Trading Agent Competition with Autonomous Economic Agents

Demonstration

David Minarsch, Marco Favorito, Ali Hosseini, Jonathan Ward*
Fetch.ai
Cambridge, UK

ABSTRACT

In this demonstration, we introduce a system that facilitates trading agent competitions.[1] Competitions mirror a *Walrasian Exchange Economy*. Each agent is endowed with a set of digital assets and preferences over them. Agents then trade these assets with each other to increase their respective utilities. They negotiate one-on-one to arrive at an optimal trade, and if successful, settle their transaction trustlessly on an emulated permissionless blockchain. This system is a precursor to a trading platform for digital assets and crypto-tokens in which agents trade on behalf of their users.[2]

KEYWORDS

Trade; Blockchain; Engineering Multi-agent Systems; Innovative Applications; Negotiation; Markets

ACM Reference Format:
David Minarsch, Marco Favorito, Ali Hosseini, Jonathan Ward. 2020. Trading Agent Competition with Autonomous Economic Agents. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 4 pages.

1 INTRODUCTION

(**Contribution**) We introduce a Trading Agent Competition (TAC) system accompanied by an open source TAC Python package. There is a rich history of trading agent competitions [9], but this work has no direct affiliation with any of them.

This system provides a competition environment in which autonomous economic agents engage in bilateral trades of digital assets. The agent's performance is evaluated according to the performance of their strategy within each market compared with the other agents that are present. In the competition, agents negotiate one-to-one in a trustless environment to arrive at a trade that is optimal in their view, and settle their transactions via a special controller agent which emulates the role of a smart contract [3] on a blockchain-based distributed ledger system [8].

In the real world, an agent would represent an individual or group of people, and would be tasked with looking after their interest by maximising a defined utility function. To achieve this, the agents must be made aware of their owners' preferences and values [1]. In the competition, each agent is explicitly given a representation of their owners' preferences over all digital assets at the beginning of each round. During the competition, the goal of each agent is to maximise its owners' benefit by engaging in profitable trades that increase their utility under the specified preferences.

(**Application**) The competition, in its current form, is a tool that simulates a trading environment for tokens, digital assets and crypto-currencies [5] whereby traders are modelled as autonomous agents. The tool helps us study and understand market behaviours, the effectiveness of market mechanisms and the strategies of the trading agents themselves [6].

In this system, the representation of crypto-traders via agents enjoys a high degree of admissibility, mainly because the aim for the agents is not to approximate the behaviour of actual human traders. Rather agents will be traders that act on behalf of human users in the next iteration of this system. As such, the competition is a precursor to a trading platform whereby agents trade digital assets and crypto-tokens on behalf of human users on, e.g., the Ethereum blockchain [10].

2 THE COMPETITION

2.1 The model

The competition follows a traditional Walrasian economy [7].

(**Agents**) The competition consists of a set $A = A_{baseline} \cup A_{model-based}$ of agents partitioned into *baseline* and *model-based* which compete in the trading competition (see section 3). There is also a special *controller* agent c that runs the competition.

(**Tokens**) There are n sets of digital assets (henceforth, *tokens*) $X = \langle X_1, \dots, X_n \rangle$ where each X_i represents one token type and each $x \in X_i$ is an instance of X_i. In general, we may have $|X_i| \neq |X_j|$, i.e. different aggregate supply across tokens. There is also a special *numeraire token* (henceforth NT) X_0 which serves as a unit of account and medium of exchange to agents.

(**Current holdings**) Each agent a's *current token holding* is $\mathbf{x}^a = \left\langle x_1^a, \dots, x_n^a \right\rangle$ where each $x_i^a \subseteq X_i$ is the set of instances of token i agent a currently possesses. Each agent a's *current NT holding* is x_0^a and it is always the case that $\sum_{a \in A} x_0^a = |A| \times NT\_amount$. Agents can have no instance of a token i at some point in time but never a negative amount, i.e. no borrowing. At the beginning of a round in the competition, every agent is endowed with at least some $base\_amount > 0$ of each token and NT_amount. Tokens can only be traded in integer amounts and are thus non-divisible.

