# Beat the Cheater: Computing Game-Theoretic Strategies for When to Kick a Gambler out of a Casino

**Troels Bjerre Sørensen, Melissa Dalis, Joshua Letchford, Dmytro Korzhyk, Vincent Conitzer**
Dept. of Computer Science
Duke University
Durham, NC 27708
{trold,mjd34,josh,dima,conitzer}@cs.duke.edu

### Abstract

Gambles in casinos are usually set up so that the casino makes a profit in expectation—as long as gamblers play honestly. However, some gamblers are able to cheat, reducing the casino's profit. How should the casino address this? A common strategy is to selectively kick gamblers out, possibly even without being sure that they were cheating. In this paper, we address the following question: Based solely on a gambler's track record, when is it optimal for the casino to kick the gambler out? Because cheaters will adapt to the casino's policy, this is a game-theoretic question. Specifically, we model the problem as a Bayesian game in which the casino is a Stackelberg leader that can commit to a (possibly randomized) policy for when to kick gamblers out, and we provide efficient algorithms for computing the optimal policy. Besides being potentially useful to casinos, we imagine that similar techniques could be useful for addressing related problems—for example, illegal trades in financial markets.

## Introduction

Consider a game between a casino and a population of gamblers that takes place over the course of a night. We focus on a single game in the casino, so that in each round $r \in \{1, \dots, R\}$, we have a set of possible outcomes with values $V = \{v_1, \dots, v_n\}$ to the gambler (where $v_1 < \cdots < v_n$, with $v_1 < 0 < v_n$), and, for each $v_i \in V$, an associated probability $P(v_i) > 0$ that the outcome occurs. We will focus on the case where $EV = \sum_i P(v_i)v_i < 0$, which corresponds to the honest gamblers losing money in expectation. Otherwise, the optimal policy is to never let anyone play.

For a given gambler, a sequence $s$ is a list of outcomes, one for each round; this sequence has a value $v_s = \sum_i s_i$, where $s_i$ is the $i$th outcome of $s$ Sequences can have length anywhere from 1 to $R$. Let $P(s) = \prod_i P(s_i)$ denote the probability that $s$ occurs (for an honest gambler).

We assume that there is a commonly known fraction $c$ of cheaters; the remaining gamblers are honest. We assume cheaters are all-powerful, i.e., a cheater is able to choose any outcome in $V$ in each round. We assume that the casino can never detect cheating directly; all it has to go on is the gambler's history of winnings so far, and every history can happen with positive probability for an honest gambler. Thus, the casino can never be completely sure that it is not kicking out an honest gambler —and kicking out an honest gambler is costly, because in expectation they would lose money to the casino for the rest of the night.

Specifically, the casino's policy is defined by a "kick probability" $p_s$ of evicting a gambler, for each possible sequence $s$. This is the probability, conditional on the gambler's track record so far being $s$ and (hence) the gambler not having been kicked out at an earlier point, that the gambler is kicked out immediately at the end of $s$. If $p_s \in \{0, 1\}$ for all $s$, then $p_s$ is a deterministic policy — but we will show randomized policies generally perform better. For notational convenience, we require $p_s = 1$ when $|s| = R$ (at the end of the night everyone must leave). Our goal is to optimize these kick probabilities, keeping in mind that cheaters will become aware of the policy and choose a best response. For example, a policy that kicks gamblers out after ten successive wins would be quite useless, because a cheater would simply choose to lose every tenth round.

While this casino problem may constitute the cleanest example to illustrate our framework, there are other possible applications. For example, in financial markets where there is a concern about hard-to-detect illegal trading behavior (such as insider trading), a clearinghouse might use our approach to decide whether or not to block a trader from continuing to trade. Traders engaging in illegal behavior are likely to learn the clearinghouse's policy beforehand, and act accordingly, potentially making smaller, less frequent, or slightly worse trades in order to avoid the block.

A natural idea is to simply do a statistical significance test: for example, kick a gambler out if his total winnings at any point are more than two standard deviations above the mean (where the mean and standard deviation are computed for an honest gambler gambling the number of rounds so far, the "null hypothesis"). Our analysis reveals that this is a suboptimal policy. The reason for this is that some honest gamblers will, early on, happen to go two deviations above the mean. Even though their winnings are small at this point, the policy would kick them out. This results in a significant expected loss for the casino, and it is pointless because cheaters will never let themselves be kicked out early in response to this policy, since later on they can accumulate much larger winnings. On the other hand, a modified policy that only checks later on whether someone is above two standard deviations is vulnerable to a cheater accumulating large winnings early on.

