Dynamics of Adversarial Co-evolution in Tax Non-Compliance Detection

Jacob Rosen
Massachusetts Institute of Technology, CSAIL
jbrosen@alum.mit.edu

Erik Hemberg
Massachusetts Institute of Technology, CSAIL
hembergerik@csail.mit.edu

Una-May O'Reilly
Massachusetts Institute of Technology, CSAIL
unamay@csail.mit.edu

ABSTRACT

In cyber security there are adversarial relationships, e.g. spammers versus spam-filter. A similar arms-race exists for tax non-compliance. The ability of individuals to exploit gaps in the "attack surface" of tax regulation reduces trust in governmental equity, drains money from public goods and reduces productivity. This relationship between utility-maximizing tax evaders and an increasingly resource constrained auditing system is codependent. We present a framework that allows a modest but representative area of US partnership taxation activity to be explored as an adversarial coevolutionary relationship. In this setting tax evaders try to minimize tax by exploiting loopholes in tax code inconsistencies and gaps. Reciprocally, auditors, with constrained resources, try to pinpoint strategies that shelter evasion. The framework relies upon *a)* a representation of the relevant partnership tax law *b)* a simulation of the auditing, tax calculation and compliance checking processes and *c)* co-optimizing taxpayer and auditor behavior via dual genetic algorithms that in turn model their coevolutionary dynamics.

CCS Concepts

•**Social and professional topics** → *Taxation;*

Keywords

Tax Evasion, Genetic Algorithm, Coevolution

1. INTRODUCTION

In *cyber security* the notion of adversaries is central. The defender aims to reduce its attack surface to thwart an adversary. Examples include spammers against spam-filters, explored in [5] as a game between classifier and adversary over the attack surface of email. Other examples is the arms race in cyber attacks, e.g. some "buffer overflow" changed to file format exploits after XP Service Pack 2 [23], and after return instructions detection in Return-Oriented-Programming on x86 changed attackers to use JMP instructions instead [3].

In tax evasion there also exists an adversarial arms race, the attacker is the tax evader, the defender is the auditor and the attack surface is the tax regulations. Financial and legal enterprises search of ambiguities in the tax code in order to discover abusive tax shelters. While tax auditors have historical examples of tax schemes to help guide examination, tax shelter promoters often adapt their strategies as existing schemes are uncovered and when changes are made to the existing tax regulations. One example is the so called BOSS tax shelter (Bond and Options Sales Strategies) that was widely promoted yet was ultimately disallowed. While audit changes were implemented to detect BOSS they were not able to detect the newly emerged variant "Son of BOSS" [25].

The auditor has limited resources and knows that there will always be exploits. Thus, for the auditor in the tax evasion arms race it becomes a matter of risk management when altering auditing and regulations, i.e. how do changes to auditing effect the behavior of the evader. A goal is to replicate, with an abstraction, the oscillatory dynamics between tax evader and auditors that occurs in the context of tax regulation. Our tax modeling framework is named Simulating Tax Evasion And Law Through Heuristics, or STEALTH, see Figure 1. To date STEALTH is focused on demonstrating the dynamics around specific United States' tax law pertaining to partnerships.

Partnership taxation has been chosen due to the evidence of adversarial relationship between the United States Internal Revenue Service (IRS) and financial advisers with an intricate knowledge of Subchapter K, the section of the Internal Revenue Code (IRC) pertaining to partnerships. Furthermore, partnerships can own an interest in another partnership, creating highly complex tiers of partnerships and taxpayers. Many partnerships in the US have millions of unitholders, and can be composed of dozens of tiers [10, 11]. Not only does the IRS lack the resources to properly identify potentially abusive behavior that occurs within these structures, but the many of the rules in Subchapter K are ambiguous when it comes to more complex financial manifestations [21, 10].

The technical challenges to modeling partnership taxation are imposing because of the tax code's complexity, the behaviors available to tax evaders and the simultaneous co-adaptive behaviors of both auditors and tax evaders. We have decomposed the challenges by creating two modules as represented in Figure 1. First we need to develop an abstraction of the relevant partnership tax law, which will allow us

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908961.2931680

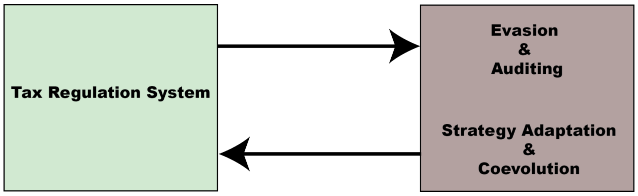


Figure 1: STEALTH modules overview

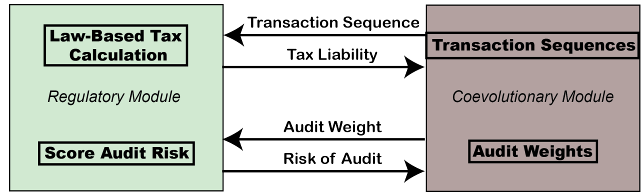


Figure 2: Tax and audit risk calculation take place in the regulatory system on the left, which is assigned to transaction sequence and audit weight individuals in the dynamic right side

to compute both tax liability and likelihood of being audited. Second, we have to develop a model of taxpayer and auditor co-adaptive behavior. In this contribution we focus on how we addressed the second challenge while we provide, brief descriptions of our solution to the first challenge that allow our readers to understand the framework and our approach in their entirety.