(**Preferences**) Agents are assigned *preferences* on tokens by the controller agent c. Each agent $a \in A$ has a transitive preference relation $\preccurlyeq_a$ which totally ranks any possible combination of token bundles. In practice, each agent a has a utility function u^a which is quasi-linear in tokens:

$$u^a(x_0^a, \mathbf{x}^a) = x_0^a + g(\mathbf{x}^a) = x_0^a + \sum_{i \in \{1,\dots,n\}} s_i^a \times f(|x_i^a|) \tag{1}$$

such that $s_i^a > 0$ and

[*] Correspondence to: david.minarsch@fetch.ai
[1] The code is available at https://github.com/fetchai/agents-tac.
[2] A video of the demonstration can be accessed at https://youtu.be/iJ0qS1RdgGw.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

$$f(|x_i^a|) = \begin{cases} \ln(|x_i^a|) & \text{if } |x_i^a| > 0 \\ -L & \text{otherwise} \end{cases} \tag{2}$$

In Equation 1, x_0^a is agent a's NT holding, s_i^a is the utility parameter a assigns to token i, $f(|x_i^a|)$ parameterizes the number of instances of token i that agent a has, and $L > 0$ is a large constant. An agent a's utility parameters for tokens are $s^a = \langle s_1^a, \ldots, s_n^a \rangle$ where s_i^a is a's utility parameter for token i. Equation 3 ensures the sum of the utility parameters for all tokens are the same across agents:

$$\sum_{i \in \{1,\ldots,n\}} s_i^a = 1 \tag{3}$$

With the specific design of the utility function in Equation 1, we implement the well studied Cobb-Douglas function [2] for $g(\mathbf{x}^a)$ which has the gross-substitutes property. This property means that an increase in the price of one token causes rational agents to demand more of the other tokens.

(**Trade cost**) We introduce a *trade cost k* to explicitly model transaction costs that incur on blockchain-based financial systems. Each transaction incurs the same cost k.

2.2 The competition setup

2.2.1 Controller Agent. In the competition, the controller agent c takes on the dual responsibilities of transaction settlement and competition management, emulating the functionalities of, respectively, a distributed ledger system and smart contracts.

2.2.2 Phases. The trading game has the following three phases:
Pre-trading: in which agents register with the controller agent.
Trading: consists of k game instances $g_1, \ldots, g_k$ s.t. at each g_i:

- The controller agent sends every participant a new draw of token endowments and preferences.
- Agents trade with each other.
- The controller agent registers and settles transactions, effectively keeping track of the token holdings of every agent and only accepting those transactions which are cryptographically signed by the transacting parties.
- After some set time period or a sustained period of no trades, whichever occurs first, the game instance finishes. The controller agent constructs a league table containing the final scores and ranking.

Post-trading: the controller agent reports the final league table which is a (weighted) average of all game instance league tables.

3 AGENTS

There are two types of agents in this competition: *baseline* and *model-based.* Agents in the competition focus on discovering how they can arrive at an optimal bundle - as defined by their preferences - through successive trades. Emphasis for agents is placed on a) finding the right agents to trade with, and b) doing the right trade with them.

After finding each other, agents in the competition negotiate one-on-one in order to reach agreement on a trade. The negotiation protocol agents use is inspired by the FIPA ACL [4]. This means, there is a multi-step dialogue during which agents send messages of the form $\mathrm{P}(c_1, \ldots, c_n)$ where P, called a *performative*, conveys the type of the message and $c_1, \ldots, c_n$ are the contents (e.g. Request(*resource*), Propose(*offer*, *price*)). Table 1 lists the allowed messages and for each message specifies its valid replies.

Table 1: Negotiation messages

Message	Contents	Replies
$cfp(q)$	$q : query$	$propose(o,p)$ $decline()$
$propose(o,p)$	$o : offer$ $p : price$	$propose(o',p')$ $accept()$
$accept()$		$match-accept()$
$decline()$		
$match\text{-}accept()$		

3.1 Baseline Agents

Baseline agents are the most basic agents provided in the TAC Python package that can participate in a competition.

