Deriving an Optimally Deceptive Policy in Two-Player Iterated Games

Elisabeth Paulson
Booz Allen Hamilton
Annapolis Junction, MD 20701
E-mail: elisabethpaulson63@gmail.com

Christopher Griffin
Department of Mathematics
United States Naval Academy
Annapolis, MD 21402
E-mail: griffinch@ieee.org

***Abstract*— We formulate the problem of determining an optimally deceptive strategy in a repeated game framework. We assume that two players are engaged in repeated play. During an initial time period, Player 1 may *deceptively train* his opponent to expect a specific strategy. The opponent computes a best response. The best response is computed on an optimally deceptive strategy that maximizes the first player's long-run payoff during actual game play. Player 1 must take into consideration not only his real payoff but also the cost of deception. We formulate the deception problem as a nonlinear optimization problem and show how a genetic algorithm can be used to compute an optimally deceptive play. In particular, we show how the cost of deception can lead to strategies that blend a target strategy (policy) and an optimally deceptive one.**

I. Introduction

The use of deception in strategic interactions creates an extra layer of complication over ordinary Nash equilibrium play. In games with a unique Nash equilibrium, deception is irrelevant as all players are expected to play their equilibrium strategy. Real-world strategic interactions rarely exhibit a single Nash equilibrium (and may exhibit no Nash equilibrium at all [1]). In the case of multiple Nash equilibria or no Nash equilibrium, deception may play a critical role in improving a players net payout.

Burgoon and Buller define deception as a "deliberate act perpetrated by a sender to engender in a receiver beliefs contrary to what the sender believes is true to put the receiver at a disadvantage" [2]. This definition clearly includes a notion of intent, and thus accidental misinformation or incomplete knowledge may not be considered deception. Bell and Whaley, in [3], [4], have developed a general theory and classification of deception. They classify deception into two categories–dissimulative and simulative. Dissimilative deception attempts to hide the true information, whereas simulative deception attempts to show false information. This work is largely qualitative.

Understanding deception and trust from a modeling perspective is becoming increasingly important as information is being created, collected, and analyzed at increasing rates. Being able to parse true information from deceptive or false information is a difficult yet important task. In [5], Santos and Johnson discuss the need for intelligent systems (AI systems), to be able to detect deception. Santos also draws a distinction between intention deception, or misinformation, and unintentional deception, such as incomplete knowledge [5]. In a multi-agent system, the introduction of misinformation can lead incorrect information entering the system, resulting in wrong decisions being reached by the decision-maker [5]. In [6], George and Carlson found that humans can only detect deception slightly more than 50% of the time, and even less when the interaction is via electronic media.

We consider repeated games as a symbolic input/output dynamical system, where behaviors are learned and used in an optimization process. Learning in repeated games has its own rich literature. Maenner [7] studies complexity of the representation of strategy and optimal responses to learned strategies. His measure of complexity, however, follows Abreu and Rubinstein [8], which uses the number of states in the automata and does not make use of the formalism developed in [9], [10], which uses entropy measures on probabilistic automata and are asymptotically true representations of underlying symbolic dynamic systems following a Markov property. Learning is also considered in [11] where learning automata are used, but no connection to symbolic input/output dynamical systems is drawn, nor is deception considered. Neyman and Okada [12] also use finite automata to study repeated two player games, but do not derive probabilistic models from data, nor do they consider deception in their data. Seiffertt et al. [13] consider mixed neural network and Partially Observable Markov Decision processes of the iterated Prisoner's dilemma; there models use a neural network approach, which is compelling, but harder to represent in a closed form set of expressions as we do in this work. Additionally, the added complexity of a neural network may be unnecessary for our underlying goal of representing a symbolic input/output dynamic system with constraints we assume. Again, deception is not considered in this work.

Gossner and Vielle [14] study a case of learning in a repeated game where the initial payoff matrix is not known. This condition is more general than the one we consider, where we assume a payoff is known and strategies are deceptive. Soo et al. [15] study a multi-armed bandit approach to two-person zero-sum Markov games, without deception. Their algorithm is for a finite horizon game, and our approach considers deception applied to infinite horizon games. Finally, it is worth noting that Fuzzy Q-learning has been used in repeated games [16]. In this work, Ishibuchi et al. studied a market equilibrium problem using a fuzzification of Q-learning strategies. They later generalize this work to

fuzzy rule based systems [17]. None of this work considers deception.

