

Detecting tax evasion: a co-evolutionary approach

Erik Hemberg¹ · Jacob Rosen² · Geoff Warner² ·
Sanith Wijesinghe² · Una-May O'Reilly¹

Published online: 28 April 2016
© Springer Science+Business Media Dordrecht 2016

Abstract We present an algorithm that can anticipate tax evasion by modeling the co-evolution of tax schemes with auditing policies. Malicious tax non-compliance, or evasion, accounts for billions of lost revenue each year. Unfortunately when tax administrators change the tax laws or auditing procedures to eliminate known fraudulent schemes another potentially more profitable scheme takes its place. Modeling both the tax schemes and auditing policies within a single framework can therefore provide major advantages. In particular we can explore the likely forms of tax schemes in response to changes in audit policies. This can serve as an early warning system to help focus enforcement efforts. In addition, the audit policies can be fine tuned to help improve tax scheme detection. We demonstrate our approach using the iBOB tax scheme and show it can capture the co-evolution between tax evasion and audit policy. Our experiments show the expected oscillatory behavior of a biological co-evolving system.

Keywords Tax evasion · Co-evolution · Grammatical evolution · Genetic algorithms · Auditing policy · Partnership tax

1 Introduction

The 2006 U.S. gross tax gap, i.e. the difference between the tax owed and the tax paid on time, was estimated at \$450 billion (IRS 2006). The Government Accountability Office (GAO) further estimates that \$91 billion of this tax gap can be attributed to income hidden in tax shelters composed of multiple “pass-through”

✉ Sanith Wijesinghe
sanith@mitre.org

¹ Massachusetts Institute of Technology, Computer Science and AI Lab, 32 Vassar Street, Cambridge, MA 02139, USA

² The MITRE Corporation, 1550 Westbranch Drive, McLean, VA 22102, USA

entities, such as partnerships, S corporations and trusts (GAO 2014a). Financial and legal enterprises scour the tax code in search of ambiguities in order to discover and promote abusive tax shelters. Such illegal tax avoidance strategies use complex transactions within networks of tax entities that are designed to reduce and obscure the tax liabilities for their individual shareholders. For the purpose of discussion here we categorize such sequences of transactions as 'tax evasion schemes' in contrast to legal tax avoidance strategies that adhere to the letter and intent of the tax code.

While tax auditors have historical examples of tax schemes to help guide examination efforts, tax shelter promoters often adapt their strategies as existing schemes are uncovered and/or 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 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" (Wright 2013). This is typical of the arms race between tax evaders and tax auditors. The significant challenges posed for enforcement efforts here have prompted recent congressional action to address some ambiguities in partnership audit and adjustment rules (Accountancy 2015).

There remains however significant challenges to enforcement efforts that arise from two primary sources. (a) *the complexity of the tax code*: tax law is not only qualitatively difficult to parse from a natural language perspective, it is quite interconnected (Li et al. 2015; Katz and Bommarito 2014) as can be quantified by the number of links between paragraphs. Furthermore calculation of certain tax quantities can be complicated and byzantine. For example, owners of a partnership need to adjust separate basis values and use one of two similar definitions of "built-in substantial income" as described by Internal Revenue Code (IRC) §734 and §743. (b) *dispersed, sensitive and obfuscated data*: tax reporting data is distributed and fraudsters purposely obscure their intentions. These individuals often obfuscate their schemes by using large hierarchies of entities, e.g. up to 100 tiers and 100,000 partners (GAO 2014b), providing as little information as possible and stalling reporting data. The auditors work with aggregated and dispersed tax data, e.g. on form 1065 (Schedule K-1)¹ many of which might be filed as separate paper attachments.

In this paper we describe a methodology that can help detect strategies for reducing tax liability by offsetting real gains in one part of a portfolio by creating artificial capital losses elsewhere, specifically those that utilize the complicated partnership tax law in Subchapter K of the IRC. While there are other strategies for abusive tax avoidance that involve related party agreements, services pricing, or state and local tax (SALT) jurisdictional items, we consider partnership tax for this initial analysis. The tax schemes here consist of sequences of transactions between entities in ownership networks that, when taken individually, appear compliant. However, when all transactions are combined they have no other purpose than to illegally mitigate tax liability and can potentially be labeled as tax evasion. Propitiously, it is possible to disallow tax evasion by one of many anti-abuse doctrines, e.g. the "economic substance" doctrine, which specifies that transactions

¹ <http://www.irs.gov/uac/Form-1065,-U.S.-Return-of-Partnership-Income>.

must contain both economic substance and a business purpose (Rostain and Regan 2014).

Tax auditors typically use protocols to rate whether an entity is suspicious enough to undergo a full audit (Andreoni et al. 1998). Such a full audit is a thorough investigation of the reported financial information to establish the correct tax liability (IRS 2014a). Our goal is to develop algorithms that can model the co-evolution of tax schemes as represented by *networks of transactions* with their corresponding *audit observables*.

Our approach to detecting tax evasion places us in the realm of AI research and legal reasoning that started in the 1970s (Buchanan and Headrick 1970). The field of AI and law (Sartor and Rotolo 2013; Bench-Capon et al. 2012) deals with simulation of norms and their emergence (Aubert and Müller 2013; Dechesne et al. 2013; Lotzmann et al. 2013). Pioneering work regarding taxes occurs in “Taxman” (McCarty 1977) and more recently in tax non-compliance modeling of real estate policy (Boer and Engers 2013). Machine learning has recently emerged as a means of detecting suspicious transactions. It relies upon known historical examples labeled to be compliant or non-compliant (DeBarr and Eyler-Walker 2006). The requirement of data labeled in this manner for training (Surden 2014; Ngai et al. 2011) limits the use of detection methods to identify already known cases of abuse. Alternatively unsupervised techniques can be used to cluster transactions based on their common resemblance. This requires a measure of similarity between different transaction types as well as knowledge to classify a cluster as abusive. In contrast our model does not require taxpayer data (it is similar to a minmax search in a game of chess) and has the added advantage of alleviating any data privacy concerns. The current paper specifically extends the prior efforts (Warner et al. 2014; Hemberg et al. 2015; Rosen et al. 2015) that describe tax evasion detection through evolutionary search.

We proceed as follows in describing how our method Simulating Tax Evasion and Law Through Heuristics (STEALTH) can be used to anticipate abusive tax avoidance schemes. In Sect. 2 we begin with a background of the partnership tax law fundamentals used in STEALTH and the related tax modeling literature. The detailed methodology of how co-evolutionary heuristics can be used to search-and-score tax schemes and audit plans is described in Sect. 3. In Sect. 4 we demonstrate the capability of STEALTH to identify an artificial basis step-up tax scheme called iBOB. Conclusions and future work are described in Sect. 5.

2 Background

We begin with a background of partnership tax law fundamentals and summarize the existing literature on modeling tax evasion.

2.1 Asset taxation

We focus on taxes incurred during the sale and trade of investment property. While an extensive set of rules apply to the calculation of taxes during such transactions (IRS 2014b), we summarize below only the terms of most interest

for our purposes: (a) *Basis* The original investment to acquire an asset, often times its purchase price. Basis is used as the baseline against which to determine capital gain or loss on asset disposition. Basis is also adjusted by any associated liabilities. (b) *Fair Market Value (FMV)* The value of an asset at a given time. (c) *Capital Gain* A capital gain results when an asset is disposed at a price higher than its adjusted basis. Depending on how long the asset has been held, the capital gain is classified as either long term (greater than 1 year) or short term (less than 1 year). Short term capital gain is taxed at ordinary income rates. Long term capital gain is taxed at a typically lower rate and varies with an individual's filing status. (d) *Capital Loss* A capital loss arises when an asset is disposed at price lower than its adjusted basis. Similar to capital gains, capital losses are also classified as short or long term. If capital losses exceed capital gains, the excess can be rolled across multiple years to match capital gains in the future.

We elaborate some basic examples of basis, its adjustment and impact on taxable gain or loss. An example of the use of basis to calculate capital gain is described below:

1. Purchase: 100 shares of IBM bought at \$200 + \$100 commission. Basis is \$20,100 (\$201/share).
2. Sale: 100 shares of IBM sold at \$210 + \$100 commission.
3. Revenue: \$20,900 (\$209/share)
4. Capital gain: $\$20,900 - \$20,100 = \$800$.

A taxpayer that incurs a loss from the sale of an asset can allocate that loss across the rest of his portfolio to lower his tax liability from other capital gains that may have been incurred. For example:

1. Taxpayer A owns a house with an FMV of \$200k, in which taxpayer A has a \$120k basis.
2. When taxpayer A sells the house, a \$80k capital gain is incurred.
3. Taxpayer A also owns a car with an FMV of \$10k, in which taxpayer A has a \$30k basis.
4. Upon sale of the car, taxpayer A only has to pay tax on \$60k because the capital loss (\$20k) from step 3 partially cancels the gain (\$80k) from the house sale.

In some cases basis can be readjusted, for example during stock inheritance:

1. 100 shares IBM stock originally purchased for \$4 in 1965
2. Inherited in 2013
3. Basis is "stepped-up" to market value on date of inheritance
4. Capital gain or loss is calculated as before on sale using the adjusted basis

2.1.1 Partnerships and carryover bases

Partnerships are legal tax entities that are governed by rules defined in subchapter K, §701–777 of the IRC. While partnerships are required to file a tax return

detailing their economic activity, they are not directly taxed. Instead, any income/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 separately for both the *inside basis* – the part of the assets' adjusted basis that is attributable to each partner, and the *outside basis* – the basis each partner holds in the partnership interest. Most times, the inside and outside basis will match but can begin to differ when partnership interests are transferred. In order to correct this mismatch, partnerships are allowed to adjust the inside basis of their assets by making a special election known as the §754 election. This basis adjustment can be made in the case of several scenarios, the two most common being: (a) sale of a partnership interest (defined in IRC §743), and/or (b) distribution of property (defined in IRC §734). Income, gain or loss calculation using adjusted basis can become complicated when individuals form partnerships by pooling resources (cash, property, labor) to conduct joint businesses. Individuals have been found to disguise gains by conducting transactions across multiple tiers of nested partnerships while claiming bogus §754 elections (Wright 2013).

