 we showed an algorithm that obtains a regret of $\tilde{O}(T^{3/4})$ with respect to medium size excuse blocks. We also showed that a regret of $\Omega(T^{3/4})$ is unavoidable. In this section we extend this lower bound and show that a regret of $\Omega(T^{3/4})$ is unavoidable not only with respect to medium size excuse blocks, but also with respect to many other natural excuse blocks.

We begin with the introduction of several classes of excuse blocks.

- *Small interval blocks.* Blocks of up to $\sqrt{T}$ consecutive rounds, starting at an arbitrary round.
- *Binary tree blocks.* T is assumed to be a power of 2. The rounds are placed on the leaves of a full binary tree of depth $\log T$, and every node of the tree represents a block that contains the rounds in its subtree.
- *Sliding window blocks.* Some arbitrary but fixed value ℓ serves as the size of the window. Every interval of ℓ consecutive rounds is a block.
- *Interval excuse blocks.* Every consecutive set of rounds of arbitrary length serves as an excuse block.

Recall that by Theorem 3, even with only two actions, a regret smaller than $\Omega(T^{3/4})$ is unavoidable with respect to medium size excuse blocks. In the proof of the following theorem the number of actions grows with T.

THEOREM 7. *Even with $O(\log T)$ actions and scalar outcome, no algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to small interval blocks.*

PROOF. Each of the $m = \Theta(\log T)$ actions is a random string of ± 1. Each $+1$ counts as payoff of $1 - 1/T^{1/4}$, whereas each -1 counts as payoff of -1.

For a small interval of length ℓ, every action has expected average payoff of $-\ell/2T^{1/4}$ and standard deviation of $\Omega(\sqrt{\ell})$. Hence when $\ell \leq \sqrt{T}$ an action has constant probability of giving positive average payoff on the interval. The probability that no action gives positive payoff is $2^{-\Omega(m)}$. Taking a union bound over all $O(T^{3/2})$ intervals, there is high probability that for every small interval there is an action with positive average payoff.

On the other hand, the player gets an average payoff of $-1/2T^{1/4}$ per round, giving total expected regret $\Omega(T^{3/4})$.

□

In the proof of the following theorem, also the number of dimensions grows with T.

THEOREM 8. *No algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to binary tree blocks.*

PROOF. Blocks of size at most $\sqrt{T}$ are already handled by Theorem 7. Hence it remains to deal with those blocks of size larger than $\sqrt{T}$. There are $O(\sqrt{T})$ such blocks in the binary tree. Each such block will be associated with two fresh dimensions and two fresh actions. Consider one such block of length $\ell > \sqrt{T}$. In the first half of the block one action offers an outcome of $(1,-1)$ in the dimensions associated with the block (and 0 in other dimensions). The other action offers an outcome of $(-1,1)$. In the second half of the block, with probability $1/2$ both actions give $(1,-1)$, and with probability $1/2$ they both give $(-1,1)$. If an action associated with an interval is played outside the interval, it gives an outcome of $(-1,-1)$. The function f computing the payoff of an average outcome vector is the negative of the ℓ_2 norm of the vector.

If on at least half the rounds the player takes actions that are associated with small intervals, the theorem follows from Theorem 7. Hence on at least half the rounds the player takes actions associated with large intervals. For every such action, the expected regret is linear in the number of rounds on which it was played. Hence to minimize regret (when measured as an ℓ_2 norm), the player needs to spread the regret (which is linear in ℓ_1 norm) on as many coordinates as possible, namely, on $O(T^{1/2})$ coordinates. This reduces the regret from $O(T)$ (had it been measured in ℓ_1 norm) to $O(T^{3/4})$. □

We note that unlike the proofs of Theorems 3 and 7, the proof of Theorem 8 only establishes a bound on the expected regret, but does not show that this bound holds with high probability.

THEOREM 9. *For every even ℓ, no algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to sliding window blocks of size ℓ. When $\ell > \sqrt{T}$, this holds even if the number of actions is a constant independent of ℓ and T.*

PROOF. The case of $\ell \leq \sqrt{T}$ is handled by Theorem 7. Hence we may assume that $\ell > \sqrt{T}$. We first prove the theorem under the assumptions that $\ell \leq T/2$ and T is divisible by ℓ.

