

# Representation and Detection of Perverse Incentives

Paul Kuliniewicz, Jeff Moser, David Zage

{kuliniew, moserjd, zage}@purdue.edu

## Overview

A perverse incentive is a reward that brings about the opposite effect from what it was intended to produce. For example, requiring users to have strong passwords might cause them to store their passwords on a piece of paper on the side of their monitor, thus making overall security worse.

In our research, we have investigated how to model perverse incentives so that they can be detected using computer algorithms.

## Problem Statement

Our goal is to define and create a computer model that will ultimately be used to detect perverse incentives that may exist in an arbitrary situation. A successful modeling tool will take a representation of a situation as input and identify under what conditions, if any, a perverse incentive can occur.

For a perverse incentive to exist, there must be at least two people involved in the situation: an *agent* who is capable of making one or more decisions, and a *client* on whose behalf the agent is supposed to act. The actions the agent takes produce *payoffs* to the agent and the client; these payoffs may be different from each other and can be negative. A good model must identify all relevant factors, including both the agent's choices and external events, that affect one or both of the payoffs.

## Experiment

We have experimented with several different models to represent perverse incentives:

- Fuzzy Cognitive Maps — The situations are represented as a directed graph in which the vertices represent factors of the situation and the edges indicate the correlations between factors. A simulation is run until every vertex reaches an equilibrium. These simulations are run multiple times using various initial conditions. A perverse incentive exists when the factor that represents the client's payoff reaches a much lower equilibrium (often reaching zero) than its initial value while the factor representing the agent's payoff increases.
- Decision Trees — Different branches in the tree represent mutually exclusive choices the agent can make or the probability that a "random" event (outside the agent's control) occurs. Each leaf of the tree identifies the payoffs to the agent and the client if that node is reached. For each possible set of choices the agent can make, the expected payoffs to the agent and the client are computed. A perverse incentive exists if the choices that maximize the agent's payoff do not also maximize the client's payoff.

- **Neural Networks** — A set of interconnected nodes representing the concepts of a situation and their “IF/THEN” associations is constructed. The nodes are given an initial value and a series of passes is run through the network, ultimately resulting in all nodes reaching equilibrium. This procedure is done multiple times using different initial node conditions. A perverse incentive exists when the node that represents the client’s payoff reaches a much lower equilibrium (often reaching zero) than its initial value while the node representing the agent’s payoff increases.
- **Closed Form Analysis** — A pair of functions represent the payoffs received by the agent and the client in terms of one or more variables that the agent can control. A perverse incentive exists if the solution that maximizes the agent’s payoff function does not also maximize the client’s payoff function.

We have used the above models to represent several canonical perverse incentive situations:

- **Mutual Fund Broker Investment Strategy** — A broker will invest in a market that is likely to go down out of fear that he or she will be fired in the unlikely event that he or she does not invest in the market and it goes up.
- **Investment in Security Technologies** — An information technology officer will choose not to invest in computer security even though it will pay off in the long term because the investment will be seen as an unnecessary expenditure in the short term.
- **Fixing Bugs** — Paying a bonus to software developers for every bug they fix encourages them to intentionally introduce new bugs in the software.
- **Fire Department Funding** — Funding a fire department based on the number of calls dispatched has the unintended effect of discouraging fire prevention programs.

## Results

We were unable to find a single method that works well for all of the perverse incentives that we considered. Each modeling technique has its own strengths and weaknesses. For example, Fuzzy Cognitive Maps and Neural Networks were easy to construct for simple scenarios, but they do not easily scale to more complex situations. Decisions Trees and Closed Form Analysis provide a more concrete analysis, but are more difficult to construct because they require detailed information about factors that are difficult to estimate. Even when our algorithmic techniques can be used to provide a solution, those algorithms run in exponential time.

## Project Continuation

Research was conducted under the guidance of Professor Mikhail Atallah. Tasks were divided informally amongst the group, with individual group members focusing on a specific type of model.

Future research will be needed to discover a model that represents perverse incentives in a more generalized manner. Another possible route would be to reduce the computational complexity of our existing algorithms.