Modeling Tax Evasion with Genetic Algorithms

Geoff Warner[1]
Sanith Wijesinghe[1]
Uma Marques[1]
Una-May O'Reilly[2]
Erik Hemberg[2]
Osama Badar[2]
[1] *The MITRE Corporation*
McLean, VA, USA
[2] *Computer Science and Artificial Intelligence Laboratory*
Massachusetts Institute of Technology
Cambridge, MA, USA

July 3, 2013

Abstract

The U.S. tax gap is estimated to exceed $450B, most of which arises from non-compliance on the part of individual taxpayers [1], [2]. Much is hidden in innovative tax shelters combining multiple business structures such as partnerships, trusts and S-corporations into complex transaction networks designed to reduce and obscure the true tax liabilities of their individual shareholders. One known gambit employed by such tax shelters is to offset real gains in one part of a portfolio by creating artificial capital losses elsewhere through the mechanism of inflated basis, a process made easier by the relatively flexible set of rules surrounding 'pass-through' entities such as partnerships [3].

The ability to anticipate the likely forms of emerging evasion schemes would represent a major tactical advantage to the IRS. To this end, we

are developing a prototype evolutionary algorithm designed to generate potential schemes of the 'inflated basis' type described above. In particular, the schemes produced by the algorithm will consist of sequences of actions undertaken by a network of tax entities that satisfy all transactional rules but nevertheless minimize tax liability. The algorithm takes as inputs a collection of asset types and tax entities, together with a rule-set governing asset exchanges between these entities. The 'fitness function' used to rank these schemes is itself a function of the reduction in tax liability they afford. Outputs consist of future generations of schemes that are evolved in time according to the mechanisms of mutation and recombination employed by genetic algorithms.

1 Background and Introduction

The U.S. tax gap, defined as the aggregate sum of the difference between what is owed in principle and what is paid in practice by all taxable entities, has recently been estimated to exceed 450 billion dollars. The bulk of this difference (2/3) is attributable to individual taxpayer non-compliance as mediated by abusive tax shelters comprised of complex transactions involving multiple business entities [1], [2]. Partnerships and other so-called "pass-through" entities are known to play a disproportionate role in these structures due to the relative flexibility of the tax rules governing transactions to which they are a party.

Tax shelters are marketed to high net worth individuals by promoters. Promoters include banks, accounting firms, investment boutiques and law firms who scour the tax code looking for exploitable loopholes. They and their confederates then arrange and execute a sequence of transactions designed to reduce their client's tax liability. On the surface these transactions satisfy all relevant tax laws; upon closer inspection, however, it becomes apparent that the transactions in question can have had no other purpose than the elimination of tax liability. Schemes of this type have long been disallowed under a common law doctrine requiring that the associated transactions have "economic substance" [4]; they are now explicitly illegal under the provisions of the 2010 Affordable Care Act [5].

These schemes come in a variety of shapes and sizes. Here we focus on an important subclass that rely on the mechanism of inflated basis to create artificial losses that are used to offset gains elsewhere in a portfolio. 'Basis' is

the set-point from which gains or losses are assessed for tax purposes; usually the basis of an asset is just the cost of acquiring it. There are, however, a complex set of rules governing how basis is computed or otherwise adjusted in the course of different transactions.

2 Approach and Methodology

Tax evasion schemes are constantly evolving. Whenever one is uncovered and measures are taken to eliminate it, others spring up to replace it. These others are often variations of the same underlying idea, though the flow of assets and the arrangement of involved entities may appear quite different than in the original scheme. One notable example of this phenomenon is the so called Son of BOSS tax shelter, which emerged in the mid-90s after its immediate predecessor, a strategy known as "shorting against the box", was rendered defunct by changes in the tax code [6].

There exists, as yet, no systematic method to anticipate the emergence of these schemes. As all such schemes are ultimately reducible to sequences of pairwise transactions between different financial entities, and as these transactions are themselves governed by a finite set of rules, it seems plausible to suppose that a computational model capable of generating candidate schemes automatically could be devised. In fact we propose that a properly designed genetic algorithm is just such a model.

Genetic algorithms are search heuristics, like hill climbing or simulated annealing, that can be applied to optimization problems. What distinguishes them from other search methods is their formal similarity to Darwinian evolution; the search process itself is mediated by a population of bit strings, or "chromosomes", which map to elements of the search space and can be manipulated in a manner reminiscent of their biological counterparts. We here undertake a brief overview of genetic algorithms and then explain our application of this methodology to the problem at hand.

2.1 Genetic Algorithms

All genetic algorithms require a genetic representation [7]. This is a method for encoding solutions in a basic mathematical structure like a bit string or parse tree. For the sake of simplicity, we focus here on bit strings of fixed length K, which form the"chromosomes" of the representation. The

representation itself consists of these chromosomes (the genotype), together with a deterministic mapping from each chromosome to an element of the search space (the phenotype). Further, all GAs require a measure of fitness on these phenotypes – loosely speaking, this constitutes the objective function of the problem at hand. The only formal requirement of the method of fitness evaluation is that it allow for an ordinal ranking of solutions, though of course it is generally better to have an absolute measure. Finally, all GAs feature some method of selection and genetic variation. Selection involves choosing chromosomes in the population for reproduction according to the relative fitness of each member of the population – the higher the fitness, the higher the probability of being selected. Variation is introduced through the use of genetic operators like crossover and mutation. Crossover is the process whereby corresponding segments of two different chromosomes are chosen at random by some method and then transposed. Mutation consists of a bitwise flip of each element of the selected bit string with some probability p. We expand somewhat on these capsule definitions in what follows.

