Tax Non-Compliance Detection Using Co-Evolution of Tax Evasion Risk and Audit Likelihood

Erik Hemberg
MIT
32 Vassar St
hembergerik@csail.mit.edu

Jacob Rosen
MIT
32 Vassar St
jbrosen@mit.edu

Geoff Warner
MITRE CORPORATION
McLean, VA 22102
gwarner@mitre.org

Sanith Wijesinghe
MITRE CORPORATION
McLean, VA 22102
sanith@mitre.org

Una-May O'Reilly
MIT
32 Vassar St
unamay@csail.mit.edu

ABSTRACT

We detect tax law abuse by simulating the co-evolution of tax evasion schemes and their discovery through audits. Tax evasion accounts for billions of dollars of lost income each year. When the IRS pursues a tax evasion scheme and changes the tax law or audit procedures, the tax evasion schemes evolve and change into undetectable forms. The arms race between tax evasion schemes and tax authorities presents a serious compliance challenge. Tax evasion schemes are sequences of transactions where each transaction is individually compliant. However, when all transactions are combined they have no other purpose than to evade tax and are thus non-compliant. Our method consists of an ownership network and a sequence of transactions, which outputs the likelihood of conducting an audit, and requires no prior tax return or audit data. We adjust audit procedures for a new generation of evolved tax evasion schemes by simulating the gradual change of tax evasion schemes and audit points, i.e. methods used for detecting non-compliance. Additionally, we identify, for a given audit scoring procedure, which tax evasion schemes will likely escape auditing. The approach is demonstrated in the context of partnership tax law and the Installment Bogus Optional Basis tax evasion scheme. The experiments show the oscillatory behavior of a co-adapting system and that it can model the co-evolution of tax evasion schemes and their detection.

Categories and Subject Descriptors

M.2.5.1 [**Social and professional topics**]: Taxation

General Terms

Algorithms, Economics

ICAIL'15, June 08 – 12, 2015, San Diego, CA, USA

DOI: http://dx.doi.org/10.1145/2746090.2746099.

Keywords

tax evasion, grammatical evolution, genetic algorithms, auditing policy, innovative applications

1. INTRODUCTION

Financial and legal enterprises promote tax non-compliance by combing through the tax code for possible areas of exploitation. Transaction sequences that reduce and obscure the tax liabilities for their individual shareholders are constructed around these technicalities. In 2014 the Government Accountability Office (GAO) estimates[7] that $91b in underrepresented income can be attributed to tax shelters implemented through "pass through" entities, such as partnerships. While tax auditors have historical examples of tax evasion schemes to help guide examination efforts, tax shelter promoters often adapt their strategies as existing schemes are uncovered and when changes are made to the existing tax regulations. One notable example is the so called BOSS tax shelter (Bond and Options Sales Strategies) that was detected and disallowed. After the changes to the regulation the audits in place could detect BOSS but were not able to detect the newly emerged variant "Son of BOSS" [20]. The arms race between tax evasion schemes and actions by tax authorities presents a significant challenge to enforcement efforts. We describe a research methodology that can help detect tax non-compliance strategies, primarily those implemented via clauses within Subchapter K, which legislates partnerships, of the IRC.

Most tax evasion schemes are sequences of transactions where each transaction is individually compliant. However, when all the transactions are combined they have no other purpose than to evade tax. The law is difficult to articulate correctly for all possible cases, there are immeasurable combinations of transactions. Thus, it is possible to disallow the uncovered cases of tax non-compliance via anti-abuse doctrines, e.g. the "economic substance" doctrine, which specifies that transactions must contain both economic substance and a business purpose [16].

Our approach for detecting tax evasion places us in the realm of AI research and legal reasoning that started in the 1970s [5] The field of AI and law [17, 3] deals with simulation of norms and their emergence [11]. Pioneering work regarding taxes occurs in "Taxman" [13]. Later, machine learning has emerged as a means of detecting non-compliant networks of transactions [6, 18]

The Internal Revenue Service (IRS) uses protocols to rate whether an entity is suspicious enough to undergo a full audit [2]. Our goal is to help tax authorities anticipate how changes to the tax code and audits drive non-compliance strategies. Our approach is to model the co-evolutionary arms race between *transaction sequences* in ownership networks with their corresponding *audit observables.*

The main challenge is how to detect tax non-compliance, i.e. figuring out how it is possible to evade tax. Tax evaders often try to obfuscate their schemes by using large hierarchies of entities, e.g. up to 100 tiers and 10.000 partners, providing as little information as possible, obfuscating information and stalling filing report. The auditors initially work with aggregated and dispersed information from the tax returns, e.g on form 1065 (Schedule K-1) for year 2014, only the occurrence of a Internal Revenue Code (IRC) §754 election has to be reported. The complicated corresponding calculations demonstrating the basis adjustments are filed as separate paper attachments. The tax law, e.g. calculation of basis, can be complicated in some paragraphs and byzantine in others. For example, owners of a partnership need to adjust, maintain and update separate basis values in relation to the owner and asset, in part described by IRC §755. Consequently, we need to determine which tax results are reached that run contrary to the purpose of the statutes that are used to produce such as result.

Our model includes a method for generating any feasible transaction sequences in a specified tax ecosystem and a practical simulation of transaction sequences that outputs the tax liability and audit score, within the specified context. Also, we introduce an optimization method that can find non-compliance, i.e. a method to search for non-compliant sequences of transactions. An important property from using tax code and audit simulation to detect unknown non-compliance is that it does not require data.

Our method of detecting tax non-compliance is similar to "10,000 monkeys using TurboTax". First we create and encode a model of the relevant parts of the law by defining actors and actions in the tax ecosystem, in order to calculate the tax liability for a sequence of transactions given an initial ownership network. Entity ownership forms the connections between entities in the network and a transaction between two entities changes the network state. Each transaction is simulated and checked for feasibility before applying law and accounting rules. Auditors might observe any transaction within the sequence and ownership network, not only the preceding transaction. There are an infinite number of transaction sequences and the order of the transactions is important when calculating tax liability. Thus, we described a compressed form represents all possible transaction sequences and ownership networks. From the compressed form it is possible to generate any combination of transaction sequences. During auditing, what to observe must be chosen conscientiously from all the possible observables. The arms race between tax evader and auditor directs the optimization, e.g. detecting novel suspicious networks of transactions. The co-evolutionary process creates the next generation of transaction sequences and audit observables and requires both to be updated. The observables that an auditor is interested in can be checked while evaluating the transactions. The model then assigns a tax non-compliance risk and an audit likelihood score to transaction networks and audit observables. Previously, Genetic Algorithms have been used for searching for tax non-compliance [19] and for co-evolutionary search.

