Repeated zero-sum games with budget

Troels Bjerre Sørensen
University of Warwick
CV4 7AL, United Kingdom
trold@dcs.warwick.ac.uk

ABSTRACT

When a zero-sum game is played once, a risk-neutral player will want to maximize his expected outcome in that single play. However, if that single play instead only determines how much one player must pay to the other, and the same game must be played again, until either player runs out of money, optimal play may differ. Optimal play may require using different strategies depending on how much money has been won or lost. Computing these strategies is rarely feasible, as the state space is often large. This can be addressed by playing the same strategy in all situations, though this will in general sacrifice optimality. Purely maximizing expectation for each round in this way can be arbitrarily bad. We therefore propose a new solution concept that has guaranteed performance bounds, and we provide an efficient algorithm for computing it. The solution concept is closely related to the Aumann-Serrano index of riskiness, that is used to evaluate different gambles against each other. The primary difference is that instead of being offered fixed gambles, the game is adversarial.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms

Algorithms, Economics, Theory

Keywords

Game playing, Game theory

1. INTRODUCTION

Game theory has often been used to prescribe good behavior in strategic interactions, and to make predictions on how participants will behave in interaction with one another. This is traditionally done by isolating a particular interaction of interest, and then modelling the interaction mathematically. The constructed model is then analyzed separately from the rest of the system, and the resulting analysis is then translated back into the system where the interaction fits in. For this approach to be successful, the right objectives must be derived from the system surrounding the interaction; otherwise the analysis will likely draw incorrect conclusions. In this paper, we examine such a situation, where a fixed finite zero-sum game is played repeatedly under a budget. The overall goal for each player is to win all the money the opponent has, and not run out of money in the process. A simple approach to analyzing this repeated game would be to analyze the zero-sum game as if it was only played once, with the objective of winning the most money in that single play. This, however, can lead to disastrous results, as we shall see in Section 4. In this paper, we show how one can analyze the underlying zero-sum game with respect to a more suitable objective.

As a motivating example, let us look at what happens when we offer a player the chance to double the payoffs of a zero-sum game. Given any zero-sum game G with payoffs in $\{-1, 1\}$ and value $v \neq 0$. Amend G by giving Player 1 the option of doubling the outcome of the game, before the game is played. If Player 1 does so, the outcomes will be in $\{-2, 2\}$ instead, but the rest of the game is unchanged. Any risk neutral Player 1 would clearly double the game, if and only if the value of the game is positive; the doubling does not change optimal strategies, but it doubles the expected value. What is perhaps more surprising is that the situation is reversed if the game has to be repeated until one player is broke; i.e., Player 1 would only double if the value was negative. To see this, first observe that two players playing the undoubled game optimally for some total amount of money, C, is simply a random walk on a line of length C. The walk moves one step to the right with probability $p = v/2 + 1/2$, and one step to the left with probability $(1 - p)$. Starting from point c_1 (being the amount of money Player 1 has), the probability of reaching the right-most point (where Player 1 has won all the money) before the left-most point is exactly

$$Pr[\text{Player 1 wins undoubled}] = \frac{\alpha^{c_1} - 1}{\alpha^C - 1} \qquad (1)$$

where $\alpha = \frac{1-p}{p}$. Playing the doubled game is essentially the same as playing the undoubled game for half as much money. Assume for simplicity that C and c_1 are both even. This means that the probability of Player 1 winning by doubling every game is

$$Pr[\text{Player 1 wins doubled}] = \frac{\alpha^{c_1/2} - 1}{\alpha^{C/2} - 1} \qquad (2)$$

This probability is greater than that (1) if and only if $\alpha > 1$, which happens exactly when $v < 0$. Thus, Player 1 should

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

double the game, if and only if he has *negative* expectation in the individual rounds of the game.

This example serves to show that if we attempt to derive good strategies for the repeated game by trying to maximize the expected outcome of the individual rounds, we will get suboptimal results. In Section 4, we will show that this suboptimality can be arbitrarily large.

1.1 Related research

In a recent paper, Miltersen and Sørensen [13] computed near optimal strategies for a full scale two-player poker tournament. The tournament format fits the description of a game being played repeatedly for a budget, but their game was different for each round; the variant of poker allowed for an all-in, which depends on how much money each player has. They concluded that simply maximizing the amount of chips won in each round was slightly worse than maximizing the probability of winning the tournament. In contrast, the present paper shows that it can be much worse to maximize the expected gain in each round. Furthermore, our results are about general zero-sum games, and not poker specific.

The conceptual ancestor of the contribution of this paper is the Aumann-Serrano index of riskiness [1], which is used to compare different gambles against each other. The main difference is that the Aumann-Serrano index of riskiness is not in an adversarial setting. The Aumann-Serrano index of riskiness of a stochastic variable X is defined as the unique γ such that $E[exp(-X/\gamma)] = 1$. Expressed in these terms, our contribution is to compute the strategy that has the most favorable Aumann-Serrano index of riskiness on the outcome of the game.