We proceed as follows: A brief survey of previous work in the space of quantitative tax evasion prediction and co-evolution is provided in Section 2. Next, Section 3 describes the two modules hinted at in the introduction, followed by some preliminary experimentation in Section 4. We finish with conclusions and future work in Section 5.

2. BACKGROUND

This project draws on models of predicting tax evasion and coevolutionary dynamics. Tax evasion or non-compliance, in the academic realm, has historically been considered a utility maximizing decision under uncertainty based on *a)* the amount of tax that one is able to illegally evade and *b)* the probability and consequences of getting caught doing so [1]. The research ranges from the effects of heterogeneity of individual preferences through agent based models [2, 19], and recently to machine learning algorithms such as neural networks, logical regression and support vector machine [7]. All past approaches attempt to answer the question: *given some method for evading taxes that is definitely illegal but stochastically detectable*, what macro-economic policies (tax rates, education, etc.) will reduce the incidence of illegal tax evasion?

Each time the IRS changes the tax code the tax evaders react by finding new loopholes, similar to foxes and hares. The system dynamics reflect a constant transition of complementary adjustments, with each predator/prey seeking advantage over the predator/prey under adjustment, studied in co-evolutionary algorithms, [13]. Co-evolutionary algorithms [6, 8] have fitness evaluations based on interactions between multiple individuals. Whereas conventional Genetic Algorithms compute objective fitness, co-evolutionary algorithms compute a subjective fitness. Competition (or cooperation) arises from these interactions in or between populations. An individual's ranking in a population can change depending on other individuals. Thus, the fitness is subjective.

Some applications of coevolutionary algorithms are in botnet detection system analysis [12], in the coevolutionary agent-based network defense lightweight event system [22] and to use evolutionary algorithms to explore the arms race of malware evolution and exploit code for vulnerability testing of anomaly detectors [14]. In moving target techniques coevolution is used to randomize system components to reduce the likelihood of a successful attack and the lifetime of an attack [17, 24, 4]. Finally, coevolution and cyber security focused software testing explore self-regenerative architecture automatically identify software vulnerabilities and create adaptations that shield or repair those vulnerabilities before attackers can exploit them [15, 16]. Next we describe how a coevolutionary algorithm is used to detect tax non-compliance.

3. METHODOLOGY

As mentioned in Section 1, STEALTH is composed of two modules: one that expresses the regulatory framework pertaining to US partnership taxation and another that simulates the adversarial dynamics that occur within such a framework. The modules are shown in Figure 2. The regulatory system contains the legal logic to calculate tax liability and assign audit risk, then passes those values to the individuals from which they were derived in the coevolutionary module. That module then contains the algorithms which replicate the desired dynamics. We will focus here more on the coevolutionary module. But next background will be provided on the regulatory system to give context to the method.

3.1 Tax Regulatory System

Crucial to the analysis of abusive tax behavior is the ability to compute the tax consequences of financial activity. Thus, many aspects of Subchapter K of the IRC were hand coded from Congressional legalese into a succinct representation. This representation needs the ability to process an arbitrary set of financial behavior for tax liability. We used, e.g.

An asset is a tuple (b, β, τ) consisting of *(1)* Adjusted Basis: A scalar $b \in \mathbb{R}^+$ *(2)* Book Value: A scalar $\beta \in \mathbb{R}^+$ *(3)* Type: A positive integer τ that whether the asset is category 0 (cash), category 1 (ordinary) and category 2 (capital).

Once we are able to calculate the tax liability resulting from a single transaction, we must determine what computer readable form a potentially abusive tax strategy takes, and how also to represent a given auditing policy. The following sections describe, respectively, our quantitative representations of tax evaders and auditors.

3.1.1 Tax Network and Transactions

We must first define the environment of interest, namely the initial conditions of a regulated unit. In the context of taxation, this amounts to a set of entities, each of which have a "portfolio" of assets, the entirety of which we refer to as the *tax network*. In abstract terms, we define the state of the network as some $\gamma \in \Gamma$, where $\gamma = \{\mathbf{e}, \mathbf{a}, d\}$. The tuple γ is composed of a set of entities $\mathbf{e} = \{e_i\}_{i=0}^{k_1}$, a set of assets $\mathbf{a} = \{a_i\}_{i=0}^{k_2}$ where $k_1, k_2 \in \mathbb{Z}_+$. The operator d determines the owner of each asset, i.e $d : A \mapsto E$, where A is the space of assets and E is the space of entities.

Transactions can then be thought of a specific type of transition from one tax network state γ_n to another γ_{n+1}. A transaction is thus described as some $t = \{e_f, e_t, a_f, a_t\}$, where $e_f, e_t \in E$ are two entities and $a_f, a_t \in A$ are two assets that are being exchanged between the two entities. Finally we can define a *transaction sequence* as $\mathbf{t} = \{t_i\}_{i=0}^{k}$, where $k \in \mathbb{Z}_+$ is the number of transactions.

3.1.2 Audit Score Sheets

Like a regulated unit is represented as a transaction sequence, there must be some abstraction of a regulator that can be fed into the simulation to determine the likelihood of conducting an audit. We thus conceive of a regulator as a certain auditing policy, which is represented as a list of events *observable* within a transaction sequence with numerical weights associated with each type of event. When a particular observable event is fed into the simulator, the overall audit likelihood is incremented by its associated weight. The list of observable events with weights is referred to as the *audit score sheet.*

A key trait of the audit score sheet is that it not only records the occurrence of each type of observable event, but it can also optionally record every possible combination of the behaviors. Thus, if there are m separate types of observable events, then an audit score sheet would be of length $2^m - 1$, representing the entire combinatoric space. This allows the detecting of more complex patterns. For a clarifying example, consider the following passage from the Internal Revenue Code §743(a).