In this paper we develop a general model for deciding optimal deception strategies in a repeated two-player game. We consider a scenario in which one player uses a method of simulative deception. For simplicity, we assume that the receiver is either unaware of, not suspicious of or incapable of dealing with the deception taking place. A few authors consider methods for detecting deception; Johnson et al. [18] builds on the work by Mawby and Mitchell [19] by classifying strategies for detecting deception into two broad groups: methods that attempt to detect evidence of deceptive behavior, and methods that search for deception information. We note that Griffin et al. show that under certain conditions detecting deception is NP-hard [20], [21], justifying our simplifying assumption.

We use the optimization technique described in [22] to derive an optimal deception strategy for generating an input/output sequence that is learned by an opponent and used to compute a (false) optimal response. In general, deriving an optimal response strategy under uncertainty can be formulated as Markov decision problem. The analysis in [22] uses a linear programming formulation of the Markov decision problem. This work is qualitatively similar to the work of Fuchs and Kargonakar [23], where the authors formulate a zero-sum game problem to reason about deception and show that a form of Jone's lemma can be extracted from the resulting duality conditions. In that study, behavior is one-shot but does consist of a full two-player interaction. In the model presented here, behavior is dynamic, there is a learning component, but we do not consider bi-directional deception. This work uses the optimal process control model presented in [22], and assumes a learning model described in [24].

The remainder of this paper is divided as follows: In Section II we provide a preliminary review of terms used in [24]. In Section III we formulate the optimal deception problem studied in this paper. We follow this in Section IV by studying a special case of our general problem when there is no cost to deception on the part of the deceiver. An example of this case is provided in Section V. Finally, the general case deception problem is studied in Section VI. Conclusions and future directions of research are provided in Section VII.

II. Preliminaries

In this section we provide the notation and preliminaries necessary for the proposed approach. Our notation is derived from [22].

A. Determining an Optimal Response in Repeated Play

We will assume that sets $\mathcal{A}_1$ and $\mathcal{A}_2$ correspond to the strategy spaces of two players in a stochastic non-cooperative game. These strategy spaces can also be thought of as finite alphabets corresponding to all possible moves that each player could make. Let $\mathcal{A}_1^l$ and $\mathcal{A}_2^l$ denote all strings of length l with symbols in $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. In this paper we will assume that there are known parameters l_1 and l_2 such that each player's next move is dependent only on the previous l_1 or l_2 moves, respectively.

A strategy for Player 1 as a function $\eta : \mathcal{A}_1^{l_1} \rightarrow \mathcal{F}_{\mathcal{A}_1}$ where $\mathcal{F}_{\mathcal{A}_1}$ is the set of probability distributions with support $\mathcal{A}_1$. Note, the function η is a *probabilistic labeled transition system* [22].

We can define a similar function for Player 2. However, since Player 2 can observe Player 1's behavior before formulating a strategy, Player 2's next move depends both on his previous l_2 moves and on Player 1's previous l_1 moves. Thus Player 2's *response* function is $\xi : \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2} \rightarrow \mathcal{F}_{\mathcal{A}_2}$. This defines a *probabilistic Mealey machine*. It is very similar to a probabilistic labeled transition system except that it defines a transition relation given both an input and output alphabet. Again see [22] or [24]. Note further, ξ is a *symbolic transfer function* [22], [24] taking symbolic inputs and providing an output.

Without loss of generality, let $Q = \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2}$ be the state space of ξ. Let $r_q : \mathcal{A}_1 \times \mathcal{A}_2 \rightarrow \mathbb{R}$ be the state parameterized reward function associated to the game. In the case of a repeated game, the function r_q is constant across q and can simply be referred to as r.

At each discrete time t we assume the players are in some state q. The players choose an action $(a, \alpha) \in \mathcal{A}_1 \times \mathcal{A}_2$ and receive reward $\beta^t r_{q(t)}(a, \alpha)$, where $\beta \in (0, 1)$ is a discounting factor. The state then becomes q'.

For all $q \in Q$ define the $m \times n$ matrix:

$$R_q = \begin{bmatrix} r_q(a_1, \alpha_1) & \cdots & r_q(a_1, \alpha_n) \\ \vdots & \ddots & \vdots \\ r_q(a_m, \alpha_1) & \cdots & r_q(a_m, \alpha_n) \end{bmatrix} \tag{1}$$

where $|\mathcal{A}_1| = m$ and $|\mathcal{A}_2| = n$. In the case where the function r is dependent not only on the current strategies, but the previous strategies then we would have a collection of functions r_q for each $q \in Q$.

