A multiobjective optimization problem for environmental deception design in game theory

Austin Lee Davis*, Brett J. Borghetti†
Department of Electrical and Computer Engineering, The Air Force Institute of Technology
Wright-Patterson AFB, OH
Email: *austin.davis@afit.edu, †brett.borghetti@afit.edu

***Abstract*—The utility of deception is often assessed as a trade-off between its potential to benefit the deceiver, the cost of implementation, and the risk of being discovered or countered. Much has been done to study deception in multi-agent conflicts, but little has been done to address the benefit-cost-risk trade-offs made by real-world deceptive planners. This article leverages game theory and multiobjective optimization to introduce a novel approach for modeling and assessing environmental deception in normal form games. Environmental deception causes the deceptive target (called the mark) to misperceive the game's payouts. The goal of the deceiver is to compute a deceptive action (literally a distortion of the payouts) that achieves an efficient trade-off between benefit, cost, and risk. This goal is formulated as new multiobjective optimization problem—the deception design problem (DDP). Solutions to the DDP provide courses of deceptive action for consideration by a deceptive agent. As the first of its kind to design game-theoretic deceptive strategies using a multiobjective evolutionary algorithm, this seminal work greatly reduces the burden of hand-crafting deceptive strategies in multi-agent conflicts. A case study based on an air-to-air combat scenario demonstrates the DDP in a two-player, normal form game.**

***Index Terms*—Deception, Counterdeception, Game Theory, Multiobjective Evolutionary Algorithm**

I. Introduction

Game theory is "the study of mathematical models of conflict" [1]. It is applied to multi-agent conflicts where an agent's utility function depends on the combined actions of all agents. Game theory can help determine strategies that maximize an agent's expected utility in such situations.

Adversarial conflicts often give rise to deception. Game theory has been used to study deception in many different ways, but there remain several gaps in the research, especially pertaining to the cost and risk of deception [2]. Cost describes the amount of effort required to cause an opponent to act according to the projected view of the conflict. Risk describes the potential loss when playing an off-equilibrium deceptive strategy against a deception-aware opponent; it depends on the accuracy of the opponent model used to counter the deception [3].

In real-world conflicts, deception is commonly realized as a trade-off between the potential benefit to the deceiver, the cost of implementation, and the risk of being discovered or countered. However, little has been done to address this trade-off in the literature [2]. This may be attributed to the difficulties in modeling benefit, cost, and risk simultaneously: It is hard to formulate these objectives in a single utility function, and parameterized utility functions are not a general solution since they are not always comparable. It, therefore, comes as no surprise that most of the game theoretic (GT) literature considers only cost-free deceptions. Furthermore, models that measure deceptive risk are entirely absent from the literature [2]. The omission of risk is especially surprising because risk assessment is a critical factor in deception planning doctrine [4].

In light of these gaps, this article focuses on deception via payout manipulation in normal form games. Payout manipulation, also called environmental deception [2], [5], is achieved when the deceiver causes the target player (called the *mark*) to misperceive one or more of the game's utility functions. This article presents a GT model of environmental deception and formulates the task of designing efficient deceptive payout matrices as a multiobjective optimization problem—the deception design problem (DDP). A solution to the DDP is a payout matrix which, if presented to the mark as the true state of the conflict, is Pareto-efficient according to benefit, cost, and risk for the deceiver. Several versions of the DDP objective functions are discussed to help researchers articulate the deceiver's goals. A case study demonstrates the DDP using a 7×7 normal-form game. This game models the decision made by two pilots in an air-to-air engagement, and is a discretized version of the game introduced in [6].

The remainder of this article is organized as follows. Section II introduces the Environmental Deception Game as the model for environmental deception used in the DDP. Section III defines the DDP as a multiobjective optimization problem. Section IV describes the methodology for solving the DDP and presents an example based on the classic Prisoner's Dilemma. Section V presents a case study using a 7×7 normal form game. Related works are described in Section VI. Concluding remarks and areas for future work are given in Section VII.

II. The Environmental Deception Game

Environmental deception is a mechanism that causes the mark to misperceive the conflicts' payouts. Since the payouts express player preferences, misperception of the payouts can

cause the mark to select a suboptimal strategy. The deceiver can take advantage of the mark's misperception by anticipating their response and responding accordingly. The Environmental Deception Game (EDG) provides a model for this kind of deception. The EDG is defined by a tuple of payout matrices: the true payout matrix and the deceived payout matrix. The true payouts describe the actual state of the conflict; the deceived payouts are generated by the deceiver to mislead the mark. The game assumes, at the end of play, players observe their own received payout, but they do not know the true payouts received by their opponent.

The EDG instance with true payouts A and and deceived payouts B is denoted $G = \langle A, B \rangle$. Let a_{ij}^p denote the payout for player p at the intersection of row i and column j in the $m \times n$ payout matrix A. The expected utility for each player depends only on the payouts in A and the players' strategy profiles, since A represents the true state of the conflict. Thus, given strategy profiles s and t for the row and column player respectively, the expected utility for the row player is

$$E_{row}(G) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^{row} s_i t_j \qquad \text{(Row Expected Utility)}$$

and the expected utility for the column player is

$$E_{col}(G) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}^{col} s_i t_j \qquad \text{(Column Expected Utility)}$$

For consistency, the row player is the deceiver and the column player is the mark. The deceiver plays according to the true payouts in A while the mark plays according to the deceived payouts in B. The mark believes both players are playing the same game, i.e. according to the Nash equilibrium of the deceived payouts. However, the deceiver plays a different game: The deceiver anticipates the mark's strategy according to the deceived payouts and computes the best response according to the true payouts. In this way, the deceiver takes advantage of the mark's misperception of the conflict.