Recall that ℓ is assumed to be divisible by 2. Partition T into $2T/\ell$ intervals of $\ell/2$ consecutive rounds. A *quadruple*

is four consecutive intervals. A quadruple is of type Q_i if it starts at an interval whose index is i modulo 4. Note that the rounds covered by two quadruples of the same type do not overlap. We shall consider only quadruples of type Q_1 and Q_3. We shall have four actions associated with type Q_1 quadruples, and four actions associated with type Q_3 quadruples. With each quadruple we associate four patterns, $(+1,+1,+1,+1)$, $(+1,-1,-1,-1)$, $(+1,-1,+1,+1)$ and $(+1,-1,+1,-1)$. Given a quadruple (say of type Q_1), with each pattern we associate at random one of the four actions associated with the type of the quadruple, and one fresh dimension (without the player knowing which action is associated with each pattern). The pattern associated with an action shows which outcome to give (in the corresponding dimension) in each of the four intervals that make up the quadruple. Observe that only the last of the four patterns ensures that in every block of ℓ consecutive rounds that lies within the quadruple, the average outcome is 0. The payoff function is the ℓ_2 norm of the average vector.

Observe that for every block of size ℓ there is some action with average payoff 0. However, the player does not know which action is associated with each pattern. Because of that, within a block of length ℓ the player has expected regret $\Omega(\ell)$. As the number of dimensions is $O(T/\ell)$, the total regret is $\Omega(\sqrt{T\ell}) \geq \Omega(T^{3/4})$ (because $\ell > \sqrt{T}$).

To remove the assumption that T is divisible by ℓ, let $T' < T$ be the largest integer smaller than T and divisible by ℓ. Make two copies of the construction above, one with eight actions on the first T' rounds (and giving -1 payoffs on the $T - T'$ last rounds), the other with eight actions on the last T' rounds (and giving -1 payoffs on the first $T - T'$ rounds).

Finally, it remains to deal with the case that $\ell > T/2$. In this case, one cannot fit even a single quadruple in T rounds. However, this does not matter. Let $T' = 2\ell > T$, which is the number of rounds that suffices in order to support a quadruple. Make two independent constructions of a single quadruple and four patterns and four actions on T' rounds, align one of them to start at the beginning of the T rounds, and the other to end at the end of the T rounds. It does not matter that the quadruples spill beyond the borders of the T rounds. □

THEOREM 10. *No algorithm has expected regret smaller than $\Omega(T^{3/4})$ with respect to the set of interval excuse blocks.*

PROOF. One can use the proof of Theorem 7 on blocks of size at most $\sqrt{T}$. To handle blocks of size $\ell > \sqrt{T}$ one would like to use the proof of Theorem 9. However, to do so for every possible value of $\sqrt{T} < \ell \leq T/2$ would be too costly, because than the total number of dimensions would no longer be $O(\sqrt{T})$, and even if the ℓ_1 regret is $\Omega(T)$, the ℓ_2 regret need not be $\Omega(T^{3/4})$. To keep the number of dimensions at most $O(\sqrt{T})$ we use the proof of Theorem 9 only with values of ℓ that are multiples of $1+\epsilon$. To handle other values of $\ell > \sqrt{T}$ we introduce some slackness in the proof of Theorem 9. Namely, rather than give a coordinate ± 1 values, the coordinate is split into two coordinates, where $+1$ corresponds to $(1, -1+\epsilon)$ and -1 corresponds to $(-1+\epsilon, 1)$. The function f is now the ℓ_2 norms of the average outcome vector, but computed only of those coordinates that have negative value (coordinates with positive value cost nothing). □

4. EXTENSIONS

In Theorem 2 we assumed a so called *expert* setting in which after each round the outcomes of all actions for that round are revealed. Theorem 2 (with somewhat different quantitative parameters) extends also to the bandit setting in which only the outcomes of actions chosen by the player are revealed.

THEOREM 11. *Let f be admissible with Lipschitz constant c, let $\mathcal{B}$ be the medium size collection of blocks, and let $\mathcal{A}$ be the class of fixed mixed strategies. Then in the bandit setting there is a randomized strategy that asymptotically matches* $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$.