For state $q = (\mathbf{u}, \mathbf{v})$ let $\xi(q) = \xi(\mathbf{u}, \mathbf{v})$ be the vector of probabilities corresponding to the probabilities that the various $\alpha \in \mathcal{A}_2$ will occur.

B. Known Symbolic Transfer Function ξ

If we are given the (formal symbolic transfer) function [24] ξ and the objective is to derive a policy η so that the long run pay-off is optimized under the β-discounting rule, then we maximize the payoff:

$$\Pi_1(\eta) = \sum_{t=0}^{\infty} \beta^t r_{q(t)}(\eta, \xi). \tag{2}$$

This problem can be coded as a Markov Decision Problem [25] in which we solve for policy η. Following [26], we may write the problem of maximizing long run reward subject to β discounting as:

$$\begin{cases} \min \ \sum_{q' \in Q} \pi_{q'}^0 v_{q'} \\ \text{s.t. } v_q \geq [R_q \xi(q)]_a + \beta \sum_{q' \in Q} \Pr(q'|q, a) v_{q'} \\ \qquad\qquad \forall q \in Q, a \in \mathcal{A}_1 \end{cases} \tag{3}$$

Here $\mathbf{v}$ is a vector in $\mathbb{R}^{|Q|}$ with elements v_q and for vector y, $[y]_a$ indicates the element of y corresponding to $a \in \mathcal{A}$. Problem 3 has known dual:

$$\begin{cases} \max & \sum_{q\in Q}\sum_{a\in\mathcal{A}} [R_q\xi(q)]_a x_{qa} \\ \text{s.t.} & \sum_{q\in Q}\sum_{a\in\mathcal{A}} \left[\delta(q,q') - \beta \Pr(q'|q,a)\right] x_{qa} = \pi^0_{q'} \ \ \forall q' \in Q \\ & x_{qa} \geq 0 \ \ \forall q \in Q, a \in \mathcal{A}_1 \end{cases} \tag{4}$$

Here x_{qa} are the dual variables corresponding to the $|Q| \times |\mathcal{A}_1|$ constraints in Problem 3 and $\delta(q,q')$ is the Dirac delta function. It should also be noted that $Pr(q'|q,a)$ is precisely $\xi(q)(\alpha)$ where $q' = \delta(q,a,\alpha)$. The following theorem follows immediately from Theorem 2.3.1 of [26].

Proposition 1: Let $\mathbf{x}^*$ be an optimal (vector) solution to Problem 4. For fixed $q \in Q$, let

$$x_q = \sum_{a\in\mathcal{A}} x_{qa} \tag{5}$$

Then the optimal policy η is given by:

$$\eta(q)(a) = \frac{x^*_{qa}}{x^*_q} \tag{6}$$

For fixed $\mathbf{u} \in \mathcal{A}_1^{l_1}$, if for all $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{A}_2^{l_2}$ we have $\eta(\mathbf{u},\mathbf{v}_1) = \eta(\mathbf{u},\mathbf{v}_2)$ then η is an open loop controller (for Player 1).

III. Deception Problem Formulation

Let $\Gamma = (\{P_1, P_2\}, \mathcal{A}_1 \times \mathcal{A}_2, \pi_1 \times \pi_2)$ be a two player game where P_1 and P_2 are the players, $\mathcal{A}_1$ and $\mathcal{A}_2$ are the set of possible actions for each player (so $\mathcal{A}_1 \times \mathcal{A}_2$ is the state space), and π_1 and π_2 are the payoff functions. Then we can definite a state $q \in \mathcal{A}_1^{l_1}$ as $a_1 a_2 \cdots a_{l_1}$ where $a_i \in \mathcal{A}_1$ and l_1 is the history length for P_1. We can do the same for P_2.

Suppose we extend Γ to a repeated game with a time-discounted payoff function $\Pi_1 \times \Pi_2$. Let $\xi : \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2} \to \mathcal{F}_{\mathcal{A}_1}$ be a *goal strategy* for Player 1. That is, the strategy he prefers to play (this may be proscribed by doctrine, policy or preference).

Suppose that Player 1 plays $y : \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2} \to \mathcal{F}_{\mathcal{A}_1}$, and generates $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2}$ where $\boldsymbol{\beta}$ is generated randomly.