> The basis of partnership property shall not be adjusted as the result of (1) a transfer of an interest in a partnership by sale or exchange or on the death of a partner unless (2) the election provided by §754 (relating to optional adjustment to a basis of partnership property) is in effect with respect to such partnership of (3) unless the partnership has a substantial built-in loss immediately after such transfer.

Each number with parentheses signifies an *observable* event. Namely, (1) The sale of a partnership interest in exchange for a *taxable* asset. (2) The partnership whose shares are being transferred has not made a §754 election. (3) The seller's basis in respect to the non-cash assets owned by the partnership exceeds their FMV by more than $250,000. An audit score sheet that encapsulated only the three observable events listed in the passage would look as follows.

Observable	**Weights(w)**	**Freq.(f)**
Partnership Interest Sale (1)	w_1	f_1
No §754 Election (2)	w_2	f_2
Substantial built-in Loss (3)	w_3	f_3
$1 \cup 2$	$w_{1\cup 2}$	$f_{1\cup 2}$
$1 \cup 3$	$w_{1\cup 3}$	$f_{1\cup 3}$
$2 \cup 3$	$w_{2\cup 3}$	$f_{2\cup 3}$
$1 \cup 2 \cup 3$	$w_{1\cup 2\cup 3}$	$f_{1\cup 2\cup 3}$

Table 1: Each row has three columns with 1) the type of observable corresponding to the three characterized observables 2) the associated audit weight and 3) the number of times it occurs in a list of transactions

While the formulation of the audit score sheet is complex, the calculation of the audit score from it is relatively simple. Suppose that there are m specific types of events that are observable, represented by $\{b_i\}_{i=0}^{n}$, where $n = 2^m - 1$. Associated with each type of event are the weights $\{\alpha_i\}_{i=0}^{n}, \alpha \in \mathbb{R}_+$ and the frequency that the event occurs within a network of transactions $\{f_i\}_{i=0}^{n}, f_i \in \mathbb{Z}_+$. We can then write the audit score, s corresponding to the audit score sheet and network of transactions as

$$s = \sum_{i=0}^{n} \alpha_i f_i \text{ where } \sum_{i=0}^{n} \alpha_i = 1$$

3.1.3 Summary

In total, the simulation is defined as a function $\mathbf{F} : \mathbf{T} \times \Gamma \times \Phi \mapsto \mathbb{R}^{\zeta} \times \mathbb{R}_+$, where $\mathbf{T}$ is the space of all transaction sequences, Γ is the space of all initial tax networks, Φ is all audit score sheets, $\mathbb{R}^{\zeta}$ represents ζ measures of taxable income and $\mathbb{R}_+$ is the audit likelihood.

3.2 Coevolutionary Dynamics

Here we describe the module which directs the appropriate adversarial dynamics. It is supported by the previously described regulatory framework, along with the quantitative representations of both tax-minimizing strategies (transaction sequences) and auditing policies (audit score sheets). This module has two subtasks. First, a *fitness* must be assigned to both transaction sequences and audit score sheets based solely on the measures of taxable income and audit score, as calculated in Section 3.1, which reflects a concrete measure of "effectiveness". Second, we must co-adapt the two opposing populations, in terms of the previously described fitness score, by searching over its highly non-linear behavioral space.

3.2.1 Population Representation

Even though individual solutions, i.e transaction sequences and audit score sheets, are both evaluated in the regulatory system, we must establish a method to express and explore the spaces of all possible transaction sequences and audit weights. Grammatical Evolution (GE) offers an elegant solution. GE is a version of the genetic algorithm with a variable length integer representation and an indirect mapping using a grammar[18], which allows us to generate complex and valid phenotypes, which in our case take the form of transaction sequences.

As shown in Figure 3, the grammar is composed of a single start symbol, terminal symbols and non-terminal symbols, as indicated by the boxes in the figure, with the start symbol at the top and terminal symbols at the bottom of each branch. Integers are fed into the top and the direction of the path is determined by taking the modulo of the current integer, at which point the next integer is selected. The process is complete when the sentence comprises only terminal symbols.

Generating transaction sequences from an integer vector, while the most complicated, is not the only possible mapping from an integer vector. Other operations can be performed by merely removing a select few integers from the vector, which is a method that we use to generate initial tax network configurations. For example, if we would like to determine some number of additional partnerships in the initial configuration between 0 and k, we simply remove the first integer and take its modulo in respect to k, then process the rest of the vector through the grammar.

Mapping an integer sequence to an audit score sheet, on the other hand, is a simple process due to the numerical qualities of the audit weights. For an audit score sheet of