2.2 Tax evasion modeling

Tax evasion can be considered a gamble like any other investment involving risk and uncertainty (Allingham and Sandmo 1972). 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 actors or agents (Bloomquist 2006). For example, in one study agents are either honest, imitative or free riders and a GA is used to update the population of agents' use of a utility function that determines the tax paying behavior of the agent (Mittone and Patelli 2000). Tax compliance has also been studied as an evolutionary coordination game (Bloomquist 2011). Another study considered evolution of tax evasion (Lipatov 2003) by an agent based model using a game theoretic approach. Cyclical behavior in compliance is seen here when the audit probability is adjusted, but no guidance is provided on how to increase compliance. In addition, attempts have been made to investigate how psychological motives vary with different audit techniques (Davis et al. 2003; Hokamp and Pickhardt 2010; Hokamp and Seibold 2014; Pickhardt and Prinz 2014). Econophysics models from statistical mechanics have also been considered, e.g. Ising models have been used to model different social behavior and investigate thresholds for efficient audits (Zaklan et al. 2009; Pickhardt and Seibold 2014; Zaklan et al. 2008). More recently, agent based models of tax compliance with varying social network structure have been analyzed. For a given enforcement regime, an environment with limited knowledge of neighbor payoffs appears to lead to higher levels of aggregate compliance than when agents are aware of neighbor strategy payoffs (Korobow et al. 2007). In addition, the effects of network topologies in the propagation of evasive behavior is found important for tax compliance (Andrei et al. 2013). Legal modeling approach of financial fraud was set up using Wigmore charts, a graphical method for legal evidence, and ontologies (Kingston et al. 2004).

Statistical machine learning techniques e.g. Decision Trees, Neural Networks, Logistic Regression, Support Vector Machines, both supervised and unsupervised, have also been used for fraud detections (Kallio and Back 2011; Bonchi et al. 1999; Jaideep and Bjorklund 2009). Notable work here by (DeBarr and Eycler-Walker 2006) used Support Vector Machines to detect tax shelters. The kernel based analysis used here identifies groups of taxpayers who appear to be participating in a tax shelter promoted by a common financial advisor. This analysis task requires estimating risk, a weighted combination of both the likelihood of abuse and the potential revenue losses. It should also be noted that statistical machine learning is applied to e-Discovery (Oard and Webber 2013) in order to retrieve similar documents.

Unlike in prior research, the STEALTH approach described here considers a *tax evasion scheme* as the unit of interest rather than an individual agents behavior. In STEALTH there are no pre-defined labels of tax evasion schemes or similarity metrics, and no modeling of assumed psychological behavior. There is no need for data, labeled or unlabeled. Instead a scoring function is derived that is able to rank the 'fitness' of different tax schemes subject to varying audit risk. We investigate how to reveal unknown schemes starting at the discrete transaction level. The research gap STEALTH contributes to here is the quantitative modeling of partnership tax law and the representation of partnership transactions and corresponding audit risk. These topics are elaborated upon next.

3 STEALTH : a method for identifying tax schemes

The STEALTH approach to the challenges of tax evasion detection is characterized across the multiple levels shown in Fig. 1: (a) the conceptual level with tax evaders and auditors, (b) the model level of the tax ecosystem, with the representation of tax schemes and audit score sheets, here we use decision tree rules to represent the tax law, (c) the simulation level that processes a transaction sequence and outputs deductible loss and audit scores, and (d) the optimization level that searches transaction sequences based on audit risk. These levels are described in more detail

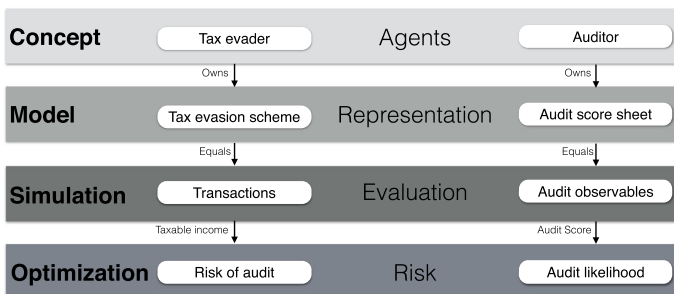


Fig. 1 Description of levels in STEALTH

below. For reference, Table 1 provides an overview of some important terms used in STEALTH . For additional formalism of these levels see Appendix 6.

3.1 STEALTH model of tax evasion

We break down the tax ecosystem 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. In broad terms, tax laws govern interactions between entities and specify any resulting tax liability that might occur as a result of a transaction. In practice adherence to tax laws is verified by the use of compliance audits. Figure 2 indicates how these activities are mapped and sequenced in STEALTH . This flow is described further below.

3.1.1 Tax transactions and ownership networks

We represent partnerships and asset exchanges using an *ownership network*. 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, each of which transfers an asset from one entity to another entity. Each transaction 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. As an example consider a tax ecosystem with four entities, two taxpayers (Taxpayer A and B) and two partnerships (Partnership P1 and P2) as shown in Fig. 3a. The nodes in the network are entities, arrows are edges for ownership relations and the dotted lines represent transactions. Taxpayer B buys a share in Partnership P1 from Taxpayer A with cash. This transaction 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 the Taxpayer B node, the Taxpayer A node is updated to show the cash in its portfolio. In addition, the transaction results in income (\$40k) for

Table 1 STEALTH glossary

	Tax evasion view	Tax auditing view
Agent	Tax evader	Auditor
Representation	Tax scheme	Audit score sheet
STEALTH input	Transaction sequences and ownership network	Audit score sheet
STEALTH output	Deductible loss (d_i)	Audit score (s)
Fitness function	$d_i(1 - s)$	$-d_i(1 - s)$
Audit score	Risk of being audited	Likelihood of auditing
Objective	Minimize audit likelihood and maximize deductible loss	Maximize likelihood of auditing a network of transactions that generate high deductible loss

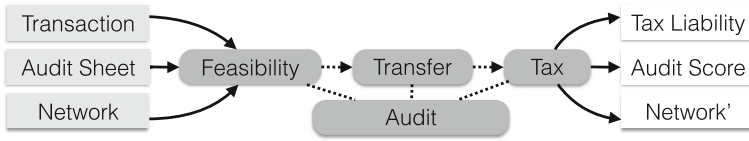


Fig. 2 STEALTH tax ecosystem model transactions

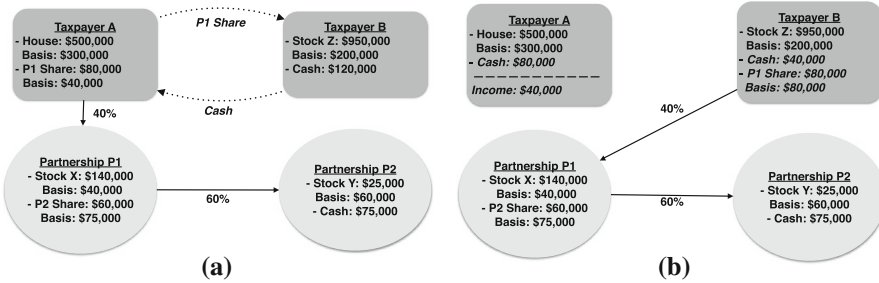


Fig. 3 Example of transaction with a network representation showing relationship edges and network state transition via a transaction. **a** Initial state and transaction. **b** Network state after transaction

Taxpayer A as the basis of the partnership share (\$40k) is lower than the price (\$80k) that Taxpayer B paid for the asset.

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 types of entities by introducing more nodes in the network and similarly we can introduce more diversity within nodes by having different types of assets.

The edges between nodes in the ownership network describe relationships in enterprise structures. These may consist of parent-child subsidiary relationships, spousal or family relationships or nested ownerships of entities (May 2012). For example, as shown in Fig. 3, Taxpayer A owns a 4 % share of a partnership P1. Similarly, partnership P1 has a 60 % share in partnership P2.

3.1.2 Integrating the tax law

Several actions are required to execute a transaction: (a) transaction feasibility checks, (b) asset transfers within the ownership network and, (c) transaction tax calculations. The tax law is imposed at each of these steps as follows:

Feasibility of a transaction Given a transaction consists of one action transferring an asset from one entity to another entity, and another action transferring an asset in the other direction, the tax law 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 and can transfer it and (c) if the receiving entity is allowed to receive the asset.

Transfer of assets Evaluate rules regarding the transfer of assets, e.g. determine how the basis of an underlying asset needs to be adjusted. A simplified decision tree rule to evaluate basis changes due to asset transfer is shown in Fig. 4.

Calculate tax Check rules regarding the tax impact of a transaction, This is implemented as a decision tree rule as shown in Fig. 5.

STEALTH can be extended not only by adding new entities and assets to the tax ecosystem, but by also adding/modifying tax rules. This requires alteration to some or all three parts of the transaction transfer actions.