Player 2 learns $\hat{y} : \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2} \to \mathcal{F}_{\mathcal{A}_1}$ using a learning algorithm such as CSSR or Baum Welch (see [10], [27]–[29]). Player 2 then develops η^* which is an optimal response to $\hat{y}$, generated by Problem 4. Then the resulting outcome of play is:

$$\widetilde{\Pi}_1 = \Pi_1(\xi, \eta^*) - \lambda||\xi - y||,$$

and

$$\widetilde{\Pi}_2 = \Pi_2(\xi, \eta^*),$$

where λ is the marginal cost of deception and $||\xi - y||$ is a metric measuring the difference between strategies ξ and y. We assume λ is constant, however in future work λ could be an increasing function in $|\boldsymbol{\alpha}|$ (the length of $\boldsymbol{\alpha}$) to reflect an increasing cost of deception over time. In this paper we define:

$$||\xi - y|| = \sum_{(w_1,w_2)\in\mathcal{A}_1\times\mathcal{A}_2} |\xi(w_1,w_2) - y(w_1,w_2)|. \tag{7}$$

though any metric will suffice.

The problem for Player 1 is to compute the optimal deception:

$$y^* = \arg\max_y \widetilde{\Pi}_1(\xi, \eta^*(y)), \tag{8}$$

where η^* is computed endogenously by Player 2, as the solution to the linear program in Expression 4.

IV. Zero cost of deception

In this section, we assume that $\lambda = 0$. Let $\eta' = \arg\max_\eta \Pi_1(\xi, \eta)$. That is, η' is the strategy for Player 2 that results in the best outcome for Player 1. This is computed from Expression 4, but using Player 1's payoffs, rather than Player 2's payoff. The ultimate goal of Player 1 is to deceive Player 2 into playing η'. In order to do so, Player 1 wishes to find a strategy y such that $||\eta^*(y) - \eta'||$ is minimized. This way, Player 1 can *train* Player 2 on strategy y so that Player 2 develops the optimal response η^* which is as close to η' as possible. The following proposition is clear from the model:

Proposition 2: Let $\eta' = \arg\max_\eta \Pi_1(\eta, \xi)$. If y^* solves:

$$\min_y ||\eta^*(y) - \eta'(\xi)||$$

then y^* is the optimal deception when $\lambda = 0$. ■

The problem of Player 1 is to find such a strategy y. When $\lambda = 0$ this can be accomplished with the genetic algorithm described in Algorithm 1. Algorithm 1 starts with a random set of possible strategies, $\{y_1, ..., y_n\}$: the *parent* set. Each strategy is evaluated according to an objective function which is computed in Algorithm 2. The *score* of each strategy is given in line 3 of Algorithm 2 by:

$$\left(\max_\eta ||\eta - \eta'||\right) - ||\eta^*(y_i) - \eta'||. \tag{9}$$

Notice that $\max_\eta ||\eta - \eta'||$ is a constant for all y_i, and is simply dependent on the chosen metric. In our case:

$$\max_\eta ||\eta - \eta'|| = 2\sum_i |\eta'_i| = 2|Q|.$$

When $\eta^* = \eta'$, the objective function is maximized.

The number of offspring for each parent is given in lines 5-9 of Algorithm 1. Thus, the number of offspring of a parent strategy is a direct result of how it scores compared to average. The new set of strategies consists of rep_i copies of strategy y_i. Finally, we use a crossover and mutation period in order to diversify the possible strategies considered. Each strategy is randomly crossed with another strategy, and mutated with probability P_{mutate}. This set of strategies then becomes the new *parent* set, and the process is repeated. This loop is repeated until all of the scores for each strategy is within an ϵ-bound of each other.

Algorithm 1 –Finding y

Input: a state space Q, alphabet $\mathcal{A}_2$, payoff function Π_2, a goal optimal strategy η', ϵ, a convergence tolerance

Genetic Algorithm

Start with a set of n possible strategies y, $\{y_1, ..., y_n\}$, where $y_i \in \mathcal{F}_{\mathcal{A}_2}$
obj $\leftarrow$ **objFunction**$(y, Q, \mathcal{A}_2, \Pi_2, \eta')$ {See Algorithm 2}
while $|\text{obj}_i - \text{obj}_j| > \epsilon \; \forall i, j$ **do**
  for $i = 1 : n$ **do**
    if $\text{obj}_i > \text{avg}_i(\text{obj}_i)$ **then**
      rep_i =Round$(\frac{\text{obj}_i}{\text{avg}_i(\text{obj}_i)})$
    else
      $\text{rep}_i = 0$
  $n \leftarrow \sum_i \text{rep}_i$
  Create the set of new strategies $y' = \{y'_1, ..., y'_n\}$ consisting of rep_i copies of y_i
  Randomly choose a crossover partner and location for each y'_i
  Mutate y'_i with probability P_{mutate}
  $y \leftarrow y'$
  obj $\leftarrow$ **objFunction**$(y, Q, \mathcal{A}_2, \Pi_2, \eta')$