Expert [Littlestone and Warmuth, 1994; Cesa-Bianchi et al., 1997] and bandit [Auer et al., 2002; Gittens, 1979] algorithms may also appear to be relevant; however, using them across rounds for a single gambler does not work, because once we kick a gambler out, we cannot let the gambler gamble again in a later round. We could use them across gamblers, letting each expert or bandit arm correspond to a policy and learning the optimal one by experimenting with many gamblers in sequence. Downsides of this include that it does not incorporate prior information, it does not give insight into the structure of optimal policies, and, perhaps most importantly, without further analysis the policy space appears too complex for this to work well.

Optimal stopping problems, such as the well-known secretary problem [Freeman, 1983], are another class of apparently related problems that would at least capture the fact that kicking a gambler out is an irreversible decision. However, we are not aware of any optimal stopping problems that are game-theoretic in nature (other than to the extent that they optimize for the worst case and in that sense are adversarial in nature). More generally, what appears to set our problem apart is that it is fundamentally a problem in game theory. We are not concerned with judging whether a gambler is a cheater *per se*, which might be the natural goal from a statistics viewpoint, but rather with maximizing the casino's profit in the face of some cheaters. Moreover, the problem is fundamentally not zero-sum: for example, throwing everyone out in the first round leaves both the casino and the cheaters worse off than the optimal solution. (The honest gamblers are not strategic players.)

## Related work

This paper fits naturally into a recent line of work on applying algorithms for computing game-theoretic solutions to security scenarios. These applications include allocating security resources across terminals in the Los Angeles International Airport, assigning Federal Air Marshals to commercial flights, choosing patrol routes for the U.S. Coast Guard to protect ports, detecting intrusion in access control systems, and so on [Alpcan and Basar, 2003; Kiekintveld et al., 2009; Pita et al., 2008; Tsai et al., 2009]. In these applications, as in ours, the player in the defensive role is modeled as a Stackelberg leader who commits to a randomized strategy, to which the player in the attacking role then responds.

In these applications, generally an argument can be made that the attacker would have an opportunity to learn the defender's strategy before attacking, which justifies the Stackelberg model; additionally, the Stackelberg model has technical advantages. It avoids the equilibrium selection issues of concepts like Nash equilibrium, and we can solve for an optimal strategy for the leader in time that is polynomial in the size of the normal form of the game [Conitzer and Sandholm, 2006; von Stengel and Zamir, 2010]. However, it is not uncommon that linear programs are exponential for security games, e.g., [Jain, Conitzer, and Tambe, 2013]. Even when the normal form is exponential in size, in some cases, an alternate formulation can be used such as the extensive form. This doesn't help in our case, as even the extensive-form formulation of our problem has an exponential number of nodes (in the number of rounds). Additionally, our formulation has both random moves and imperfect information. It has been shown that either of these individually would be sufficient to make the problem NP-hard [Letchford and Conitzer, 2010].

## Computing the optimal policy

In this section, we show how to compute the optimal policy for the casino. It is possible to solve small instances of the problem using linear programming. While this provides a direct solution to the problem, it is practically infeasible for solving anything but small toy examples. This is due to the fact that both the number of variables and the number of constraints are exponential in $R$, with a variable per probability in the policy and a constraint per sequence. We therefore now proceed to define the optimal policies in a more concise way, and prove that they are indeed optimal.

As we shall see shortly when we formalize everything, the optimal policy is completely characterized by how much a cheater is allowed to win in expectation. This amount we will call the casino's initial *goodwill* towards a gambler, which we in general denote by $g$. By updating this value according to wins and losses as the night progresses, the casino keeps track of how much more the gambler can be allowed to win in expectation. If the gambler wins, his goodwill goes down by the amount won, and if he loses, it goes up by the amount lost. If a gambler runs out of goodwill, he will be kicked out of the casino for sure. If a gambler's goodwill gets dangerously low, the casino can raise his goodwill by kicking him out with some probability. This may sound a little strange, but remember that the goodwill represents how much we allow a cheater to win *in expectation*. By kicking the gamblers with history $s$ out with probability $p_s < 1$, we can allow those that are not kicked out to win a factor $1/(1-p_s)$ more, and still provide the same guarantee in expectation against cheaters. As we shall see shortly, if the goodwill is less than $v_n$, the optimal kicking probability raises the goodwill so that it becomes equal to $v_n$, and if the goodwill is at least $v_n$, we do not kick at all.