We proceed as follows in describing how our method, called Simulating Coevolution and Tax Evasion (SCOTE) can detect tax non-compliance. Section 2 has background of partnership tax law used in SCOTE, related literature and methods. In section 3 methodology is provided. In section 4 we demonstrate the capability of SCOTE to identify an artificial basis step-up tax scheme called iBOB. Finally, there are conclusions and future work in Section 5.

2. BACKGROUND

We begin with a background of partnership tax law, related literature on AI and tax evasion and co-evolutionary heuristics for optimization.

2.1 Partnership tax law fundamentals

We focus on taxes incurred during the sale of investment property. An extensive set of rules applies to the calculation of taxes during such transactions [9]. The terms of interest are: *a)* **Basis**: The original investment to acquire an asset, often its purchase price. Basis is used to determine gain or loss on asset disposition *b)* **Fair Market Value**: The value of an asset at a given time. *c)* **Gain**: A result of when an asset is disposed at a price higher than its adjusted basis. *d)* **Loss**: A result of when an asset is disposed at a price lower than its adjusted basis. If losses exceed gains, the excess can be deducted from an individual's tax return to reduce their overall tax liability up to a pre-defined threshold. Net losses in excess of this threshold can be rolled across multiple years. The use of basis to calculate taxable gain is described in figure 3a.

Individuals have been found to disguise gains by conducting transactions across multiple tiers of nested partnerships [20]. Taxable gain or deductible loss calculation using basis can become complicated when individuals form *partnerships* by pooling resources (e.g. cash, property) to conduct joint businesses. Partnerships are legal tax entities that are governed by rules defined in sub-chapter K, §701–777 of the IRC. While partnerships are required to file a tax return detailing their economic activity they are not directly taxed and instead any gain/loss is "passed through" to their immediate owners in proportion to their ownership percentages. In order to determine the corresponding tax liability for each of the partners, the basis of the original assets contributed to the partnership must be tracked. The basis must be tracked separately for both the "inside basis" –the share of basis assigned to each partner for all assets held by partnership, and the "outside basis" –the basis each partner holds in their partnership interest. Most times, the inside and outside basis will match but can begin to differ when partnership interests are transferred. In order to remove this mis-match, partnerships are allowed to adjust the inside basis of their assets using a special election known as the §754 election. A basis adjustment occurs in the case of two separate scenarios: a) sale of a partnership interest, defined in IRC §743, and/or b) distribution of property, defined in IRC §734.

2.2 Detecting Tax Non-Compliance

Tax evasion can be considered a gamble like any other investment involving risk and uncertainty [1]. This insight forms the basis for many agent based modeling approaches that divert from standard microeconomic notions of tax evasion and instead consider the individual preferences of heterogeneous agents [4, 15]. Alternate techniques use machine learning algorithms, e.g. Decision Trees, Neural Networks, Logistic Regression, Support Vector Machines [6].

In biology, co-evolution describes situations where two or more species reciprocally affect each other's evolution. The notion of adversarial co-evolution from biology can be used for the circumstances of the auditors, e.g. 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.

Co-evolutionary optimization heuristics [10] evaluate solutions based on interactions between populations of multiple solutions. The Genetic Algorithm (GA) [8] is used for co-evolutionary optimization. It is a general-purpose stochastic adaptive learning heuristic that searches and scores solutions in parallel. The GA draws inspiration from the fundamental principles of population adaptation through inheritance, selection and genetic variation in neo-Darwinian evolution. In the GA solutions are represented as fixed length bitstrings, evaluation accredits a score(fitness) to a solution, good(fit) solutions are selected as parents, and new solutions are created by inheritance with variation, see Figure 1.

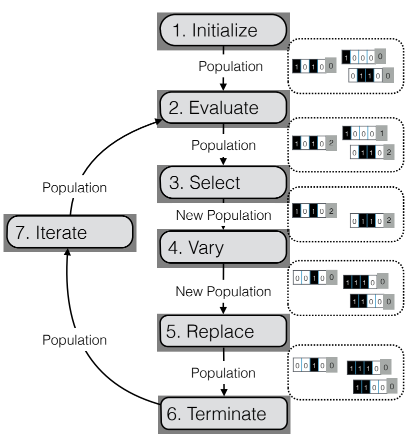


Figure 1: Overview of the flow of a Genetic Algorithm. A population of bitstrings is created as fixed length bitstrings, evaluation accredits a score(fitness to a solution, good(fit) solutions are selected as parents, and new solutions are created by inheritance with variation.

3. SCOTE – A METHOD FOR DETECTING TAX LAW NON-COMPLIANCE

The levels of SCOTE are: *a)* the representation level of the tax ecosystem, with the representation of tax evasion schemes and audit score sheets, here we use decision rule trees to represent law *b)* the simulation level that processes a transaction sequence and outputs tax liability and audit score *c)* the optimization level that searches transaction sequences based on audit risk. The levels are described in more detail in this section, along with a formalization of SCOTE.

3.1 Model of tax non-compliance

We have broken the tax ecosystem down into three fundamental components; **tax entities** (e.g. taxpayers, partnerships) their **assets** (e.g. cash, real-estate, securities) and the corresponding **transactions** that occur amongst them. Tax laws govern interactions between entities and specify any resulting tax liability that might occur as a result. Adherence to these tax laws is verified by the use of compliance audits.

The tax ecosystem model components, as an ownership network and sequence of transactions that are suited for a computational search of non-compliance. Figure 2 show how the input of a transaction, audit score sheet and ownership network are checked for feasibility, transfer of assets, calculation of tax and audit, and then outputs tax liability, audit score and an updated ownership network.