1.2 Structure of the paper

The rest of the paper is structured as follows. In Section 2 we introduce the formal model of the games we are discussing in this paper. In Section 3, we review existing theory that provide exact optimal strategies for the games, and discuss why this is not a feasible approach. In Section 4, we show why maximizing expectation in each round can be arbitrarily far from optimal. In Section 5, we describe the main contribution of the paper in the form of a new solution concept and an algorithm for computing it. In Section 6, we derive a bound on the performance of the introduced solution concept. In Section 7, we apply the theory to the game Kuhn poker with a budget. In Section 8, we provide a way to estimate the parameter of the introduced solution concept for games that are too large to repeatedly solve. In Section 9, we discuss two natural extensions of the theory. In Section 10 we compare the introduced solution concept to existing concepts, and discuss future research.

2. MODEL

In this section, we will formalize the model we are using in this paper. First we need some notation and terminology from classic game theory. Details can be found in any introductory textbook on game theory. The underlying game to be played is given as a finite zero-sum game:

DEFINITION 1 (FINITE ZERO-SUM GAME).
A finite zero-sum game is given by $m \times n$ matrix A with integer entries. It is played by Player 1 and Player 2 simultaneously choosing a row i and a column j respectively, after which Player 2 pays A_{ij} to Player 1.

The definition above has a non-standard assumption that the outcome of the finite zero-sum games are integer. This is only to make analysis easier, and everything in this paper can be done with rational valued outcomes as well, as is discussed in section 9.

The players can use *mixed strategies*, that are probability distributions over rows and columns respectively. For zero-sum games, there is a well defined value that each player can guarantee himself, and the associated strategies that provide this guarantee:

DEFINITION 2 (MINIMAX VALUE AND STRATEGIES).
The minimax value of a finite zero-sum game given by the matrix $A \in \mathbb{Z}^{m\times n}$ is:

$$val(A) = \max_{x\in\Delta^m} \min_{y\in\Delta^n} x^\top Ay = \min_{y\in\Delta^n} \max_{x\in\Delta^m} x^\top Ay$$

The minimax strategies for Player 1 are the maximizing x's in the expression above:

$$\operatorname*{argmax}_{x\in\Delta^m} \min_{y\in\Delta^n} x^\top Ay$$

Likewise, the minimax strategies for Player 2 are

$$\operatorname*{argmin}_{y\in\Delta^n} \max_{x\in\Delta^m} x^\top Ay$$

In the setting we are examining in this paper, the game is played repeatedly between two players, each starting with some amount of money, c_1 and c_2. The game progresses over a number of rounds, each of which is a play of a finite zero-sum game. After a round is played, money changes hands according to the strategies (i, j) chosen by the two players, and the game continues with $c_1 = c_1 + A_{ij}$ and $c_2 = c_2 - A_{ij}$, unless either player has run out of money. If that happens the game ends, and the player who is out of money has lost the game, and his opponent has won. Notice that the total amount of money stays constant throughout the repeated game. Denote this constant by $C = c_1 + c_2$. It is of course not possible to win more money from the other player than he has, so the last round might not have full payment off all of A_{ij}. Each player naturally wants to maximize the probability of ending up with all the money, thereby winning the whole game. As in [13], this is not necessarily the same as maximizing the amount of money in each round. In the next section, we will quantify exactly how bad this can be.

Before we can continue, we need to handle certain special cases of games.

DEFINITION 3 (DEGENERATE GAME).
We call a game degenerate, if either player has a strategy that never loses any money against any strategy of the opponent. We also call a game degenerate, if it has equilibria with deterministic outcome 0.

If the first is the case, then one of the players doesn't have any chance of winning the game, if his opponent tries to prevent it. This is not the same as the opponent always being able to win, and as such, the objective of the game is not necessarily clear; it depends on whether infinite play is truly an acceptable outcome. We have chosen to sidestep this complication, as it is caused by a degenerate input game. For the same reason, we will assume that the game does not have equilibria with deterministic outcome 0, as this also opens the possibility of infinite play.

3. EVERETT'S RECURSIVE GAMES

In this section we will describe how to play the repeated game optimally, and discuss why this is often infeasible. If the total amount of money is known beforehand, the game can be modelled and solved as a *recursive game*, introduced by Everett [7]. These recursive games should not be confused with the largely unrelated recursive games by Etessami and Yannakakis [6]. Everett's recursive games is a generalization of *concurrent reachability games* [5] and of *simple stochastic games* [3, 4]. A recursive game consists of a set of *game elements*, each of which are finite zero-sum games, with the added possibility of a outcome being a reference to another game element. If a normal outcome is reached, the game ends with the associated value as the zero-sum outcome. If one of the special outcomes is reached, the play must continue at the referenced game element. This opens up the possibility of the game never ending, which we assign the value 0.

This model fits the repeated game setting in the following way. There will be a game element for every possible division of money between the two players, and each game element will be indexed by how much money Player 1 has. The outcomes of each game element will be references to the neighboring game elements, such that outcome A_{ij} from game element indexed c_1 will be a reference to game element indexed $c_1 + A_{ij}$.

Everett proved that these games can be played ϵ-optimally using *stationary strategies*. A stationary strategy consists of one strategy per game element, and it is played by always using the strategy associated with the current game element. This means that there is nothing to be gained for a player by remembering what game elements have been visited prior to playing a particular game element.

Everett showed that a *critical value* can be assigned to each game element, similar to the minimax value of a finite zero-sum game, such that each player can guarantee an expected outcome arbitrarily close to the assigned value, if the play was started at that game element. The vector of the critical values for all game elements is called the *critical vector*.