We now formally define both goodwill and optimal policies. Since the optimal policy is defined from goodwill, the following definition of goodwill already has the policy substituted in:

**Definition 1 (Goodwill).** *Given an initial amount of goodwill, $G \geq 0$, the goodwill towards a gambler with history $s$ is given by $g(s, G)$:*

$$g(\emptyset, G) = G \tag{1}$$

$$g(s.v, G) = \max(g(s, G), v_n) - v \tag{2}$$

*where $s.v$ is the composition of the history $s$ with the outcome $v$ at the end.*

and we immediately define the corresponding policy as:

**Theorem 1.** *The following policy, $p_G$, maximizes the casino's expected utility against the honest gamblers, under the constraint of limiting the loss against cheaters to at most $G$:*

$$p_{G,s} = \max(0, 1 - g(s, G)/v_n) \tag{3}$$

That is, the kicking probability is 1 when there is 0 goodwill, 0 when there is at least $v_n$ goodwill, and it varies linearly between 1 and 0 in the interval $[0, v_n]$.

To prove the theorem, we will first construct an expression, $h(g, r)$, which captures how much the casino wins in expectation from an *honest* gambler over the remaining $r$ rounds, by using the policy defined in Theorem 1 for sequences with $g$ remaining goodwill. We will then prove that no policy can win more from the honest gamblers, without letting cheaters win more than $g$ in expectation, thereby proving Theorem 1. Thus, $h(g, r)$ captures the trade off between earnings on the honest gamblers and losses on the cheaters. Once this is established, the casino's optimization problem becomes:

$$\max_g \{(1 - c) \cdot h(g, R) - c \cdot g\} \tag{4}$$

The base case for $h(g, r)$ is when there are no rounds left. In that case, the casino closes, and no one wins any more. Hence, $h(g, 0) = 0$. Similarly, if we cannot allow a cheater to win anything, we have to kick everyone out, so no one wins any more, and $h(0, r) = 0$. In all other cases, we can express $h(g, r)$ in terms of $h(\cdot, r - 1)$ by letting the honest gambler play for another round, while reducing the expected winning by a factor equal to the probability of not being immediately kicked out by the policy. That gives the following recurrence:

**Proposition 1.** *The expected utility to the casino using the policy defined in Theorem 1, over the remaining $r$ rounds, against an honest gambler with remaining goodwill $g$ is given by:*

$$h(g,r) = \begin{cases} 0, & \text{if } r = 0 \\ \frac{g}{v_n} \sum_{v \in V} P(v) \cdot (h(v_n - v, r-1) - v), & \text{if } g < v_n \\ \sum_{v \in V} P(v) \cdot (h(g - v, r-1) - v), & \text{otherwise} \end{cases} \tag{5}$$

We need to prove a few lemmas about $h(g, r)$, before we can prove that the policy used to compute it is optimal. First, we need a lemma that says that with more rounds, the casino can get more money from the honest gamblers:

**Lemma 2.** *For any fixed $g$, the function $h(g, r)$ is non-decreasing in $r$. Formally: $h(g, r + 1) \geq h(g, r)$.*

*Proof.* The proof goes by induction on $r$. The base case is $r = 0$, for which $h(g, r) = h(g, 0) = 0$ by definition. Furthermore, for $g \geq v_n$, we have that

$$h(g, 1) = \sum_{v \in V} P(v) \cdot (h(g - v, 0) - v) = -EV > 0 = h(g, 0) \tag{6}$$

Since this also holds for $g = v_n$, and $h(g, r)$ is linear in $g$ for $g \in [0; v_n]$, we have that $h(g, 1) \geq h(g, 0)$ for all $g$. Now, assume for induction that $h(g, r) \geq h(g, r - 1)$. For $g \geq v_n$, we have that

$$h(g, r + 1) = \sum_{v \in V} P(v) \cdot (h(g - v, r) - v) \tag{7}$$

$$\geq \sum_{v \in V} P(v) \cdot (h(g - v, r - 1) - v) \tag{8}$$

$$= h(g, r) \tag{9}$$

Again, since this also holds for $g = v_n$, and $h(g, r)$ and $h(g, r + 1)$ are linear in $g$ for $g \in [0; v_n]$, we have that $h(g, r + 1) \geq h(g, r)$ for all $g$, which concludes the proof. □

We also need the following lemma:

**Lemma 3.** *For any fixed $r$, the function $h(g, r)$ is concave in $g$.*

*Proof.* The proof goes by induction on $r$. The base case is $r = 0$, for which $h(g, r) = h(g, 0) = 0$ by definition, so the statement trivially holds. Now assume for induction that the statement holds for $r - 1$. There are three cases in this proof; one for each of $g < v_n$, $g > v_n$, and $g = v_n$.