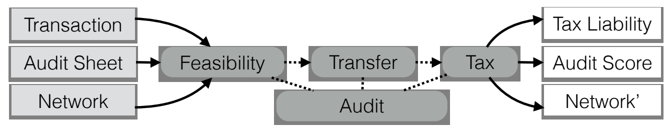


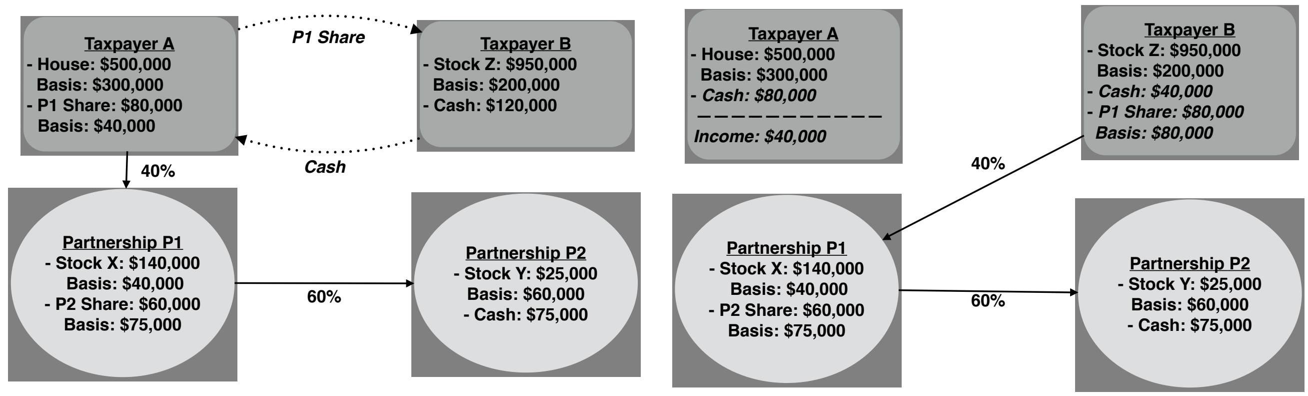
Figure 2: SCOTE tax ecosystem model has transaction, audit score sheet and ownership network. These are checked for feasibility, transfer of assets, calculation of tax and audit, that yields a tax liability, audit score and updated ownership network.

3.1.1 Tax transactions and ownership networks

We represent the partnership and asset activity in an ownership *network*, it provides transaction level chronological snapshots of activity regarding assets in the portfolios and their bases. The *nodes* in the network represent tax entities, while the *edges* represent ownership relations between those entities. A transaction consists of a pair of actions in opposite directions which both change the state of the network. An action transfers an asset from one entity to another entity. Each action alters the state of the network by updating the stateful variables in the nodes. Moreover, each entity has a portfolio of assets that it owns. An asset is transferred from the portfolio of one entity to the portfolio of another entity.

For example: Take a tax ecosystem with four entities, two taxpayers (Taxpayer A and B) and two partnerships (Partnership P1 and P2) as in Figure 3a, the nodes in the network are entities, arrows are edges for ownership relations and the dotted lines represent the transaction. Taxpayer B buys a share in Partnership P1 from Taxpayer A with cash. In the transaction, which consists of two actions, Action 1 transfers cash from Taxpayer B to Taxpayer A and Action 2 transfers a partnership share from Taxpayer A to Taxpayer B. When the partnership share is transferred to node Taxpayer B, the node Taxpayer A is updated to show the cash in its portfolio. In addition, the transaction results in income for Taxpayer A, the basis is lower than the price that Taxpayer B paid for the asset, $\$40,000 = \$80,000 - \$40,000$.

The network representation of our tax ecosystem allows us to record snapshots of a sequence of transactions between multiple entities and calculate tax incurred per entity, per transaction. Transactions happen sequentially, at any given time a transaction will only take place between two given nodes. The network representation also makes our design modular. We can add different kinds of entities by introducing more nodes in the network and similarly we can introduce more diversity within nodes by having different kinds of assets.



(a) Initial state and transaction

(b) State after transaction

Figure 3: Example of transaction with a network representation showing relationship edges and network state transition via a transaction.

3.1.2 Audit score sheets

Audits play two roles in SCOTE. First we use them to direct and co-adapt transaction sequences towards evasion (see Section 3.3.1). Second, they are used to discern the nature of the non-compliance. The IRS issues tax guidance on matters related to regulations, revenue rulings and revenue procedures, using a number of announcements and notices. These collective communications can be used to clarify the intent of the tax code and determine specific transactions and/or transaction types deemed to be in violation of certain regulatory statutes. In SCOTE, the IRS audits are modeled based on this public information. E.g. , in 2004 the IRC was altered in §743(a) to read

> 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 basis of partnership property) is in effect with respect to such partnership or (3) unless the partnership has a substantial built-in loss immediately after such transfer.

We have added numbers in parenthesis to signify *observable* events and the amendment is captured in SCOTE as follows: (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 is a procedure that examines a sequence of transactions for particular events that are tracked. To represent audits in SCOTE we use a list of audit points (weights), corresponding to all detectable events that can occur when a network of transactions is executed. An audit score sheet is a collection of audit points, each corresponding to a different type of event that may be present in a transaction. The higher the audit points associated with a certain type of event, the more suspicious that type of event is. We also constrain the sum of audit points to equal one, in order to mirror the limited resources available for auditing. The audit score associated with a sequence of transactions and an audit score sheet is defined as the sum of all of the audit points present in a sequence of transactions, multiplied by their respective frequencies. Visually, audit score sheets can be represented by a spreadsheet, with each row corresponding to a different type of audit observable as shown in Table 1. One can imagine a hypothetical auditor going through a network of transactions and incrementing the frequency in the far right column whenever each type of event is observed.

Table 1: Each row has the observable, the associated audit point and the number of times it occurs in a sequence of transactions

Observable	Point	Frequency
$Activity_1$	$Point_1$	$Frequency_1$
...	...	...
$Activity_n$	$Point_n$	$Frequency_n$

Using this formulation, we interpret an audit score as the *likelihood* that a sequence of transactions will be audited.

The representation of audit points relies on the presence of "observable" events. An observable event is one that is possible to detect in the tax ecosystem model, but not necessarily by the IRS. E.g, if a taxpayer purchases a share in a partnership for cash, SCOTE will process that as a transaction involving a partnership asset, as well as all parties involved in the transaction, while an actual audit would require more effort to acquire the information. Three possible types of events are observable within SCOTE, but are not currently used by the IRS for auditing decisions. *1)* Events that are possible to observe *2)* Events that may be impossible or very difficult to observe, but can be captured with an observable proxy *3)* Events that are impossible to observe due to either logistical or legal reasons.