Denote by σ_X^p the strategy profile for player p according to the Nash equilibrium of the matrix X. Since the mark plays according to the payout matrix B, the mark's strategy profile is denoted, σ_B^{mark}. The deceiver's best response to σ_B^{mark} is called the *deceptive strategy* and denoted σ_D . The deceptive strategy $\sigma_D = \mathbf{X} = (x_1, x_2, \ldots, x_m)$ is computed by solving the following linear program:

$$\begin{aligned} \mathbf{X} \quad = \quad & \text{maximize}_x \quad \mathbf{C}x \\ & \text{subject to} \quad \sum_{i=1}^{m} x_i = 1, \\ & \qquad\qquad 0 \le x_i \le 1, \quad i = 1, \ldots, m \end{aligned}$$

where $\mathbf{C} = \{c_1, c_2, \ldots c_m\}$ and $c_i = \sum_{j=1}^{n} \left((\sigma_B^{mark})_j \cdot a_{ij}^{deceiver} \right)$ for $i = 1, \ldots, m$, with $(\sigma_B^{mark})_j$ being the j^{th} element of mark's strategy profile at the Nash equilibrium according to the deceived payouts B, and $a_{ij}^{deceiver}$ being the deceiver's payout in the ith row and jth column of the true payout matrix A.

Likewise, the mark's best response to σ_D is called the *counterdeception strategy* and denoted σ_{CD}. The mark's counterdeception strategy $\sigma_{CD} = \mathbf{Y} = (y_1, y_2, \ldots, y_n)$ is computed by solving the following linear program:

$$\begin{aligned} \mathbf{Y} \quad = \quad & \text{maximize}_y \quad \mathbf{D}y \\ & \text{subject to} \quad \sum_{j=1}^{n} y_j = 1, \\ & \qquad\qquad 0 \le y_j \le 1, \quad j = 1, \ldots, n \end{aligned}$$

where $\mathbf{D} = \{d_1, d_2, \ldots d_n\}$ and $d_j = \sum_{i=1}^{m} \left((\sigma_D)_i \cdot a_{ij}^{mark} \right)$ for $j = 1, \ldots, n$, with σ_D being the deceiver's's best response to the mark's strategy profile at the Nash equilibrium according to the deceived payouts in B, and a_{ij}^{mark} being the mark's payout in the ith row and jth column of the true payout matrix A.

III. The Deception Design Problem

The DDP is intended as a means of analyzing the benefit-cost-risk trade-off of environmental deception in multi-agent conflicts. As such, it is designed to answer the following question: Suppose one agent could successfully deceive another through environmental deception; how should such a capability be used to achieve an efficient trade-off between benefit, cost, and risk? Military utility analysts often consider questions of this kind in the earliest phases of a capability's acquisition life-cycle. They assume the capability exists and then try to predict its impact on operations. In the same way, this article is interested not in engineering a capability that performs environmental deception. Instead, it frames a discussion on efficient employment of such a capability and provides a means of predicting its impact. The Environmental Deception Game predicts the outcome of a conflict with one-sided misperception of the payouts. The DDP asks: If the row player (deceiver) could influence those misperceptions, how should they be changed?

The DDP makes two principal assumptions with regards to uncertainty. First, the deceiver's environmental deception is assumed to be successful, i.e. the mark correctly perceives the deceived payouts B and then plays according to the Nash equilibrium of B. Second, the deceiver is assumed to know the true payouts of the game. Relaxation of these assumptions is discussed in Section VII.

With this in mind, the DDP for two players is defined as follows. Given a payout matrix for a two-player game, A, compute a payout matrix, B, for the environmental deception game $G = \langle A, B \rangle$ that maximizes benefit (f_B) and minimizes cost (f_C) and risk (f_R). The functions f_B, f_C, f_R are defined below. For notational convenience, the solution where $G = \langle A, A \rangle$, is called the trivial solution and is denoted $s_\emptyset$.

A. Benefit

Benefit describes the increase in expected utility for using B as the environmental deception. Benefit is defined as

$$f_B(G) = E_{row}(G) - E_{row}(A), \qquad \text{(Benefit)}$$

where $E_{row}(G)$ is the deceiver's expected value in G and $E_{row}(A)$ is the expected value of the original game. Benefit compares the value of playing the environmental deception game G against playing the non-deceptive game according to A. The value of the trivial solution $s_\emptyset$ is $f_B(s_\emptyset) = 0$, since $G = \langle A, A \rangle$ implies $E_{row}(G) = E_{row}(A)$. On the other hand, when $A \neq B$, $f_B(G) \geq 0$, since misperception by the mark can cause him to deviate from the Nash equilibrium of the true game. The mark's unilateral deviation provides an opportunity for the deceiver to increase his expected utility over that of the Nash equilibrium by playing a best response. However, the deception provides zero benefit if the misperception of the mark does not induce changes in his equilibrium strategy.

B. Cost

The cost function describes the amount of effort that must be expended to cause the mark to believe the deception. It is assumed that the cost of the deception varies according to the difference between the true payout matrix and the deceived payout matrix, i.e. the distance between the two. Thus, it is sensible to use a metric function to define cost:

$$f_C(G) = \mathbf{dist}(A, B), \tag{Cost}$$

where $\mathbf{dist}(A, B) : \mathcal{M} \times \mathcal{M} \to [0, \infty)$ is a metric, $\mathcal{M}$ is the set of all real-valued $m \times n$ payout matrices, and $[0, \infty)$ is the set of non-negative real numbers. Table I presents several cost metrics and domains in which they might be useful. The metrics in Table I are defined so that the cost of deceiving any particular payout is treated the same as those in any other payout. In some circumstances, it may be appropriate to measure cost differently based on the particular payout being changed, i.e. when the cost varies based on both the change to the true payouts *and* the outcome payout being modified. This approach is useful if it were possible to deceive on some payouts cost-free while other payouts in the same game are not cost-free.