If we assign value 0 to game element 0, and value 1 to game element C, the critical value of a game element will be exactly the probability that Player 1 can guarantee himself of winning the repeated game.

In general, there is no easy way to check whether a given vector is an upper bound to the critical vector of a given recursive game. However, Everett gave a property that can be checked in polynomial time that would hold for a subset of the upper bounds and another property that would hold for a subset of the lower bounds. Before we can formally state the property, we need to define the value mapping:

Definition 4 (Value vector and value mapping). *Let G be a recursive game with n game elements. A value vector $\vec{v} \in \mathbb{R}^n$ for G is a vector with one value for each game element of G. The value mapping $\mathbb{M} : \mathbb{R}^n \to \mathbb{R}^n$ of G, mapping value vectors to value vectors, is the minimax evaluation of each game element, where the non-terminal outcomes have been replaced with the values given by the input vector.*

The properties rely on the following relations among value vectors:

$$\vec{u} \succeq \vec{v} \Leftrightarrow \left\{ \begin{array}{ll} \vec{u}_i > \vec{v}_i & \text{if } \vec{v}_i > 0 \\ \vec{u}_i \geq \vec{v}_i & \text{if } \vec{v}_i \leq 0 \end{array} \right\} \forall i$$

$$\vec{u} \preceq \vec{v} \Leftrightarrow \left\{ \begin{array}{ll} \vec{u}_i < \vec{v}_i & \text{if } \vec{v}_i < 0 \\ \vec{u}_i \leq \vec{v}_i & \text{if } \vec{v}_i \geq 0 \end{array} \right\} \forall i$$

Everett proved the following Theorem:

Theorem 5 (Everett, 1957). *If $\mathbb{M}(\vec{v}) \succeq \vec{v}$, then v is a lower bound on the critical vector. Furthermore, the stationary strategy for Player 1 obtained by finding the optimal strategy in each game element, with arcs to other game elements replaced by the corresponding values in $\vec{v}$, has guaranteed expected payoff* at least *$\vec{v}_g$ for play starting in g. If $\mathbb{M}(\vec{v}) \preceq \vec{v}$, then $\vec{v}$ is an upper bound on the critical vector.*

In short, if the value mapping increases the value of each entry of a value vector, then the new values is a lower bound on the critical vector. It is this property we will use later in the paper to give performance guarantees on the introduced solution concept.

For general recursive games, the only known algorithm for computing the exact critical vector is that of Hansen et.al. [9]. This algorithm runs in time doubly exponential in the size of the game, and outputs the values as algebraic numbers in isolating interval representation. However, we can approximate the critical vector efficiently using value iteration. This approach does not work for general recursive games, as shown by Everett, but as discussed below, it works for our special case.

In this context, value iteration means repeatedly applying the value mapping to a value vector, until it converges. An easy way to detect convergence is to run two value iterations, one starting from a trivial upper bound and the other starting from a trivial lower bound. In our case, as the values are probabilities, a vector of 0s would serve as a lower bound, while a vector of 1s would work as an upper bound. Notice that $\mathbb{M}(\vec{v})$ is monotone in $\vec{v}$, so if $\vec{v}$ is an upper (resp. lower) bound to the critical vector, then so is $\mathbb{M}(\vec{v})$. Thus, if the two vectors are close after a number of iterations, then we have a good approximation to the critical vector. Now we only need to show that the two vectors will in fact get close. Notice that the value iteration on the lower bound after T steps corresponds exactly to the time limited game, where Player 2 is declared the winner if the game doesn't end naturally before T steps. Similarly with Player 1 for the upper bound. Since the game is non-degenerate, both Players have positive probability of winning money from any given game element. Thus, the probability of the game not having ended after T steps goes to 0 as T goes to infinity. For a more thorough analysis of this type of algorithms, see [8].

While this approach gives a provably optimal strategy for the repeated game, it might not be a feasible approach for several reasons. First of all, the explicit modelling requires one game element for every non-degenerate division of money between the two players, which is $C - 1$ game elements. This number might be prohibitively large from a computational point of view, as more game elements requires more work. It also has the problem that each player must remember one strategy per game element, as the strategies in general will be different. Finally, it might be that a player does not know how much money his opponent has, in which case the explicit modelling given above is not possible.

DEFINITION 6 (OBLIVIOUS STRATEGY).
A positional strategy is oblivious *if it associates the same strategy to all game elements.*

Using an oblivious strategies, the player needs only remember a single strategy, but it will in general not be optimal in the recursive game. To the best of our knowledge, there is no algorithm for computing the best oblivious strategy for this setting.