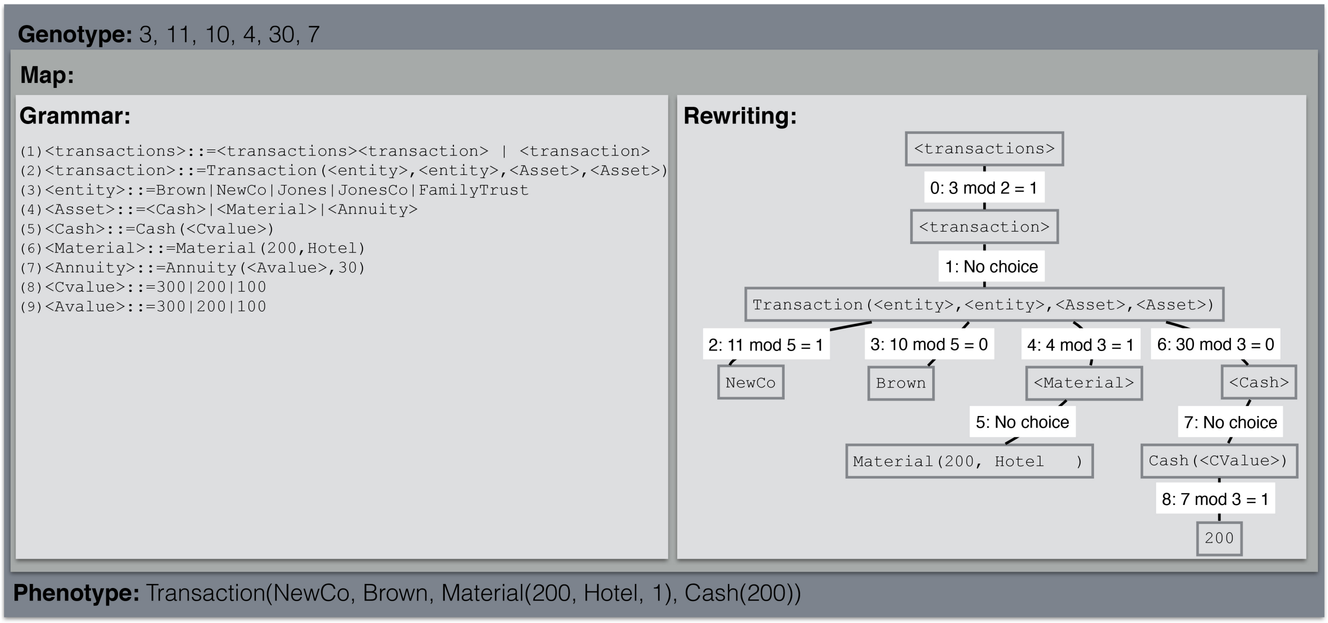


Figure 3: Example of how GE rewrites a list of integers (Genotype) into a list of transactions (Phenotype) with a BNF grammar. The mod value is based on the number of production choices for the non-terminal symbol.

length m we simply take an integer vector of length m and divide each integer in the vector by the sum of all of the elements. This creates m positive real numbers that sum to one.

3.2.2 Objective Functions

Crucial to our analysis is the question of what makes an effective tax strategy, taking into account both tax liability and likelihood of being detected by various auditing policies. Similarly, what constitutes an effective auditing policy? Neither question is trivial, nor can they be generalized to encompass every use case. That being said, a generic heuristic for determining effectiveness in a specific scenario can be applied to help formulate a good objective function.

By *objective function*, we mean some mapping between the numerical traits associated with a transaction sequence or audit score sheet, and some measure of effectiveness or desirability. Section 3.1 gives us the tools to calculate taxable income for all ζ of the entities in the simulation, and an audit score. Given these two numerical constructs, we are tasked with formulating objective functions for both transaction sequences and audit score sheets, respectively h_e and h_s, both defined as a mapping from measure of taxable income and audit likelihood, to a single real number.

An effective transaction sequence, from the perspective of an adviser or taxpayer, is one that results in a low level of taxable income, while attracting a low likelihood of being audited, both outputs of the simulation described in Section 3.1. Thus, one can conceive of a transaction sequence's effectiveness as shown in Figure 4, with taxable income on the x-axis and audit likelihood on the y-axis. A highly effective transaction sequence would be in the lower left corner, incurring relatively low levels of tax liability and attracting little attention. Conversely, a transaction sequence that produces low levels of tax liability but a *high* likelihood of being audited, as shown by the top left corner, would be extremely undesirable. While there is nothing inherently wrong about low levels of taxable income, we evaluate transaction sequences that all accomplish relatively similar economic goals. Thus any lower variations in taxable income can be indicative of, at the very least, tax implications that were never intended by policy-makers.

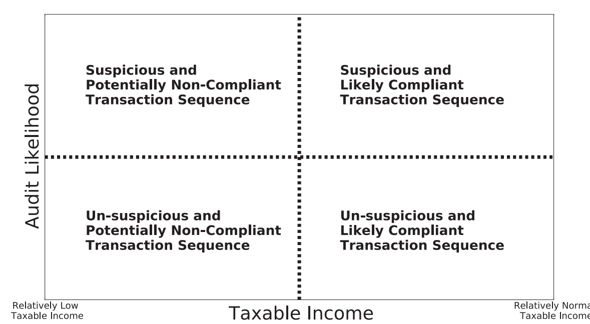


Figure 4: Traits of transaction sequences as a function of their taxable income and the audit likelihood they generate in respect to a certain audit score sheet

Auditing policies face a different heuristic for calculating effectiveness, due to their resource constraints. That is, while transaction sequences are only concerned about a one-off evaluation against an audit score sheet, auditing policies must take into account the amount of resources that it takes to audit. Figure 5 demonstrates an example of effective auditing policies as distinct from incorrect and wasteful ones. A good auditing policy, indicated by the solid line, applies a low audit likelihood to transactions sequences generated relatively normal levels of taxable income and a high likelihood to similar, low taxable income transaction sequences. Bad auditing policies are the exact opposite, assigning high audit likelihood to transaction sequences with normal levels of tax liability and ignoring those resulting in low taxable income.