Table I
EXAMPLE COST METRICS

Cost Metric	Problem Domain
0	Cost-free deceptions
$\sum_{ijp} \left\lvert a_{ij}^p - b_{ij}^p \right\rvert$	Cost of deception is constant regardless of payout value
$\sum_{ijp} \sqrt{\left(a_{ij}^p - b_{ij}^p\right)^2}$	Small payout manipulations cost proportionately less than large payout manipulations
$\sum_{ijp} \left\lvert \int_{a_{ij}^p}^{b_{ij}^p} \frac{2}{1-\lvert 2x-1 \rvert} dx \right\rvert$	Payouts are constrained to a range (e.g. [0,1]), and cost grows asymptotically near the limits of a payout's range

C. Risk

Deception sometimes involves an element of surprise, but often the best deception is the tacit one—the one that remains undiscovered even after many encounters. At what point does such a tacit deception pose a risk to the deceiver? It poses a risk if the mark discovers the deception and can formulate a counterdeception strategy before the deceiver changes his own strategy. Considering risk in a game where the deception is assumed to be successful may appear at first glance to be nonsense, but there is a potential for risk if the deception is revealed through the gameplay itself. If the mark is surprised by the outcome, he may question whether or not he perceived the conflict accurately. Surprise could spur an update to his belief state that uncovers the deception. If he discovers the true payouts unbeknownst to the deceiver, he could formulate the counterdeception strategy (σ_{CD}) that defeats the deceiver in subsequent encounters.

Risk assessments usually involve calculating two critical components: the magnitude of a potential consequence and the likelihood of its occurring [7]. For DDP, the magnitude of the consequence depends on the deceiver's level of exposure when playing an off-equilibrium strategy. Consequence is measured as the deceiver's value loss in the worst-case scenario, i.e. when the deceiver assumes the mark is successfully deceived, but the mark counters the deception perfectly. Thus, consequence is expressed as:

$$R_C = E_{row}(A) - \left(\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^{deceiver} (\sigma_D)_i (\sigma_{CD})_j \right) \tag{Consequence}$$

where $(\sigma_{CD})_j$ is the j^{th} element of the mark's counterdeception strategy and $(\sigma_D)_i$ is the i^{th} element of the deceiver's deception strategy.

DDP assumes the deception is initially successful and that the deception is revealed if the payouts do not match the mark's perception of the game. The likelihood component of risk is based on the probability that the game's actual outcome and the mark's perception of the conflict are inconsistent, i.e. that the deception is revealed when the game is played. For the deception to be revealed, two conditions must be satisfied. First, the mark must receive a payout. Given strategy profile $s = (s_1, \ldots, s_m)$ for row and strategy profile $t = (t_1, \ldots, t_n)$ for column, the probability of receiving outcome a_{ij} equals $s_i \cdot t_j$ and is denoted $p_{ij}(s, t)$. The second condition is that the payout received by the mark must contradict his perception of the outcome's payout. The discrete indicator function $\mathbf{I} : \mathbb{R} \times \mathbb{R} \to \{0, 1\}$ is used to indicate whether or not the mark's perception of the game agrees with the payout he receives. If $x = y$, then $I(x, y) = 0$. Otherwise, $I(x, y) = 1$. Thus, likelihood is defined by the following equation:

$$R_L = \sum_{ij} \mathbf{I}\left(a_{ij}^{mark}, b_{ij}^{mark}\right) \cdot p_{ij}(\sigma_D, \sigma_B^{mark}) \tag{Likelihood}$$

Given an environmental deception game G with likelihood of detection $R_L(G)$ and consequence $R_C(G)$, risk is expressed as the product:

$$f_R(G) = R_L(G) \cdot R_C(G) \tag{Risk}$$

To show how risk is affected by consequence, consider the situation where the deceiver changes all the payouts i.e. when $\forall i, j, I(a_{ij}, b_{ij}) = 1$. In this case, regardless of the strategies adopted by the players, the likelihood of discovery equals

$$\sum_{ij} 1 \cdot p_{ij}(s, t) = 1 \cdot \sum_{ij} p_{ij}(s, t) = 1$$

Thus, the deception is certain to be discovered. If the deception strategy σ_D is off-equilibrium, then the deceiver is exposed to the counterdeception strategy σ_{CD} and $R_C > 0$. In this case, $f_R = R_L \cdot R_C = 1 \cdot R_C > 0$. On the other hand, if the deception strategy σ_D is on-equilibrium, the deceiver does not change his strategy from the original game. Thus, the consequence is zero. As a result, $f_R = R_L \cdot R_C = 1 \cdot 0 = 0$.

To show how risk is affected by likelihood, consider the situation where the deceptive strategy σ_D is off-equilibrium. Then the consequence is non-zero. If the deceiver achieves the deception by changing only those payouts outside the deception game's equilibrium, then the players only receive payouts that correspond to the original game's payout values. In this case, the likelihood of discovery is zero. Thus, $f_R = R_L \cdot R_C = 0 \cdot R_C = 0$. On the other hand, if the payouts in the deceived game's equilibrium have been changed, then $R_L > 0$ and $f_R = R_L \cdot R_C > 0$.

A special case arises in when the true payouts contain one or more correlated equilibrium. A correlated equilibrium is a generalization of the Nash equilibrium whereby no player has incentive to deviate from the equilibrium assuming the other player selects a strategy based on the same public signal—in this case, the deceptive payouts. Thus, if the true payouts have a correlated equilibrium and σ_D plays the correlated equilibrium, then the counterdeception strategy is to play the correlated equilibrium, as well. By playing a deception that follows a favorable correlated equilibrium, the deceiver is no worse-off for being discovered since (assuming rationality of the mark) the deceptive payouts force the mark's strategy choice. In this way, it is possible for R_C to be less than zero. Thus, for games where the only equilibrium is the Nash equilibrium, R_C has a straightforward interpretation: it is the expected value loss when the mark plays the counterdeception strategy. In games with correlated equilibria other than Nash, a negative value for R_C must be interpreted as forcing cooperation between the players.