3.1.3 Audit score sheets

In STEALTH an audit is a procedure that examines a sequence of transactions to help identify suspicious events. Audits play two roles in STEALTH . First we use

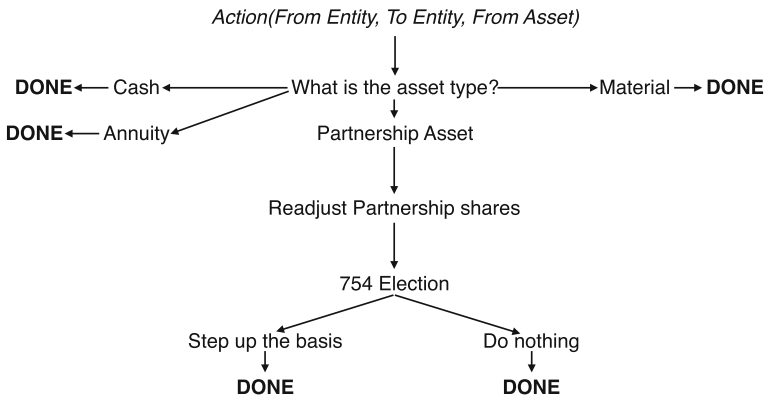


Fig. 4 A decision tree rule to evaluate asset basis changes

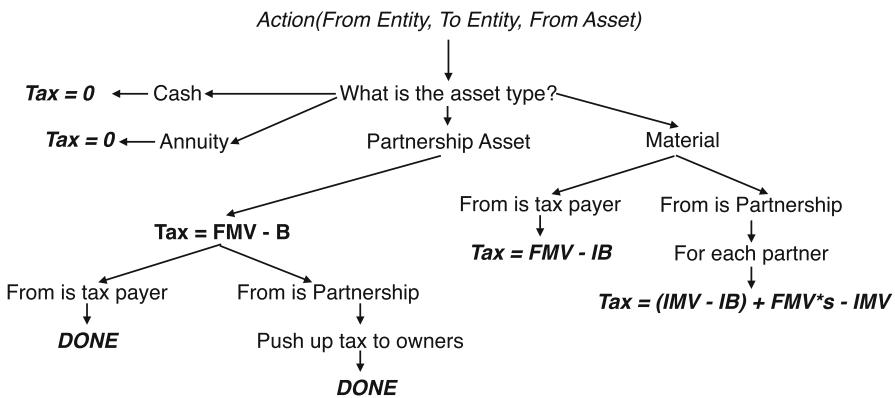


Fig. 5 A decision tree rule that shows the tax calculation on an asset transfer

them to help direct and co-adapt transaction sequences towards non-compliance (see Sect. 3.4). Second, they are used to quantify the degree of evasion.

In addition to amendments to the IRC, 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. Audits in STEALTH are modeled based on this public information. E.g. in 2004 the IRC §743 (a) was altered 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.

This amendment is captured in STEALTH using the following *observable* events: (a) the sale of a partnership interest in exchange for a *taxable* asset, (b) the partnership whose shares are being transferred has not made a §754 election, and (c) the seller's basis with respect to the non-cash assets owned by the partnership exceeds their FMV by more than \$250,000, the threshold for substantial built-in loss.

To represent audits in STEALTH we use a list of audit points (weights), corresponding to all observable events that can occur when a set 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. In order to mirror the limited resources available for auditing we also constrain the sum of audit points to equal one.

The audit score associated with an audit score sheet, is defined as the sum of all of the audit points present in a sequence of transactions, multiplied by their

Table 2 Each row has three columns with (1) the type of observable corresponding to the three characterized observables from the IRS notice, (2) the associated audit point and (3) the number of times it occurs in a list of transactions

Observable	Points	Frequency
1	Point ₁	Frequency ₁
2	Point ₂	Frequency ₂
3	Point ₃	Frequency ₃
1 ∪ 2	Point _{1∪2}	Frequency _{1∪2}
1 ∪ 3	Point _{1∪3}	Frequency _{1∪3}
2 ∪ 3	Point _{2∪3}	Frequency _{2∪3}
1 ∪ 2 ∪ 3	Point _{1∪2∪3}	Frequency _{1∪2∪3}

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 2. One can imagine a hypothetical auditor going through a sequence of transactions and incrementing the frequency in the far right column whenever each type of event is observed.

Using this formulation, we interpret an audit score as the *likelihood* that a sequence of transactions will be audited. That is, the more types of transactions associated with high levels of suspicion there are in a sequence of transactions, the higher the audit score will be.

The events on the audit score sheet can range from basic facts about a transaction, such as whether a material asset is being exchanged, to more complex aspects of the model state, such as ownership linkages between multiple entities. Note 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 auditor. For example, if a taxpayer purchases a share in a partnership for cash, STEALTH will process that as a transaction involving a partnership asset, as well as tracking all parties involved in the transaction.

The usefulness of the audit point representation is not only to suggest important qualifiers to the auditor, but to evaluate how hypothetical auditing behavior effects future schemes. Thus, even if an event is completely unobservable by the auditor, it can still be useful for STEALTH .

3.2 Simulation with tax ecosystem model

At the core of STEALTH is a simulation of the tax ecosystem model, shown in Fig. 6. The simulation first initializes the tax ecosystem, which is a set of interconnected taxpayers and partnerships, and takes as input a transaction sequence and an audit score sheet with associated points for each observable.

Each transaction needs to be analyzed for legality/feasibility before it can be executed. A simple check is to validate whether or not an entity has an asset before

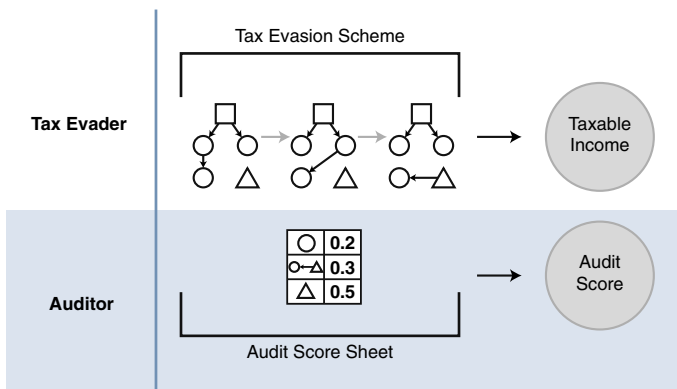


Fig. 6 STEALTH tax ecosystem simulator

it can be transferred. Feasibility checks are divided into two broad categories namely *impossible transactions* and *economically unviable transactions*. Once validated the model can perform the actual transactions between entities and calculate the tax/deductible loss associated with the feasible and taxable transactions. The new state of the ownership network is also then updated.

Simultaneously, each transaction sequence is assigned an audit score by multiplying the observed financial activity against the prescribed audit score sheet. The audit points can specify patterns that represent combinations of financial activity that may indicate abuse to an auditor, such as when a certain type of transaction occurs between two linked entities.

3.3 Optimization architecture

The optimization in STEALTH is orchestrated by the adversarial relationship between tax schemes and audits. STEALTH performs co-evolution of a population of tax 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 Fig. 7. That is, each tax scheme “selects” some audit score sheets to calculate its fitness against and vice-versa.

3.3.1 Co-evolution as a search heuristic

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 ambiguities. The auditor and the tax evaders are *co-evolving* as interacting species, much like foxes and hares. The auditor searches for beneficiaries of abusive tax shelters while the beneficiaries seek

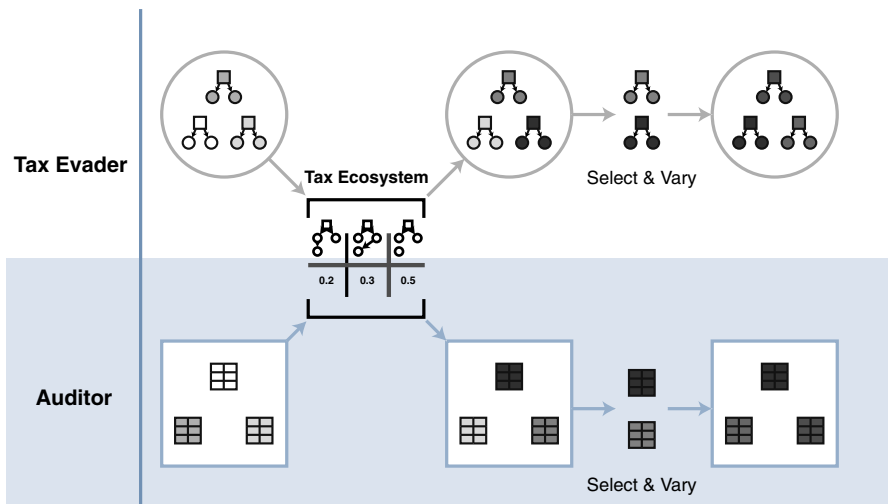


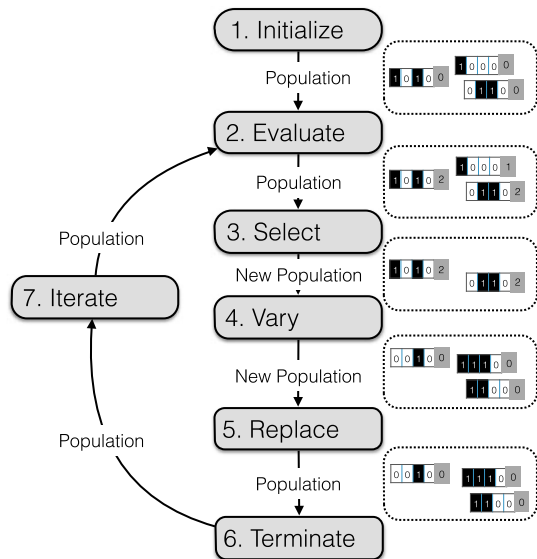
Fig. 7 Concurrent optimization of high likelihood audit scores and low risk tax schemes

to evade the auditor. At its core, the overall dynamics of the system reflect a constantly transitioning series of complementary adjustments, with each predator/prey seeking to bring advantage to the predator/prey under adjustment. A principle that arguably explains the constant evolutionary arms race between the iterative tax shelter pattern and IRS reaction is the “Red Queen Principle” (Van 1973): a species (in this case the IRS or the tax evaders) must continually adapt to maintain its relative fitness among the species it co-evolves with.

One algorithm which is used for co-evolutionary modeling is the Genetic Algorithm (GA) (Goldberg 1989). It is a stochastic, adaptive learning heuristic which searches and scores a number of solutions (individuals) 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 “individuals” are represented as fixed length bit strings and evaluated for fitness, good ones are selected as parents, and new ones are created by inheritance with variation as illustrated in Fig. 8.