**case $g < v_n$:**
In this case, we have that $h(g, r) = g/v_n \cdot h(v_n, r)$, i. e. , a linear function in $g$, and it is trivially concave.

**case $g > v_n$:**
In this case, $h(g, r) = \sum_{v \in V} P(v) \cdot (h(g - v, r - 1) - v)$. Since $h(g, r - 1)$ is concave for all $g$ by induction, so is $h(g - v, r - 1) - v$. Thus $h(g, r)$ is a linear combination of concave functions, and is therefore concave.

**case $g = v_n$:**
The function is only semi-differentiable at $v_n$, so to prove concavity, we have to prove that the right derivative is not greater than the left derivative.

$$\frac{\partial_+}{\partial g} h(g, r) = \frac{\partial_+}{\partial g} \sum_{v \in V} P(v) \cdot (h(g - v, r - 1) - v) \tag{10}$$

$$= \sum_{v \in V} P(v) \cdot \frac{\partial_+}{\partial g} h(g - v, r - 1) \tag{11}$$

$$\geq \sum_{v \in V} P(v) \cdot h(v_n, r - 1)/v_n \tag{12}$$

$$\geq h(v_n, r)/v_n = \frac{\partial_-}{\partial g} h(g, r) \tag{13}$$

The bound at (12) is due to $h(v_n, r-1)/v_n$ being the derivative of $h(g, r - 1)$ for $g \in (0; v_n)$. This works as an upper bound on the derivative, since it is the leftmost part of the domain, and the function is concave. The bound at (13) is due to Lemma 2. Thus, the function is also concave at $g = v_n$. Thus the induction hypothesis holds for $r$, which concludes the proof. □

We are now ready to prove the main theorem.

*Proof of Theorem 1:* First, we prove that no policy can kick a gambler out with a lower probability than what the theorem prescribes, and still guarantee than a cheater gets no more than $g$. This follows almost directly from the definition. If the current goodwill is at least $v_n$, then the policy prescribes kicking with probability 0, so it is only relevant to look at $g < v_n$. In this case, if a policy were to kick with a probability lower than $1-g/v_n$, then a cheater can get more than $g$ in expectation, by immediately choosing outcome $v_n$. Thus, the policy in Theorem 1 provides a lower bound on kicking probabilities that guarantee to limit cheaters to winning at most $g$.

Secondly, we need to prove that among such "loss limiting" policies, the one guaranteeing the highest utility against honest gamblers is exactly the policy from Theorem 1. We prove this by induction on $r$. The base case is $r = 0$, for which there is only one possible policy. Assume for induction that the policy from Theorem 1 is optimal with $r-1$ rounds left. This means that $h(g, r-1)$ has the optimal value over all policies. For any given $g$, we can decide to kick with a higher probability with $r$ rounds left. Since $h(g, r)$ captures what we get by kicking with the lowest allowed probability, we can express our optimization problem as:

$$\max_{p\in[0,1)} \{(1-p)\cdot h(g/(1-p), r)\} \tag{14}$$

where $p$ represents an additional probability of kicking, before reverting to the policy from Theorem 1. We can exclude the special case $p = 1$, since this kicks everyone out, guaranteeing an outcome of 0, which cannot be optimal for $g > 0$. The expression in (14) is maximized at $p = 0$, since:

$$\begin{aligned} h(g,r) &= h(p\cdot 0 + (1-p)\cdot g/(1-p), r) && (15)\\ &\geq p\cdot h(0,r) + (1-p)\cdot h(g/(1-p), r) && (16)\\ &= (1-p)\cdot h(g/(1-p)) && (17)\end{aligned}$$

The first equality is valid, since $p \neq 1$. The second is Jensen's inequality, since $h(g, r)$ is concave in $g$. The last equality is due to the fact that $h(0, r) = 0$. Thus, the optimal additional kick probability is 0, meaning that the minimal kicking probability prescribed by Theorem 1 is optimal. $\square$

With Theorem 1 proven, we know that $h(g, r)$ exactly captures the tradeoff between the casino's earnings on honest gamblers versus how much they lose to the cheaters.