PROOF. We first adapt the statement and proof of Theorem 6 to the bandit setting. The only change in the proof is a blackbox replacement of the algorithm from Theorem 4 by the algorithm of Theorem 5. This changes the value achieved to $g - O(c\sqrt{m}\log(T/\delta)/T^{1/4})$. [3]

Now we adapt the proof of Theorem 2 to the bandit setting. The difficulty is that in the bandit setting the algorithm does not know after block B_j whether there was a fixed mixed strategy whose average outcome vector lies within K_i. Hence instead we employ a different strategy. If in block B_j the movement away from the respective K_i along the direction d^j turned out to be significantly larger than expected (namely, D^j increased by more than $O(c\sqrt{mB}\log(T/\delta))$), then the algorithm infers (and this inference is correct with probability at least $1-\delta/T$) that block B^j can serve an excuse block for failing to approach K_i, and raises i by 1. The analysis then proceeds exactly as in the proof of Theorem 2. □

How robust are our algorithms? In what follows we emphasize some of the issues that are not stated in the formal results, which can perhaps allow the reader to better understand the robustness of our algorithms.

First, notice that the division into blocks of size $\sqrt{T}$ can be replaced by division into other block sizes, and in fact, not all block sizes need to be the same. This will only change the quantitative guarantees on the regret term, but the qualitative guarantee (of being a term that tends to 0 as the number of rounds grow) will remain. For example, our bounds apply in a realistic setting of a campaign on an ad exchange, which runs for a year, when a block corresponds to a day, in which bidding is done approximately every minute.

Also, our guarantee for being no worse than the best expert on the worst block, can be replaced by referring to the k worst blocks; this is important from a pragmatic perspective as one may claim that some blocks (e.g. days) may be really bad as far as payoffs or budget spent are concerned due to the nature of supply arrival (in the ads example) or demand arrival (in the pricing example). Suppose that for some value g it happens that k of the excuse blocks are such that no fixed mixed strategy can attain g on these blocks. Then our algorithms with no change are still useful, provided that k is not too large. The bounds in Theorem 6 suffer an additive loss of at most $2kB/T$ (because in k blocks the algorithm

[3]The $\sqrt{m}$ term is an exploration cost associated with the bandit algorithm. For simplicity of notation, we equated the space of actions with the space of bandits. We note that in cases where the number of bandits exceeds number of actions (for example, if bandits represent complicated strategies over a small set of actions) the exploration term can be decreased.

might behave erratically), which for $k \leq T^{1/4}$ is smaller than the loss that they already allow. The bounds in Theorems 2 and 11 suffer a larger additive loss of $O(k/T^{1/4})$ (because the algorithm may end up trying to approach K_{i+k} instead of K_i), which is still tolerable when $k = o(T^{1/4})$. The possibility of such sources of error suggest that in practice, in dormant blocks (see proof of Theorem 6), rather than taking arbitrary actions, it may be advisable to try to pick a direction in which f increases and use the expert algorithm (of Theorem 4) in an attempt to move in that direction.

One might also think that our definitions imply that our algorithms would necessarily divide resources such as budget equally among blocks. However, this is not the case: our algorithms may well spend budget at higher rate in some blocks, and compensate for this by spending budget at slower rates at other blocks.

Finally, our algorithm is well defined, and moreover, may potentially work great, even if there is no block in which our experts do well; this is due to the fact that the derivatives of the multi-dimensional changes in different blocks may cancel each other, so that the overall average output may have high value even if the average output of each individual block has low value.

5. A GAME-THEORETIC SETTING

The low regret guarantees given by our algorithms are appropriate for multiround optimization settings, but not necessarily so for multiround games. The distinction is that in multiround optimization settings we think of the sequence of outcomes of every action as being fixed in advance (though unknown to the player), whereas in multiround games the outcomes depend on actions of other players, and the choice of actions that the other players make may depend on the actions of the optimizing player. This point is well illustrated by considering repeated play of the well known prisoner's dilemma game. For our notion of regret, defecting in every round is an optimal policy. But if this game is played against another player whose strategy is to play at each round whatever the other player played in the previous round, the optimal strategy would be to always cooperate (except for the last round). Hence despite being optimal with respect to our notion of regret, always defecting is a strategy that might carry high regret in the multiround game setting (when the player learns after the game that had he chosen to cooperate his payoff would have been much larger). This aspect is not captured by our modeling (and neither by the standard experts and bandits models). On the other hand, for multiround 0-sum games, the multiround optimization framework is appropriate.