3.1.3 Design of tax law and its application to transactions

A transaction is evaluated by abstracting the tax law into three parts. The parts that the tax law checks are:

Feasibility of a transaction A transaction consists of one action transferring an asset from one entity to another entity, and the other action transferring an asset in the other direction. A transaction is checked to determine *a)* if two assets can be exchanged for each other *b)* if the entity owns the asset that it is attempting to transfer *c)* if the receiving entity is allowed to receive the asset.

Transfer of assets Check the rules regarding the transfer of assets, e.g. how the basis of the underlying assets is adjusted when a partnership asset is transferred. A simplified decision rule tree regarding basis alteration from a transfer of an asset is shown in Figure 4.

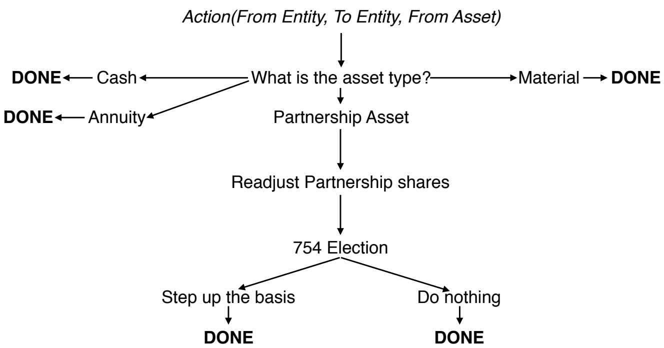


Figure 4: A decision rule tree that shows the checks on an asset transfer in a transaction.

Calculate the tax from the transaction Check the rules regarding the tax calculation of the transaction, e.g. when an entity exchanges an annuity no taxable gain or loss can be incurred, see figure 5 for a decision rule tree visualization.

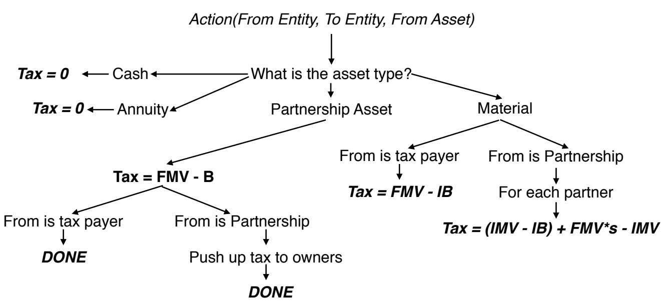


Figure 5: A decision rule tree that shows the tax calculation on an asset transfer.

SCOTE is extensible not only with regards to the entities and assets in the tax ecosystem, but also when and where a tax law is expressed, and to what degree the tax law is simplified. Modification of tax law requires alteration to all three parts.

3.2 Simulating the tax ecosystem model

At the heart of SCOTE is the tax ecosystem simulation of the model, see figure 6. It initializes the model of our tax ecosystem and evaluates transactions between entities. It derives the tax liability incurred by these transactions. The input to the simulation is an ownership network, an audit score sheet with associated points for each audit observable and a transaction sequence. The transaction sequences are complete as they detail every interaction between nodes in the ownership network. The tax ecosystem model checks to see if transactions are valid and taxable. A transaction updates the state of the ownership network, and concurrently the audit score and the tax liability are calculated. As each transaction in the network is executed, the tax simulator calculates the audit score for the types of financial events indicated on the audit score sheet. The entire transaction sequence is evaluated by iterating over it.

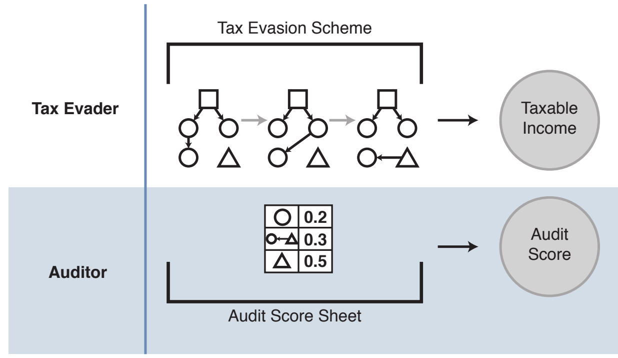


Figure 6: SCOTE simulating tax ecosystem

3.3 Optimization architecture

The optimization in SCOTE is directed by the adversarial dependence between tax evasion and audits. SCOTE performs a co-evolution of a population of tax evasion schemes with a population of audit score sheets, both of which are evaluated in every step against a sub-population of the opposite agent type, as shown in Figure 7. That is, each tax evasion scheme "selects" some audit score sheets to calculate its fitness. Similarly, each audit score sheet selects some tax evasion schemes to calculate its fitness.

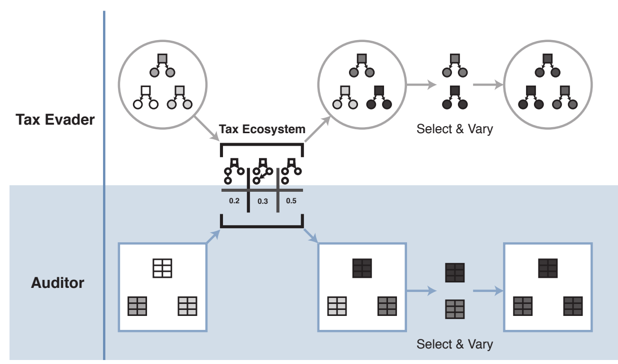


Figure 7: Concurrent optimization of high likelihood audit scores and low risk tax evasion schemes.

The GA performs an optimization on sequences of transactions to find the specific sequence of transactions that maximize a fitness score. A tax scheme generated by the GA is represented by a list of integers. A parser is used to read these integers and generate a network of transactions with the help of a grammar, as done by GE, see appendix 6. The transactions consist of a list of Java interpretable objects that are input to a tax ecosystem simulation to calculate the resulting taxable gain. The mechanics of co-evolution are identical to a standard GA, except the fitness for each individual is calculated with a k size subset of the opposite population.

3.3.1 *Rewarding co-evolution*

The goal of the tax evader is to minimize audit likelihood and maximize deductible loss. The reward of individual solutions in co-evolutionary search can be assigned with so called fitness functions. First, each set of transactions generates a *deductible loss*. Second, an audit score sheet generates an *audit score*, based on a sequence of transactions, which represents the likelihood that a scheme will be audited, i.e. the risk of being audited.