## Computing optimal initial goodwill

From the previous section, we know that we can focus on computing $h(g, r)$. We can do this efficiently using dynamic programming by using the following lemma:

**Lemma 4.** *$h(g, r)$ is a piecewise linear function in g, with end-points in $M = \{k \cdot gcd(V) \mid k \in \mathbb{N}_0\}$, where $gcd(V)$ is the greatest common divisor of the $v_i$'s. The optimal goodwill is therefore also to be found in the set $M$.*

*Proof.* The proof goes by induction on $r$. The base case, $r = 0$, is trivially satisfied, as $h(g, 0) = 0$. Assume for induction that the lemma holds for $r-1$. For $g \geq v_n$, we have that $h(g, r) = \sum_{v\in V} P(v) \cdot (h(g - v, r - 1) - v)$. This is a weighted average of piecewise linear functions with end-points in $M$. Therefore $h$ will also be a piecewise linear function. Furthermore, it will only have an end-point, whenever one of the terms $h(g - v, r - 1)$ is at an end-point. By induction, this can only happen at a point in $M$. By definition of $M$, if $g - v$ is in $M$, then so is $g$. Thus $h(g, r)$ can only have end-points in $M$. Since this holds for $g \geq v_n$, and $h(g, r)$ is linear in $g$ for $g \in [0; v_n]$, the lemma holds for all values of $g$. $\square$

This means that we can focus on computing the values of $h(g, r)$ for $g \in M$. Furthermore, we do not need to compute it for $g \geq r \cdot v_n$, since that is more goodwill than a gambler can lose on the remainder of the game, and thus $h(g, r) = -EV \cdot r$. We can thus construct a table of the values of $h(g, r)$ for $g \in M \cap [0, Rv_n]$ and $r \in \{1, \ldots, R\}$. This table has $R^2 v_n/gcd(V)$ entries, each requiring $O(|V|)$ work. See the experiments section for estimates of constants.

## Deterministic policies

In general, optimal policies require the use of randomization, but in some cases, the optimal policies will always be deterministic. The deciding factor is the structure of the set of outcomes, $V$.

**Definition 2 (Inverse jackpot).** *A set of outcomes, $V$, is called an inverse jackpot if*

$$V \subseteq \{v_n \cdot z \mid z \in \mathbb{Z}\} \quad , \tag{18}$$

*i.e., $gcd(V) = v_n$.*

In other words, all outcomes other than $v_n$ are some non-positive integer multiple of $v_n$; you either win a little, or lose big. Inverse jackpots distinguish themselves by the following property:

**Corollary 1.** *If the set of outcomes is an inverse jackpot, then there is a deterministic policy that is optimal.*

*Proof.* The first part follows directly from Lemma 4, which implies that optimal goodwill is an integer multiple of $v_n$ for inverse jackpots, and therefore so is the resulting goodwill of all sequences. By Theorem 1, all kicking probabilities will either be 0 or 1. $\square$

Gambles that are not inverse jackpots will not admit optimal deterministic policies, for any sufficiently large $R$, assuming $P(v_n) < 1$ and $EV < 0$, but we will not prove that here.

It is possible to find the optimal deterministic policy, using the same techniques as for randomized policies from the previous section. The calculating differs in two places. Most importantly, restricting $p$ to be either 0 or 1 in the definition of $h$ leads to the following form:

$$h_d(g,r) = \begin{cases} 0, & \text{if } r = 0 \text{ or } g < v_n \\ \sum_{v\in V} p(v)\cdot(h(g-v, r-1) - v), & \text{otherwise}\end{cases} \tag{19}$$

Secondly, the cheater's utility against the deterministic policy derived from goodwill $G$ is not necessarily $G$, as this requires the existence of a sequence $s$ of outcomes of length at most $|R|$ with at least one $v_n$ and $val(s) = G$. This can either be solved directly, or the reachable utilities can be computed by dynamic programming.

## Approximating the optimal policy

The previous section made several assumptions that might not fit the problem at hand. It might be the case that $R$ is too large for it to be feasible to compute the table for the dynamic program, or it might be that $v_n/gcd(V)$ is prohibitively large. Slightly more obscure, if $V$ is not given as rational numbers, it might be the case that $V$ does not have a greatest common divisor, e.g., if $V = \{-\sqrt{2}, 1\}$. It would still be possible to do a similar dynamic programming approach, but the table would in general be exponentially large in the number of such independent $v_i$'s, which is worst case all of $|V|$.