4. FAILURE OF MINIMAX

In this section, we will prove why simply maximizing immediate outcome of each round can be arbitrarily far from optimal. We will do this by explicit construction of a finite zero-sum game with a unique minimax strategy that wins with probability zero, where the optimal strategy would win with probability arbitrarily close to 1. Let $n \geq 4$ be an even number. Now construct $A \in \mathbb{Z}^{2n \times n}$ in the following way:

$$A_{ij} = \begin{cases} -1 & \text{if } i = j \\ 0 & \text{if } i \neq j \wedge i \leq n \\ 1 & \text{if } i > n \wedge [(i+j) \bmod n \leq n/2 - 1] \\ -1 & \text{otherwise} \end{cases}$$

Both players have unique minimax strategies; Player 2 uniformly mixes over all n columns, while Player 1 uniformly mixes over the n first rows. The one-round game thus has value $-1/n$. However, the row player would never win the repeated game using this strategy, as the strategy never wins any money. If the row player instead uniformly mixed over the n last rows, he would win a coin with probability $1/2 - 1/n$, and lose a coin with probability $1/2 + 1/n$. Player 2's minimax strategies are still a best response, even in the repeated setting. The expected gain in each round is lowered to $-2/n$, but the probability of winning something is now just below $1/2$ instead of 0. If the repeated game is started from $(c_1 = k - 1, c_2 = 1)$, the probability of Player 1 winning with the second strategy will be almost $1 - 1/k$, while the first strategy will win with probability 0. Pick n and k large enough, and the minimax strategies turn an almost sure win into a certain loss.

5. RISKINESS OF A GAME

In this section we will describe the main contribution of the paper. In short, we show how to find the right exponential utility function for a given game, such that minimax strategies with respect to that utility function will have good guaranteed bounds on the performance in the repeated game. Utility functions are functions from outcomes to real values, describing a player's satisfaction with a particular outcome. They are commonly used to explain how both the seller and the buyer of insurance policies can be satisfied by the transaction, even though the underlying transaction is zero-sum. A risk-averse (concave utility function) insuree is happy to pay an insurance premium that is more than the expected loss, in exchange for less variance of the outcome. Similarly, a risk-seeking gambler (convex utility function) will buy lottery tickets, even though the expected outcome is lower than the price of the ticket. For our purpose, the players want to maximize the probability of winning the tournament, which in general is not linear in the money won in each round. We therefore want to find the right function to serve as a proxy for the probability of winning the tournament. Let us first define the exponentiation of a game:

DEFINITION 7 (EXPONENTIATION OF A GAME).
Given a zero-sum game $A \in \mathbb{Z}^{m \times n}$ and a positive constant α, define A^α to be

$$A^\alpha_{ij} = \begin{cases} \frac{\alpha^{A_{ij}} - 1}{\ln \alpha} & \text{, if } \alpha \neq 1 \\ A_{ij} & \text{, if } \alpha = 1 \end{cases}$$

Notice that the entries of A^α are continuous in α . If $\alpha = 1$, then $A^\alpha = A$, corresponding to the players being completely risk neutral. If $\alpha > 1$, the utility function is convex for Player 1 and concave for Player 2, making Player 1 risk seeking, while Player 2 will be risk averse. The situation is the opposite for $\alpha < 1$. We are now looking for a suitable α for the game at hand.

PROPOSITION 8.
Given a non-degenerate zero-sum game $A \in \mathbb{Z}^{m \times n}$, there exists an α^* such that $val(A^{\alpha^*}) = 0$.

PROOF SKETCH. Since all entries of A^α are continuous functions of α, we know that $val(A^\alpha)$ is also continuous in α. Since A is non-degenerate, Player 1 has a strategy that guarantees at least some fixed strictly positive probability of a positive outcome. As all positive entries of A^α approach ∞ as $\alpha \to \infty$, and all negative entries approach 0, we have the $val(A^{\alpha_{hi}}) > 0$ for some sufficiently large α_{hi}. Likewise, Player 2 has a strategy that guarantees at least some fixed strictly positive probability of a negative outcome. As all positive entries of A^α approach 0 as $\alpha \to 0$, and all negative entries approach $-\infty$, we have the $val(A^{\alpha_{lo}}) < 0$ for some sufficiently small α_{lo}. Combined with continuity of $val(A^\alpha)$ in α, the intermediate value theorem gives us that there exists some α^* such that $val(A^{\alpha^*}) = 0$. □

PROPOSITION 9.
Given a non-degenerate zero-sum game $A \in \mathbb{Z}^{m \times n}$, there is only one α^* such that $val(A^{\alpha^*}) = 0$.

PROOF SKETCH. We need to prove that $val(A^{\alpha^*})$ is strictly monotone in α, from which the proposition follows. Given $\alpha_1 < \alpha_2$, we must prove that $val(A^{\alpha_1}) < val(A^{\alpha_2})$. Notice first that the payoff of all strategy combinations strictly increases in α, unless they result in a deterministic outcome of 0. Let x be the strategy for Player 1 that guarantees $val(A^{\alpha_1})$ in A^{α_1}. Unless Player 2 has a response that guarantees a deterministic outcome of 0, the value of all responses of Player 2 will be strictly higher in A^{α_2} than in A^{α_1}, and Player 1 can thus use x to get a higher value in A^{α_2} than in A^{α_1}, from which it follows that $val(A^{\alpha_1}) < val(A^{\alpha_2})$. If $val(A^{\alpha_1}) < 0$, Player 2 has no desire to use such a 0-strategy, but the payoff of all other strategies against x in A^{α_2} is higher than in A^{α_1}. If $val(A^{\alpha_1}) > 0$, Player 2 does not have a 0-strategy, and therefore $val(A^{\alpha_1}) < val(A^\alpha_2)$. The only case left is if $val(A^{\alpha_1}) = 0$. As the base game A does not have equilibria with deterministic outcome 0, the outcome of A^{α_1} cannot be deterministic outcome 0, therefore $val(A^{\alpha_2}) > val(A^{\alpha_1})$. □

The assumption that A does not have equilibria with deterministic outcome 0 is crucial for the uniqueness of α^*.

