Incentivai

concept paper

Piotr Grudzien
http://incentivai.co
piotr@incentivai.co

March 10, 2018

Abstract

Currently, there is no equivalent of AB testing for economy-based mechanism design encoded in smart contracts (or otherwise). In this paper, I propose a solution for simulating such systems prior to deployment for real-world use. Reinforcement Learning (RL) agents are trained to test mechanism design under conditions similar to what is expected after deployment into widespread adoption. It can be used for testing the incentive structure being generated, discovery of failure modes, collusion between agents occurring both on- and off-chain.

1 Motivation

Suppose, last night there was a street demonstration and one newspaper claims that 10,000 people took part, another that it was 100,000. A true value exists, yet there is no reliable way of discovering it apart from working hard on analysing several possibly biased and inaccurate sources. Additionally, the person or entity doing the research, has to themselves be trustworthy, unbiased and accurate. One possible alternative approach is to use prediction markets.

Imagine a plugin that works silently in your web browser and provides reliable, unbiased validation of any piece of news or information you come across. What one experiences is browsing a truth-augmented Internet, one where every fact is validated and every source is accompanied by a score reflecting how reliable they have been in the past. Most importantly, however, nobody owns it but rather it is collectively run by a group of users who are guaranteed to act in good faith. Being a bad actor is expensive and is not sustainable in the long run.

How to achieve that?

The task is to design an incentive mechanism such that the dominant strategy is to make the best guess at the true value rather than try to guess how the majority will vote, not participate at all or perform any form of dishonest behaviour. While this sounds like a very difficult task, there are several possibilities one has at their disposal: pay-off system that rewards voting with majority, reputation system that penalises following the crowd, favours 51% majority to 99% majority, and many others. Pay-off and reputation systems are hyperparameters of the design that one wants to tune. What makes the task even more challenging is the impossibility of reliably testing it.

Incentivai takes a step towards the solution by proposing to use AI agents to simulate how people would behave in the real world.

2 Problem statement

Economy-based mechanisms implement financial incentives aimed at encouraging desired behaviours. Examples include prediction markets[1, 2, 3], proof of stake consensus protocols[4], public-content platforms[5], *blind oracles*[1] or decentralised reputation systems. Many of them can be implemented in a decentralised manner using blockchain technology. It could potentially also be used for testing various cryptoeconomic protocols like sharding[6], plasma[7], state channels[8], Truebit[9] or decentralised exchanges[10]. To the best of my knowledge, limited amount of mechanism design testing is currently being done prior to deployment of such systems. Such testing would include (and not be limited to):

1. Testing if correct incentives are being created (e.g. *is honest participation beneficial?*)
2. Preventing ways of benefiting from acting against the mechanism (e.g. *are users incentivised to influence other users by offerring bribes?*)
3. Discouraging collusion and centralisation of power (e.g. *are users incentivised to act independently or delegate their vote to a centralised pool?*

The above list is inexhaustive and should be expanded to touch on several issues that can be characterised as *known unknowns* as well as *unknown unknowns*. The key observation is that the possibility of their occurrence can, by definition, never be proved or disproved. Formal verification approaches, while applicable to testing smart contract logic[14, 15, 16], can provide no guarantees. Users can never be prevented from freely acting and cooperating in unexpected ways both on- and off-chain. This essentially makes the rule set infinite and makes any formal verification impossible.

Smart contract mechanism design[11, 12, 13] should, apart from considering a list of known attack vectors and dishonest behaviours, consider the following:

1. Discussion of the extent to which the list is exhaustive
2. Justification of particular design decisions made in terms of their optimality
3. Robustness to potential unknown failure modes
4. Robustness in *high-stress scenarios*
   - high transaction volume and value
   - possible external incentives for users to destabilise the system
   - increased potential gain from successfully attacking the system
5. Robustness to deviations from rational behaviour (see section 6)

Incentivai, or any other testing system, will never be able to rule out all possible failure modes and guarantee correct incentive structures. What it can do, however, is discover a lot about both of those for any economy-based smart contract *with no risk* and *prior to* its deployment. As of mid-February 2018 almost 15,000[2] verified smart contracts alone have already been deployed to just the Ethereum main network.

3 The AB testing analogy

Web design can be thought of as a machine learning problem. Historical data can be split temporaly into a training and test set. All design decisions can be abstracted away into a set of hyperparameters that need to be optimised to maximise objective functions such as click-through rate or ad revenue. Finally, effects of feedback loops and data non-stationarity can be mitigated by use of controlled experiments such as AB testing.