We therefore turn our attention to approximations. If the number of rounds is large enough, the gambling of an honest gambler can be approximated by a Brownian motion with drift [Durrett, 2010]. We can use this to approximate the optimal initial goodwill for a randomized policy.

### Continuous time

If the game is continuous over a period of time with very small and frequent gambles, we can use Brownian motion to find the optimal threshold at which to kick out a gambler. Define $S_n$ as the total earnings up to game $n$, $\tau$ as the stopping time when the gambler gets kicked out of the casino, and $T$ as the final time the gambler could keep playing until if he never gets kicked out. Let $X^n$ be the outcome of the $n$th game for an honest gambler. We know that $X^n$ is normally distributed as follows:

$$X^n \sim N\Big(\frac{\mu}{n}, \frac{\sigma^2}{n}\Big) \tag{20}$$

Thus the sum of the earnings from time 1 to $n$ is given as:

$$\sum_{i=1}^{n} X_i \sim N(\mu, \sigma^2) \tag{21}$$

Then, summing over time 1 to $n$ and games 1 to $t$, we get that for $t \in \mathbb{Z}$

$$S_{tn} = \sum_{j=1}^{t}\sum_{i=1}^{n} X^n_{j,i} \sim N\Big(\frac{\mu n t}{n}, \frac{\sigma^2 n t}{n}\Big) = N(\mu t, \sigma^2 t) \tag{22}$$

We define the standard normal variable $z^n_t$ as:

$$z^n_t = \frac{X_{tn} - E_{tn}}{\sqrt{\text{var}(S_{tn})}} \tag{23}$$

To model this as a Brownian motion with drift, we get

$$\sqrt{t} z^n_t = \frac{\sqrt{t}(X_{tn} - E_{tn})}{\sigma\sqrt{t}} = \frac{S_{tn}}{\sigma} - \frac{\mu t}{\sigma} \tag{24}$$

So the probability that an honest gambler has not won more than a given limit $\lambda$ by time $t$ is

$$P(\tau_\lambda > t) = 1 - P(\tau_\lambda \le t) = 1 - P(z_i \le \lambda, \forall i \le t) \tag{25}$$

The probability density function [Herlemont, 2009] is given as follows

$$P^{\mu,\sigma^2}(t,\lambda) = N\left(\frac{\lambda - \mu t}{\sigma\sqrt{t}}\right) - \exp\left(\frac{2\mu\lambda}{\sigma^2}\right) N\left(\frac{-\lambda - \mu t}{\sigma\sqrt{t}}\right) \tag{26}$$

where $N$ is the cumulative distribution function of the standard normal distribution. In other words, $P^{\mu,\sigma^2}(t,\lambda)$ expresses the probability that an honest gambler has not been kicked out at time $t$. That means that the expected rate at which the casino is getting money at time $t$ from an honest gambler is $-\mu P^{\mu,\sigma^2}(t,\lambda)$. The expected winnings over the entire evening is the integral of the rate:

$$H(\lambda, T) = \int_{t=0}^{T} -\mu P^{\mu,\sigma^2}(t,\lambda) \tag{27}$$

Because of the $N$ terms in the expression for $P^{\mu,\sigma^2}(t,\lambda)$, this has to be evaluated numerically, which can be done to high precision by most mathematics software. In the experiments section, we used Mathematica for the task.

The value of this integral, $H(\lambda, T)$, gives the amount that an honest gambler is expected to lose over time length $T$, when gamblers are kicked out at any time they have won $\lambda$. It thus plays the same role as $h(g, r)$ in the discrete case. A cheater will just get $\lambda$. The casino is therefore interested in optimizing:

$$\max_{\lambda}\{-c \cdot \lambda + (1-c) \cdot H(\lambda, T)\} \tag{28}$$

This expression is concave in $\lambda$, so it can be maximized by a simple ternary search, evaluating $H$ numerically at each step. In the next section, we evaluate using optimal $\lambda$ as the input to the policy defined by Theorem 1.

## Experiments

In this section we provide the results of experiments that compare the casino utilities from different kicking policies, and the scalability of the linear and dynamic programs. We use an example game in which the gambling population consists of 30% cheaters and 70% honest gamblers. The game is played for $R$ rounds and has $K = |V|$ outcomes, each of which are chosen uniformly at random from $[-2000, 1000]$, with $P(v) = 1/K$ for all outcomes.

