Designing and Validating an Agent-based Commodity Trading Simulation

Shih-Fen Cheng       Yee Pin Lim
School of Information Systems
Singapore Management University
{sfcheng, sharonlim}@smu.edu.sg

Abstract

In this paper, an agent-based commodity trading simulation platform is presented, with focus on simulation design and validation. We propose a novel combination of event-based approach and event study method for market dynamics generation and validation. In our event-based approach, the simulation is progressed by announcing news events that affect various aspects of the commodity supply chain. Upon receiving these events, market agents that play different roles, e.g., producers, consumers, and speculators, would adjust their views on the market and act accordingly. Their actions would be based on their roles and also their private information, and collectively the market dynamics will be shaped. The generated market dynamics can then be validated by a variant of the event study method. We demonstrate how the methodology works with several numerical experiments and conclude by highlighting the practical significance of such simulation platform.

1 Introduction

Commodity trading is probably one of the most ancient economic activity, and the spot markets for commodities are in existence since the dawn of the human history. Over the centuries, the scope for the so-called “commodity” has grown from agricultural commodities to include metals and energy; in recent years, even “virtual entity” like carbon emission rights is considered to be a type of commodity and is traded actively just like traditional commodities. The nature of trading has also evolved from primitive barter exchange (direct exchange of goods or services without monetary instrument) to more sophisticated forward contracting between producers and consumers (agreement to buy or sell at a fixed price at a future period), to formal Futures exchanges with clearing houses guaranteeing transactions. Since the scope of the commodities is casted so wide and the available financial instruments are so rich, managing the commodity trades effectively has become more and more important but also extremely challenging.

What makes trading in commodity markets unique and challenging, despite their similarities to stock and bond markets, is the physical transactions that are behind all the financial trades in any form of commodity market. Although the volume in the financial trades (commodity derivatives) has already overtaken the physical trades, physical transactions are still critically important. This is because the balance of supply and demand and the resulting spot prices in the physical transactions are still the fundamental forces that are
behind the commodity market, and no matter how sophisticated the used financial instruments are, all of them still need to closely reference these spot prices.

Which is why trading in commodity market is challenging: physical transactions are affected by not only supply and demand of the commodity, but also all the physical elements that link together the supply and demand sides. For example, in a global commodity supply chain, transportation disruption or freight rate changes will propagate through the chain and generate regional imbalances in supply or demand; the resulting impact could be felt in all the related industries and commodity classes. Other factors, like new legislations, changes in regulatory policies, and abnormal weather, could also exert complicated and significant impacts on the commodity market. Therefore, to trade successfully in a particular commodity market, one needs to be very familiar with the physical properties and the supply chain of that commodity. These requirements are the primary barriers in training successful commodity traders. On the other hand, these sophisticated requirements probably also help to explain why fully automated trading has not taken over the commodity trading yet, as is already happening in the stock, bond, and foreign exchange markets.

This is what motivates our research in the commodity trading simulation. On one hand, we would like to create a commodity trading simulation that is realistic enough so that novice commodity traders could be trained effectively. On the other hand, we are also interested in studying human trader’s trading behaviors in the face of complicated environment, with the ultimate goal of making software agents trade just as human traders in the commodity market.

Despite the fact that there is a vast amount of literature in economics and finance on commodity price modeling, we find them not suitable for our purposes. One of the major missing features in these models is the explicit links between physical events and the price dynamics. These links are important because one of the highly valued skills in trading commodity is the correct readings of the physical events and also the ability to carry out appropriate trades with these understandings. To address this need, we thus propose an event-based simulation model in which the price dynamics is created by a series of well-defined physical events. By introducing causality links between events and price dynamics into the simulation, we also grant ourselves the ability to create scenarios that are rarely seen but important, e.g., the recent commodity boom and the subsequent market crash.

Creating event-based market dynamics could be quite daunting and complicated if it is to be accomplished by a centralized model. In this paper we propose a constructive approach which is widely adopted by researchers in the area of agent-based computational economics. Stated at very high level, our idea is to introduce multiple agents, each of whom plays a specific role in the physical market, and when an event is announced in the simulation, each agent, depending on its role, private information, and also the event properties, will react accordingly. By acting jointly, corresponding market dynamics could be created. As demonstrated in our experiment, it is shown that complicated price dynamics could be generated with fairly simple agent strategies.

One of the major challenge of our approach is the validation of the generated price dynamics as a result of the events that are introduced. Unlike other agent-based simulations, real-world data necessary for the validation usually is not available. This is because the events that are introduced to our simulations could be completely fictitious, and the sequence in which they appear could also be arbitrary. One way to address this is to perform validation based on the data generated by the simulation itself. To achieve this, the event-study method which is widely applied in financial or economic analysis is introduced. We will demonstrate how to adopt and modify this well-studied method so that we could address our unique needs.