Algorithm 2 –Objective Function for $\lambda = 0$

obj =**objFunction**$(y, Q, \mathcal{A}_2, \Pi_2, \eta')$

for i=1:n **do**
  Solve Problem 4 for $\eta^*(y_i)$
  $\text{obj}_i \leftarrow (\max_\eta ||\eta - \eta'||) - ||\eta^* - \eta'||$ {This transformation is done because we want to maximize obj_i instead of minimize}

V. Example

In this section we explore a numerical example where the game is Prisoner's Dilemma, and Player 1 wishes to use the *tit-for-two-tats* strategy during live play. This is strategy ξ. It is known that the forgiving nature of *tit-for-two-tats* can be exploited (see e.g., [22]) and therefore a deception will be employed.

For this strategy, $l_1 = 2$, meaning that Player 1's action depends on the previous two actions by both players. We also assume that play continues for infinite time. This strategy is shown in Figure 1 as a probabilistic state transition function dependent on Player 2's strategy. Capital letters are Player 1's actions and lowercase letters are Player 2's actions. Table I shows the payoff function for each player. As usual, Π_1 and Π_2 can be computed by taking the first and second values of Π, respectively.

TABLE I
Payoff Matrix Π

	d	c
D	-1,-1	3,-2
C	-2,3	1,1

Given desired strategy ξ, Player 1 can determine the strategy η' such that $\Pi_1(\eta', \xi)$ is maximized by solving Problem 4. For this specific problem, assuming an initial probability distribution of $\pi^0_{\{C,c\}} = 1$, Solving Problem 4 results in:

$$\eta'(\{CC, cc\})(c) = 1$$

Notice that the optimal response function is only specified for states that were reached. Since our initial probability distribution was $\pi^0_{\{CC,cc\}} = 1$, the game stays in $\{CC, cc\}$ forever. In order to determine η' completely, we must solve Problem 4 starting from every state. This yields the optimal response function:

$$\eta'(q)(c) = 1 \;\; \forall q \in \mathcal{A}_1^{l_1} \times \mathcal{A}_2^{l_2}$$

which means that η' is the strategy *always cooperate*. This is very sensible; it is in Player 1's best interest while playing *tit-for-two-tats* if Player 2 always cooperates.

Thus, Player 1 wishes to train Player 2 on some strategy y such that

$$\eta^*(y) = \eta' = \text{always cooperate.}$$

As it happens, the best response to the tit-for-tat strategy is to always cooperate. This strategy has history length of one. The tit-for-tat strategy is shown in Figure 2. Thus, in our example, Algorithm 1 could return

$$y = \text{tit-for-tat}$$

Note: there may be other strategies y for which $\eta^*(q) = C$ for all states q besides tit-for-tat. These are each alternative optimal solutions.

Once y is determined, Player 1 generates $(\boldsymbol{\alpha}, \boldsymbol{\beta}) \in \mathcal{A}_1^* \times \mathcal{A}_2^*$ using strategy y. A short example is shown in Table II for the tit-for-tat strategy. Using the observed sequence $(\boldsymbol{\alpha}, \boldsymbol{\beta})$, Player 2 learns $\hat{y}$ and then chooses an optimal response such that $\widetilde{\Pi_2} = \Pi_2(\eta^*, \xi)$ is maximized by solving Problem 4. Of course, this optimal response was already predetermined by Player 1 to be as close to η' as possible since y was determined by running Algorithm 1.

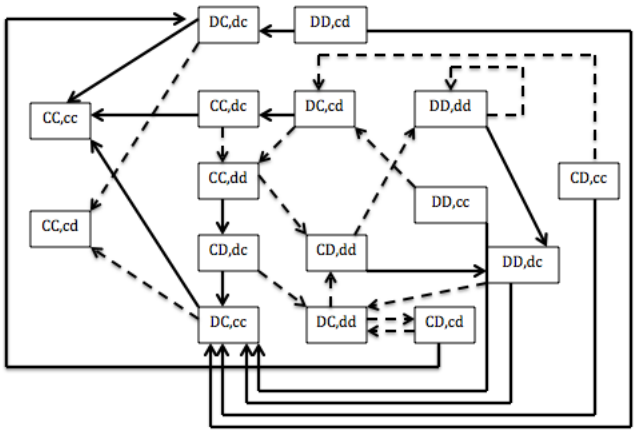

Fig. 1. The "tit-for-two-tat" strategy for Player 1. The states depend on Player 2's actions. A solid line shows the state transition when Player 2 cooperates and a dashed line shows the state transitions when Player 2 defects.