The GA performs a search on networks 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. The transactions consist of a list of Java interpretable objects that are input to a tax ecosystem model to calculate the resulting taxable gain. There are two populations of individual solutions: sequences of transactions and sets of audit points. 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. Simultaneous changes of the networks of transactions and audit points are shown in Fig. 7. In order to generate both a final tax value and an audit score, the tax simulator in

Fig. 8 Overview of the flow of a Genetic Algorithm



STEALTH must take both a network of transactions *and* audit points as inputs. As each transaction in the network is executed, the tax simulator updates the audit score, given the audit points associated with the observed events.

3.4 Rewarding co-evolution

The reward assigned to individual solutions in co-evolutionary search can be quantified with the help of a “fitness” function. From the perspective of a tax evader, two terms effect a tax scheme’s fitness function. First, the fitness should be positively correlated with the *deductible loss*, in other words a tax scheme is only effective if it results in a high deductible loss. The second term in the function represents the likelihood of the audit disallowing the tax benefits gained from the scheme. This term takes into account the likelihood of an audit (the *audit score*) and the amount of tax that is evaded. Ideally the audit score should be low to reduce the likelihood of an audit.

The objective of auditors is essentially the opposite. They seek to maximize the likelihood of audit for a sequence of transactions with high deductible loss, relative to other transaction sequences observed in the population. By considering fitness in this manner, we are able to take into account both the effectiveness of tax schemes from a purely tax perspective, as well as from a risk perspective.

Further details of the fitness functions are described in Appendix 6. In Sect. 4 we investigate how STEALTH concurrently searches for co-adapting tax schemes and audit scores.

4 Experiments with STEALTH

We demonstrate how tax schemes and audit scores co-evolve in STEALTH by using a known artificial basis step-up tax scheme. The aim of the experiments is to demonstrate that STEALTH can search simultaneously for tax schemes and audit scores as they mutually adapt to one another over time. The results demonstrate that STEALTH has the required fundamental components and processes for detecting and anticipating tax evasion.

4.1 iBOB : an artificial basis step-up scheme

For the purposes of these experiments, we consider a particular known tax scheme called Installment Sale Bogus Optional Basis (iBOB). In iBOB , a devious taxpayer arranges a network of transactions designed to reduce his deductible loss 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 §743 (b) of the tax code. In this way, he manages to eliminate taxable gain while ostensibly remaining within the bounds of the tax law (GAO 2013).

The sequence of transactions for the iBOB scheme are enumerated below and also shown graphically in Fig. 9.

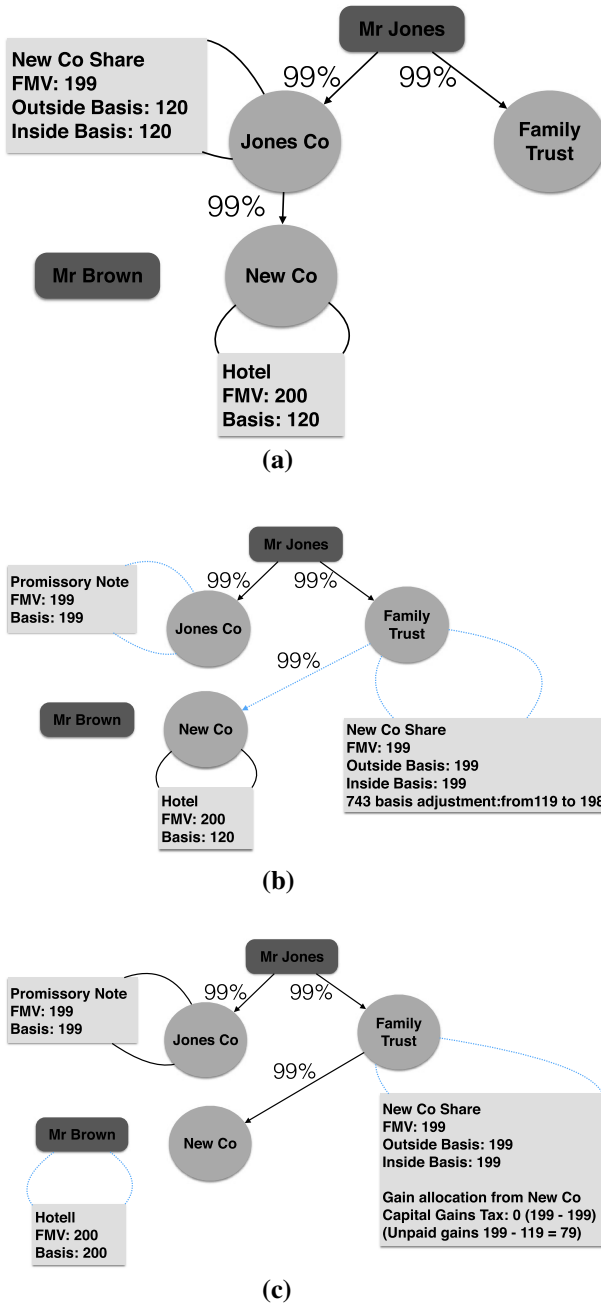


Fig. 9 The steps in the iBOB abusive tax avoidance scheme. The basis of an asset is artificially stepped up and tax is avoided by using “pass-through” entities. **a** iBOB step 1. **b** iBOB step 2. **c** iBOB step 3

1. 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 step 1 (Fig. 9a), 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. $\$198 - \$119 = \$79$. Mr. Jones can evade or indefinitely defer this tax by artificially stepping up the inside basis of the hotel to \$198.
2. In the next step (Fig. 9b), we see that FamilyTrust, which Mr. Jones controls, decides to buy JonesCo's partnership share in NewCo for a promissory note with a face value of \$198. 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 can now step up its share of inside basis in the hotel to \$198.
3. When NewCo sells the Hotel to Mr. Brown for \$200 (Fig. 9b), 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 STEALTH experiments

To run STEALTH we need to specify the initial ownership network of tax entities, the grammar for transactions, the audit score sheet and the co-evolutionary search parameters.

We initialize a network with two tax filers, 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 (see Appendix 6) used by STEALTH is detailed in Fig. 10. The first recursive rule in the grammar shows that the search space (language) is bounded only by the length of the input (genome) used to map integers to transactions. We also note that the search space can be increased and biased by altering the structure of the grammar.

In addition to iBOB, we note two additional patterns of transaction activity that can result in zero immediate tax liability for Mr Jones. The first of these involves the

```

<transactions> ::= <transactions><transaction> | <transaction>
<transaction> ::= Transaction(<entity>, <entity>, <Asset>, <Asset>)
<entity> ::= Brown | NewCo | Jones | JonesCo | FamilyTrust
<Asset> ::= <Cash> | <Material> | <Annuity> | <PartnershipAsset>
<Cash> ::= Cash(<Cvalue>)
<Material> ::= Material(200, Hotel, 1)
<Annuity> ::= Annuity(<Avalue>, 30)
<PartnershipAsset> ::= PartnershipAsset(99, <Pname>)
<Share> ::= Share(<Sshare>)
<Cvalue> ::= 200 | 300 | 100
<Avalue> ::= 200 | 300 | 100
<Pname> ::= NewCo | JonesCo | FamilyTrust
<Sshare> ::= 30 | 50 | 20

```

Fig. 10 iBOB BNF grammar for STEALTH

transfer of a partnership interest between two “linked” entities in the same enterprise structure, 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 (or JonesCo and NewCo), and “doubly linked” transactions, as may occur between Mr Jones and NewCo. These types of transactions result in zero immediate tax liability for all parties, but would almost certainly be audited. The second such transaction involves the use of Annuities such as promissory notes that are taxed only at the time of payment. As with “linked” transactions, defaulting on Annuity payments is nominally legal and results in zero tax liability but can be very suspicious for auditors.

4.3 Results: co-evolution of iBOB

We conducted a number of experiments to verify that the co-evolutionary dynamics of STEALTH are consistent and suffice to find existing tax schemes. We expect that as these schemes evolve to accommodate existing audit priorities, the audit points will themselves evolve to detect the new abusive behavior. This should result in either convergent or oscillatory dynamics.

Experiment 1 shows that when there is no audit observable that can capture the tax scheme, the non observable tax scheme population converges to the tax scheme that cannot be observed. Experiment 2 shows that if the tax scheme is observable, then the audit points will converge to detect it. Finally in Experiment 3 we add an extra constraint to the audit observables such that exactly one of the audit points must be zero. This constraint aids the representation of limited audit resources.

In co-evolution, we are interested in not only the end-result tax scheme and audit score distribution but also the dynamics that emerge between the two. Evaluating the fitness levels of both populations and subsets of those populations over each sequential generation can more accurately display the effects that the evolution of one population has on the other. This is an important goal when trying to anticipate tax schemes and audit points.

For the experiments considered here, STEALTH represents audit scores as the sum of four audit points between 0 and 1. The initial average distribution of audit scores in the population for each experiment is shown in Table 3. The value of each

Table 3 Each row lists an audit observable. Each column lists an experiment and the corresponding initial average distribution of audit points in the population

Audit observable	Experiment 1	Experiment 2	Experiment 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

Table 4 Parameters for the 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 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

audit point can be thought of as the relative importance of the associated transaction to the auditor.

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 enumerated in Table 4.