This paper is organized as follows. Section 2 provides an overview on our trading simulation system, highlighting the roles of different components. Section 3 describes the most important component in our system — the market agents. Our focus will be the trading logic behind different types of market agents. Section 4 introduces the general concept and steps of an event study approach. We also describe the statistical tests based on the general event study method that we use in validating the occurrence and the strength of an event. In Section 5, we demonstrate by numerical examples on how various types of events could be
detected. In Section 6, we further demonstrate how to validate that correct market responses have indeed been generated for events of different strengths. Finally, we will conclude the paper in Section 7 by highlighting standing issues and future research directions.

2 System Overview

There are three important components in our commodity trading simulation (see Figure 1): a) human traders; b) market server (servicing market mechanism and dispatching events); c) market agents (including hedgers and speculators, which will be described in detail later).

![Figure 1: The architecture of the commodity trading simulation.](image)

Since one of our design goals is to provide an intuitive and straightforward trading simulation for training novices, we have proposed a highly simplified scenario that is composed of only a futures market for some specific commodity (the exact type of the commodity can be specified by the user). Although the spot market is purposely hidden from the human traders, it is included in our consideration when creating market agents. To streamline the trading, we greatly simplify the market making process (dropping all the tedious steps in finalizing a transaction) and assume that the matchings of all transactions are instantaneous without default and are handled by a standard Continuous Double Auction (CDA). In most of the simulations, the participating human traders are assumed to be pure speculators with no intention to be involved in the physical transactions. For simplicity, we assume that for now we will ignore the daily settlement of futures contracts. This implies that margin calls will not be modeled and the cash flows will be computed only when a transaction is made (establish or close out certain position). However, we do require that human traders should not exceed their position limits at all times and all positions should be closed out before the simulation ends.

The market mechanism, event dispatcher, action monitor, and all the required communication infrastructures are developed based on AB3D [2], a generic market game server.

The list of events is predetermined by the scenario designer, where each event is defined by the following parameters (a sample event can be seen in Figure 2):
Crude Oil Rises After OPEC Cut Output in January

Time

t0 ts te

News announced

Effective time window

Realized function
of impact

Figure 2: A typical event with all the important parameters.

**Title and content:** This information provides qualitative information regarding the event and is mainly for the human traders.

**Arrival time:** The time \(t^0\) when the event is delivered and visible to all agents (both human traders and market agents). It is assumed that all agents receive event at the same time, without discrimination.

**Impact:** This parameter specifies the type and the strength of the event. The strength of an event is specified by an integer in \([1, 5]\), where 5 indicates the strongest event and 1 indicates the weakest event. An event could be either bullish or bearish, and is indicated by the sign of the impact. Events with positive and negative impacts would be bullish and bearish respectively. Note that agents might generate completely different responses to the same event, and the “realized function of impact” plotted in Figure 2 is the aggregated response from all market agents.

**Effective time window:** The event is only effective within the time window \([t^s, t^e]\). With these two parameters we could create events with short-term or long-term impacts. Also, by overlapping a series of events we could model the escalation of a major event (e.g., the impact on crude oil price exerted by the progression of the war in Iraq).

For all agents, an event only comes into existence when its designated arrival time has passed; no agent could peek into the future list of events. Once an event arrives, only the “title and content” will be made available to the human traders. Both “impact” and “action time” are only available to the market agents. Besides qualitative event information, human traders also have access to the latest market information in the form of bid quotes (however, they cannot peek into the order book). As stated earlier, the only constraints applied to their behaviors are that they need to obey their trading limits at all times, and they have to unload all their positions at the end of the simulation.

### 3 Designing Market Agents

As in most other training-oriented trading simulations, fidelity and realism are some of the most important features we would like to achieve. To create simulations with high fidelity, a number of academic and commercial applications has deliberately created a linkage between trading simulations and the real market (e.g., see UMOO at [http://www.umoo.com/](http://www.umoo.com/) and FACTSim at [http://www.factsim.org/](http://www.factsim.org/)). In these cases, real-time market data feeds are what constitute the market dynamics. However, this type of design, although popular, is not suitable for our purpose. Again, the reason here is that our primary focus is on explicitly incorporating the impacts of physical events in forming the market dynamics, and for this purpose, the real market data is far too noisy (this is the same reason why we didn’t adopt a well-studied econometric model for generating market dynamics). To be consistent with the event-based model we
proposed earlier, we break down important market actors into independent agents, embed appropriate trading strategy into each agent, and then let these agents interact in order to create market dynamics collectively. Therefore, the design of market agents is the most critical part of our commodity trading simulation.

Using agents in modeling complex economic or financial systems is not new, in fact, a large number of literature has been devoted to the subject of “Agent-based Computational Economics” (ACE) [3, 4, 5]. The ACE models is probably best explained in Tesfatsion’s own words [5]:

“The defining characteristic of ACE models is their constructive grounding in the interactions of agents, ... Starting from an initially specified system state, the motion of the state through time is determined by endogenously generated agent interactions.”

Our model follows similar constructive principle, creating various agent roles that interact to generate market dynamics. In our market model, we place the modeling emphasis on hedgers and speculators. Their internal respond models, which take event occurrences and market states (prices) as inputs, will determine how the market evolves. The roles and the models of hedgers and speculators will be explained in detail in the following two subsections.