We take into account both the effectiveness of a tax evasion scheme from a tax perspective and a risk perspective. Two terms in the function effect a tax evasion scheme's fitness. First, fitness is positively correlated with the deductible loss. A tax evasion scheme is only effective if it results in a low tax liability. The second term in the function represents the likelihood of the audit disallowing the tax benefits gained from the scheme, which takes into account the likelihood of an audit (audit score). There should be little reduction in fitness if little tax is evaded because there is a lower likelihood that an audit will result in a negative ruling. Conversely, the objective of the auditing behaviors is the opposite of the objective for the evaders.

3.4 Formal Description of SCOTE

This section describes SCOTE in a more formal notation in order to define the scope of the approach.

3.4.1 Model of tax ecosystem

The ownership network at a given time is defined as a list of entities, each of which owns a set of assets. At any point, the state of the network can be described as some $\gamma \in \Gamma$, where $\gamma = \{\mathbf{e}, \mathbf{a}, d\}$, where $\mathbf{e} = \{e_i\}_{i=0}^{k_1}$ is the set of entities, $\mathbf{a} = \{a_i\}_{i=0}^{k_2}$ is the set of all assets and $k_1, k_2 \in \mathbb{Z}_+, e_i \in E, a_i \in A$. 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.

Next we define a sequence of transactions as a vector $\mathbf{t} = \{t_i\}_{i=0}^{k}$ for some $k \in \mathbb{Z}_+$, $t \in \mathbf{T}$ is the space of all transactions. A transaction is defined as $t = \{e_f, e_t, a_f, a_t\}$, 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.

For audits, suppose that there are n specific types of events that are observable, represented by $\{b_i\}_{i=0}^{n}$. Associated with each type of event are the audit points $\{\alpha_i\}_{i=0}^{n}, \alpha \in \mathbb{R}$ 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

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

We observe that laws governing a given transaction depend on the "type" of assets and entities being exchanged. For example, the laws governing the exchange of a hotel for cash between two taxpayers are different from those governing the contribution of an annuity to a partnership in exchange for a share. Thus, we can determine the laws governing a given transaction by the combination of both asset and entity types.

3.4.2 Simulation of tax ecosystem

Consider the abstract transaction $t = (e_f, e_t, a_f, a_t)$, which states that entity e_f gives e_t the asset a_f in exchange for a_t. Define $\hat{E}$ to be the finite set of entity *types*, and $\hat{A}$ to be the finite set of asset *types*. We can then write the set of all transactions as a union of disjoint subsets $\mathbf{T} = \cup_{i=0}^{n} T_i$, where each subset contains all transactions of a certain combination of asset and entity types. The steps that follow are.

1. a transaction type t is first checked to see if it is within the bounds of the legal/feasible region by first determining to which subset T_i it belongs. We define $\mu : T_i \mapsto \Phi$ as a map from a subset $T_i \in \mathbf{T}$ to Φ that determines the laws ϕ that govern the transaction, given its combination of asset/entity types.

2. the transfers in the two actions composing the transaction represent the transition of the network state γ_t to γ_{t+1} and γ_{t+1} to γ_{t+2} according to the map $\tau : \mathbf{T} \times \Gamma \mapsto \Gamma$

3. taxable gain/loss calculation takes a transaction t and a network state γ_t and maps it to a deductible loss value d_L for each taxable entity and an updated network state, $P : \mathbf{T} \times \Gamma \mapsto \mathbb{R} \times \Gamma$

3.4.3 Optimization of tax ecosystem

We can describe the process by which sequences of transactions and initial ownership network are generated by defining a grammar $\Xi_t : \mathbb{Z}_+^n \mapsto \mathbf{T} \times \Gamma$ that maps a list of n integers to an element in the set of sequences of transaction ($\mathbf{T}$) and an element in the set of all ownership networks (Γ). Thus, for any $\mathbf{x} \in \mathbb{Z}_+^n$, $\Xi_t(\mathbf{x}) = (\mathbf{t}, \gamma_0)$ where $\mathbf{t} \in \mathbf{T}$ is a sequence of transactions and $\gamma_0 \in \Gamma$ is an initial network.

We can now define the space of auditing observables as Ψ, where for some $m \in \mathbb{Z}_+$,

$$\Psi = \{\{b_i\}_{i=0}^{m} : b_i \in [0,1] \text{ and } \sum_{i=0}^{m} b_i = 1\} \subset \mathbb{R}_+^m$$

The grammar $\Xi_a : \mathbb{Z}_+^m \mapsto \Psi$ maps a vector $\mathbf{y} \in \mathbb{Z}_+^m$ to an element in the set of auditing behavior.

The tax ecosystem is defined as a function $\mathbf{F} : \mathbf{T} \times \Gamma \times \Psi \mapsto \mathbb{R}_+^2$ that takes as input a sequence of transactions, an initial network state and auditing observables, and generates a network state and audit score. Contained within the network state is the *deductible loss* d_L. 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 $\psi \in \Psi$, $\mathbf{F}(\mathbf{t}, \gamma_0, \psi) = (d_L, s)$

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 $i \in [0, k]$, $F_i(t_i, \gamma_i, \psi) = (\gamma_{i+1}, s_i) \quad \text{where} \quad s = s_k$

The goal of the tax evader is to minimize audit likelihood and maximize recognizable loss. First of all, each set of transactions generates a *deductible loss*, d_L. Secondly an audit score sheet generates an *audit score*, s based on a network of transactions, which represents the likelihood that a scheme will be audited, i.e. the risk of being audited. Thus, we can represent the fitness function, h_e for a tax evasion scheme, given a specific audit score sheet, as $h_e = d_L(1 - s)$

The goal of the auditor is to maximize the likelihood of an audit of a network of transactions with high deductible loss. The fitness function for an audit score sheet given a specific tax evasion scheme is the same as that shown above, but with the opposite sign $h_a = -h_e = -d_L(1-s)$. An audit score sheet is fit for a specific evasion scheme if either *1)* there is a high level of taxable gain *2)* if there is a high likelihood that if not much tax is collected, then the scheme will be audited

We describe how to judge the fitness of a network of transactions $\mathbf{t}$ and an auditing behavior ψ based on the deductible loss d_L and audit score s generated from the tax ecosystem model $\mathbf{F}$. We can now also define the fitness function $h : \mathbb{R}_+^2 \mapsto \mathbb{R}$ as such $h_e(d_L, s) = d_L(1-s)$

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 evaders.