A particular multiround 0-sum game setting with vector outcomes was studied in [3]. In that setting there is a *master* player who plays on behalf of several players in a repeated game, and wishes to guarantee good performance to all the players under his control, independent of the actions of the other players. While in [3] the player observes the actions taken by the opponent but not the obtained payoff, we consider here the complementary problem, where the obtained payoff is observed, but not the action. This situation can be modeled as a two player game between players P_1 (the *master* player) and P_2. The master player, P_1, plays on behalf of n players, and therefore the outcome of a round can be viewed as an n-dimensional vector of payoffs, with one coordinate for each player he represents. Payoff matrices are unknown and cannot be inferred (since the actions of P_2 are not observable). The master player can only observe the realized payoff vector (the outcome) after each round of play. As the payoffs for P_2 are never observed by P_1, we wish to obtain guarantees for P_1 that hold regardless of the payoff matrix for P_2, and the worst case from this respect is the 0-sum setting. For symmetric games, a natural goal for P_1 is to maximize the sum of payoffs of his n players, and this task can be cast as a bandit problem. For nonsymmetric games, the situation becomes more complicated. The master player wishes to achieve some goal over the obtained payoff vector, which corresponds to some master payoff function f (that returns a real value v given a payoff vector V). Our work establishes sufficient conditions for a value v to be approachable in the repeated game. In particular, by Theorem 2 (or rather Theorem 11, we shall not distinguish between them here), v is approachable if the master payoff function is quasiconcave, and every medium sized block has an action that achieves v. A natural goal for P_1 is to guarantee good performance to every player he represents. In this case, f is the *minimum* function, given by $f(V) = \min(V_1, \ldots, V_n)$, where V_i is the payoff for player i, averaged over all rounds. Theorem 2 establishes that P_1 has a strategy that guarantees good performance to every player if every medium sized block of rounds has an action that exhibits good performance with respect to every player.

Two remarks are in order here. One is that it is natural and desirable to make use of the extra power offered by Theorem 2 to compare with fixed mixed strategies rather than fixed actions. This is because in any given block, it could be that each action is bad for some player, but a mixed strategy offers a vector of payoffs that is not bad for any player.

The other remark is that in the game setting there will be a natural class of vectors $\mathcal{V}$ such that for every vector $V \in \mathcal{V}$, in each medium size block there is a fixed mixed strategy that achieves V. Moreover, in every block of rounds (regardless of the number of rounds in the block and regardless of whether they are consecutive or not) there is a fixed mixed strategy achieving this vector. This class contains those vectors that in each coordinate have the *product-minimax value* [3] for the corresponding player. The product-minimax value of a player in a game is defined as the value that a player can guarantee herself if the other players announce their most harmful product mixed strategy first, and in response she chooses her best strategy. This value is at least as high as the minimax value and is often higher. In particular, if we let v be the minimum product-minimax value across all players the master represents, then our results imply that v is approachable for the master for the function $f(V) = \min(V_1, \ldots, V_n)$.

6. OPEN QUESTIONS

We do not know if the lower bounds of Section 3 hold when the dimension and/or the number of actions is a constant independent of T (except for those theorems that state so explicitly). When block size can be large (e.g., as large as T) this relates also to the question of whether the bound of $\Omega(T^{3/4})$ is only on the expected regret or also a high probability bound. If the number of actions is k and the complete block can serve as an excuse block then the probability of having no regret is at least $1/k$.

Another situation in which the $\Omega(T^{3/4})$ lower bound need not hold is when every set of rounds is an excuse block (and not only intervals). This case may come up in situations like those discussed in Section 5. The number of possible excuse blocks is then exponential in T rather than polynomial in T which makes the construction of negative examples difficult.