To prevent ambiguity in the results, each solution lists the risk objective function value as well as the likelihood R_L and consequence R_C.

IV. Methodology

This section outlines the process used to perform the case study in Section V. A small demonstration using the prisoner's dilemma provides a concrete example.

A. Process

Solving an instance of the DDP involves computing the best-known a set of non-dominated solutions, called PF_{known}. The Speed-constrained Multiobjective Particle Swarm Optimization (SMPSO) algorithm is used [8]. The SMPSO algorithm is discussed in Section VI.

The authors implemented the DDP and its associated objective functions using the MOEA Framework [9]—a Java library for multiobjective evolutionary algorithms. The equilibrium for each game is computed using the Enumeration of Extreme Equilibrium algorithm [10] and the linear programs are implemented using the JOptimizer version 3.4.0—a Java library for solving convex optimization problems. The true and deceived payouts are stored as $2 \times n \times m$ arrays. To account for finite representation of the solutions, the deceived payouts are rounded to the nearest $10^{-5\text{th}}$ decimal place before evaluating the objective functions.

Table II
SMPSO Settings

Setting	Value
Number of seeds	1000
Population size	100
Generations	250
Polynomial Mutation rate	1/100
Polynomial Mutation distribution index	20

SMPSO is executed 1,000 times using a unique random seed for each run. The settings for SMPSO are listed in Table II. Each run produces a set of nondominated solutions including their associated fitness values, the deceiver's deceptive strategy σ_D, and the mark's optimal counteredeception strategy σ_{CD}. All 1,000 solution sets are inserted into an intermediate set, which is then reduced to PF_{known} by removing all dominated solutions. Furthermore, any solution dominated by the trivial solution, $s_\emptyset$, is removed. The solutions in PF_{known} are then decoded back into payout matrix form and analyzed.

B. Example: Prisoner's Dilemma

An instance of the prisoner's dilemma illustrates the process. The prisoner's dilemma is a classic problem presented in introductory game theory courses; the payouts are shown in Figure IV-B.

It is often introduced as a conflict between two prisoners with regards to a plea bargain. The prisoners agreed that if caught, they would not confess the crime. However, they have been caught and separated so that neither knows what the other will choose. Each must decide to either cooperate with their fellow prisoner or to defect, i.e. to confess their crime to the police. If both cooperate, they face only a one-year sentence for a lesser crime. If they both defect, their confessions earn them a two years sentence. But if only one prisoner defects, his confession will be used against the other: The defecting prisoner is released immediately, and his fellow prisoner will face the maximum sentence of three-years. The game exhibits a pure strategy Nash equilibrium wherein both prisoners confess and receive two-year sentences. This equilibrium is underlined in Table **??**.

In this example, the cost of deception is assumed to be constant regardless of the payout value. Therefore, the L^1-

	Cooperate	Defect
Cooperate	(-1,-1)	(-3,0)
Defect	(0,-3)	<u>(-2,-2)</u>

Figure 1. The Prisoner's Dilemma. Payouts are shown as ordered pairs (i, j) where i is the row player's payout and j is the column player's payout. The pure strategy Nash equilibrium is for both players to defect.

	Cooperate	Defect
Cooperate	(-1,**0**)	(-3,0)
Defect	(0,**-1.999**)	(**-1**,-2)

Figure 2. DDP Example Solutions. Deceived payout matrix B for solution s_1 to the Prisoner's Dilemma. Deviations from the true payouts are shown in bold. The equilibrium of the EDG is underlined.

norm is used as the cost metric, i.e.

$$f_C(G) = \sum_{ijp} \left| a_{ij}^{(p)} - b_{ij}^{(p)} \right|.$$

After executing SMPSO and removing all dominated solutions from the solutions sets, PF_{known}, contains only a single solution s_1. The deceived payouts B defined by s_1 are shown in Figure IV-B; differences from the true payouts are shown in bold. s_1 exhibits a pure strategy Nash equilibrium at (Defect, Cooperate), underlined in Figure IV-B. The deceiver's best response is to Defect, i.e. $\sigma_D = (0, 1.0)$. The expected utility of the original game matrix A is -2 since both players receive a two-year sentence. In the environmental deception game $G = \langle A, B \rangle$, the deceiver's expected utility is 0 and the mark's expected utility is -3. Since the deceiver improves his expected utility by two years, the benefit of deception is $f_B(G) = 2$.

Since the L^1-norm is used as the cost metric, the total cost of s_1 is:

$$f_C(G) = \sum_{ijp} \left| a_{ij}^{(p)} - b_{ij}^{(p)} \right| = (1 + 1.001 + 1) = 3.001$$

Given the mark is deceived, the mark plays Cooperate while the deceiver plays Defect. As a result, the mark expects a payout of -1.999, but receives -3 instead. This result is certain since the equilibrium of G is a pure strategy. Therefore, the likelihood component of risk $R_L = 1.0$. On the other hand, the mark's counterdeception strategy is to defect, i.e. $\sigma_{CD} = (0, 1.0)$. If the mark plays σ_{CD}, then the game returns to the original equilibrium, (Defect, Defect). Since the deceiver plays exactly as he would in the deception-free game, the consequence component of risk $R_C = 0$. Thus, the risk for performing this deception is $f_R(s_1) = 1.0 \cdot 0 = 0$. As an interesting result, this means that in a single-shot prisoner's dilemma, there is no reason not to deceive your opponent.