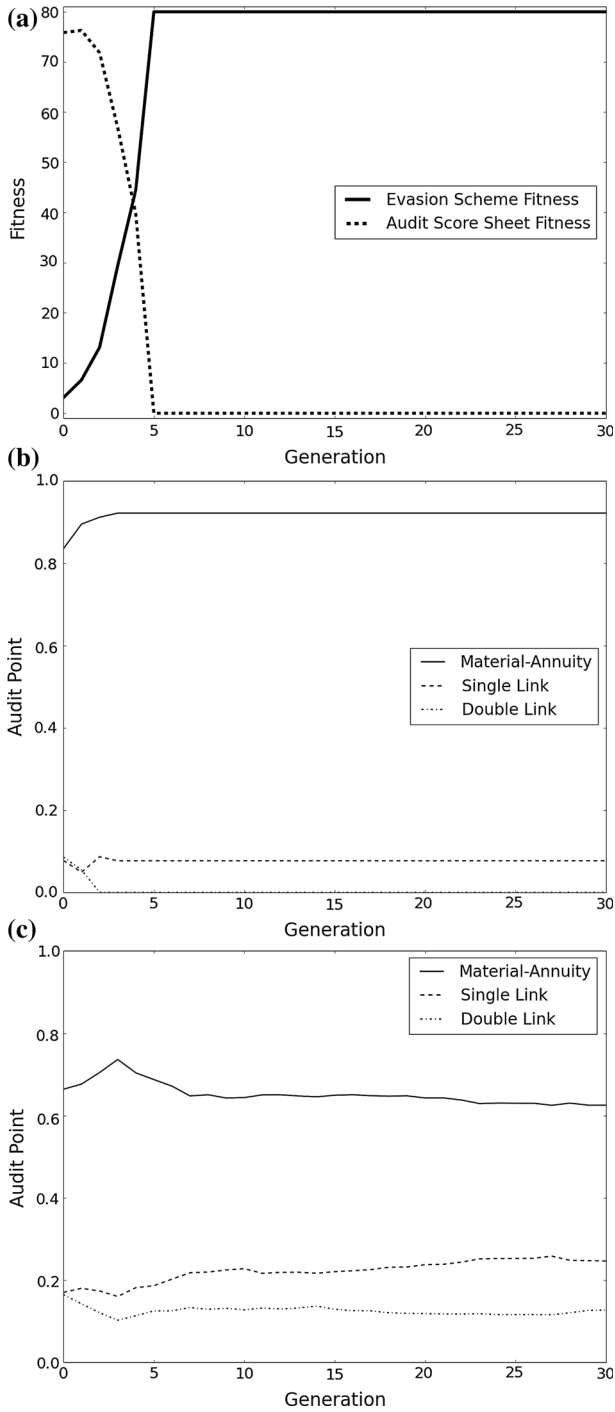
4.3.1 Experiment 1: audit observables that do not detect iBOB

For this experiment we purposefully left out audit points that can detect iBOB . Thus, once iBOB is found it will propagate through the tax scheme population on the merit of being the best tax scheme that is unauditible.

Figure 11 shows plots of tax scheme and audit score sheet fitness. Figure 11a shows the fitnesses of both the tax schemes and the audit scores for the best individual from each generation for a single iteration. Note, while iBOB was found in only 34 % of the iterations here, we expect most iterations to converge to iBOB as the number of generations increases. The average audit point values displayed in Fig. 11b show the variation in the points over time. Note here the audit points are unchanged and in equilibrium once iBOB has been found. The average audit points of the best solutions for all the iterations are shown in Fig. 11c, where we see that transactions that exchange a material for an annuity are assigned a higher audit point.

Material-annuity transactions have a significantly higher audit point because they occur more frequently than the other two transactions. That is, any transaction in which the Hotel is exchanged for an annuity mitigates all of the taxable gain on the ultimate sale of the hotel because annuities are non-taxable. Furthermore, a double link transaction requires that a material-annuity transaction takes place because Jones has to purchase the Hotel from NewCo with an annuity. Thus, the likelihood of a tax scheme involving a material-annuity transaction is higher than the likelihood of a single or double linked transaction based scheme. This results in a higher average audit point assigned to material-annuity transactions because it is the most common way to mitigate taxable gain in our example.

A clear pattern emerges when iBOB is evolved: initially, the pool of tax schemes gravitates towards a sequence of transactions that contains suspicious activity,



◀ **Fig. 11** Evolution of iBOB in STEALTH Experiment 1. **a** Best fitness for one run. **b** Distribution of audit points for one run. **c** Distribution of audit points averaged over runs

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.

Two distinct metastable 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 audit score fitness to converge to its maximum. Alternatively, the pools of both tax 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. Many runs show oscillations or long-lived transients, as these show the kind of predator-prey dynamics we expect, and illustrate that the search can sometimes get stuck in a ‘metastable state’.

The transient behaviors that occur before the simulation ultimately settles into iBOB. The transients can exhibit oscillations into unstable or weakly metastable equilibria, and can sometimes get stuck for a while in one of these equilibria where the scheme fitness is low and the audit fitness is high. 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 (1,2,3...) must eventually give way to iBOB.

4.3.2 Experiment 2: audit observables that can detect iBOB

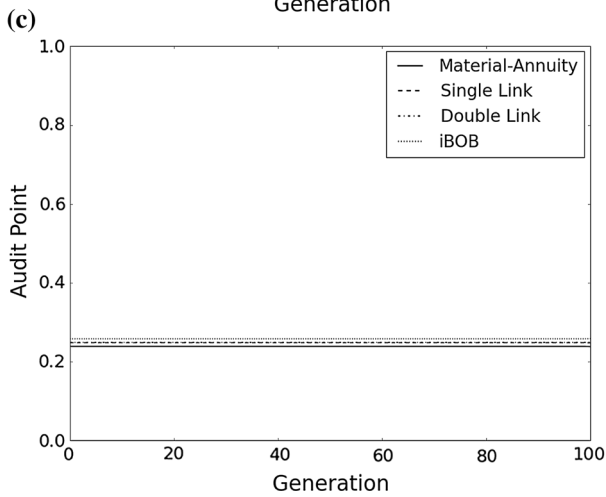
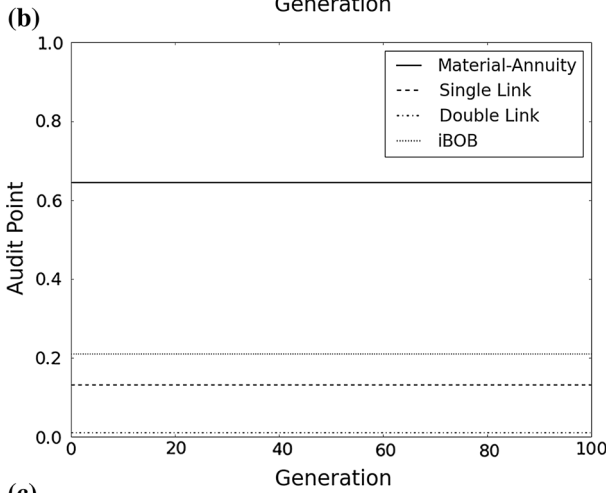
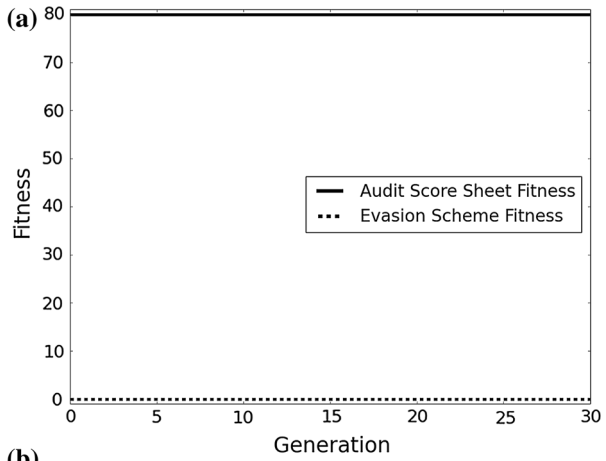
In this experiment we include an audit point that can detect iBOB. Thus, iBOB should not be able to propagate through the tax scheme population. Because the audit score sheets were previously unable to detect iBOB, the fitness of the tax schemes would only oscillate until a single iBOB scheme was introduced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

Figure 12a displays the fitnesses of both the tax 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 recognizable loss, the fitness is always minimal for the tax schemes and maximal for audit score sheets. The corresponding audit points for the iteration are all constant as shown in Fig. 12b. Furthermore, Fig. 12c shows that the average audit points of the best individuals over all the runs corresponds to the expected values at the initial state.

We conclude that the observed co-evolutionary dynamics of STEALTH are consistent with expectations for this example.

4.3.3 Experiment 3: sustained oscillatory dynamics of fitness values in STEALTH

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



◀ **Fig. 12** Evolution of iBOB in STEALTH experiment 2. **a** Best fitness for one run. **b** Distribution of audit points for one run. **c** Distribution of audit points averaged over runs