7. REFERENCES

[1] J. Y. Audibert, U. P. Est, and S. Bubeck. Minimax policies for adversarial and stochastic bandits, in. In *Proceedings of the 22nd Annual Conference on Learning Theory, Omnipress*, pages 773–818, 2004.

[2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, Jan. 2003.

[3] Y. Azar, U. Feige, M. Tennenholtz, and M. Feldman. Mastering multi-player games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '12, pages 897–904, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[4] M. Babaioff, S. Dughmi, R. Kleinberg, and A. Slivkins. Dynamic pricing with limited supply. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 74–91, New York, NY, USA, 2012. ACM.

[5] O. Besbes and A. Zeevi. On the minimax complexity of pricing in a changing environment. *Oper. Res.*, 59(1):66–79, Jan. 2011.

[6] D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.

[7] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online learning in online auctions. In *SODA*, pages 202–204, 2003.

[8] V. Dani and T. P. Hayes. How to beat the adaptive multi-armed bandit. *CoRR*, abs/cs/0602053, 2006.

[9] E. Even-Dar, R. Kleinberg, S. Mannor, and Y. Mansour. Online learning with global cost functions. In *22nd Annual Conference on Learning Theory, COLT*, 2009.

[10] M. Feldman, A. Kalai, and M. Tennenholtz. Playing games without observing payoffs. In *ICS*, pages 106–110, 2010.

[11] A. Kalai and S. Vempala. Efficient algorithms for online decision problems. In *COLT*, pages 26–40, 2003.

[12] R. Kleinberg and T. Leighton. The value of knowing a demand curve: Bounds on regret for online posted-price auctions. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '03, pages 594–, Washington, DC, USA, 2003. IEEE Computer Society.

[13] N. Littlestone and M. Warmuth. The weighted majority algorithm. *Informat. Comput.*, 108:212–261, 1994.

[14] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

APPENDIX

A. THE SUFFIX EXCUSE BLOCKS

We restate and prove Proposition 1 and Theorem 1.

Proposition 1 *If f is admissible and the outcome is one dimensional, then there is a strategy that asymptotically matches $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$, where $\mathcal{B}$ is the suffix blocks and $\mathcal{A}$ are the pure strategies.*

Theorem 4 does not directly prove Proposition 1 because in Proposition 1 the player does not need to maximize the outcome, but rather some function f of the average outcome. This function need not be nondecreasing. However, Theorem 4 can be used as a blackbox in order to prove Proposition 1, which we now prove.

PROOF. As f is admissible it is quasiconcave. When the outcome is scalar, this implies that f is unimodular. That is, there is some value $p \in [-1,1]$ such that $f(x)$ is nondecreasing in the range $x \in [-1,p]$ and nonincreasing in the range $x \in [p,1]$. Furthermore, f is continuous with Lipschitz constant $c > 0$, and hence for every $x \in [-1,1]$, $f(p) - f(x) \leq c|x-p|$.

For the sake of devising a strategy for the player, ignore the function f and treat the outcome of a round as a payoff. Consider two strategies for the player. Strategy *max* attempts to maximize the sum of payoffs over all rounds, and is the strategy from the proof of Theorem 4. Strategy *min* attempts to minimize the sum of payoffs over all rounds, and is the strategy one obtains from the proof of Theorem 4 by negating all payoffs. Let p_t denote the payoff obtained by the player in round t, and let $P_t = \frac{\sum_{i\leq t} p_i}{t}$ (with $P_0 = 0$). Then the *zooming in* strategy is as follows: for every $t \geq 0$, in round $t+1$ use strategy *max* if $P_t \leq p$, and use strategy *min* if $P_t > p$.

Let v be the minimum over all suffixes of the value of f obtained by averaging the outcomes of the expert who maximizes this value over the respective suffix. We now prove that for every $\delta > 0$, there is probability at least $1-\delta$ that the expected value obtained by the zooming in strategy is at least $v - O(c\sqrt{\frac{\log(mT/\delta)}{T}})$.