4. EXPERIMENTS WITH SCOTE

We demonstrate how transaction schemes and audit scores co-evolve in SCOTE by using an artificial basis step-up scheme. The aim of the experiments is to demonstrate that SCOTE can optimize simultaneously for tax evasion schemes and audit scores as they mutually adapt to one another over time. The results demonstrate that SCOTE has the fundamental components and processes for detecting tax non-compliance.

4.1 iBOB– An artificial basis step-up scheme

For the purposes of these experiments, we consider 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. 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 [20].

The sequence of transactions, shown graphically in Figure 8, 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 Parameter settings in SCOTE experiments

To run SCOTE we need to specify the co-evolutionary optimization parameters, initial network of tax entities, the grammar for transactions and the audit score sheet's observables.

The parameters that govern the GA are displayed in Table 2. We ran 100 independent iterations of the co-evolutionary GA. 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 fitness function is the one described in Section 3.4.3.

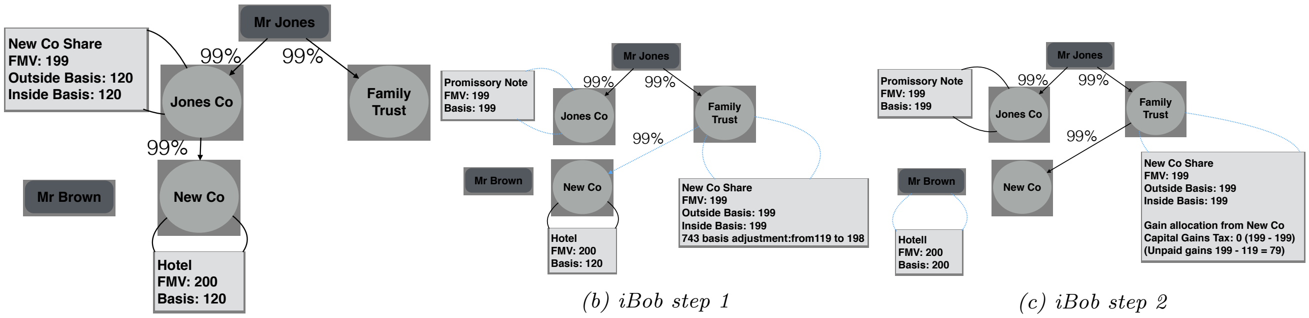
Table 2: co-evolutionary optimization parameters for SCOTE

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 determining most fit individuals	2
`Number chosen`	fraction of other population that each individual is tested against	0.5
`Population size`	number of individuals in each population	100
`Generations`	number of times populations are evaluated	100

We initialize a network with two Taxpayers, Mr. Jones and Mr. Brown, and three partnerships, JonesCo, NewCo, and FamilyTrust. These entities have portfolios of assets that include Cash, an Annuity, a Hotel, and various partnership shares. The assets can have different fair market values. The Backus-Naur Form (BNF) grammar used by SCOTE is detailed in figure 10.

In addition to the sequence for *iBOB*, we note two additional patterns of transaction activity that can sometimes result in zero tax liability for Mr Jones. The first of these involves the transfer of a partnership interest between two linked entities in the same enterprise structure [12], usually resulting in a basis adjustment due to an earlier §754 election. By *linked* we mean a transaction in which the two parties are connected by an ownership relationship. In the IBOB context, these include "singly linked" transactions, such as those that may occur between Mr Jones and JonesCo, and *doubly linked* transactions, as may occur between Mr Jones and NewCo. These types of transactions result in zero tax liability for all parties, but would almost certainly be audited. The second such transaction involves the use of *annuities* such as promissory notes, because annuities allow for tax deferral, which always works to the taxpayer's benefit. As with linked transactions, defaulting on annuity payments is nominally legal and results in zero tax liability, but can be very suspicious for auditors.

The experiments: *1,* (Section 4.3.1), show that when there is no audit observable that can capture the evasive scheme, the non observable tax evasion scheme population converges to the tax evasion scheme that cannot be detected *2,* (Section 4.3.2) shows that if the tax evasion scheme is observable, then the audit points will converge. *3,* (Section 4.3.3) shows that if not all schemes are observable the



(a) iBob step 0

(b) iBob step 1

(c) iBob step 2

Figure 8: 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.

evasion schemes and audits fitnesses oscillate, this is enabled by a constraint to the audit observables, one of the audit observables has to be zero, to model limited audit resources.

For the experiments considered here, the model represents audit scores as the sum of four audit points between 0 and 1 as shown in Table 3. The value of each audit point can be thought of as the relative importance of the associated behavior to the IRS.

Table 3: Each row lists an audit observable. Each column lists an experiment and their average initial distribution of weights in the population.

Audit observable	**Exp 1**	**Exp 2**	**Exp 3**
iBOB	0	0.25	0.25
Annuity	0.33	0.25	0.25
Linked	0.33	0.25	0.25
Double Linked	0.33	0.25	0.25

4.3 Results – Co-evolution of iBOB

We verified the co-evolutionary dynamics of SCOTE and their sufficiency to find existing tax evasion schemes. We expect that as these schemes evolve to accommodate existing audit priorities, the audit points will themselves evolve to detect the new evasion behavior. This should result in either convergent or oscillatory dynamics.

4.3.1 Experiment 1: Audit observables not covering the tax evasion schemes

For this experiment we purposefully left out audit observables that can detect iBOB. Thus, once iBOB is found it should propagate through the tax evasion scheme population on the merit of being the best tax evasion scheme and unauditable. In fact iBOB was found in 34% of the iterations, and we expect most iterations to converge to iBOB if the number of generations were increased. In addition, the audit points are unchanged and in equilibrium once iBOB has been found. See Figure 9a for a plot of tax evasion and audit score sheet best fitness from one iteration. We note that transactions that exchange a material for an annuity are assigned a higher audit point.

There is a clear pattern when iBOB is evolved: initially, the pool of tax evasion schemes gravitates towards a network of transactions that contains suspicious activity, which the audit scores are able to detect. Only after the audit scores evolve to reduce the fitness of such schemes does iBOB become dominant.