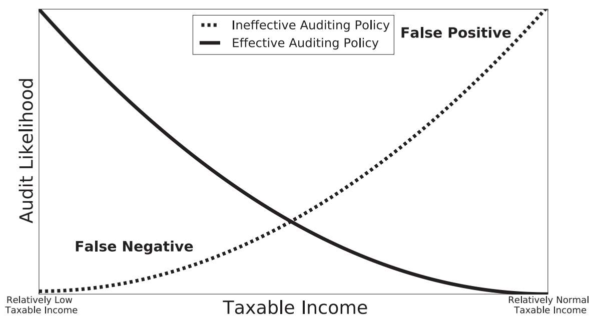


Figure 5: An example of a "good" audit score sheet and a "bad" audit score sheet, shown by the audit likelihood that they would generate in respect to different types of taxable income

These heuristics and associated plots are merely a means to formulate proper objective functions for both a tax-minimizing strategy and an audit plan, given a transaction sequence, initial tax network and audit score sheet.

3.2.3 Adaptation – Coevolutionary Genetic Algorithm

The second subtask is to specify a means by which a large and highly non-linear space of transaction sequence-audit score sheet pairs can be co-adapted. Once objectives establish a notion of effectiveness, the evolutionary algorithm determines *a)* which transaction sequences can minimize tax liability while circumventing an audit and *b)* which auditing policies assign high audit likelihood to relatively low tax liability schemes while ignoring non-suspicious behavior. It thus attempts to anticipate new forms of potentially abusive tax behavior as well as desirable, or at least likely, regulator response to it.

Upon establishing these mappings, any non-linear search can be performed on the space. Because of the predator-prey relationship between non-compliance schemes and auditing policy, we chose to use a *co-evolutionary genetic algorithm*. Specifically, we evolve with a so-called *one-to-many interaction scheme*, in which the two populations evolve in parallel. Each individual in the two populations are evaluated against a subset of the opposing population, which can be chosen by a number of different decision heuristics.

Our coevolutionary algorithm:

1) **initializes** both populations

2) **evaluates** each individual against a subset of of the other population to determine their objective score

3) **selects** the best individuals in each population

4) creates new populations by **crossing over**(combining) the chosen individuals

5) introduces slight **mutation** into that new population

6) **repeats** steps $2-5$ over some *generations* until there is some halting condition.

Specifically, every generation, each individual in the transaction sequence population selects a subset of the audit score sheet of the population to evaluate against. After all sequences are evaluated, the process is repeated with the opposite population: each audit score sheet chooses a subset of the transaction sequence population to evaluate, see [20] for more details.

Recall from Section 3.1 that the regulatory system is a function $\mathbf{F} : \mathbf{T} \times \Gamma \times \Psi \mapsto \mathbb{R}^{\zeta} \times \mathbb{R}_{+}$ that takes as input a sequence of transactions, an initial network state and auditing observables, and generates the taxable income for all relevant entities and audit score. In other words, for any $\mathbf{t} \in \mathbf{T}$ and $\gamma_0 \in \Gamma$ generated from the same vector of integers $\mathbf{x}$ and accompanying auditing observables $\psi \in \Psi$, $\mathbf{F}(\mathbf{t}, \gamma_0, \psi) = (\boldsymbol{\ell}, s)$, where $\boldsymbol{\ell}$ is a vector of real numbers of length ζ that represents taxable income for all entities and s is the audit score.

The function $\mathbf{F}$ can be broken up into a network of transition functions that has the same length as the number of transactions in the transaction set contained within the function call (k). Each transition function generates a new network state and an audit score. So for all $i \in [0, k]$, $F_i(t_i, \gamma_i, \psi) = (\gamma_{i+1}, s_i) \quad \text{where} \quad s = s_k$

3.2.4 Summary

Recall from Section 3.2.2 that the objective functions for transaction sequences and audit score sheets are, respectively, h_e and h_s, both maps from $\mathbb{R}^{\zeta} \times \mathbb{R}_{+}$. Additionally, define $\Xi_t : \mathbb{Z}_{+}^{n} \mapsto \mathbf{T} \times \Gamma$ and $\Xi_a : \mathbb{Z}_{+}^{m} \mapsto \mathbb{R}_{+}^{m}$ as, respectively, the transaction sequence and audit score sheet mapping functions described in Section 3.2.1 Now it is possible to fully define the maximizing objectives of networks of transactions as

$$\arg\max_{\mathbf{x}^* \in X} \left[h_e\left(\mathbf{F}\left(\Xi_t(\mathbf{x}^*), \Xi_a(\mathbf{y})\right)\right)\right] = \arg\max_{\mathbf{t}^* \in \mathbf{T}, \gamma_0^* \in \Gamma} \left[h_e\left(\mathbf{F}\left(\mathbf{t}^*, \gamma_0^*, \psi\right)\right)\right]$$

over all $\mathbf{y} \in B(\hat{\mathbf{y}}, r_1)$ for some $\hat{\mathbf{y}} \in \mathbb{Z}_{+}^{m}$, where $B(\hat{\mathbf{y}}, r_1)$ is a *ball* of radius $r_1 \in \mathbb{R}_{+}$ around $\hat{\mathbf{y}}$. This represents the fact that the goal of the GA is to find local maxima around some subset of auditing behavior, rather than attempting to search the entire Φ space. Conversely, the objective for the auditing behaviors is to maximize the *positive* h_a function, the opposite of the objective for the transactions, i.e. the goal is

$$\arg\max_{\mathbf{y}^* \in \mathbb{Z}_{+}^{m}} \left[h_a\left(\mathbf{F}\left(\Xi_t(\mathbf{x}), \Xi_a(\mathbf{y}^*)\right)\right)\right] = \arg\max_{\psi^* \in \Psi} \left[h_a\left(\mathbf{F}\left(\mathbf{t}, \gamma_0, \psi^*\right)\right)\right]$$

over all $\mathbf{x} \in B(\hat{\mathbf{x}}, r_2)$ for some $\hat{\mathbf{x}} \in \hat{X}$, where $B(\hat{\mathbf{x}}, r_2)$ is a *ball* of radius $r_2 \in \mathbb{R}_{+}$ around $\hat{\mathbf{x}}$. Similar to the previous objective function, this represents the fact that the EA only searches for local maxima around a subset of all transaction sets and initial model states.