TABLE II
Input and Output Strings

$\boldsymbol{\beta}$	c	c	d	d	c	d	d	d	c	c
$\boldsymbol{\alpha}$	-	C	C	D	D	C	D	D	D	C

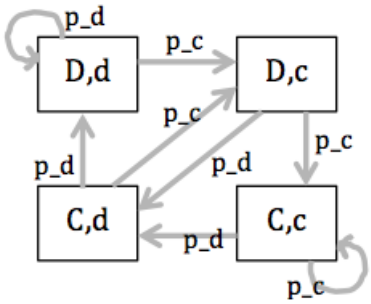

Fig. 2. The tit-for-tat strategy for Player 1. The states depend on Player 2's actions which are given as probabilities.

When the two players play against each other, Player 1 plays ξ and Player 2 plays η^*. In this case, since $y =$"tit-for-tat" results in $\eta^* = \eta'$, Player 2 ends up playing "always cooperate." Meanwhile, Player 1 plays ξ which is "tit-for-two-tat." The resulting play is simply stationary in state $\{CC, cc\}$. The payoff for both players is $\sum_{n=0}^{\infty} \beta^n = 1/(1-\beta)$.

If Player 1 had not deceived, i.e., if Player 2 could optimize against "tit-for-two-tat" instead of optimizing against "tit-for-tat". In this case, the optimal strategy is for Player 2 to alternate between cooperating and defecting, taking advantage of the forgiving tit-for-two-tat strategy. Using this alternating strategy instead, the game play alternates between state $\{CC, cc\}$ and $\{CC, cd\}$. In this case, when $\beta = .9$, the long run payoff to Player 2 is 14.73 and the long run payoff for Player 1 is -8.95.

VI. Nonzero cost of deception

When $\lambda \neq 0$, i.e., we assume that there is a cost of deception, the strategy y chosen by Player 1 must change so that

$$\widetilde{\Pi}_1 = \Pi_1(\xi, \eta^*) - \lambda||\xi - y||$$

is maximized. That is, Player 1 now must balance the game-play payoff with how much they are willing to deviate from strategy ξ. Such deception costs occur when negative training (i.e., generating $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$) coincides with ordinary practice by Player 1.

In this case, the genetic algorithm in Algorithm 1 can still be used, however the objective function must change. We can no longer simply seek a strategy minimizing $||\eta^*(y) - \eta'||$.

The new objective function is shown in Algorithm 3. For every strategy y_i, we must compute the payoff for that strategy, $\Pi_1(y_i, \eta^*(y_i))$ (Line 3), as well as the cost of deception, $||\xi - y_i||$ (Line 4). The overall score of a certain strategy is the payoff minus the cost of deception (Line 5). The strategy that scores the best must balance these two components. Because the values of obj_i could be negative, we also perform the transformation in Line 6 to ensure strict positivity.

We return to the tit-for-two-tat example. When $\lambda = 0$, we know that Algorithm 3 will return a strategy y which results in $\eta^*(y) =$"always cooperate". (Note: tit-for-tat is one such strategy, however there could be another strategy z such that $\eta^*(z) =$"always cooperate" as well. Algorithm 1 could return any such strategy.) In this case, we were able to find a strategy y such that $\eta' = \eta^*(y)$ exactly. By Proposition 2, the strategy y is our optimal deception with $\lambda = 0$.

Algorithm 3 –Objective Function for $\lambda \neq 0$

obj =**objFunction**$(y, \xi, Q, \mathcal{A}_2, \Pi_2, \Pi_1)$

1: **for** i=1:n **do**
2: Solve Problem 4 for $\eta^*(y_i)$
3: $\text{payoff}_i \leftarrow \Pi_1(y_i, \eta^*(y_i))$
4: $\text{cost}_i \leftarrow ||\xi - y_i||$
5: $\text{obj}_i \leftarrow \text{payoff}_i - \text{cost}_i$
6: $\text{obj}_i \leftarrow \text{obj}_i - \min_i(\text{obj}_i) + 1$ {This ensures that the objective function is positive and nonzero for each strategy y_i}

However, when $\lambda \neq 0$, Algorithms 1 and 3 produce slightly different strategies y' for varying values of λ. This is illustrated in Table III. Each strategy y' also results in a different strategy $\eta^*(y')$. Algorithms 1 and 3 now search for a strategy that not only maximizes $\Pi_1(y', \eta'^*(y'))$ but also simultaneously minimizes $\lambda||\xi - y||$. Therefore we might expect a slightly smaller value of $\Pi_1(y', \eta'^*(y'))$, but a strategy more similar to ξ, so that $||y' - \xi||$ is smaller. Table III shows the resulting strategies y' produced by Algorithms 1 and 3 for $\lambda = 1, 1.5, 2$, and 20 as well as the goal strategy ξ.