One could attempt to use a similar framework for smart contract mechanism design problem. Historical data could be gathered after it is deployed to a test or real network. Mechanism design could be expressed as a set of tunable hyperparameters. Modular smart contract design could make it de facto modifiable[17].

However, the AB testing analogy breaks down for several important reasons:

[1] *Blind oracle* is a term I use to describe prediction markets for data for which a true value exists however does not *become available in future*. Example: *number of people who took part in a street protest as described in the Motivation section.*

[2] https://etherscan.io/contractsVerified

1. Testing iterations would be extremely time-consuming and expensive
2. Initial failures would discourage users from future participation
3. Most importantly, all code running on a blockchain is necessarily public and users are not only aware of the testing itself but also of the particular flavour of the contract they are interacting with. That in my view changes their behaviour in a way that cannot be neglected and any insights would not be relevant to a later scenario when the contract is released for full use never to be altered. Actions of keenly volunteering early adopters are wildly different from those of large groups of users with lots to gain and lots to lose.

4 Incentivai structure

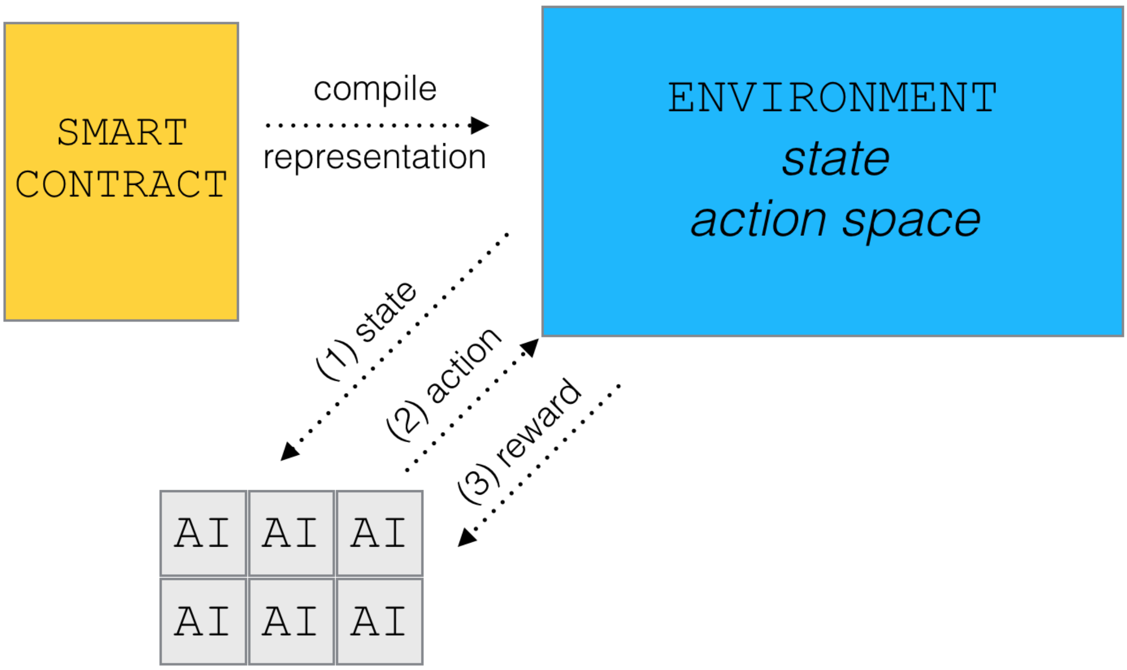

Figure 1: Incentivai: basic structure

The basic structure of the Incentivai system is similar to that of a typical Reinforcement Learning system[18]. There is an environment and a number of agents. Agents are trained (e.g. using an approach similar to [19]) to interact with the environment and optimise their specified objective function. The goal is to observe the behaviour of trained agents and verify whether it is in line with the idea behind the mechanism that was designed.

4.1 Environment representation customisation

One the challenges of building a system like Incentivai is to accurately represent the way users would interact with the mechanism in the real world in a way that can be leveraged by RL algorithms.

Let us take a simple example of a Vickrey auction[20]. In this case, the state could be represented by a single scalar number - user's private valuation. Action space could be discretised into a number of bidding options.