Without this assumption, $val(A^\alpha)$ is only weakly monotone in α. To properly handle such degenerate games with multiple such α^*, we need the following slightly more general definition of the suitable value of α^*.

DEFINITION 10 (RISKINESS OF A GAME).
Given a zero-sum game $A \in \mathbb{Z}^{m\times n}$, the riskiness for Player 1 in A is the largest α^* such that $val(A^{\alpha^*}) = 0$.

If the game is non-degenerate, the is only one A^* satisfying the condition. If that is the case, we will simply call it the riskiness of the game.

DEFINITION 11 (RISK-AWARE STRATEGIES).
Given a non-degenerate zero-sum game $A \in \mathbb{Z}^{m\times n}$, the risk-aware strategies of A are the minimax strategies of A^{α^*}, where α^* is the riskiness for Player 1 in A.

Even with just weak monotonicity, we can compute the riskiness for Player 1 using the algorithm given in Algorithm 1.

Algorithm 1 Computes risk-aware strategies

```
α_hi ← 1
α_lo ← 1
while val(A^{α_lo}) > 0 do
  α_lo ← α_lo/2
end while
while val(A^{α_hi}) ≤ 0 do
  α_hi ← α_hi * 2
end while
while α_hi − α_lo > ε do
  α ← (α_lo + α_hi)/2
  if val(A^α) > 0 then
    α_hi ← α
  else
    α_lo ← α
  end if
end while
return minimax strategies of A^{α_hi}
```

6. PERFORMANCE GUARANTEE

In this section, we will show what performance guarantees we get in the repeated game, if both players have a finite budget. This is done by expressing value vectors that satisfy Everett's conditions, proving that they are true bounds on the critical vector, and that the computed strategies have the promised performance.

THEOREM 12. *Given a game $A \in [-min; max]^{M\times N}$ with riskiness α, using the risk-aware strategy will guarantee Player 1 a winning probability of:*

$$\frac{\alpha^{c_1} - 1}{\alpha^{c_1+c_2+max-1} - 1}$$

when the game is started from money division (c_1, c_2).

PROOF. We need to prove that the values vector described above satisfies Everett's lower-bound condition. For all *internal* game elements, with indices in $[min; C - max]$, we can use the following argument. Given a fixed game element indexed c_1, the values assigned to the game elements around it are:

$$\cdots \frac{\alpha^{i-1} - 1}{\alpha^{C+max-1} - 1}, \quad \frac{\alpha^{i} - 1}{\alpha^{C+max-1} - 1}, \quad \frac{\alpha^{i+1} - 1}{\alpha^{C+max-1} - 1}, \quad \cdots$$

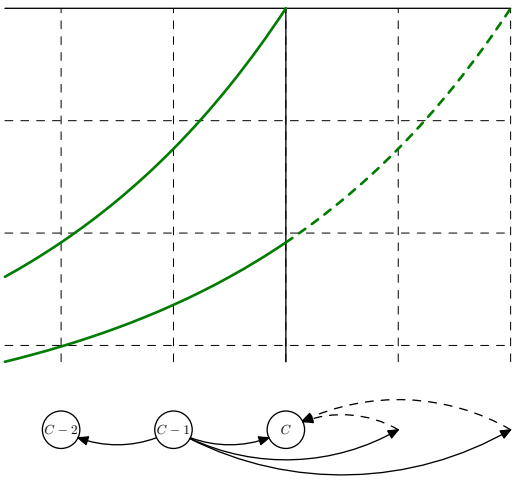

Figure 1: Shifting the value vector to satisfy Everett's condition for high indexed game element.

Optimal strategies are not changed by positive affine transformations of the utility function. Notice that if we multiply the neighborhood values by the positive constant

$$\frac{\alpha^{C+max-1} - 1}{\alpha^i \ln \alpha}$$

and add the constant

$$\frac{\alpha^{-i} - 1}{\ln \alpha}$$

the neighborhood becomes

$$\cdots \frac{\alpha^{-1} - 1}{\ln \alpha}, \quad \frac{\alpha^{0} - 1}{\ln \alpha}, \quad \frac{\alpha^{1} - 1}{\ln \alpha}, \quad \cdots$$

which is exactly the exponentiated game with parameter α. This exponentiated game has positive value for all $\alpha > \alpha^*$, implying that the value mapping on the vector given in Theorem 12 increases the value of game elements with indices in $[min; C - max]$.

The outlying game elements, where either player might not have enough money to pay A_{ij}, for some (i, j), we need to argue differently. For game elements with indices in $[1; min - 1]$, some of the low values in the neighborhood is rounded up to 0, compared to what the exponentiated game looks like. But since we are increasing the value of some game elements in the neighborhood, Everett's condition will still hold, as the value mapping is monotone. The situation is different for the indices $[C - max + 1; C - 1]$, as they have neighborhoods extending beyond game element C. If we were to fit the exponentiated utility function in with $val_C = 1$, we would have to round the value down of the higher payoffs, thereby possibly violating Everett's condition. We therefore have to shift the value vector such that the non-existing game elements indexed $C + max - 1$ gets value 1, as shown in figure 1. Fitting the exponential function to the points $(0, 0)$ and $(C + max - 1, 1)$, we get exactly the expression in Theorem 12. By definition of the riskiness of Player 1, we now have that Everett's first condition is satisfied for all $\alpha > \alpha^*$ for all game elements, and we therefore have the limit as a lower bound. □

We can also use the same expression to give an upper bound on the performance, not just of these strategies, but of any strategy; even non-oblivious.