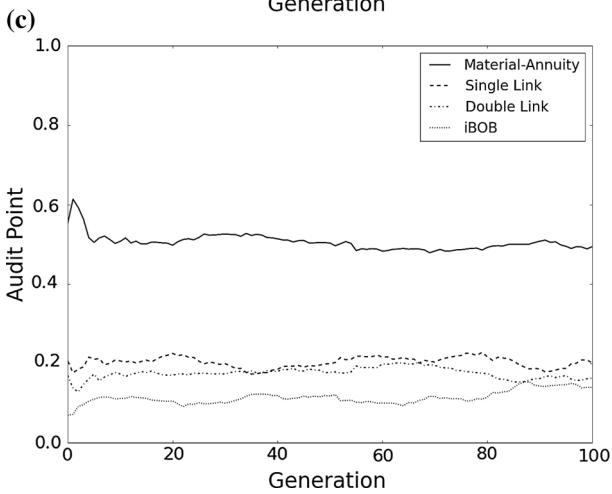
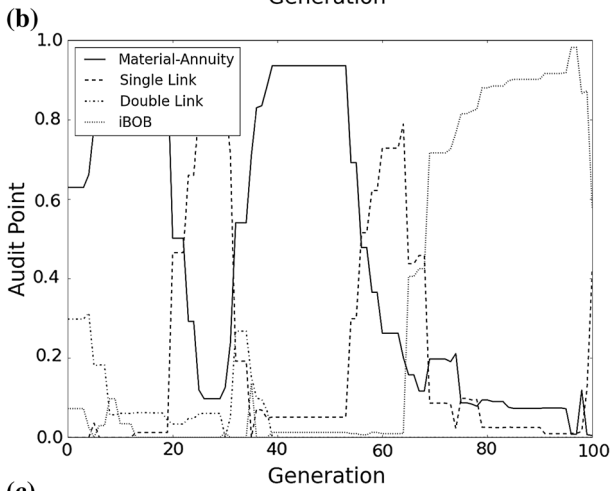
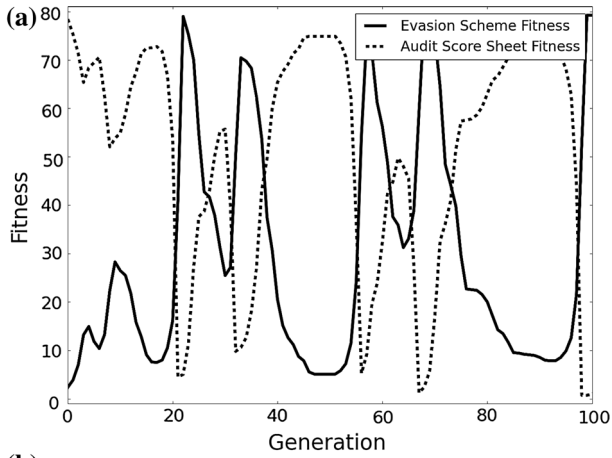
equilibrium. This is a necessary step because a primary assumption underlying our model is that tax schemes and audit score sheets are engaged in a perpetual co-evolutionary process in which no global attractor exists. Because the audit score sheets were unable to detect iBOB in Experiment 1, the fitness of the tax schemes would only oscillate until a single iBOB scheme was introduced into the population, at which point it would quickly propagate. At the same time, simply allowing the audit score sheets to detect iBOB would result in rapidly convergent dynamics, as demonstrated in Experiment 2.

To generate sustaining 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 scheme will evolve that utilizes the type of behavior that is currently not detectable by the majority of audit score sheets. The effective tax scheme will propagate within its population until the audit score sheets gradually evolve to detect the now dominant behavior.

Figure 13a displays the fitnesses of both the tax schemes and the audit scores from the best individual from each generation during a single iteration. In this scenario, since the audit points cannot completely cover all the transactions that can create large deductible loss the fitness oscillates between minimal for the tax schemes and maximal for audit score sheets and vice versa. The audit points corresponding to this iteration also oscillate as shown in Fig. 13b. In Fig. 13c we see for the reasons listed in Sect. 4.3.1 that the highest average audit points of the best individuals over all the iterations are for transactions involving annuities.

In Fig. 13, there is at first a high level of fitness among tax 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 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 scheme fitness. At closer inspection we see that 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 sequences. The audit score sheets eventually evolve to detect the type of behavior that is present in the new dominant tax 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 schemes are possible.

While the experiments were designed to generate oscillatory behavior, the results are promising because they show realistic dynamics between the tax schemes and the corresponding auditing priorities. Specifically we can see that once a single new tax scheme emerges that is not currently detectable by the auditor, it propagates



◀ **Fig. 13** Evolution of iBOB in STEALTH experiment 3. **a** Best fitness for one run. **b** Distribution of audit points for one run. **c** Distribution of audit points averaged over runs

throughout the population very quickly, as evident by the steep upward slope in the average scheme fitness plot. Conversely, the audit score sheets take a longer time to adapt to the new tax scheme. This dynamic mirrors the reality of actual audits.

4.3.4 *HiBOB: iBOB with hierarchical enterprise structures*

In an attempt to find potentially new evasion schemes, we added a hierarchy of multiple partnerships to the SCOTE tax ecosystem with the initial ownership structure shown in Fig. 14a. Here, NewCo owns 99 % of FunCo, who in turn owns

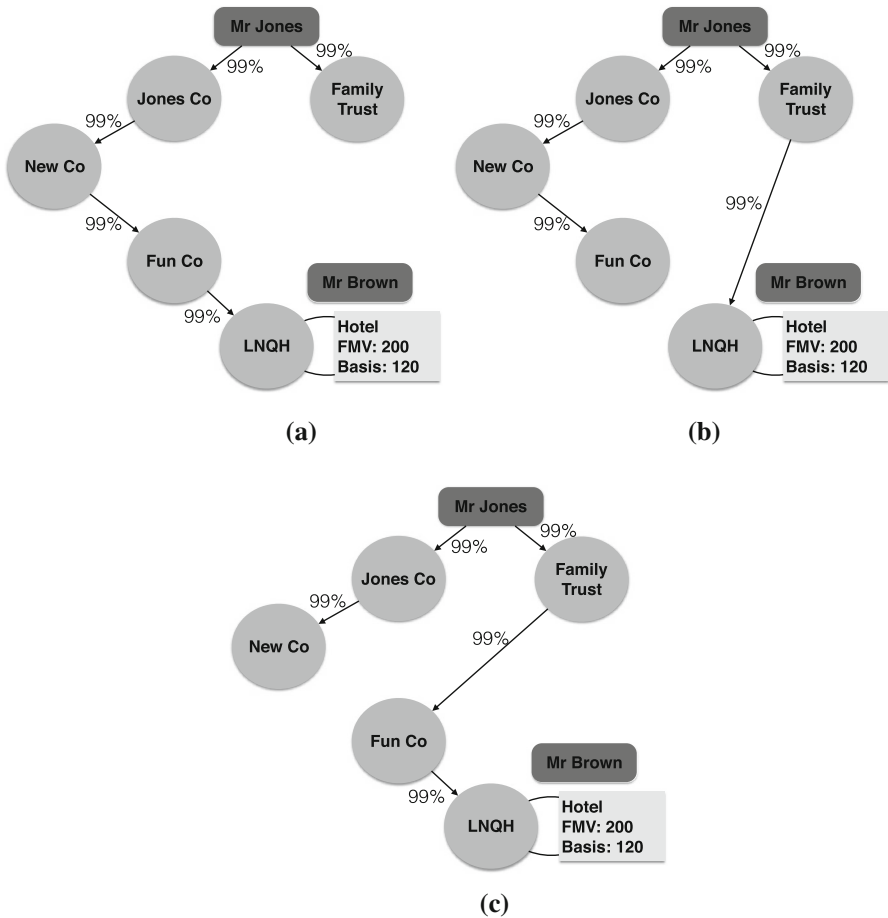


Fig. 14 The steps in the HiBOB tax scheme, a hierarchical extension of iBOB. **a** HiBOB base. **b** HiBOB1. **c** HiBOB2

99 % of LQNH, who in turn now owns the Hotel. We postulated the emergence of 2 interesting transaction sequences utilizing this hierarchy, one that is evasive and the other benign:

HiBOB1 HiBOB1 is almost identical to the original iBOB: FamilyTrust purchases FunCo's share in LQNH for an annuity and invokes a §754 election, mitigating all capital gains on the eventual sale of the Hotel, as shown in Fig. 14b. *HiBOB1* is a derivative of the original iBOB because they both result in zero tax liability and involve the sale of an entity's stake in a partnership that directly owns the material asset being sold.

HiBOB2 HiBOB2 is characterized by the purchase of a partnership asset by FamilyTrust further up the partnership chain. That is, instead of FamilyTrust purchasing a share in LQNH, the entity that owns the Hotel, they instead purchase NewCo's share in FunCo, as shown in Fig. 14c. Because §754 elections only affect the assets of the partnership that is being purchased, the inside basis of the Hotel in respect to FamilyTrust is not adjusted and there is *subsequently no tax benefit*. This is true of all transactions that purchase partnership assets that do not directly own the Hotel. For example, the same result would be accomplished if FamilyTrust purchased JonesCo's share in NewCo.

We ran the simulation 100 times with a population of size 500 for 100 generations with fixed audit weights to determine the ability of STEALTH to discover these more complex transaction sequences. The audit observables were fixed as for Experiment 1. *HiBOB1* was found by the model over the course of all 100 runs. However *HiBOB2* was *not* found by STEALTH in any of the runs. While this result demonstrates the ability of STEALTH to distinguish between the *HiBOB* types it also indicates the need for a smoother solution landscape to allow for exploration of a larger variety of potential schemes.

5 Conclusions and future work

We developed a novel search heuristic called STEALTH that can simulate the co-evolution of abusive tax schemes and audit scores. STEALTH is based on an intuitive time evolving graph-based representation; entities are represented as nodes with assets, edges are ownership relations between entities and transactions are transfers of assets between nodes. STEALTH does not mine data and is fundamentally based on evaluating and exploring rules. As such, STEALTH is a complement to traditional data driven machine learning techniques. Since the STEALTH output provides a readable and intuitive understanding of how transactions can be sequenced to achieve tax evasion it can be used to inform both supervised and unsupervised learning methods. Moreover, STEALTH has the advantage that it allows search of both transactions given a fixed audit score sheet

and the fine tuning of audit scores given fixed transactions. Thus it can be used to determine audit points that are successful in finding suspicious large loss deductions.

While promising, the experiments that have been conducted using our current implementation have been limited. Given iBOB as an initial starting point and assigning its most basic manifestation a high audit likelihood, a genetic algorithm is able to find variants such as the HiBOB scheme in a significant portions of runs. That being said, we constrain the search in a few non-trivial ways, such as by limiting the generative grammar. Limiting the grammar, while indicative of the model's inability to find an acceptable local maxima without help, demonstrates how the addition of expert knowledge can be incorporated into the search without changing the source code. By choosing the types of assets that are exchanged, the number of partnerships that can be formed and which entities engage in transactions, our search can be catered to the situation at hand, greatly increasing its efficiency.