V. Case Study

This section presents a case study to demonstrate the DDP in a more complex game. Specifically, the case study uses a 7×7 normal form game based on the output of an air-to-air combat simulator, the Missile Support Time (MST) game [6], shown in Figure 3.

The conflict involves two identical aircraft (AC) approaching each other at the edge of their engagement range. They each fire a missile at their opponent. The pilots can increase their probability of kill (P_K) by supporting their missile with updates from their AC's guidance system. The pilots can enhance their chances of survival by performing evasive

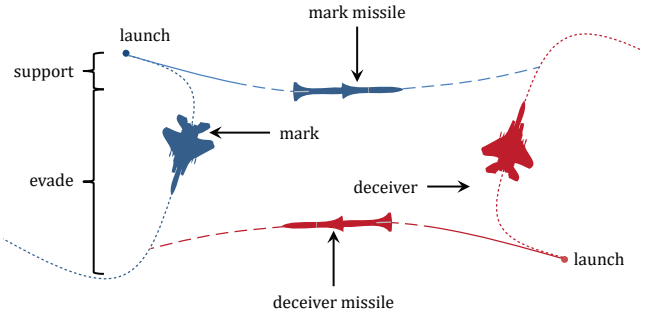

Figure 3. Missile Support Scenario. The aircraft must decide how long they will support their missile before evading the oncoming missile.

Table III
Parameter vectors for the deceiver (β_D) and mark (β_M)

Variable	Parameter	Deceiver $\boldsymbol{\beta_D}$	Mark $\boldsymbol{\beta_M}$
constant	β_0	-3.440	-3.529
x	β_1	0.289	-0.013
y	β_2	-0.131	0.300
x^2	β_3	-0.009	0.011
y^2	β_4	0.012	-0.009
xy	β_5	0.003	0.003

maneuvers. A pilot can either evade or support, but not both. To evade, the pilot must break his lock on the opponent and cease supporting his missile. So, the pilots must decide how long they will support before they evade. The longer they support, the more likely they are to kill their target, but this exposes them to a higher probability of being shot down themselves. Conversely, if they evade too soon, they are more likely to survive the engagement, but their missile will not likely hit their opponent.

This case study investigates the possibility of using deception to cause one pilot to either support too long, or evade too soon. The payouts for each pilot are defined by a regression model fitted to the output of a discrete-event air combat simulator [6]. The regression model is of the form

$$p(x, y; \beta) = \frac{\exp(q(x, y; \beta))}{1 + \exp(q(x, y; \beta))}$$

where x is the support time of the deceiver, y is the support time of the mark, and $q(x, y; \beta)$ is a quadratic function of decision variables x and y, i.e.

$$q(x, y; \beta) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy.$$

The resulting P_K values are based on the β parameter vectors for the deceiver (β_D) and mark (β_M) presented in Table III.

The payouts for the players are nearly symmetric, and the payouts for the deceiver are shown in Figure 4.

Thus, given that the deceiver supports for x seconds and the mark supports for y seconds, the deceiver's P_K is $p(x, y; \beta_{\mathbf{D}})$ and the mark's P_K is $p(x, y; \beta_{\mathbf{M}})$. Since the probability of survival equals $(1 - P_K)$, the probability of survival for the deceiver is $(1 - p(x, y; \beta_{\mathbf{M}}))$, and the probability of survival for the mark is $(1 - p(x, y; \beta_{\mathbf{D}}))$. A weighted sum defines

each player's payout as a trade-off between achieving a kill and surviving the engagement, i.e.

$$u_{deceiver}(x,y) = \omega_D p(x,y;\beta_{\mathbf{D}}) + (1-\omega_D)(1-p(x,y;\beta_{\mathbf{M}}))$$

and

$$u_{mark}(x,y) = \omega_M p(x,y;\beta_{\mathbf{M}}) + (1-\omega_M)(1-p(x,y;\beta_{\mathbf{D}}))$$

where $0 \leq \omega_M$ and $0 \leq \omega_D$. For this case study, ω_D and ω_M are set to 0.5.

The strategies x and y are discretized so the game can be represented in normal form. The resulting support times range from 0 seconds to 15 seconds at 2.5 second intervals, i.e. the pilots can support for either 0s, 2.5s, 5s, and so on. The maximum support time of 15 seconds is based on the maximum flight time of the simulated missiles. Under these conditions, the game exhibits a pure strategy Nash equilibrium when each player supports for 10 seconds before evading. As in the prisoner's dilemma example, this case study uses the L^1-norm as the cost function.

SMPSO found 71,415 solutions total during the 1,000 runs. After the dominated solutions are eliminated, PF_{known} contained seven non-dominated solutions. The solutions are enumerated in Table IV, sorted by benefit f_B. The first column lists the index of each solution. The next seven columns show the mark's equilibrium strategy based on the deceived payouts of the solution. Thus, the deceived payouts for s_1 exhibit a pure strategy Nash equilibrium where the mark supports for 0 seconds. The deception strategy for the deceiver is listed in the σ_D column, and the counterdeception strategy for the mark is listed in column σ_{CD}. For s_1, the best response to the mark is for the deceiver to support for 15 seconds; the best counterdeception is for the mark to support for 10 seconds. Equilibrium strategies throughout the table are rounded to the nearest 100$^{\text{th}}$ decimal place for presentation purposes. The objective function values for each solution are listed in the adjacent three columns with the most desirable values for each function shown in bold. Since risk is the product of consequence R_C and likelihood R_L, the final two columns display the consequence and likelihood components for each solution.