We first make some preliminary calculations that will later be used in our analysis. For $0 \leq t \leq T-1$, let u_t denote the maximum over all actions of the average outcome on rounds $[t+1,T]$, and let ℓ_t denote the minimum over all actions of the average outcome on these rounds. Suppose that a player uses strategy *max* from round $t+1$ until round T (regardless of whether the *zooming in* strategy actually dictates using *max* on these rounds). Then we denote by U_t the event that in these rounds the sum of outcomes obtained by the *max* strategy is below $u_t(T-t) - O(\sqrt{T\log(mT/\delta)})$. Theorem 4 implies that $Pr[U_t] \leq \frac{\delta}{2T}$. Likewise, denote by L_t the event that in rounds $[t+1,T]$ the sum of outcomes obtained by the *min* strategy is above $l_t(T-t) + O(\sqrt{T\log(mT/\delta)})$. Theorem 4 implies that $Pr[L_t] \leq \frac{\delta}{2T}$. The union bound implies that with probability at least $1-\delta$ there is no value of t for which either U_t or L_t happens.

We now return to the analysis of the *zooming in* strategy. Let $-1 \leq q_1 \leq p \leq q_2 \leq 1$ be such that $[q_1,q_2]$ is the maximal range of values for which $f(x) \geq v$ (by quasiconcavity of f there must be such an interval). Partition values into three ranges: *under* ($< q_1$), *good* ($\in [q_1,q_2]$) and *over* ($> q_2$). Consider what happens when the zooming in strategy is played.

As t increases, the corresponding average value P_t may drift. For some values of t, it changes range (say from *good* to *under*). Let t_0 be the latest round for which the range in which P_{t_0} lies is different from that in which P_{t_0-1} lies. Consider the range in which P_{t_0} lies. If this range is *good* then we are done, because then $f(P_T) \geq v$. Hence we may assume that this range is either *under* or *over*, and without loss of generality let it be *under*. This implies that also P_T is *under*, and it is left to estimate by how much. In round t_0 the player got a payoff p_{t_0} that caused P_{t_0} to drift into under. Hence $-1 \leq p_{t_0} < q_1$. In the suffix starting from round t_0+1, there is an expert whose average outcome lies in $[q_1, q_2]$. As was shown above, even if the value t_0 is chosen in an adversarial manner, there is probability at least $1-\delta$ that the strategy *max* of the player on these rounds gives a sum of outcomes of at least $(T-t_0)q_1 - O(\sqrt{T\log(mT/\delta)})$. In the first t_0-1 rounds the sum of outcomes was at least $(t_0-1)q_1$. Hence the average outcome is at least $q_1 - O(\sqrt{\frac{\log(mT/\delta)}{T}})$ (and at most q_1), and hence the value obtained by the player is at least $v - O(c\sqrt{\frac{\log(mT/\delta)}{T}})$. □

Remark. Our proof of Proposition 1 shows that with probability at least $1-\delta$ the regret is at most $O(c\sqrt{\frac{\log(mT/\delta)}{T}})$. The T factor within the log was a consequence of using Theorem 4 as a blackbox and applying a union bound on $2T$ events. A smaller regret of $O(c\sqrt{\frac{\log(m/\delta)}{T}})$ can be obtained by a more sophisticated argument that we sketch here. A common approach of devising algorithms in the expert setting is via the multiplicative weight update approach. Within that framework, one can consider fractional expert algorithms (rather than randomized ones), and for them the regret is $O(\sqrt{\frac{\log m}{T}})$ (this is a deterministic statement). The dependence on δ only enters once one uses randomized rounding to transform the fractional algorithms into randomized ones. For the randomized rounding process, with probability $1-\delta$ the error introduced in T rounds is at most $\sqrt{1/T\log\delta}$. But in fact, this is be true for every prefix of rounds simultaneously. (Suppose otherwise. Then after the first time an exceptionally high value is reached, there is probability $1/2$ of maintaining this value until the end.) By symmetry, a similar property holds for all suffixes. Hence there is no need of suffering a union bound over $O(T)$ events. (A formal proof based on this remark becomes simpler if one modifies the *zooming in* strategy to switch between *max* and *min* only at points when the fractional expert algorithm would have switched. However, then it does not apply to the bandit setting.)

We now restate and prove Theorem 1.