The canonical GA exhibits the following iterative structure ([7], [8]), which we here describe in seven steps. First, an initial population of N chromosomes is generated, usually randomly. Second, each member of this population is subjected to an evaluation of its fitness. The process of fitness evaluation may be so simple it needs only the computation of a basic formula, as in the traveling salesman problem, or so complex that it requires its own simulation, as in the design of a bridge or a jet engine. In the third step, pairs of chromosomes are selected from the population for crossover and mutation. The selection method must favor fitter members of the population; one popular approach, and the one we adopt here, is called tournament selection. In tournament selection, k members are drawn at random from the population, and the fittest of these is selected. Fourth, the crossover and mutation operators are applied to the selected pair. Steps three and four are repeated until $N - e$ children have been produced, where e is the size of the elite population (that is, the group composed of the e fittest members of our original population). Fifth, the old population is replaced by the new; this latter is comprised of the $N - e$ children that were produced by iterating steps 3 and 4, together with the e fittest members of the old population. Sixth, a test condition is evaluated in order to determine whether to halt. Seventh, assuming the test condition is false, we return to step 2 with our new population.

2.2 An Application of GAs to Tax Evasion

We employ a variant of the evolutionary algorithm approach known as grammatical evolution [8]. The principal difference between this method and other similar algorithms is in the genetic representation. In GE, chromosomes consist of lists of integers; each integer is called a "codon". Phenotypes are lists of instructions that can be interpreted and executed by other modules of the algorithm. The mapping from genotype to phenotype proceeds by means of a context-free grammar. A grammar consists of a set of symbols, called "terminal" and "non-terminal" symbols, together with a set of overwrite or production rules. The non-terminal set always includes a "start" symbol. Production rules prescribe the manner in which a particular non-terminal symbol may be replaced by combinations of terminal and non-terminal symbols. In our case, the output of production rules results in a list of executable instructions (or "schemes") which act on previously instantiated Asset and Entity objects.

The algorithm begins by instantiating a number of Asset and Entity java objects. The Asset class may include stocks, promissory notes, loans, cash, options or whatever other securities or instruments are deemed necessary. Entities include individual taxpayers, partnerships, trusts, corporations, banks, etc. These objects track all the book-keeping associated with transactions, including ownership, the market value of assets, the basis of assets, inside and outside basis of all assets in relation to the partners in a partnership, debt obligations, and tax owed on any particular exchange.

Whenever a particular chromosome is being evaluated for fitness, a parser looks at the leftmost non-terminal symbol and replaces it using a production rule determined by the value of the corresponding codon. When only terminal symbols are left, they form a list of instructions which is then passed to an interpreter module for execution. The interpreter uses the Asset and Entity objects on the heap to carry out the instructions, most of which involve pairwise exchanges of Asset objects between Entities, and makes sure to apply whatever tax rules may be applicable to the exchange.

At present, fitness is evaluated purely by taking the difference between the tax that should have been paid before any transactions were undertaken, and the tax that is assessed afterwards. Fitter schemes have larger values of this difference. In future we hope to go beyond this fitness measure by incorporating elements of risk and cost to any particular scheme.

3 Conclusion and Next Steps

We have built a functioning end-to-end codebase capable of executing all the steps of the canonical genetic algorithm described above. In particular, we have developed a genetic representation that we believe is flexible enough to generate complex tax evasion schemes. This was achieved by separating the representation into two parts: first, a set of Asset and Entity objects responsible for tracking any state changes associated with transactions, and second, a grammar mapping chromosomes into a set of transaction instructions to be executed by the associated objects.

At present, the number of Asset and Entity objects is small, and the grammar is functional but still relatively primitive. We continue to develop these aspects of the genetic representation by incorporating more assets and entities and by expanding the space of possible transactions. The principal challenge to these efforts is the sheer complexity of potential transactions and the rules governing them. Rather than attempt a wholesale reconstruction of this space, our current objective is to incorporate only the smallest ruleset necessary to reproduce a known scheme like Son of BOSS or iBOB [9]. Once this goal has been reached, we will turn our attention to the larger space required to anticipate as yet undiscovered evasion strategies.

References

[1] http://www.gao.gov/assets/600/590215.pdf

[2] http://www.irs.gov/uac/IRS-The-Tax-Gap

[3] http://www.irs.gov/pub/irs-pdf/p3744.pdf

[4] Robertson, John F.; Quinn, Tina; and Carr, Rebecca. Codification of the Economic Substance Doctrine. *Journal of Business Administraton Online*, vol. 9 no. 2, 2010.

[5] http://www.irs.gov/Businesses/Guidance-for-Examiners-and-Managers-on-the-Codified-Economic-Substance-Doctrine-and-Related-Penalties

[6] Wright, Del. Financial Alchemy: How Tax Shelter Promoters Use Financial Products to Bedevil the IRS (and How the IRS Helps Them), forthcoming, *Ariz. St. L. J.* 2013.

[7] Goldberg, David E. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Boston: Addison-Wesley, 1989.

[8] Brabazon, Anthony and O'Neill, Michael. *Biologically Inspired Algorithms for Financial Modeling*, Heidelberg: Springer-Verlag, 2010.

[9] http://www.gao.gov/new.items/d10968.pdf