A more complex representation would need to be used for the *blind oracles* mechanism described in sections 1 and 2. As a starting point, statistics of votes cast so far could be used as features included in the state vector. A corresponding action space would contain bidding options (e.g. True and False in a binary validation scenario) and possibly an option to abstain from bidding.

Another step, could be a model of user's private knowledge. Suppose the true value is 0 and agents have an option to sample from their *truth function* which is $\mathcal{N}(\mu, \sigma^2)$. Action space could also be expanded by providing agents with an option to incur extra cost to reduce their private bias μ or variance σ^2. This could be a model of extra cost incurred by putting an effort into research. Another option could be for agents to follow actions of other agents depending on their bias and variance and subject to variable cost.

Further extension could be to allow agents to communicate and cooperate[21]. In that scenario, the action space would be expanded to let them announce how they are planning to vote and possibly the value of the bribe they are offerring. Anologously, state representation would be expanded to make that information observable for all agents.

4.2 Smart contract environment representation

The above serves as a simple example of customisation that might be needed for many real world scenarios to be adapted to the Incentivai framework. However, one very useful characteristic of blockchain smart contracts (e.g. as implemented on the Ethereum blockchain[17]) is their fully defined interface which can be readily translated into an RL action space (see section 8). In that scenario, the state vector would represent the current state of deployed smart contracts and action space their combined interfaces encoding how users can interact with them. State transition and action cost implementation can and ought to be cloned exactly from how it is running on the public blockchain. Well defined interface, structure and state transitions are one of the main reasons why mechanisms implemented on a blockchain could reliably be tested using AI agents.

It needs to be noted, however, that the above approach does not take into account possible off-chain interactions between agents. That needs to be modelled separately and translated into extensions to state and action space representations (as in the simple example in section 4.1).

4.3 Objective function

Design of the objective function should be customised to the particular problem at hand, however total net profit (individual or within a cooperating group of agents) should usually be a useful starting point. Other possible objective functions could incorporate aiming at harming a particular subgroup of users, arriving at false or biased results in a system like *blind oracle* (relative to the *truth function*, see section 4.1) or rendering the system useless by preventing it from reaching its goals or promoting an incentive structure that is wildly incompatible with what was intended.

5 Experimental procedure

A typical experimental iteration in the Incentivai framework would involve the following steps:

1. Environment representation design
   - state
   - action space
   - reward structure
   - objective function
2. Training of agents' policies
3. Quantitative and qualitative analysis of trained agents' behaviour in the environment
4. Refinement of representation design (back to step 1)

The above iterative procedure should generate a number of insights regarding possible failure modes as well as behaviour observed as a result of the incentive structure.

6 Behavioural analysis

Perhaps people in the real world would not necessarily behave in the same way as trained AI agents? Excellent remark.

Indeed, the above design and analysis are valid under the *rational agent assumption*. The Incentivai system could follow the approach proposed in [23, 24] to model assumptions such as *quantal response* or *limited iterative strategic reasoning*. That would essentially add an extra modeling layer which allows for relaxing the rational agent assumption:

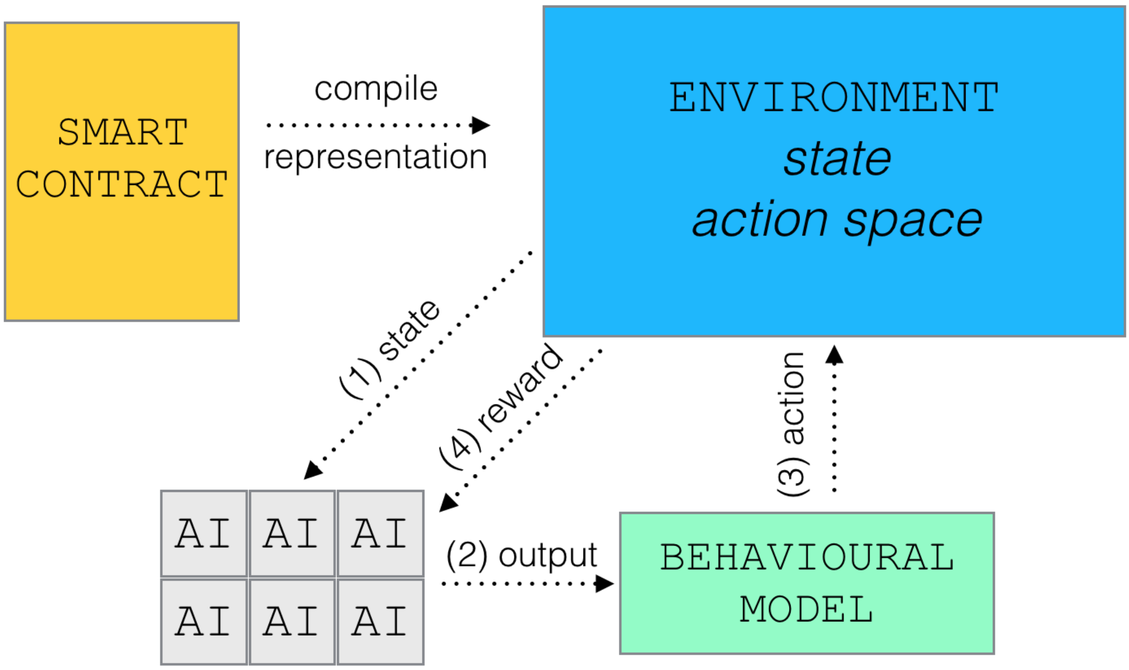

Figure 2: Incentivai: structure including behavioural model

7 Deep Reinforcement Learning research challenges

Deep Reinforcement Learning is an active field of research and work on the Incentivai project will need to go hand in hand with its cutting edge developments. A particularly important and challenging aspect is the correct choice of state and action space representation. It needs to both capture all the aspects of the mechanism at test but also aid agent learning. This is an active area of research, development and experimentation within the Incentivai project.

As suggested in [25, 26], other approaches such as supervised learning or Monte Carlo tree search could replace RL in certain scenarios.

8 Further developments

This concept paper presents a high-level overview of basic ideas behind Incentivai - the system for testing smart contract mechanism design by means of simulation using RL agents. As development progresses and more mechanisms get tested, more generic solutions and universal best practices will be developed (for best practices relating to RL, see [27, 28, 29, 30, 31, 32]). Crucially, that should facilitate growth of Incentivai as an ecosystem of components that can be reused across projects.

One necessary component of future work is *compilers* that translate most popular smart contract languages (e.g. Solidity[3]) directly into an Incentivai-compatible representation. The basic idea would be to map internal states and interfaces of smart contracts to state and action space representations, respectively.

9 Conclusion

Economy-based smart contracts are an essential component of any blockchain ecosystem and beyond. In author's opinion, they have great potential of social impact as exemplified in section 1.

One of the many challenges on the path towards widespread adoption of such systems is understanding and trust of the global population. Cases when trust was lost due to human error are not unheard of [4]. Errors in mechanism design are particularly difficult to predict, model and guard against yet no less dangerous than security vulnerabilities.

Incentivai is taking a step towards answering questions such as *is there a beneficial way to cheat?* or *what is the true incentive structure?* more robustly. In my view, doing so is necessary to make reasonable attempts at deploying systems such as *blind oracle* on public blockchain.

Finally, perhaps somewhere down the line, Incentivai could be used to provide insights into other, much more open-ended and underdefined incentive systems such as supermarket clubcards, promotion points and extra offers, all the way to election systems or state policies.

References

[1] Paul Sztorc, *Truthcoin, Peer-to-Peer Oracle System and Prediction Marketplace*, http://bitcoinhivemind.com/papers/truthcoin-whitepaper.pdf, 2015

[2] Jack Peterson, Joseph Krug, Micah Zoltu, Austin K. Williams, and Stephanie Alexander, *Augur: a Decentralized Oracle and Prediction Market Platform*, http://www.augur.net/whitepaper.pdf, 2018

[3] Gnosis team, *Gnosis whitepaper*, https://gnosis.pm/resources/default/pdf/gnosis-whitepaper-DEC2017.pdf, 2017

[4] Buterin, Griffith, *Casper the Friendly Finality Gadget*, https://arxiv.org/pdf/1710.09437.pdf, 2017

[5] Steem team, *Steem, An incentivized, blockchain-based, public content platform*, https://steem.io/SteemWhitePaper.pdf, 2017

[6] Ethereum Foundation and various contributors, *Sharding FAQ*, https://github.com/ethereum/wiki/wiki/Sharding-FAQ, 2018

[7] Poon, Buterin, *Plasma: Scalable Autonomous Smart Contracts*, https://plasma.io/plasma.pdf, 2017