THEOREM 13. *Given a non-degenerate game with matrix $A \in [-min; max]^{m \times n}$ with riskiness α, no strategy for Player 1 can guarantee more than*

$$\frac{\alpha^{c_1+min-1} - 1}{\alpha^{C+min-1} - 1}$$

in the repeated game starting from division (c_1, c_2).

PROOF OF THEOREM 13. To prove this, we need a simple observation about the game from the opponents point of view, namely that the riskiness is inverted for the other player, i.e.,

$$risk(A) = risk(-A^\top)^{-1}$$

Combining this with Theorem 12 we get a strategy for Player 2 that wins with probability

$$\frac{\alpha^{-c_2} - 1}{\alpha^{-C-min+1} - 1}$$

when the game is started from division (c_1, c_2). Since at most one player can win, Player 1 cannot guarantee a higher probability of winning than 1 minus the guarantee Player 2 has.

$$\begin{aligned} Pr[\text{Player 1 win}] &\leq 1 - \frac{\alpha^{-c_2} - 1}{\alpha^{-C-min+1} - 1} \\ &= \frac{\alpha^{-C-min+1} - \alpha^{-c_2}}{\alpha^{-C-min+1} - 1} \\ &= \frac{\alpha^{c_1+min-1} - 1}{\alpha^{C+min-1} - 1} \end{aligned}$$

□

We can use the upper bound to give a bound on how far from optimal the risk-aware strategies are. Assume for simplicity that $min = max$, i.e., that that most negative outcome is minus the most positive outcome of the game. The difference between the upper and lower bounds is greatest at game element C-1 when $\alpha > 1$, and at game element 1 when $\alpha < 1$. Assume wlog the latter is the case:

$$\text{Gap} = \frac{\alpha^{1+min-1} - 1}{\alpha^{C+min-1} - 1} - \frac{\alpha^1 - 1}{\alpha^{C+max-1} - 1} = \frac{\alpha^{min} - \alpha}{\alpha^{C+min-1} - 1}$$

Notice that the gap is 0 when $min = max = 1$. That is, the risk-aware strategies are optimal for games with outcomes in $\{-1, 0, 1\}$. The gap in the general case approached $\alpha - \alpha^{min}$ for $C \to \infty$ for $\alpha < 0$.

7. EXPERIMENTS

As an example of how well the risk-aware strategies perform, let us look at the game of Kuhn poker [12]. The game is a heavily simplified version of poker, played between two players. The rules are as follows. A deck of three cards (King, Queen and Jack) is shuffled, and the two players receive one card each. Both players put one coin in the pot as an ante. The players now use normal poker betting protocol to decide whether to bet an additional coin, i.e., Player 1 can check or bet. If he bet, Player 2 can either call or fold. If Player 1 checked, Player 2 can either check or bet. In case Player 2 bets, Player 1 has to decide whether to call or fold. If a player folds, he forfeits the hand and loses the ante he paid in the beginning of the hand. If neither player folds, the hidden cards are revealed, and the higher card wins the ante and the bets (if any).

The game has a unique equilibrium:

- Player 1 bets with King, checks with Queen, and bets (bluffs) with probability 1/3 with Jack. If Player 2 bets, Player 1 will call with probability 2/3 with Queen, and always folds with Jack.
- If Player 1 checks, Player 2 uses same strategy as Player 1 does for the first move. If Player 1 bets, Player 2 calls with King, calls with probability 1/3 with Queen, and always folds with Jack.

The game has value $-1/18$, i.e., Player 1 is expected to lose a little every round. Using the algorithm outlined earlier in the paper, we compute the riskiness of the game to be $\alpha \approx 1.062$. Solving the game exponentiated with this α, we find that the optimal strategy has changed in the following way:

- Player 1 increases the probability of bluffing with Jack to 37.6%, but slightly lowers the probability of calling with Queen to 64.6%.
- Player 2 lowers the probability of bluffing with Jack to 29.5%, but increases the probability of calling with Queens to 37.3%.

We can evaluate the two pairs of strategies against an optimal counter strategy in the following way. If we fix the strategy of one player to the strategy we want to evaluate, the other player is left with a one-player game, which we can easily solve as a Markov Decision Process; simply do value iteration with only one player. The values of these counter strategies are given in Figures 2, 3, and 4.

For larger values of C, the minimax performance gap grows to around 20%, while the risk-aware performance gap falls to less than 3%. This corresponds well with the theoretical bound on the performance gap of 5.5%, calculated using the difference between the upper and lower bound in the previous section.