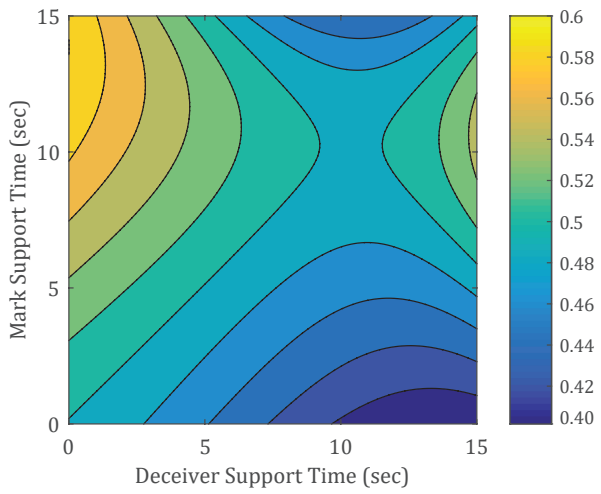

Figure 4. Deceiver Payouts for Missile Support Time game. The payouts for the mark and deceiver are nearly symmetric. The game exhibits a pure strategy Nash equilibrium wherein both players support for ten seconds.

The most beneficial solution, s_1, provided an increase in the deceiver's expected utility of 0.095. The payout manipulations cause the mark to support for 0 seconds (i.e. the fire and forget strategy). The deceiver can take advantage of this and support his missile for a full 15 seconds. So, the EDG results in the outcome at (15s, 0s).

s_1 is the highest cost solution: the absolute deviations from the true payouts summed to 3.263. s_1 modified 97 of the 98 ($= 7 \times 7 \times 2$) payout values in the game. The only payout that remained untouched was the payout for the mark at (15s, 0s), i.e. the expected outcome when the mark plays σ_B^{mark} and the deceiver plays σ_D. The solution increased the values of 58 payouts and decreased the values of 39 payouts. The solution applies a large portion of the cost (+0.149) to the payout in outcome (15s, 7.5s) for the mark. Since s_1 leaves the mark's payouts in (15s, 0s) untouched, it achieves zero likelihood of detection (R_L). However, if by some alternative means, the mark discovers the deception, the counterdeception strategy is to support for 10s; the consequence (R_C) in this case is 0.048 for the deceiver. Recall $f_R = R_L \cdot R_C$. Thus, $f_R(s_1) = 0 \cdot 0.05 = 0$.

The risk to the deceiver was zero in five of the seven cases. Four of the zero-risk strategies (s_4, s_5, s_6, s_7) achieved zero risk because the deception strategy σ_D is to support for 10 seconds. Since supporting for 10 seconds is the equilibrium strategy of the original game, the deceiver faces no consequence for playing these strategies. s_1 is the only other zero-risk solution. No solution had both zero consequence and zero likelihood of detection. Still, the observed risk values are lower than anticipated; it was surprising that so many zero- and low-risk solutions could be located in a game with 98 ($= 7 \times 7 \times 2$) free variables.

In all cases, σ_D implied the deceiver should support for at least 10s before evading, i.e. at least as long as in the deception-free game. Surprisingly, the mark's counterdeception strategy, σ_{CD}, is to revert to the true game's equilibrium. However, that σ_{CD} reverts back to the original equilibrium is not true for every possible deceptive strategy—only the non-dominated solutions found in this case study. If the deceiver supported for less than 10 seconds, the mark responds best with a longer, off-equilibrium support time. For instance, if the deceiver supports for 0 seconds, the counterdeception strategy is $\sigma_{CD} = 15s$.

The tendency to support longer when deceiving is influenced by the pilot preferences weights, ω_D and ω_M. However, the tendency to cause the mark to support for a shorter amount of time applies generally. If the deceiving pilot favors kills over survival, then the deception improves utility when the pilot can support longer. This is possible if the mark evades sooner. On the other hand, if the deceiving pilot favors survival over kills, then the deception improves utility when the mark supports for less time. Thus, in both cases, the tendency is to deceive the mark into evading sooner.

The lowest-cost solution was s_7 with $f_C(s_7) = 3.103$;

Table IV
SOLUTIONS TO THE MISSILE SUPPORT GAME.

Index	Fake Game Strategy Profile of Mark (σ_B^{mark})							σ_D	σ_{CD}	f_B	f_C	f_R	R_C	R_L
	0s	2.5s	5s	7.5s	10s	12.5s	15s							
s_1	1.00	-	-	-	-	-	-	15s	10s	**0.095**	3.263	-	0.048	-
s_2	1.00	-	-	-	-	-	-	15s	10s	**0.095**	3.159	0.048	0.048	1.000
s_3	0.70	-	0.30	-	-	-	-	12.5s	10s	0.076	3.152	0.008	0.008	1.000
s_4	-	-	-	-	-	-	1.00	10s	10s	0.056	3.151	-	-	1.000
s_5	-	-	0.67	-	-	-	0.33	10s	10s	0.039	3.148	-	-	0.328
s_6	-	-	0.67	-	-	0.01	0.32	10s	10s	0.039	3.133	-	-	1.000
s_7	-	-	0.64	0.30	-	0.06	-	10s	10s	0.024	**3.103**	-	-	1.000

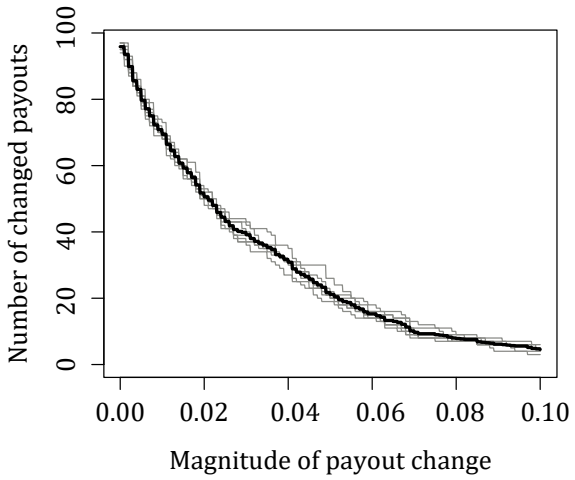

Figure 5. Number of changed payouts in DDP. The number of changed payouts are shown according to the magnitude of the payout's change. The light-grey lines are the traces for solutions $s_1, s_2, \ldots, s_7$. The black line shows the average of all ten solutions.