### Casino utility

First, we consider how the casino utility compares between the different kicking algorithms for $K = 100$, and $R = \{10, 20, ..., 100\}$.

The performance of the different policies can be seen in Figure 1. All performances are relative to the optimal randomized policy, which is the top-most line. Just below, barely distinguishable from optimal, is the randomized policy with goodwill derived from the Brownian motion model. On all data points, it is above 99.9% of optimal. The third line from the top is how much Brownian motion underestimates the true optimal utility, and it is thus not a performance of a kicking policy. The fourth line is the optimal deterministic kicking policy. Finally, the last line is the deterministic policy with goodwill derived from the Brownian motion.

Unsurprisingly, both the deterministic and randomized policies result in greater casino utility by using the calculated goodwills from those policies rather than the Brownian motion approximation of the optimal goodwill. However, as

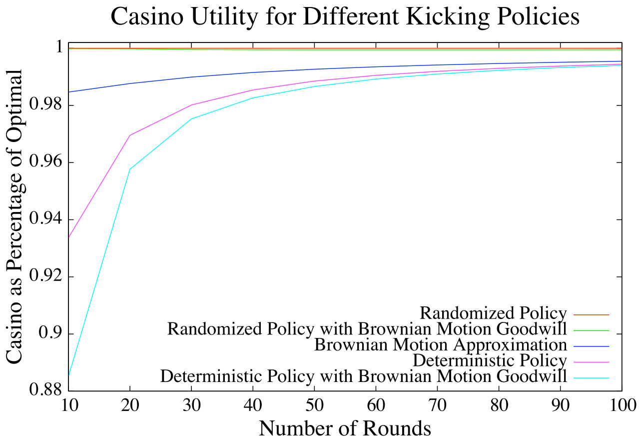


Figure 1: Casino Utility for Randomized, Deterministic, and Brownian Motion-approximated kicking policies

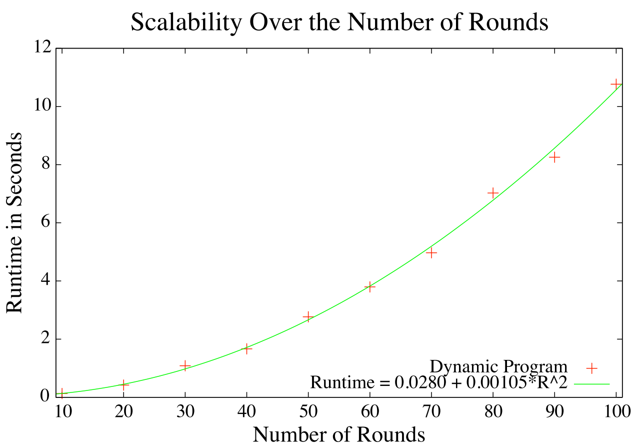


Figure 2: Scalability in number of rounds

the number of rounds increases, the random and deterministic policies result in increasingly similar casino utilities. The randomized policy with the Brownian motion approximated goodwill becomes very slightly worse as the number of rounds increases, and the deterministic policy with Brownian motion approximated goodwill improves significantly. The benefit of using the Brownian motion approximation is that the ternary search with numerical integration is much faster to compute than the full dynamic program.

## Efficiency of computation

We ran performance tests on the LP approach as well as the dynamic programming approach. The LP approach was evaluated for two outcomes, for which it becomes impractical to solve for $R \geq 16$. Performance would be worse for more outcomes. The dynamic program was tested with $K, R \in \{10, 20, \ldots, 100\}$. Each of the 100 combinations were timed as an average over 10 runs. We found $time = 50.49 + 0.01132 \cdot K \cdot R^2$. The p-value of this model less than $2.2e-16$, and the adjusted R-squared is 0.9894.

## Summary and future work

We modeled the casino's optimal policy for maximizing its earnings by limiting the impact of cheaters, assuming that the cheaters have full control over the outcome of the game. We presented a dynamic programming approach, which efficiently computes the casino's optimal policy by finding the optimal initial goodwill. It is easy to execute the policy based on goodwill of a gambler; subtract his winnings from his goodwill. If his goodwill drops below $v_n$, kick him out with probability $1 - goodwill/v_n$. If he wasn't kicked out, round his goodwill up to $v_n$ and let him continue gambling. This runtime of the dynamic program is linear in the number of outcomes and quadratic in the number of rounds.