TABLE III
STRATEGIES y' WHEN $\lambda = 1, 1.5, 2, 20$

state	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2$	$\lambda = 20$	ξ
$\{CC, cc\}$	c	c	c	c	c
$\{CC, cd\}$	d	d	d	c	c
$\{CC, dc\}$	c	c	c	c	c
$\{CC, dd\}$	d	d	d	d	d
$\{CD, cc\}$	c	c	c	c	c
$\{CD, cd\}$	c	c	c	c	c
$\{CD, dc\}$	c	c	c	c	c
$\{CD, dd\}$	d	d	d	d	d
$\{DC, cc\}$	c	c	c	c	c
$\{DC, cd\}$	d	d	c	c	c
$\{DC, dc\}$	c	c	c	c	c
$\{DC, dd\}$	d	d	d	d	d
$\{DD, cc\}$	c	c	c	c	c
$\{DD, cd\}$	d	c	c	c	c
$\{DD, dc\}$	c	c	c	c	c
$\{DD, dd\}$	d	d	d	d	d

From Table III we see that for $\lambda = 1, 1.5$, and 2, y' lies *between* y and ξ in the sense that $||y - \xi|| > ||y' - \xi|| > 0$. As the value λ increases, the strategy resulting from Algorithm 3 approaches ξ and eventually becomes ξ at a critical value λ^*. As λ increases, the payoff for Player 1 is more highly weighted by the cost of deception. That is, as $\lambda \to \infty$,

$$\widetilde{\Pi} = \Pi_1(\xi, \eta^*) - \lambda||\xi - y|| \approx -\lambda||\xi - y||$$

Thus, as λ increases, Algorithm 1 along with Algorithm 3 will result in a strategy that is more and more similar to ξ in order to minimize $||\xi - y||$. However, since these strategies are discrete, there is a finite number λ^* at which Algorithms 1 and 3 will result in ξ exactly. In this example, the critical value is approximately $\lambda^* \approx 20$ and we can see in Table III

that when $\lambda = 20$, the resulting strategy y' is exactly equal to ξ, which is tit-for-two-tat.

VII. Conclusion and Future Directions

In this paper we developed a model that uses optimization and learning in symbolic time series to create optimal strategies for deception in a repeated two-player games. Using this model, a numerical example was presented which explored deception in the repeated Prisoner's Dilemma game. Future directions for this work include exploring how the length of the learning period affects the effectiveness of deception. Similarly, we hope to determine how long the learning period must be in order to effectively infer the strategy that is being utilized in that period. We also wish to apply this model to more complicated game structures, and possible to real-world war-game scenarios. Another area of future research includes either limiting the game-play to a finite number of stages instead of assuming infinite play, or allowing the deceived player to adjust his strategy during the game-play period as he realizes that he is being deceived. This may be a more realistic model formulation because it does not assume that the deceived player is completely naive. Finally, theoretical exploration of the optimal deception problem to identify solution methods beyond genetic algorithms is of interest.

References

[1] C. Griffin and G. Kesidis, "Behavior in a shared resource game with cooperative, greedy, and vigilante players," in *Information Sciences and Systems (CISS), 2014 48th Annual Conference on*, March 2014, pp. 1–6.

[2] J. K. Burgoon and D. B. Buller, "Interpersonal deception: Iii. effects of deceit on perceived communication and nonverbal behavior dynamics," *Journal of Nonverbal Behavior*, vol. 18, no. 2, pp. 155–184, 1994.

[3] J. Bell and B. Whaley, *Cheating and Deception*. Transaction Publ., 1991. [Online]. Available: https://books.google.com/books?id=ojmwSoW8g7IC

[4] B. Whaley, "Toward a general theory of deception," *Journal of Strategic Studies*, vol. 5, no. 1, pp. 178–192, 1982. [Online]. Available: http://dx.doi.org/10.1080/01402398208437106

[5] E. Santos Jr and G. Johnson Jr, "Toward detecting deception in intelligent systems," in *Defense and Security*. International Society for Optics and Photonics, 2004, pp. 130–141.