4. RESULTS

Ideally, we would like to be able to show that, with the proper specifications, dynamics between dominant tax strategies and dominant auditing policies can be replicated in a computational setting. That is, we see audit score sheets changing to assign high audit likelihood to certain transaction sequence behavior that produces relatively low taxable income. Then in turn, we see the population of transaction sequences changing to favor transaction sequences that continue to produce low levels of taxable income, but using techniques that are not deemed suspicious by the dominant audit score sheets in the opposing population.

4.1 iBOB Description

We demonstrate STEALTH using a particular known tax evasion scheme called Installment Bogus Optional Basis (iBOB). In iBOB, a taxpayer arranges a network of transactions designed to reduce his tax liability upon the eventual sale of an asset owned by one of his subsidiaries [9]. He does this by stepping up the basis of this asset according to the rules set forth in §755 of the IRC. In this way, he manages to eliminate taxable gain while ostensibly remaining within the bounds of the tax law [25].

The sequence of transactions, shown graphically in Figure 6, for the iBOB scheme are enumerated:

0. In the initial ownership network Mr. Jones is a 99% partner in JonesCo and FamilyTrust, whereas JonesCo is itself a 99% partner in another partnership, NewCo. NewCo owns a hotel with a current fair market value (FMV) of $200. If NewCo decides to sell the hotel at time step 1, Mr. Jones will incur a tax from this sale. The tax that Mr. Jones owes is the difference between the FMV at which the hotel was sold and his share of inside basis in this hotel, i.e. $199-$119 = $80. Mr. Jones can evade this tax by artificially stepping up the inside basis of the hotel to $199.

1. In the first transaction, we see that FamilyTrust, which Mr. Jones controls, decides to buy JonesCo's partnership share in NewCo for a promissory note with a current value of $199. Of course, FamilyTrust has no intention of paying off this note, as any such payments entail a tax burden upon NewCo. Having already made a 754 election, FamilyTrust steps up its inside basis in the hotel to $199.

2. When NewCo sells the Hotel to Mr.Brown for $200, Mr. Jones does not incur any tax, as the difference between the current market value and his share of inside basis in the hotel is now zero.

4.2 Setup

We ran 100 independent iterations of the co-evolutionary GA for 100 generations each with tax scheme and audit score populations of size 100. We chose 0.5 of the tax scheme population for evaluating the fitness of the solution in the other audit score population and vice-versa. The parameters that govern the GA simulation are displayed in Table 2.

Table 2: Parameters for STEALTH iBOB experiments

Parameter	Description	Value
`Mutation rate`	probability of integer change in individual	0.1
`Crossover rate`	probability of combining two individual integer strings	0.7
`Tournament Size`	number of competitors when selecting individuals	2
`Number chosen`	fraction of other population each individual is evaluated	0.5
`Population size`	number of individuals in each population	100
`Generations`	number of times populations are evaluated	100

We are doing an initial exploration of the problem and the choice of parameters and operators are a first attempt. Each population has the same operator and parameter settings. In the initialization integers are randomly chosen. Individuals are selected from the population using tournament selection. In the crossover operation two individuals are combined into two new individual by randomly picking a single point and swap after the point. The grammar used to map the integers in an individual is shown in Figure 3. The mutation operation of an individual chooses a new random integer at a random position. The fitness of an individual is the average over a number of randomly chosen individuals from the other population. The objectives functions used for these experiments were simple single objective and opposites of one another. Given that Jones's taxable income was the only one of interest, $\zeta = 1$, and we can define his taxable income as just ℓ. Thus the objective function for a transaction sequence is $h_e(\ell, s) = -\ell(1-s)$. Conversely, the objective function for audit score sheets are $h_a(\ell, s) = -\ell(1-s)$.

4.3 Coevolution of Auditors & Evaders in iBOB

Fitnesses from various subpopulations of the transaction sequence population from a selected run we investigate are shown in Figure 7. A sharp increase in the fitness of the "best" transaction sequence indicated the discovery of new way to minimize the payment of taxes. As soon as that occurs, the fitness of the best 10% of transaction sequences increases to the maximum, shortly followed by the fitness of all of the sequences in the population. This agrees with the real-life observations of abusive tax shelters, where tax-minimizing schemes quickly propagate amongst the industry once discovered [25, 21]. Also encouraging is the combined decline amongst all subpopulation fitnesses, indicative of the evolution of an audit score (not shown) sheet that increases the audit likelihood of a transaction sequence exhibiting the previously discovered scheme