A special case for this repeated game is when gambles are small enough that the casino's and gambler's utility is smoother. In this case, we can use Brownian motion to calculate an honest gambler's expected earnings in order to efficiently approximate the optimal kicking policy for the casino. This Brownian motion will have negative drift since an honest gambler's expected utility is negative if the casino is profitable. We showed how to use this Brownian motion model to quickly get a very good approximation of the optimal initial goodwill. Since the goodwill defines the policy, an approximation translates to an easy to execute policy.

We exploited the casino's power of evicting cheaters from the casino when it reaches our policy's threshold. But there are also other ways for the casino to strategize so that the cheater doesn't always cheat. For example, instead of deciding simply whether or not to kick a cheater out, the casino might also decide to perform an investigation on a gambler that comes with a cost as well as a probability that the investigation provides the correct conclusion about a gambler. In other domains such as catching insider trading, performing investigations might be more reasonable than without evidence stopping a trader from continuing to trade.

Future work should also be done to break down some of the assumptions in this paper about the cheaters and the honest gamblers. Rather than a cheater that has complete control over the outcome of the game, we could have cheaters that only have some control over the game. In this case, there might be different types of cheaters whose probability of a win comes from different distributions, depending on how successful of a cheater he can be. Another assumption was gamblers would never choose to leave the casino, and that they only leave if the casino kicks them out, or when the casino closes for the night. However, it might be more realistic that honest gamblers are willing to continue playing until they lose a certain amount of money, at which point they might leave to cut their losses.

We also want to apply this framework to other scenarios in which the leader has the power to end the game at any point in order to disincentivize an adversary. One could apply this to financial markets in which clearinghouses or policy makers want to reduce the frequency and size of insider trading. When a trader reaches a certain threshold that an honest trader would likely never reach, security may want to prevent the trader from continuing to trade. This is much like the framework presented in this paper in which the casino wants to prevent cheaters from winning too much.

## Acknowledgements

We would like to thank the helpful comments of the AAAI anonymous reviewers. We would also like to thank Prof. Susan Rodger for useful feedback on a previous version of the paper. We thank ARO and NSF for support under grants W911NF-12-1-0550, W911NF-11-1-0332, IIS-0953756, CCF-1101659, and CCF-1337215.


## References


Alpcan, T., and Basar, T. 2003. A game theoretic approach to decision and analysis in network intrusion detection. In *42nd IEEE Conf. on Decision and Control*.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. 2002. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing* 32(1):48–77.

Cesa-Bianchi, N.; Freund, Y.; Haussler, D.; Helmbold, D.; Schapire, R.; and Warmuth, M. 1997. How to use expert advice. *Journal of the ACM* 44(3):427–485.

Conitzer, V., and Sandholm, T. 2006. Computing the optimal strategy to commit to. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*, 82–90.

Durrett, R. 2010. Essentials of stochastic processes.

Freeman, P. R. 1983. The secretary problem and its extensions: a review. *International Statistical Review* 51(2):189–206.

Gittens, J. C. 1979. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society. Series B (Methodological)* 41(2):148–177.

Herlemont, D. 2009. Stochastic processes and related distributions. www.yats.com/doc/stochastic-processes-en.pdf.

Jain, M.; Conitzer, V.; and Tambe, M. 2013. Security scheduling for real-world networks. In *AAMAS-13*.

Kiekintveld, C.; Jain, M.; Tsai, J.; Pita, J.; Tambe, M.; and Ordóñez, F. 2009. Computing optimal randomized resource allocations for massive security games. In *AAMAS-9*.

Letchford, J., and Conitzer, V. 2010. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce (EC-10)*, 83–92.

Littlestone, N., and Warmuth, M. K. 1994. The weighted majority algorithm. *Information and Computation* 108(2):212–261.

Pita, J.; Jain, M.; Western, C.; Portway, C.; Tambe, M.; Ordóñez, F.; Kraus, S.; and Parachuri, P. 2008. Deployed armor protection: The application of a game-theoretic model for security at the los angeles international airport. In *AAMAS-08 (Industry Track)*.

Tsai, J.; Rathi, S.; Kiekintveld, C.; Ordóñez, F.; and Tambe, M. 2009. Iris a tool for strategic security allocation in transportation networks. In *AAMAS-09 (Industry Track)*.

von Stengel, B., and Zamir, S. 2010. Leadership games with convex strategy sets. *Games and Economic Behavior* 69:446–457.