however, this solution also had the lowest benefit, as well. One interesting observation is that there is no incentive for the algorithm to reduce the number of altered payouts, only the total magnitude of the changes. It's not surprising that this algorithm produces solutions where very few payout values remained unchanged. Specifically, each solution changed at least 94 payout values. Plotting the number of changed payouts according to the magnitude of change shows that half of the payouts are modified by less than 0.02 utility points. This relationship is shown in Figure 5. The light-grey lines in Figure 5 are the traces of each solution, and the black line is the average for all seven solutions. The relationship between the magnitude of manipulation and the number of changed payouts indicates that the solutions produced by SMPSO may be noisy.

VI. RELATED WORKS

The benefit and cost objective functions resemble those found in [5], which presents an algorithm for computing desirable costly environmental deceptions in 3×3 normal form games. The risk objective function was based on the principles of risk assessment [7]. The effects of misrepresenting preferences was studied in two-player, 2×2 games [11] and in simple three-player voting games [12]. That the deception and counterdeception strategies are pure strategies coincide well with the results of [13] which shows that a pure strategy is often the best response to an opponent who misperceives the conflict. The environmental deception game is partially inspired by Hypergame theory [14]. It differs in that one player is assumed to know the true state of the conflict; Hypergame theory does not make this assumption in general.

The algorithm used to compute PF_{known}, SMPSO, is based on Particle Swarm Optimization (PSO), which is inspired by the flocking behavior of birds [15]. Although the no free lunch theorem implies that no one single optimization algorithm works best on all problems [16], PSOs have been proven to be effective tools for solving both continuous nonlinear and discrete binary optimization problems [15], [17–19]. In PSO algorithms, each solution is called a *particle*, and the population of solutions is called the *swarm*. Each particle i has a position $\vec{x_i}$ and velocity $\vec{v_i}$. The PSO updates the position at each generation t using the formula

$$\vec{x_i}(t) = \vec{x_i}(t-1) + \vec{v_i}(t)$$

where $\vec{v_i}(t)$ is given by

$$\vec{v_i} = w \cdot \vec{v_i}(t-1) + C_1 \cdot r_1 \cdot (\vec{x_{p_i}} - \vec{x_i}) + C_2 \cdot r_2 \cdot (\vec{x_{g_i}} - \vec{x_i}),$$

and $\vec{x_{p_i}}$ is the best solution according to the ith particle, $\vec{x_{g_i}}$ is the best solution known to the swarm, w is the inertia weight, and C_1 and C_2 dictate the influence of inertia and swarm particle attraction on the velocity of x_i.

SMPSO [8] constrains the velocity of each variable (in each particle) to prevent the particle velocities from "exploding" [20]. Constraining the velocity allows the algorithm to perform well even on difficult multiobjective problems (e.g. ZDT4 [21]). The pseudocode for SMPSO is given in Algorithm 1.

VII. CONCLUSION AND FUTURE WORK

This article presents a novel approach to fill a gap in the GT literature [2] by introducing a multiobjective optimization problem to compute efficient environmental deceptions. Section II introduces a GT model for environmental deception in situations where an opponent's perception of the conflict's payouts can be altered by the deceiver. The DDP is introduced in

Algorithm 1 SMPSO pseudocode [8]

```
initializeSwarm()
initilizeLeadersArchive()
generation = 0
while generation < maxGenerations do
    computeSpeed()
    updatePosition()
    polynomialMutation()
    fitnessEvaluation()
    updateLeadersArchive()
    updateParticlesMemory()
    generation++
end while
return LeadersArchive()
```

Section III as a multiobjective optimization problem to design efficient environmental deceptions in terms of its benefit, cost, and risk. Benefit is measured as the value difference between the non-deceptive and deceptive games. Cost is computed using a metric to measure the distance between the original payouts and the deceived payouts. Risk is the product of the likelihood of discovery and the potential consequence of being countered [7]. When the true game payouts exhibit correlated equilibrium other than the Nash equilibrium, the measure of risk is useful, but it can be harder to interpret. However, if risk likelihood and consequence are treated as separate objective functions, this difficulty is eliminated.

Section IV outlines the process used to compute solutions to the DDP during the case study in Section V. The solutions are obtained by executing a multiobjective evolutionary algorithm, SMPSO, 1000 times. Each time, a population of 100 particles is evolved over 250 generations. The solutions are added to an archive during execution. Afterward, the dominated solutions are removed from the archive. The remaining solutions constitute the known Pareto front.

The case study in Section V used the Missile Support Time (MST) game [6]. The known Pareto front for the MST consists of seven nontrivial solutions. All of these solutions are beneficial to the deceiver. Of the seven solutions, five exhibited zero risk. This indicates that SMPSO is able to effectively locate zero-risk solutions in this complex problem domain. Four solutions are zero-risk because there is no consequence for being discovered, i.e. the deception strategy coincides with the deceiver's strategy in the deception-free game. The fifth zero-risk solution has a zero-valued risk likelihood component. Thus, the mark's expected outcome in the deceived payout matrix coincides exactly with the expected outcome in the true payout matrix. Concerning the solutions' costs, it is possible that some cost observed in the resulting solutions is an artifact of the algorithm, but this observation requires further investigation.