8. ESTIMATING RISKINESS

Some games are so large that even solving them once requires months of computation [10] and it is therefore practically impossible to solve it repeatedly in order to do binary search and compute the riskiness of the game. It would therefore be useful to estimate the riskiness of the game beforehand, and then only compute minimax strategies of the exponentiated game once with the estimated riskiness. This can be done by observing that two fixed strategies played obliviously against each other results in a random walk on line of possible divisions of money between the players, just as in the introductory example. This process can be approximated by a Wiener process with drift, if we know just a little about the typical play in the game. The following theorem can be found in any textbook on stochastic processes:

THEOREM 14. *A Wiener process with parameter σ^2 and drift μ on a line of length C, starting at point i has probability of reaching the right endpoint before the left equal to*

$$\frac{\alpha^i - 1}{\alpha^C - 1}$$

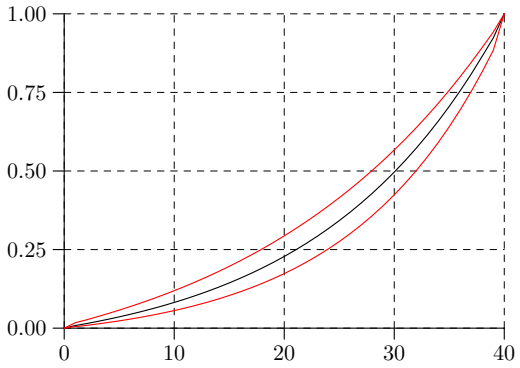

Figure 2: Upper line is the value of Player 1's best response to Player 2's minimax, lower line is value of Player 2's best response Player 1's minimax, middle line is the critical vector

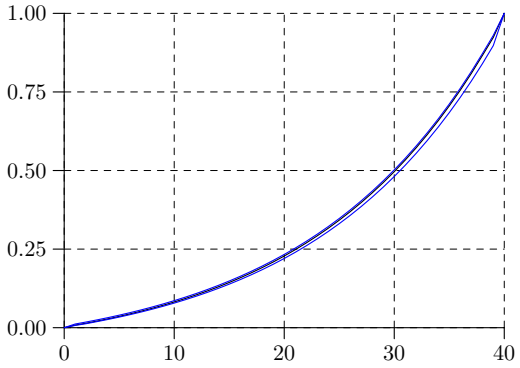

Figure 3: Upper line is the value of Player 1's best response to Player 2's risk-aware strategy, lower line is value of Player 2's best response Player 1's risk-aware strategy, middle line is critical the vector

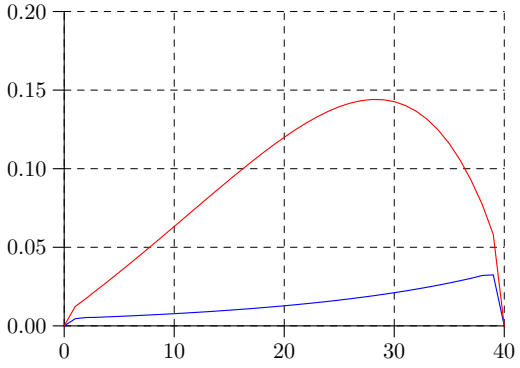

Figure 4: Upper line is the difference between the best response values against minimax. Lower line is the difference between the best response values against the risk-aware strategies.

where $\alpha = exp(-2\frac{\mu}{\sigma})$

Thus, if we know the typical outcome distribution of the game, we can estimate the riskiness as $\alpha = exp(-2\frac{\mu}{\sigma})$.

In 2007, the computer poker research group at the University of Alberta organized the First Man-Machine Poker Competition [10, p.79], where two professional poker players played against four different poker playing programs, collectively called Polaris. Out of the four sessions, the humans won two sessions, drew one, and lost one. The only program they lost to was named "Mr. Orange", and it was constructed by solving the game with respect to a modified utility function. The utility function was as follows:

$$u(v) = \begin{cases} v & \text{if } v \leq 0 \\ 1.07 \cdot v & \text{if } v > 0 \end{cases}$$

In other words, the program saw the game as if any winnings where 7% higher, while losses where left unmodified. A side effect of this choice was that the game was no longer a zero-sum game, since the 7% was not paid by the loser. The resulting program was very aggressive; according to Laak [10] (one of the human players), Mr. Orange "...was like a crazed, cocaine-driven maniac with an ax".

If we were to use the results of this paper to suggest an alternative utility function, we could use empirical observations about the game to estimate the riskiness of the game. Heads Up Limit Texas Hold'em has been observed [2] to have a standard deviation of 6.856 sb/hand. Currently, the best minimax algorithms produce poker playing programs that lose around 0.1 sb/hand against an optimal opponent [11]. We can then use the discussion in the previous section to figure out what riskiness balances out the 0.1 sb/hand suboptimality. Using the formula, we get the $\alpha = exp(-2 \cdot (-0.1)/6.856) \approx 1.03$, and the resulting modified utility function becomes

$$u'(v) = \frac{1.03^v - 1}{\ln 1.03}$$

Notice that $u(v)$ and $u'(v)$ are very close for typical values $v \in [-\sigma; \sigma]$. While the setting for the Man-Machine match was not exactly the same as our model, the similarity does provide some hope to the applicability of the approach.