Theorem 1 *Even for admissible f and when $\mathcal{A}$ are the pure strategies, no strategy asymptotically matches $\min_{B\in\mathcal{B}}[\max_{A\in\mathcal{A}} f_A(B)]$ for $\mathcal{B}$ that contains all single rounds, all prefixes and all suffixes.*

Proof. In our proof there will be four actions, that we call P_1, P_2, S_1 and S_2 (P for *prefix*, S for *suffix*). The outcome vectors will be 4-dimensional. For convenience of the presentation, their ℓ_2 norm will be bounded by 3 rather than by 1. Likewise, the total number of rounds will be denoted by $4T$ rather than T. For a vector $x = (x_1, x_2, x_3, x_4)$, the value function will be $f(x) = \min[x_1, x_2, x_3, x_4, 0]$, which is nonpositive. Observe that f is continuous with Lipschitz constant 1, and quasiconcave, as required from value functions. We first prove the theorem when the excuse blocks are all prefixes and all suffixes, and later extend the proof to capture also all single rounds.

We shall consider four possible input sequences, that we shall denote by $I_{11}, I_{12}, I_{21}, I_{22}$. The first subscript determines which of the prefix actions attains a value of 0 on all prefix blocks, whereas the second subscript determines which of the suffix actions attains a value of 0 on all suffix blocks.

Action P_1 gives an outcome of $(3,0,0,0)$ on each of the first T rounds. Action P_2 gives an outcome of $(0,3,0,0)$ on each of the first T rounds. On the last $3T$ rounds, both P_1 and P_2 give an outcome of $(-1,0,0,0)$ for inputs sequences I_{11} and I_{12}, and an outcome of $(0,-1,0,0)$ for inputs sequences I_{21} and I_{22}.

Action S_1 gives an outcome of $(0,0,-1,0)$ on each of the first $3T$ rounds. Action S_2 gives an outcome of $(0,0,0,-1)$ on each of the first $3T$ rounds. On the last T rounds, both S_1 and S_2 gives an outcome of $(0,0,3,0)$ for inputs sequences I_{11} and I_{21}, and an outcome of $(0,0,0,3)$ for inputs sequences I_{12} and I_{22}.

One can readily verify that for input sequence I_{ij} (with $1 \leq i,j \leq 2$), for every prefix the average outcome of action P_i has value 0, and for every suffix the average outcome of action S_j has value 0.

Consider now a player faced with an input sequence chosen at random from the four possible input sequences I_{ij} (the player does not know which one). In the first T rounds, at least one of the actions P_1 or P_2 is played at most $T/2$ times. Without loss of generality, let P_1 be this action. Then after T rounds the accumulated outcome vector has value at most $3T/2$ on its first coordinate. Then, with probability half, it turns out that the input sequence is one of I_{11} or I_{12}. As a consequence, in the middle $2T$ rounds, the player may choose an action from P_1 or P_2 at most $5T/3$ times, as otherwise the first coordinate of the accumulated output vector becomes $-T/6$, and can never recover from being negative. Hence at least one of the actions S_1 or S_2 is played at least $T/6$ times. Without loss of generality, let S_1 be this action. Then in the input sequence turns out to be I_{12} the third coordinate of the aggregate outcome vector ends up being at most $-T/6$, and the overall value of the outcome vector is at most $-\frac{T}{6}\frac{1}{4T} = -1/24$. As this happens with probability at least $1/4$, the expected value obtained by the player is negative and bounded away from 0, proving the theorem when the excuse blocks are all prefixes and all suffixes.

Extending the proof to allow also for single round excuse blocks is fairly straightforward. One option is to add two auxiliary actions A_1 and A_2, where in each round one of them (chosen at random independently in each round) gives an outcome vector of $(0,0,0,0)$ and the other an outcome vector of $(-2,-2,0,0)$. The player gains nothing by playing these actions because their expected outcome vector is $(-1,-1,0,0)$ which is worse than playing P_1. However now in each round some outcome vector has value 0. Another option is not to add auxiliary actions, but instead, in every round to replace the outcome of one of the $P_1/P_2/S_1/S_2$ actions (chosen at random) by $(0,0,0,0)$. □