[8] Coleman, *State Channels*, http://www.jeffcoleman.ca/state-channels/, 2015

[9] Teutsch, Reitwießler, *A scalable verification solution for blockchains*, https://people.cs.uchicago.edu/ teutsch/papers/truebit.pdf, 2017

[10] Warren, Bandeali, *0x: An open protocol for decentralized exchange on the Ethereum blockchain*, https://0xproject.com/pdfs/0x_white_paper.pdf, 2017

[11] Tim Roughgarden, *Algorithmic Game Theory, Stanford course*, https://www.youtube.com/watch?v=z1QZqYuiGa8, 2013

[3] http://solidity.readthedocs.io/en/latest/

[4] https://en.wikipedia.org/wiki/The_DAO_(organization)

[12] Tim Roughgarden, *Advanced Mechanism Design, Stanford course*, https://www.youtube.com/watch?v=1MG2k83bZmI, 2014

[13] Tim Roughgarden, *Approximately Optimal Mechanism Design: Motivation, Examples, and Lessons Learned*, https://arxiv.org/pdf/1406.6773.pdf, 2014

[14] Mavridou, Laszka, *Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach*, https://arxiv.org/pdf/1711.09327.pdf, 2017

[15] Matthew Di Ferrante, *Mechanism Design Security in Smart Contracts*, https://medium.com/@matthewdif/mechanism-design-security-in-smart-contracts-87f08555b38b, 2017

[16] Bernhard Mueller, *How Formal Verification Can Ensure Flawless Smart Contracts*, https://media.consensys.net/how-formal-verification-can-ensure-flawless-smart-contracts-cbda8ad99bd1, 2017

[17] Several contributors, *A Next-Generation Smart Contract and Decentralized Application Platform, Decentralized Autonomous, Organizations*, https://github.com/ethereum/wiki/wiki/White-Paper#decentralized-autonomous-organizations, 2018

[18] Sutton, Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998

[19] Mnih1, Kavukcuoglu1, Silver, et al., *Human-level control through deep reinforcement learning*, Nature, vol 518, 2015

[20] Vickrey, *Counterspeculation, Auctions, and Competitive Sealed Tenders*, The Journal of Finance. 16 (1): 8–37, 1961

[21] Hadfield-Menell, Dragan, Abbeel, Russell, *Cooperative Inverse Reinforcement Learning*, https://arxiv.org/pdf/1606.03137.pdf, 2016

[22] Wood, *ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER*, https://ethereum.github.io/yellowpaper/paper.pdf, 2018

[23] Hartford, Wright, Leyton-Brown, *Deep Learning for Predicting Human Strategic Behaviour*, https://papers.nips.cc/paper/6509-deep-learning-for-predicting-human-strategic-behavior, 2016

[24] Leyton-Brown, *Modeling Human Strategic Behavior: Behavioral Economics to Deep Learning*, https://www.youtube.com/watch?v=w4yvpOLd-5M, 2017

[25] Bellemare, Naddaf, Veness, Bowling *The Arcade Learning Environment: An Evaluation Platform for General Agents*, http://www.marcgbellemare.info/static/publications/bellemare13arcade.pdf, 2013

[26] *Deep Reinforcement Learning Doesn't Work Yet*, https://www.alexirpan.com/2018/02/14/rl-hard.html, 2018

[27] Karpathy, *CS231n Convolutional Neural Networks for Visual Recognition*, http://cs231n.github.io/neural-networks-3/

[28] Mnih, Kavukcuoglu, Silver, Graves, Antonoglou, Wierstra, Riedmiller, *Playing Atari with Deep Reinforcement Learning*, https://www.cs.toronto.edu/ vmnih/docs/dqn.pdf, 2013

[29] Schulman, *The Nuts and Bolts of Deep RL Research (video)*, https://www.youtube.com/watch?v=8EcdaCk9KaQ, 2017

[30] Schulman, *The Nuts and Bolts of Deep RL Research (slides)*, http://rll.berkeley.edu/deeprlcourse/docs/nuts-and-bolts.pdf, 2017

[31] Silver, *UCL course on RL*, http://www0.cs.ucl.ac.uk/staff/D.Silver/web/Teaching.html, 2015

[32] Silver, Hubert, Schrittwieser, *Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm*, https://arxiv.org/pdf/1712.01815.pdf, 2017