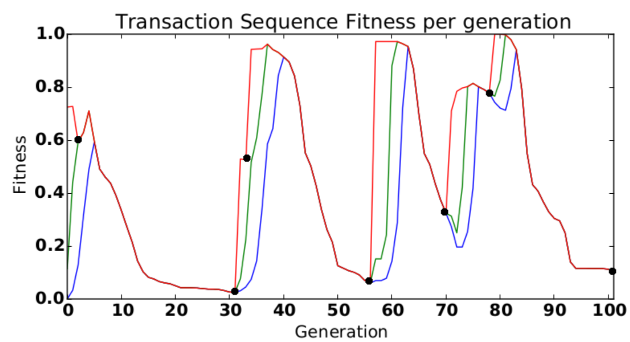
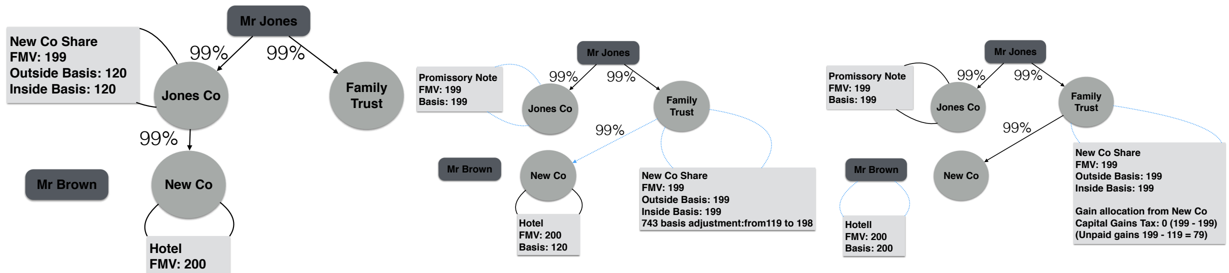


Figure 7: From a selected run. Fitness of best transaction sequences (red), mean of top ten sequences (green) and mean of population (blue). The dots signify points at which a novel tax-minimizing strategy is evolved.

Figure 8 below shows a nuanced picture of the audit score sheet population's response to the general trend in the transaction sequence population from the selected run. The colored background shows the audit weight distribution of the most fit audit score sheet in the population. Conversely, colored the lines show the proportion of the transaction sequence population that uses the scheme of the corresponding color. Thus we can see how the proportion of certain tax schemes follow the existence of the highest fitness audit score sheet.

We observe that an audit score sheet capable of sufficiently auditing a certain type of tax scheme can co-exist with that scheme for some time until the frequency of that tax strategy starts to decline. This demonstrates *a)* the successful audit score sheet taking time to propagate amongst its population and *b)* the jagged fitness landscape of the transaction



(a) iBob step 0

(b) iBob step 1

(c) iBob step 2

Figure 6: The steps in the iBOB tax evasion scheme. The basis of an asset is artificially stepped up and tax is avoided by using "pass-through" entities.

sequences, both of which are mildly reminiscent of the fast-slow dynamics mentioned in Section 1. That is, audit score sheets have a shallow but smooth fitness landscape, allowing successful auditing policies to be "seen" easily, but with a slow dissemination. Conversely, dominant tax-minimization strategies have a jagged and more stochastic discovery process, but successful schemes propagate rapidly once found.

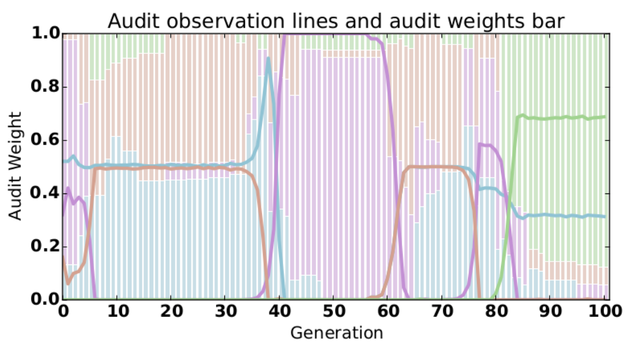


Figure 8: From a selected run. Audit weights of best audit score sheet and proportions of various transaction sequence scheme types in population

Thus the dynamics we set out to prove our model was capable of replicating were achieved. Successful transaction sequences are those that generate low levels of taxable income for Jones, as well as exhibiting behavior that is not adequately represented in the audit score sheet population. Soon enough, the objective functions of the auditing policies begin to associated that behavior with low taxable income relative to other transaction sequences that accomplish the same economic purpose and assign an audit weight to that behavior. The effectiveness of that tax strategy then decreases until a new tax-minimizing strategy is found which once again evades all (or most) existing auditing policies. That strategy then rapidly spreads amongst the transaction sequence population and the process continues.

There are calibrations that can improve the fidelity of the experiments. For example, while transaction sequences are clearly more responsive to a successful individual in their population than audit score sheets, the time scale gives too much credit to the propagation of audit score sheets. For example, Figure 7 shows that a successful tax strategy enjoys only about $5 - 10$ generations of unbridled prosperity until an auditing policy evolves and propagates that reduces its effectiveness. Transaction sequences take about the same amount of generations to figure out a new dominant tax strategy, the only tangible difference is the speed at which it propagates through the population. Thus, there must be further calibration in the model to reflect the differences in time scale.

5. CONCLUSION AND FUTURE WORK

The purpose of the presented methodology is to replicate the co-evolutionary relationship between tax evaders and auditors, using US partnership taxation as an initial example. This was accomplished by separating the effort into three parts: *(1)* representing the rule system in order to calculate benefit that the advisor can offer to their client *(2)* simulating interactions between the advisor's strategy and the relevant regulatory authority, and *(3)* optimizing for behavior on both ends of the relationship to investigate potential areas of exploration.