Note that, two distinct meta-stable states emerge when the basic iBOB is not found. The most common is when a suspicious scheme is evolved in an early generation, which the audit scores can effectively detect early on, causing the scheme fitness to converge towards its minimum and the tax code fitness to converge to its maximum. Alternatively, the pools of both evasion schemes and audit scores oscillate in respect to each other for the duration of the run, implying a process of suspicious schemes emerging and audit scores evolving to detect them, causing another suspicious scheme to become dominant. Runs with oscillations or long-lived transients, show the kind of predator-prey dynamics we expect, and illustrate that the search can sometimes get stuck in a 'meta-stable state', as it were. We know the only stable configuration is one in which iBOB dominates the population – any "oscillations", whatever the intervals between transient peaks, and whatever the number of those peaks must eventually give way to iBOB.

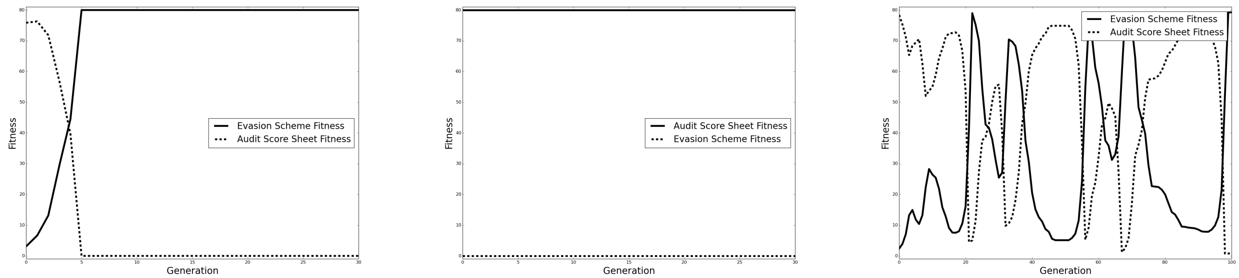
4.3.2 Experiment 2: Audit observables covering the tax evasion scheme

In this experiment we include an audit observable that can detect iBOB. Thus, iBOB should not be able to propagate through the tax evasion scheme population. Because the audit score sheets were previously unable to detect iBOB, the fitness of the tax evasion schemes would only oscillate until a single iBOB scheme was introduced into the population, at which point it would quickly propagate.

Figure 9b displays the fitnesses of both the tax evasion schemes and the audit score sheets from the best individual from each generation from one iteration. Since the audit points completely cover all transactions that can create large deductible loss, the fitness is always minimal for the evasion schemes and maximal for audit score sheets. We conclude that the observed co-evolutionary dynamics of SCOTE are consistent with what we expect if SCOTE is functioning correctly.

4.3.3 Experiment 3: Co-evolution when prioritizing audit observables

Our goal with this experiment was to generate sustained oscillatory dynamics, since we have shown in previous experiments that oscillations in tax evasion scheme fitness are possible for a short amount of time before converging to



(a) Audit observables not covering iBOB *(b) Audit observables covering iBOB* *(c) Prioritizing audit observables iBOB*

Figure 9: Example run for best fitness over generation during the evolution of tax evasion schemes for iBOB with SCOTE

equilibrium. This is a necessary step because an assumption underlying our model is that tax evasion schemes and audit scores sheets are engaged in a perpetual arms race process in which no global attractor exists. Because the audit score sheets were in Experiment 1 unable to detect iBOB, the fitness of the tax evasion schemes would only oscillate until a single iBOB scheme was introduced into the population, at which point it would quickly propagate.

To generate sustained oscillations, we augment the audit score sheets to assign the lowest audit point a value of zero, so that there will always be at least one scheme that is not detectable by the auditor. Our hypothesis is that once the population of audit score sheets begins to converge, a tax evasion scheme will evolve that utilizes the type of behavior that is currently not detectable by the majority of audit score sheets. The effective tax evasion scheme will propagate within its population until the audit score sheets gradually evolve to detect the now dominant behavior.

Figure 9c displays the fitnesses of both the tax evasion schemes and the audit scores from the best individual from each generation during a single iteration. In this scenario, since the audit observables cannot completely cover all the transactions that can create large deductible losses, the fitness oscillates between minimal for the evasion schemes and maximal for audit score sheets and vice versa.

There is at first a high level of fitness among tax evasion schemes across all runs, but the initial dominant scheme is quickly detected by the corresponding audit score sheet population, which decreases the overall fitness. Over time, new tax evasion schemes emerge in some of the runs that are initially not detectable by the corresponding audit score sheet population, which generates a rapid upward surge in tax evasion fitness. Conversely, the audit score sheets take a longer time to adapt to the new tax evasion scheme. The audit score sheets eventually evolve the audit points to detect the type of behavior that is present in the new dominant tax evasion schemes, but the process is more gradual. These results confirm our hypothesis that under the correct conditions, sustained oscillatory dynamics in the fitness of tax evasion schemes are possible.

5. CONCLUSIONS & FUTURE WORK

When the IRS pursues a tax evasion scheme and changes the tax law, the tax evasion schemes evolve and new ones appear. We showed that SCOTE generates oscillatory dynamics between tax evasion scheme effectiveness and the audit score sheets' ability to detect them. SCOTE is based on a network-based representation, entities are nodes with assets and edges are ownership relations between entities and transactions are transfers of assets between nodes, i.e. state changes of the network. With SCOTE, transactions are concurrently tested against a list of audit points and a tax evasion scheme respectively. The result of the simulation gives a tax evasion risk measure for the transactions and a likelihood of identification for the audit scores. We can both fine-tune the audit scores for existing tax evasion schemes and find efficient tax evasion schemes, given a list of audit points.

Our approach to tax non-compliance by using audit points we avoid legal complications compared to trying to explicitly change the legal code. Moreover, SCOTE does not require data and thus simplifies individual privacy concerns. SCOTE is a complement to data driven machine learning techniques and the features and data generated by it can be used to inform supervised and unsupervised learning methods. Finally, the model SCOTE is based on gives a readable and intuitive understanding of how transactions lead to tax non-compliance. Drawbacks are that currently SCOTE has a very simplified view of transactions, audit points and law.

Our next step is to increase the complexity of both the transactions that compose the evasion schemes and the types of activities that are detectable by the audit score sheets. Most important is to consult domain experts about some generated evasion schemes to see if they are novel. The validity of the approach can be tested on actual tax report data. Further, new evasion schemes can be implemented and sensitivity of SCOTE can be investigated. To automatically detect patterns that emerge from the tax evasion schemes and use them as audit scores will be another step.