[6] J. F. George and J. Carlson, ""group support systems and deceptive communication"," in *Proceedings of the 32nd Hawaii International Conference on System Sciences*. IEEE, January 1999.

[7] E. Maenner, "Adaptation and complexity in repeated games," *Games and Economic Behavior*, vol. 63, pp. 166–187, 2008.

[8] D. Abreu and D. Rubinstein, "The structure of nash equilibria in repeated games with finite automata," *Econometrica*, vol. 56, pp. 1259–1282, 1988.

[9] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *Proc. ACM International Conference on Uncertainty in Artificial Intelligence*, 2004.

[10] C. Shalizi and J. Crutchfield, "Compuational mechanics: Pattern and prediction, structure and simplicity," *J. Statistical Physics*, vol. 104, no. 3/4, 2001.

[11] A. S. Poznyak and K. Najim, "Learning through reinforcement for n-person repeated constrained games," *IEEE Transactions on Systems, Man and Cybernetics-Part B*, vol. 32, no. 6, pp. 759–771, December 2002.

[12] A. Neyman and D. Okada, "Two-person repeated games with finite automata," *International Journal of Game Theory*, vol. 29, pp. 309–325, 2000.

[13] J. Seiffertt, S. Mulder, R. Dua, and D. C. Wunsch, "Neural networks and markov models for the iterated prisoner's dilemma," in *Proc. International Joint Conference on Neural Networks,*, Atlanta, GA, June 14-19 2009.

[14] O. Gossner and N. Vieille, "Strategic learning in games with symmetric information," *Games and Economic Behavior*, vol. 42, pp. 25–47, 2003.

[15] H. Soo, J. Hu, M. C. Fu, and S. I. Marcus, "Adaptive adversarial multi-armed bandit approach to two-person zero-sum markov games," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 463–468, 2010.

[16] H. Ishibuchi, T. Nakashima, H. Miyamoto, and C.-H. Oh, "Fuzzy q-learning for a multi-player non-cooperative repeated game," in *Proc. Sixth International Conference on Fuzzy Systems*, vol. 3, 1997, pp. 1573–1579.

[17] H. Ishibuchi, R. Sakamoto, and T. Nakashima, "Adaptation of fuzzy rule-based systems for game playing," in *IEEE International Fuzzy Systems Conference*, 2001.

[18] P. Johnson and S. Grazioli, "Fraud detection: Intentionality and deception in cognition," *Accounting, Organizations and Society*, vol. 25, pp. 355–392, 1993.

[19] R. Mawby and R. W. Mitchell, *Deception: Perspectives on Human and Nonhuman Deceit*. State University of New York Press, 1986, ch. Feints and Ruses: An Analysis of Deception in Sports.

[20] C. Griffin and K. Moore, "A framework for modeling decision making and deception with semantic information," in *Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, may 2012, pp. 68 –74.

[21] A. Squicciarini and C. Griffin, "Why and how to deceive: game results with sociological evidence," *Social Network Analysis and Mining*, vol. 4, no. 1, pp. 1–13, 2014.

[22] C. Griffin and E. Paulson, "Optimal process control of symbolic transfer functions," in *Proc. Feedback Computing '15*, Seattle, WA, April 13 2015.

[23] Z. Fuchs and P. Khargonekar, "Games, deception, and jones' lemma," in *American Control Conference (ACC), 2011*, June 2011, pp. 4532–4537.

[24] C. Griffin, R. R. Brooks, and J. Schwier, "Determining a purely symbolic transfer function from symbol streams: Theory and algorithms," in *Proc. American Control Conference*, Seattle, WA, June 11-13 2008, pp. 4065–4067.

[25] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY: John Wiley and Sons, 1994.

[26] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. New York, NY, USA: Springer-Verlag, 1997.

[27] L. Baum and J. Egon, "An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology," *Bull. Amer. Meteorol. Soc.*, vol. 73, pp. 360–363, 1967.

[28] L. Baum and G. Sell, "Growth functions for transformations on manifolds," *Pac. J. Math.*, vol. 27, no. 2, pp. 211–227, 1968.

[29] E. Paulson and C. Griffin, "Computational complexity of the minimum state probabilistic finite state learning problem on finite data sets," December 2014, submitted to Journal of Applied Mathematics and Computation.