A baseline agent a uses its utility function $u^a(\mathbf{x}^a, x_0^a)$ to compare different states and make decisions. In general, a baseline agent a compares two states with token and NT holdings of respectively $\mathbf{x}^a, x_0^a$ and $\tilde{\mathbf{x}}^a, \tilde{x_0}^a$, by calculating the *marginal utility* $u^a(\tilde{\mathbf{x}}^a, \tilde{x}_0^a)$ - $u^a(\mathbf{x}^a, x_0^a) = g(\tilde{x}^a) - g(\mathbf{x}^a) + \tilde{x_0}^a - x_0^a$. For an exchange that results in agent a's token and NT holdings to change from $\mathbf{x}^a$ to $\tilde{x}^a$, respectively x_0^a to $\tilde{x}_0^a$, to pay off, the change in the marginal utility has to be positive.

A baseline agent a simultaneously offers to buy and sell (if possible) an instance of every token i in her current holding $\langle x_1^a, \ldots, x_n^a \rangle$. The price p the agent is willing to sell is $p \geq g(\mathbf{x}^a) - g(\tilde{\mathbf{x}}^a)$, where $\tilde{x}^a$ is the current token holding $\mathbf{x}^a$ minus the instance to be sold. On the other hand, the price p' the agent is willing to pay to acquire an instance is: $p \leq g(\tilde{x}^a) - g(\mathbf{x}^a)$, where $\tilde{\mathbf{x}}^a$ is the current token holding $\mathbf{x}^a$ plus the token instance to be bought.

3.2 Model-based Agents

A model-based agent uses information it gains from acceptances and declines of her proposals to create a multi-armed bandit model of price for each token. It uses this price model to offer the price it assumes has the highest likelihood of becoming a successful trade.

The agents start with a uniform beta distribution (i.e. where $\alpha = \beta = 1.0$) for each price bin. During the competition, each time an agent gets a successful or failed trade involving a specific price bin and token, she updates the distributions. The prices are then sampled from the model by checking for the distribution with the highest success probability.

4 CONCLUSION

We provided a system that allows autonomous agents to engage in bilateral trades of digital assets.

Two agent strategies are included in the system's package, i.e. baseline and model-based. However, the system allows agents with other strategies to be developed and entered into the competition.

As the next step, the system will be converted into a trading platform for digital assets, in which the role of agents change from simulating traders to performing trades autonomously on behalf of their users.

REFERENCES

[1] Katie Atkinson and Trevor Bench-Capon. 2016. States, goals and values: Revisiting practical reasoning. *Argument & Computation* 7, 2 - 3 (November 2016), 135 – 154.

[2] Murray Brown. 2017. *The New Palgrave Dictionary of Economics*. Palgrave Macmillan UK. 1–4 pages.

[3] Christopher Clack, Vikram Bakshi, and Lee Braine. 2016. Smart Contract Templates: foundations, design landscape and research directions. *arxiv:1608.00771. 2016* (08 2016).

[4] IEEE FIPA Standards Committee. 2001. *Communicative Act Library Specification*. Technical Report. Foundation for Intelligent Physical Agents.

[5] Garrick Hileman and Michel Rauchs. 2017. Global cryptocurrency benchmarking study. *Cambridge Centre for Alternative Finance* 33 (2017).

[6] Graham Kendall and Yan Su. 2003. The co-evolution of trading strategies in a multi-agent based simulated stock market through the integration of individual learning and social learning. *Proceedings of IEEE* (2003), 2298–2305.

[7] Alfred Marshall. 2013. *Principles of economics*. Palgrave Macmillan UK.

[8] Roger Maull, Phil Godsiff, Catherine Mulligan, Alan Brown, and Beth Kewell. 2017. Distributed ledger technology: Applications and implications. *Strategic Change* 26, 5 (2017), 481–489.

[9] Michael P. Wellman, Amy Greenwald, and Peter Stone. 2007. *Autonomous Bidding Agents, Strategies and Lessons from the Trading Agent Competition*. The MIT Press.

[10] Gavin Wood et al. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper* 151, 2014 (2014), 1–32.