6. APPENDIX

Grammatical Evolution (GE) is a version of the Genetic Algorithm with a variable length integer representation and a compressed form of indirect mapping using a grammar [14]. GE has an explicit mapping step (genotype-to-phenotype) and biases the search by changing the grammar, e.g. alter the search space size and reduce source code modification. The grammar rewrites the input (genotype) to the output (phenotype), as shown in Figure 10. Recursive rules in the grammar indicate that the search space (language) is bounded only by the length of the input (genome) used in rewriting.

In GE, the compressed form of the search space is rep-

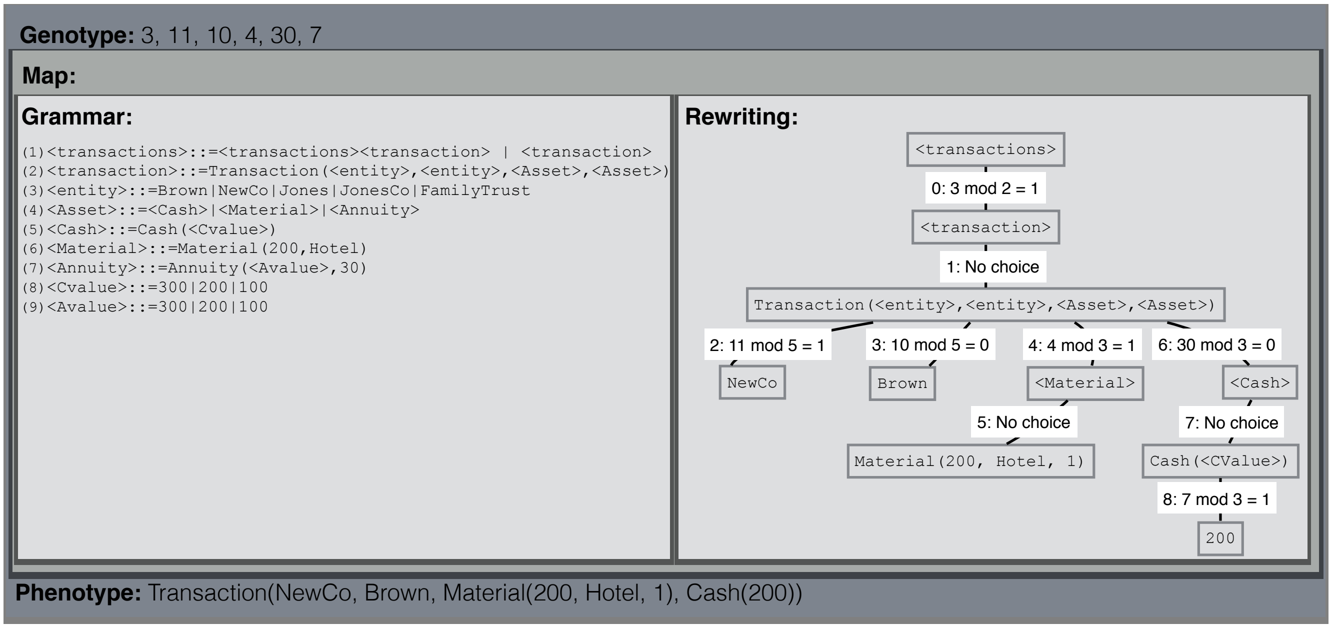


Figure 10: Example of how GE rewrites a list of integers (Genotype) into a list of transactions (Phenotype) with a BNF grammar.

resented by a Backus-Naur Form (BNF) grammar which defines the language that describes the possible output sentences. a BNF grammar has terminal symbols, non-terminal symbols, a start symbol and production rules for rewriting non-terminal symbols. The grammar is used in a generative approach and the production rules are applied to each non-terminal, beginning with the start symbol, until a complete program is formed. The list of integers (genotype) rewrites the start symbol into a sentence. An integer from the list of integers is used to choose a production rule from the current non-terminal symbol by taking the current integer input and the modulo of the current number of production choices. Each time a production from a rule with more than one production choice is selected to rewrite a non-terminal, the next integer is read and the system traverses the genome. The rewriting is complete when the sentence comprises only terminal symbols.

References

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

[2] Andreoni J, Erard B, Feinstein J (1998) Tax compliance. Journal of economic literature pp 818–860

[3] Bench-Capon T, Araszkiewicz M, Ashley K, Atkinson K, Bex F, Borges F, Bourcier D, Bourgine P, Conrad JG, Francesconi E, et al (2012) A history of ai and law in 50 papers: 25 years of the international conference on ai and law. Artificial Intelligence and Law 20(3):215–319

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

[5] Buchanan BG, Headrick TE (1970) Some speculation about artificial intelligence and legal reasoning. Stanford Law Review pp 40–62

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

[7] GAO (2014) `http://www.gao.gov/assets/670/663185.pdf`

[8] Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-wesley

[9] IRS (2014) Sales and trades of investment property. Http://www.irs.gov/publications/p550/ch04.html

[10] de Jong ED, Stanley KO, Wiegand RP (2007) Introductory tutorial on coevolution. In: GECCO (Companion), pp 3133–3157

[11] Lotzmann U, Möhring M, Troitzsch KG (2013) Simulating the emergence of norms in different scenarios. Artificial intelligence and law 21(1):109–138

[12] May LR (2012) Using link analysis to identify indirect and multi-tiered ownership structures. In: SOI Tax Stats - 2012 IRS-TPC Research Conference

[13] McCarty LT (1977) Reflections on" taxman": An experiment in artificial intelligence and legal reasoning. Harvard Law Review pp 837–893

[14] O'Neill M, Ryan C (2003) Grammatical evolution: evolutionary automatic programming in an arbitrary language, vol 4. Springer

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

[16] Rostain T, Regan MC (2014) Confidence Games: Lawyers, Accountants, and the Tax Shelter Industry, vol 1. MIT Press

[17] Sartor G, Rotolo A (2013) Ai and law. In: Agreement Technologies, Springer, pp 199–207

[18] Surden H (2014) Machine learning and law. Wash L Rev 89:87–217

[19] Warner G, Wijesinghe S, Marques U, Badar O, Rosen J, Hemberg E, O'Reilly UM (2014) Modeling tax evasion with genetic algorithms. Economics of Governance pp 1–14, DOI 10.1007/s10101-014-0152-7, URL `http://dx.doi.org/10.1007/s10101-014-0152-7`

[20] Wright Jr D (2013) Financial alchemy: How tax shelter promoters use financial products to bedevil the irs (and how the irs helps them)