9. EXTENSIONS

The most natural extension is to remove the requirement that outcomes must be integer. If we allow them to rational numbers instead, we can still do the same analysis. The explicit modelling of the game would require C/gcd game elements, where gcd is the greatest common divisor of all the outcomes of the game. This could be a very large number of game elements, but as our approach uses oblivious strategies, we are not hindered by this. The easiest way to prove performance guarantees for rational valued games is by scaling all number of the game (outcomes and amount of money) up with a constant, so that everything becomes integers. To do this, we need to know how scaling affects riskiness of a game.

PROPOSITION 15. *Multiplying all outcomes of a game A by constant k results in the riskiness becoming the k'th root of the previous riskiness:*

$$risk(k \cdot A) = risk(A)^{1/k}$$

PROOF. The property follows directly by the fact that exponentiating with the k'th root cancels out the scaling, except for a constant scaling of the whole utility function, and therefore $val((k \cdot A)^{\alpha^{1/k}} = k \cdot val(A^{\alpha}) = 0$. □

Using this property, we can scale to integers, get the performance guarantee, and scale back to the original game. Doing this, we get the lower bound on winning probability for Player 1 to be

$$\frac{\alpha^{c_1} - 1}{\alpha^{C+max-gcd} - 1}$$

when the game starts from money division (c_1, c_2). In other words, the only change is that the -1 we got from the overlap for high index game elements is replaced with an expression that only depends on the input game. Notice that Algorithm 1 does not rely on the outcomes being integer, so it can be used directly on games with fractional outcomes; the scaling is only for the analysis.

Another interesting extension of the result is to the case where only one of the players has a restricted budget, while his opponent has infinite resources. The goal for the budget constrained player is to build his fortune, while avoiding going bankrupt in the process. Of course, for this to be interesting, the budget constrained player must have positive expectation; otherwise he will lose with probability 1. Assume wlog that it is Player 1 that is budget constrained, and that the value of the underlying game is positive. This implies that the game has low riskiness. We can now observe that in this case the performance guarantee given by Theorem 12 for a fixed c_1 converges to $1 - \alpha^{c_1}$ for $c_2 \to \infty$. Thus, the risk-aware strategies apply to one-sided budgets as well.

10. DISCUSSION AND FUTURE RESEARCH

An interesting observation on the riskiness estimate in section 8 is that the riskiness is closely tied to the ratio of mean over standard deviation. In portfolio management, this ratio is commonly called the Sharpe ratio [14] (with risk free rate 0). In many idealized settings, it is exactly the Sharpe ratio one wants to maximize. It does, however, have important shortcomings. Primarily, it relies on the outcomes being normally distributed, which is not in general the case for the scenarios we have examined in this paper. To the best of our knowledge, there is no known algorithm for maximizing the Sharpe ratio of a zero-sum game. However, we can observe that our proposed solution concept outperforms Sharpe ratio maximization on simple examples. To see why this is the case, examine the doubling game from the introduction. Both the doubled and the undoubled games have the same Sharpe ratio, so a slight perturbation would make a Sharpe ratio maximizer choose the wrong action. One can easily construct games where the scaling is with a larger constant to exacerbate the problem.

In this paper, we have only examined a single setup, where two players where playing under a budget. The general idea of deriving a utility function to use for solving a sub-problem leads to many open problems. For instance, in the Man-Machine poker tournament discussed in section 8, the true objective was not to bankrupt the opponent (you cannot; he has unlimited bankroll), but rather to have the most money after a fixed number of rounds has been played. This again leads to a non-linear utility function, but the exact function to be used in unclear.

11. REFERENCES

[1] R. J. Aumann and R. Serrano. An economic index of riskiness. *Journal of Political Economy*, 116:810–836, 2008.

[2] D. Billings. *Algorithms and Assessment in Computer Poker*. PhD thesis, University of Alberta, 2006.

[3] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[4] A. Condon. On algorithms for simple stochastic games. *Advances in Computational Complexity Theory, DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 13:51–73, 1993.

[5] L. de Alfaro, T. A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188 – 217, 2007. Expressiveness in Concurrency.

[6] K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. In *Proceedings of International Colloqium on Automata, Languages and Programming (ICALP)*, pages 324–335, 2006.

[7] H. Everett. Recursive games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games Vol. III*, volume 39 of *Annals of Mathematical Studies*. Princeton University Press, 1957.

[8] K. A. Hansen, R. Ibsen-Jensen, and P. B. Miltersen. The complexity of solving reachability games using value and strategy iteration. In *Proceedings of the 6th International Computer Science Symposium in Russia, CSR*, 2011.

[9] K. A. Hansen, M. Kouchý, N. Lauritzen, P. B. Miltersen, and E. P. Tsigaridas. Exact algorithms for solving stochastic games. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, 2011.

[10] M. Johanson. Robust strategies and counter-strategies: Building a champion level computer poker player. Master's thesis, University of Alberta, 2007.

[11] M. Johanson, M. Bowling, K. Waugh, and M. Zinkevich. Accelerating best response calculation in large extensive games. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 258–265, 2011.

[12] H. Kuhn. A simplified two-person poker. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the theory of games I*, volume 24 of *Annals of Mathematical Studies*. Princeton University Press, 1950.

[13] P. B. Miltersen and T. B. Sørensen. A near-optimal strategy for a heads-up no-limit texas hold'em poker tournament. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.

[14] W. F. Sharpe. Mutual fund performance. *Journal of Business*, 1:119–138, 1966.