We showed that STEALTH can generate oscillatory dynamics between tax schemes and the audit score sheets designed to detect them. Co-evolutionary algorithms often generate intricate run-time behaviors that make it difficult to monitor progress towards a goal. Furthermore, potential gradient loss (disengagement) of the search can occur here. This happens when the distribution of fitness values in at least one population is almost flat. For example, if audit points are made too "tough", no transaction sequences are able to pass the audit score sheet. Additional precautions need to be taken to avoid mediocre stable states or relative over-generalization, which favors versatile components over those of the optimal solution.

The current drawbacks of STEALTH is that it has a very simplified view of transactions, audit points and the tax law. Our next step is to increase the complexity of both the transactions that compose the tax schemes and the types of activities that are detectable by the audit score sheets. Examples here include logic to handle (a) other deferred payment structures and loans/repurchase agreements, (b) asset 'leakage' such as when transactions post to international destinations and (c) automatic detection of patterns that emerge from tax schemes to help derive audit score sheets.

A sensitivity analysis of the STEALTH algorithm is also required to ascertain the stability of convergence to variations in input parameters and run-time settings. Calculation of externalities such as the Fair Market Value can also have significant uncertainty that can impact auditing cost and efficiency. A sweep of the distribution of values here would help evaluate the robustness of audit sheet thresholds.

Ultimately we seek to evaluate the schemes generated by STEALTH against tax return data and solicit feedback from domain experts.

6 Appendix

6.1 Genetic Algorithm

Co-evolutionary search heuristics (de Jong et al. 2007; Ficici and Bucci 2007; Wiegand and Potter 2006; Stanley and Miikkulainen 2004) evaluate an individual solution based on interactions between populations of multiple solutions. The individual solution may appear good in one context and poor in another context, e.g. one solution's ranking in a population can change depending on other solutions.

The control flow of a Genetic Algorithm search is shown in Fig. 8 The steps in a single iteration (generation) are:

1. *Initialize* Input the initial solutions, e.g. uniformly randomly generated input sequences.
2. *Evaluate* The individual solutions are evaluated and assigned a score (fitness) according to some function.
3. *Select* Some individuals from the current population are included in a new population.
4. *Variation* Individuals in the new population are modified by some operators, e.g. crossover and mutation.
5. *Replacement* Update the current population with the new population.
6. *Termination* Stop if a termination criteria is met.
7. *Iterate* Return to step 2.

6.1.1 Grammatical evolution

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 (O'Neill and Ryan 2003). We can map to a transaction sequence by means of a grammar which conveniently expresses all possible transaction sequences compactly. 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 Fig. 15. 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 represented 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

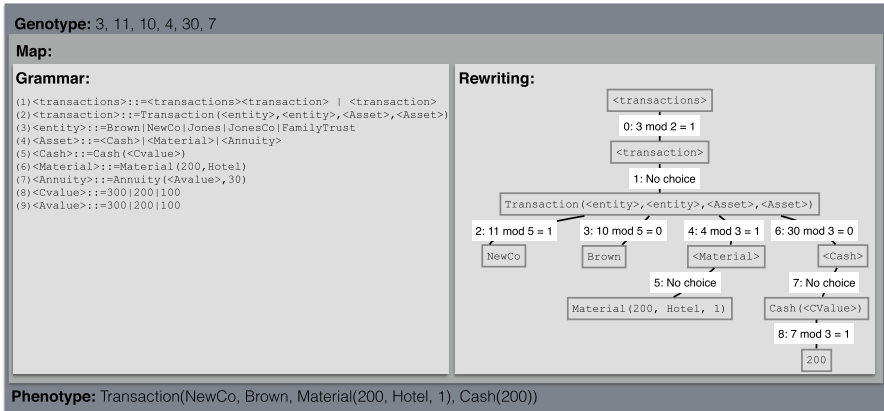


Fig. 15 Example of mapping a list of integers (Genotype) into a list of transactions (Phenotype) by using grammatical evolution

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.

In Fig. 15 there is an example of the rewriting of an integer list (genotype) to a sentence (phenotype) describing a transaction between two entities that exchange assets.

1. We pick the first rule in the grammar as the start symbol, in this case (1) <transactions>.
2. Expand the left most non-terminal symbol in our sentence <transactions>. We take the current integer input 3 and the modulo of the number of production choices 2, which is 1, thus we pick <transaction> the production choice at position 1 (the indexing starts at 0) and rewrite the <transactions> with <transaction>.
3. Again expand the left most non-terminal symbol <transaction>. There is only one production choice here, so it is rewritten to Transaction(<entity>, <entity>, <Asset>, <Asset>).
4. Again expand the left most non-terminal symbol <entity>. We take the current integer input 11 and the modulo of the number of production choices 5, which is 1, thus we pick NewCo. The sentence is now Transaction(NewCo, <entity>, <Asset>, <Asset>).
5. The left most non-terminal symbol is again <entity>. We take the current integer input 10 and the modulo of the number of production choices 5, which is 0, thus we pick Brown. The sentence is now Transaction(NewCo, Brown, <Asset>, <Asset>).
6. The left most non-terminal symbol is now <Asset>. We take the current integer input 4 and the modulo of the number of production choices 3, which is 1, thus we pick <Material>. The sentence is now Transaction(NewCo, Brown, <Material>, <Asset>).

7. The left most non-terminal symbol is now $\langle \text{Material} \rangle$. There are no choices for $\langle \text{Material} \rangle$ so we rewrite it with $\text{Material}(200, \text{Hotel}, 1)$. The sentence is now $\text{Transaction}(\text{NewCo}, \text{Brown}, \text{Material}(200, \text{Hotel}, 1), \langle \text{Asset} \rangle)$.
8. The left most non-terminal symbol is again $\langle \text{Asset} \rangle$. We take the current integer input 30 and the modulo of the number of production choices 3, which is 0, thus we pick $\langle \text{Cash} \rangle$. The sentence is now $\text{Transaction}(\text{NewCo}, \text{Brown}, \text{Material}(200, \text{Hotel}, 1), \langle \text{Cash} \rangle)$.
9. The left most non-terminal symbol is now $\langle \text{Material} \rangle$. There are no choices for $\langle \text{Cash} \rangle$ so we rewrite it with $\text{Cash}(\langle \text{CValue} \rangle)$. The sentence is now $\text{Transaction}(\text{NewCo}, \text{Brown}, \text{Material}(200, \text{Hotel}, 1), \text{Cash}(\langle \text{CValue} \rangle))$.
10. The left most non-terminal symbol is $\text{Cash}(\langle \text{CValue} \rangle)$. We take the current integer input 7 and the modulo of the number of production choices 3, which is 1, thus we pick 200. The sentence is now $\text{Transaction}(\text{NewCo}, \text{Brown}, \text{Material}(200, \text{Hotel}, 1), \text{Cash}(200))$.
11. There are no more non-terminal symbols left to rewrite and our string rewriting is done.

6.2 STEALTH formalism

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

6.2.1 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

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

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.

6.2.2 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 $\mu : T_i \rightarrow \Phi$ 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 $\tau : \mathbf{T} \times \Gamma \rightarrow \Gamma$
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, $P : \mathbf{T} \times \Gamma \rightarrow \mathbb{R} \times \Gamma$

6.2.3 Optimization of tax ecosystem

We can describe the process by which sequences of transactions and initial ownership network are generated by defining a grammar $\mathcal{E}_t : \mathbb{Z}_+^n \mapsto \mathbf{T} \times \Gamma$ 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 $\mathbf{x} \in \mathbb{Z}_+^n$, $\mathcal{E}_t(\mathbf{x}) = (\mathbf{t}, \gamma_0)$ 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 $m \in \mathbb{Z}_+$,

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

The grammar $\mathcal{E}_a : \mathbb{Z}_+^m \mapsto \Psi$ 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 $i \in [0, k]$, $F_i(t_i, \gamma_i, \psi) = (\gamma_{i+1}, s_i)$ where $s = s_k$

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 scheme is the same as that shown above, but with the opposite sign $h_a = -h_e = -d_L(1 - s)$ (where a negative deductible loss value indicates positive taxable gain). Recalling that each audit score sheet evaluates a population of transaction sequences, an individual can determine sequences with relatively low levels of taxable gain. Thus, an audit score sheet is fit in the event that it assigns: 1) high audit likelihood to transaction sequences with relatively low levels of taxable gain and 2) low audit likelihood to sequences with normal or high levels of taxable gain.

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

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

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

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

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.