Through experimentation, we show that the co-evolutionary relationship can indeed be replicated, given the proper specifications. Some further parameter calibrations are required in order to capture certain time scale effects, but the qualitative dynamics are present. Transaction sequences can be shown to respond to both tax minimizing behavior and risk of being audited. Similarly, auditing policies respond to and isolate behavior which generates lower than expected taxable income.

There is still much to explore of this methodology. While our representation of US partnership tax code is in itself a novel discovery, its most exploited aspects remain the most crudely approximated by our formulations. Specifically we would like to explore non-recourse liabilities and depreciation deduction schedules as a means to minimize taxable income. Another key aspect of validating the co-evolutionary search method is by gaining access to actual auditing data. This is a non-trivial process that requires security clearance.

Additional future work is in analyzing the dynamics of the algorithm, i.e. how the solutions in the populations are coevolving during the co-evolutionary search, e.g. longer runs in terms of generations, comparisons with previous populations. One feature of co-evolutionary algorithms which is important to investigate further is it generates intricate runtime behaviors and makes it difficult to monitor progress towards the goal, e.g. over-specialization, under specialization and stalling. Finally, we need to explore the parameters and operators of the coevlutionary algorithm, use of multiple objectives and archives in the coevolution, e.g. the number of individuals chosen for evaluation in the opposing population, reduce elitism.

References

[1] M. G. Allingham and A. Sandmo. Income tax evasion: a theoretical analysis. *Journal of Public Economics*, 1(3-4):323–338, 1972.

[2] K. Bloomquist. Tax compliance as an evolutionary coordination game: an agent-based approach. *Public Finance Review*, 39(1):25–49, 2011.

[3] S. Checkoway and H. Shacham. Escape from return-oriented programming: Return-oriented programming without returns (on the x86). Technical Report CS2010-0954, UC San Diego, Feb. 2010.

[4] R. Colbaugh and K. Glass. Moving target defense for adaptive adversaries. In *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on*, pages 50–55. IEEE, 2013.

[5] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 99–108. ACM, 2004.

[6] E. D. de Jong, K. O. Stanley, and R. P. Wiegand. Introductory tutorial on coevolution. In *GECCO (Companion)*, pages 3133–3157, 2007.

[7] D. DeBarr and Z. Eyler-Walker. Closing the gap: automated screening of tax returns to identify egregious tax shelters. *ACM SIGKDD Explorations Newsletter*, 8(1):11–16, 2006.

[8] S. G. Ficici and A. Bucci. Advanced tutorial on coevolution. In *GECCO (Companion)*, pages 3172–3204, 2007.

[9] GAO. Gao-14-453. http://www.youtube.com/watch?v=O8VDUStvxMY, 2013.

[10] GAO. http://www.gao.gov/assets/670/663185.pdf, 2014.

[11] GAO. http://www.gao.gov/assets/670/661772.pdf, 2014.

[12] F. Haddadi and A. N. Zincir-Heywood. Botnet detection system analysis on the effect of botnet evolution and feature representation. In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 893–900. ACM, 2015.

[13] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. *Physica D: Nonlinear Phenomena*, 42(1):228–234, 1990.

[14] H. G. Kayacık, A. N. Zincir-Heywood, and M. I. Heywood. Can a good offense be a good defense? vulnerability testing of anomaly detectors through an artificial arms race. *Applied Soft Computing*, 11(7):4366–4383, 2011.

[15] C. Le Goues, A. Nguyen-Tuong, H. Chen, J. W. Davidson, S. Forrest, J. D. Hiser, J. C. Knight, and M. Van Gundy. Moving target defenses in the helix self-regenerative architecture. In *Moving Target Defense II*, pages 117–149. Springer, 2013.

[16] R. J. M. Musliner D. J., Friedman Scott E. Automated fault analysis and filter generation for adaptive cybersecurity. In *Proc. 6th Int'l Conf. on Adaptive and Self-Adaptive Systems and Applications*, 2014.

[17] H. Okhravi, T. Hobson, D. Bigelow, and W. Streilein. Finding focus in the blur of moving-target techniques. *Security Privacy, IEEE*, 12(2):16–26, Mar 2014.

[18] M. O'Neill and C. Ryan. *Grammatical evolution: evolutionary automatic programming in an arbitrary language*, volume 4. Springer, 2003.

[19] M. Pickhardt and A. Prinz. Behavioral dynamics of tax evasion–a survey. *Journal of Economic Psychology*, 40:1–19, 2014.

[20] J. Rosen. Computer aided tax avoidance policy analysis. Master's thesis, Massachusetts Institute Of Technology, 2015.

[21] T. Rostain and M. C. Regan. *Confidence Games: Lawyers, Accountants, and the Tax Shelter Industry*, volume 1. MIT Press, 2014.

[22] G. Rush, D. R. Tauritz, and A. D. Kent. Coevolutionary agent-based network defense lightweight event system (candles). In *Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference*, pages 859–866. ACM, 2015.

[23] G. Willard. Understanding the co-evolution of cyber defenses and attacks to achieve enhanced cybersecurity. *Warfare*, 14:17–31, 2015.

[24] M. L. Winterrose and K. M. Carter. Strategic evolution of adversaries against temporal platform diversity active cyber defenses. In *Proceedings of the 2014 Symposium on Agent Directed Simulation*, page 9. Society for Computer Simulation International, 2014.

[25] D. Wright Jr. Financial alchemy: How tax shelter promoters use financial products to bedevil the irs (and how the irs helps them). *Ariz. St. LJ*, 45:611, 2013.