This article provides several opportunities for future exploration by relaxing the two main assumptions made by the DDP. First, the DDP assumes that the deceiver's environmental deception is successful. An environmental deception is successful if two conditions are satisfied: 1.) the mark correctly perceives the deceived payouts provided by the deceiver, and 2.) the mark chooses to play the Nash equilibrium. This assumption could be relaxed to consider situations where one of these two conditions are not satisfied. If a relaxation allows the mark to incorrectly perceives the deceived payouts, researchers should characterize the degree to which the mark's perception deviate from the payouts provided by the deceiver. If a relaxation allows the mark to select a strategy other than the Nash equilibrium, it is necessary to define a reasonable alternative.

The DDP also assumes that the deceiver accurately perceives the true payouts of the conflict. This assumption could be relaxed to explore the effects of misperception on the part of the deceiver. A more general form of the environmental deception game would need to be defined. A possible solution would be constructed of three matrices: One for the true payouts, one for the deceiver's perception of the conflict, and one for the deceived payouts presented to the mark. The strategies of the mark and deceiver could be selected as they are in the version of the environmental deception game; however, the expected utility for each player would be computed according to the true payouts (vice the deceiver's perception of the true payouts). In this way, the environmental deception game defined above becomes a special case of this more general version. If the DDP is used as a tool for post hoc analysis, this approach might be able to address the issue of misperception on the part of the deceiver. In an ad hoc situation, this approach could be used to perform sensitivity analysis on the solutions.

Finally, all solutions to the case study manipulate at least 94 out of 98 payouts, but most of the payouts change by less than 0.02. This case study used the L^1-norm as a cost metric. Researchers should evaluate the impact of the cost metric on the number of changed payouts, e.g. by comparing solutions with an L^1-norm cost to those that use L^p-norm cost with $0 < p < 1$. However, the cost metric should be carefully selected based on the problem domain. Therefore, future work should evaluate alternative techniques for reducing the total number of changed payouts, as well. One simple technique is to introduce an additional objective function that counts the number of changed payouts; however, the introduction of a fourth dimension in objective space will likely increase the number of non-dominated solutions considerably. Alternatively, future work can evaluate the performance of other optimization algorithms. Another technique is to adjust the cost (or benefit) objective function according to the number of changed payouts. One final approach, borrowed from the Ridge Regression shrinkage method [22], penalizes cost additively based on the magnitude of changed payouts. For instance, suppose the maximum allowable deviation is $\pm\delta_{\max}$. Then the cost can be penalized by adding the shrinkage penalty, $\lambda \sum_{ij}(1 - \delta_{ij}/\delta_{\max})^2$, where δ_{ij} is the absolute deviation in row i and column j, and lambda is a tuning parameter. The shrinkage penalty is inversely proportional to the size of the deviation; it effectively shrinks the deviations toward zero. The tuning parameter λ controls the relative impact of the penalty on the resulting solutions. One final approach is to reduce solution noise mid- or post- execution. A greedy approach could reduce the smallest deviations first

unless and until a change in the equilibrium is observed before moving to subsequently larger deviations.

REFERENCES

[1] R. B. Myerson, *Game Theory: Analysis of Conflict*. Harvard University Press, 1991.

[2] A. L. Davis, "Deception in game theory: A survey and multiobjective model," Master's thesis, Air Force Institute of Technology, 2016.

[3] B. J. Borghetti, "The environment value of an opponent model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 623–633, 2010.

[4] US Dept. of Defense, "Joint doctrine for military deception," 2012.

[5] H. Poston, "Generation of strategies for environmental deception in two-player normal-form games," Thesis, Air Force Institute of Technology, 2015.

[6] J. Poropudas and K. Virtanen, "Game-theoretic validation and analysis of air combat simulation models," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 5, pp. 1057–1070, Sep. 2010.

[7] E. Verzuh, *The fast forward MBA in project management*, 3rd ed. Hoboken, NJ: John Wiley & Sons Inc, 2008.

[8] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "SMPSO: A new pso-based metaheuristic for multi-objective optimization," in *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'2009)*. Nashville, TN, USA: IEEE Press, Mar. 2009, pp. 66–73.

[9] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization," *Evolutionary Computation*, vol. 20, no. 3, pp. 423–452, Sep. 2012.

[10] G. Rosenberg, "Enumeration of all extreme equilibria of bimatrix games with integer pivoting and improved degeneracy check," *CDAM Research Report LSE-CDAM-2005-18, London School of Economics*, vol. 42, no. 2010, 2005.

[11] S. J. Brams, "Deception in 2x2 games," *J. Peace Sci.*, vol. 2, pp. 171–203, 1977.

[12] S. J. Brams and F. C. Zagare, "Deception in simple voting games," *Soc. Sci. Res.*, vol. 6, pp. 257–272, 1977.

[13] R. R. Vane, III, "Using Hypergames to Select Plans in Competitive Environments," Doctoral Dissertation, George Mason University, 2000.

[14] P. Bennett, "Toward a theory of hypergames," *Omega*, vol. 5, no. 6, pp. 749–751, 1977.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov. 1995, pp. 1942–1948 vol.4.

[16] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary COmputation1*, vol. 1, no. 1, pp. 67–82, 1997.

[17] J. Kennedy and R. Eberhart, *Swarm Intelligence*. San Fransisco, California: Morgan Kaufmann Publishers, 2001.

[18] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, vol. 5, Oct. 1997, pp. 4104–4108 vol.5.

[19] A. P. Engelbrecht, *Computational Intelligence: An Introduction*, 2nd ed. Hoboken, NJ: John Wiley & Sons, Ltd, 2003.

[20] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba, "Multi-objective particle swarm optimizers: An experimental comparison," in *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux, Eds. Nantes, France: Springer. Lecture Notes in Computer Science Vol. 5467, Apr. 2009, pp. 495–509.

[21] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000.

[22] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.