6.3 STEALTH software architecture

The UML diagrams in Figs. 16 and 17 shows the central classes defined in STEALTH and their relationships. The Graph class contains all the entities and assets. The GraphTransformer transitions the network from one state to the

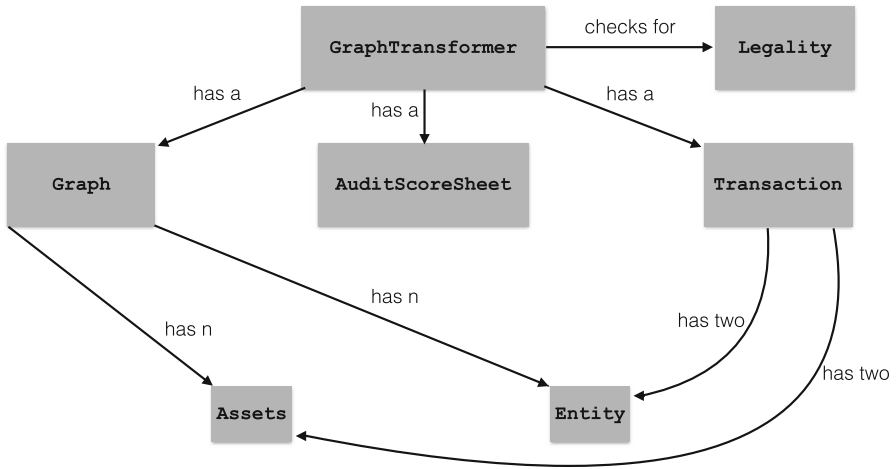


Fig. 16 STEALTH UML diagram of the central classes

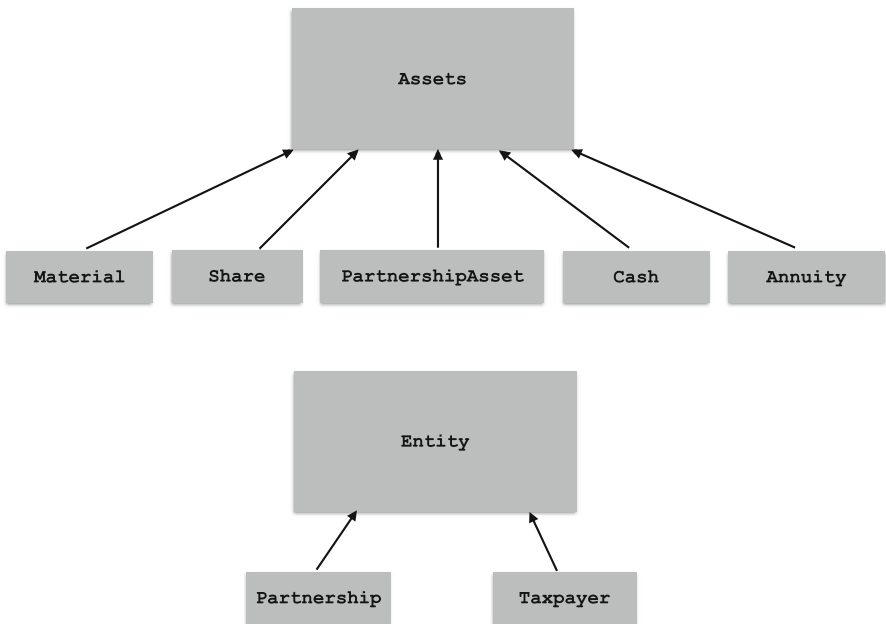


Fig. 17 STEALTH UML diagram assets and entities classes

other by performing a Transaction consisting of two Assets and Entities, checking the Legality of it and calculating the corresponding entries in the *AuditScoreSheet*.

References

- Accountancy JO (2015) <http://www.journalofaccountancy.com/news/2015/oct/partnership-audit-and-adjustment-rules-201513294.html>
- Allingham MG, Sandmo A (1972) Income tax evasion: a theoretical analysis. *J Public Econ* 1(3–4):323–338
- Andrei AL, Comer K, Koehler M (2013) An agent-based model of network effects on tax compliance and evasion. *J Econ Psychol* 40:119–133
- Andreoni J, Erard B, Feinstein J (1998) Tax compliance. *J Econ Lit* 36:818–860
- Aubert S, Müller JP (2013) Incorporating institutions, norms and territories in a generic model to simulate the management of renewable resources. *Artif Intell Law* 21(1):47–78
- Bench-Capon T, Araszkiwicz M, Ashley K, Atkinson K, Bex F, Borges F, Bourcier D, Bourguine 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. *Artif Intell Law* 20(3):215–319
- Bloomquist K (2011) Tax compliance as an evolutionary coordination game: an agent-based approach. *Public Finance Rev* 39(1):25–49
- Bloomquist KM (2006) A comparison of agent-based models of income tax evasion. *Soc Sci Comput Rev* 24(4):411–425
- Boer A, van Engers T (2013) Agile: a problem-based model of regulatory policy making. *Artif Intell Law* 21(4):399–423
- Bonchi F, Giannotti F, Mainetto G, Pedreschi D (1999) A classification-based methodology for planning audit strategies in fraud detection. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 175–184
- Buchanan BG, Headrick TE (1970) Some speculation about artificial intelligence and legal reasoning. *Stanford Law Review*, pp 40–62
- Davis JS, Hecht G, Perkins JD (2003) Social behaviors, enforcement, and tax compliance dynamics. *Account Rev* 78(1):39–69
- DeBarr D, Eyler-Walker Z (2006) Closing the gap: automated screening of tax returns to identify egregious tax shelters. *ACM SIGKDD Explor Newsl* 8(1):11–16
- Dechesne F, Di Tosto G, Dignum V, Dignum F (2013) No smoking here: values, norms and culture in multi-agent systems. *Artif Intell Law* 21(1):79–107
- de Jong ED, Stanley KO, Wiegand RP (2007) Introductory tutorial on coevolution. In: GECCO (Companion), pp 3133–3157
- Ficici SG, Bucci A (2007) Advanced tutorial on coevolution. In: GECCO (Companion), pp 3172–3204
- GAO (2013) Gao-14-453. <http://www.youtube.com/watch?v=O8VDUStvxMY>
- GAO (2014a) <http://www.gao.gov/assets/670/663185.pdf>
- GAO (2014b) <http://www.gao.gov/assets/670/661772.pdf>
- Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-wesley, Boston
- Hemberg E, Rosen J, Warner G, Wijesinghe S, O'Reilly UM (2015) Tax non-compliance detection using co-evolution of tax evasion risk and audit likelihood. In: ICAIL
- Hokamp S, Pickhardt M (2010) Income tax evasion in a society of heterogeneous agents-evidence from an agent-based model. *Int Econ J* 24(4):541–553
- Hokamp S, Seibold G (2014) How much rationality tolerates the shadow economy?—an agent-based eonophysics approach. In: Advances in social simulation. Springer, Berlin, pp 119–128
- IRS (2006) The tax gap. <http://www.irs.gov/uac/IRS-The-Tax-Gap>
- IRS (2014a) Irs audits. <http://www.irs.gov/Businesses/Small-Businesses-&Self-Employed/IRS-Audits>
- IRS (2014b) Sales and trades of investment property. <http://www.irs.gov/publications/p550/ch04.html>
- Jaideep KWHNP, Bjorklund SGTE (2009) Data mining based tax audit selection: a case study from minnesota department of revenue. In: Proceedings of the third international workshop on data mining case studies. ACM, pp 23–35

- Kallio M, Back B (2011) The self-organizing map in selecting companies for tax audit. In: Emerging Themes in information systems and organization studies. Physica-Verlag HD, pp 347–358
- Katz DM, Bommarito MJ II (2014) Measuring the complexity of the law: the united states code. *Artif Intell Law* 22(4):337–374
- Kingston J, Schafer B, Vandenberghe W (2004) Towards a financial fraud ontology: a legal modelling approach. *Artif Intell Law* 12(4):419–446
- Korobow A, Johnson C, Axtell R (2007) An agent-based model of tax compliance with social networks. *National Tax J*, pp 589–610
- Li WP, Azar P, Larochelle D, Hill P, Lo AW (2015) Law is code: a software engineering approach to analyzing the united states code. *J Bus Technol Law* 10:297
- Lipatov V (2003) Evolution of tax evasion. Technical report, University Library of Munich, Germany
- Lotzmann U, Möhring M, Troitzsch KG (2013) Simulating the emergence of norms in different scenarios. *Artif Intell Law* 21(1):109–138
- May LR (2012) Using link analysis to identify indirect and multi-tiered ownership structures. In: SOI Tax Stats—2012 IRS-TPC research conference
- McCarty LT (1977) Reflections on “taxman”: an experiment in artificial intelligence and legal reasoning. *Harvard Law Review* pp 837–893
- Mittone L, Patelli P (2000) Imitative behaviour in tax evasion. In: Economic simulations in swarm: agent-based modelling and object oriented programming. Springer, US, pp 133–158
- Ngai E, Hu Y, Wong Y, Chen Y, Sun X (2011) The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature. *Decis Support Syst* 50(3):559–569
- Oard DW, Webber W (2013) Information retrieval for e-discovery. *Inf Retr* 7(2–3):99–237
- O’Neill M, Ryan C (2003) Grammatical evolution: evolutionary automatic programming in an arbitrary language, vol 4. Springer, Berlin
- Pickhardt M, Prinz A (2014) Behavioral dynamics of tax evasion—a survey. *J Econ Psychol* 40:1–19
- Pickhardt M, Seibold G (2014) Income tax evasion dynamics: evidence from an agent-based econophysics model. *J Econ Psychol* 40:147–160
- Rosen J, Hemberg E, Warner G, Wijesinghe S, O’Reilly UM (2015) Computer aided tax evasion policy analysis: directed search using autonomous agents. In: AAMAS
- Rostain T, Regan MC (2014) Confidence games: lawyers, accountants, and the tax shelter industry, vol 1. MIT Press, Massachusetts
- Sartor G, Rotolo A (2013) AI and law. In: Agreement technologies. Springer, Netherlands, pp 199–207
- Stanley KO, Miikkulainen R (2004) Competitive coevolution through evolutionary complexification. *J Artif Intell Res JAIR* 21:63–100
- Surden H (2014) Machine learning and law. *Wash L Rev* 89:87–217
- Van V et al (1973) A new evolutionary law. *Evol Theory* 1:1–30
- Warner G, Wijesinghe S, Marques U, Badar O, Rosen J, Hemberg E, O’Reilly UM (2014) Modeling tax evasion with genetic algorithms. *Econ Gov*. doi:[10.1007/s10101-014-0152-7](https://doi.org/10.1007/s10101-014-0152-7)
- Wiegand RP, Potter MA (2006) Robustness in cooperative coevolution. In: Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM, pp 369–376
- Wright Jr D (2013) Financial alchemy: how tax shelter promoters use financial products to bedevil the irs (and how the irs helps them). *Ariz State Law J* 45(2)
- Zaklan G, Lima FW, Westerhoff F (2008) Controlling tax evasion fluctuations. *Phys A Stat Mech Appl* 387(23):5857–5861
- Zaklan G, Westerhoff F, Stauffer D (2009) Analysing tax evasion dynamics via the ising model. *J Econ Interact Coord* 4(1):1–14