It should be noted that the framework in which we develop our market agents is open and flexible. Therefore if necessary, ourselves or any third-party could easily develop new types of agents and add it to the mix.

3.1 Hedger Model

Hedgers are the original users of the futures market. They are usually producers or consumers of the commodity who would like to lock in at some specific prices and quantities well before the time of production (for producers) or usages (for consumers). These producers and consumers provide liquidity and are the main drivers of the supply and demand in the market.

To properly incorporate producers and consumers in our model, we assume that they exhibit stationary behaviors, i.e., the rate of their production and consumption will be stationary. Since we assume that only one futures market exists for this commodity, this assumption implies that all producers and consumers have to constantly establish new hedges in this market, and their collective actions will create the market dynamics accordingly. We further assume that all producers and consumers will employ a simple hedge-and-forget strategy, meaning that they will establish new hedges based on their own needs (new produces or usages), the current market condition, and their expectation; once the hedges are established, they will hold them to the end (in other words, no dynamic hedge will be considered).

A pseudo-code implementation of the hedger’s model based on the above simplifications is listed in Algorithm 1. The listed code is meant to be generic so that both producers and consumers can use it, and the agent-specific information is specified as parameters. In Algorithm 1, agent role, perceived price, lower and upper bounds on capacity, and bidding interval are all specified as parameters.

Hedgers will continue to submit bids at agent-specific intervals $int_i$ (line 13) as long as the simulation is still running. As stated earlier, the critical decisions to be made in the bidding routine are the price and the quantity of the bid. The quantity of the bid reflects agent’s hedging need and will be randomly drawn from a uniform distribution $U[Q_{i_{min}}^i, Q_{i_{max}}^i]$, in which $Q_{i_{min}}^i$ and $Q_{i_{max}}^i$ represent lower and upper bounds on agent $i$’s demand (supply) capacity. By manipulating $Q_{i_{min}}^i$ and $Q_{i_{max}}^i$ across all hedger agents, we could control how sensitive would the market reacts to human traders’ actions. Larger aggregated $Q$’s usually corresponds to slower movement and vice versa.

The price of the bid will depend on individual hedger agent’s latest price forecast, which is computed based on the current spot price (which is internal to the agent, and not explicitly modeled) and also the estimated impact of the latest event. We first assume that a perfect forecast can be made and derive the perfect forecast. We will then discuss how agents could approximate this perfect forecast.
Algorithm 1 Hedger’s bidding routine.

```plaintext
HEDGERAGENT(role_i, \mu_i, Q^\text{min}_i, Q^\text{max}_i, int_i)

1: while IS SIMULATION RUNNING() is true do
2:     q ← DISCRETEUNIFORM(Q^\text{min}_i, Q^\text{max}_i)
3:     (P_a, P_b) ← GET LATEST QUOTE()
4:     P_e = \mu + \lambda((P_a + P_b)/2 - \mu)
5:     P_f = P_e + GET EVENT IMPACT() + \sigma \text{NORMAL}(0, 1)
6:     if role_i is producer then
7:         p = \text{NORMAL}((P_f + P_a)/2, 1)
8:         SUBMIT BID(p, −q)
9:     else
10:        p = \text{NORMAL}((P_f + P_b)/2, 1)
11:       SUBMIT BID(p, q)
12:    end if
13:    SLEEP(int_i)
14: end while
```

Commodity prices tend to be mean-reverting at the level of marginal cost of production. This has been shown to be true in a wide variety of commodities (e.g., see [6]). To model commodity prices, we thus have to look for models that are mean-reverting. One such model is the Ornstein-Uhlenbeck (OU) process, which is originally proposed by Vasicek [7] for modeling interest rates. Because of its mean-reverting feature, the OU process has been a popular choice for modeling commodity prices in recent years. Besides mean-reversion, events also play an important role in our simulation. To incorporate the event impacts into the model, we follow the jump diffusion model proposed by Merton [8]. Finally, to properly implement the resulting price evolution model in our discrete-time simulation, we prefer a discrete-time model (as opposed to the original continuous-time model). Our price forecasting model is thus a discrete-time variant described by Blanco and Soronow [9].

In our price forecasting model, the forecast at time $t$ is determined by:

$$P_f(t) = P_e(t) + n(t) + \sigma \epsilon(t),$$

(1)

where $P_e(t)$ is the equilibrium price derived from the market price information, and it follows the mean reversion model:

$$P_e(t) = \mu + \lambda(P_a(t) + P_b(t)/2 - \mu),$$

(2)

where $\mu$ is the perceived price, $\lambda$ is the weight for mean correction, and the current market price is estimated by the average of $P_b(t)$ (bid price, which is the price of the highest untransacted long bid) and $P_a(t)$ (ask price, which is the price of the lowest untransacted short bid). $n(t)$ is the estimated impact of recent events at time $t$, and is both event and agent specific. The definition of $n(t)$ depends on how event impact is specified. Suppose the impact of the event is constant over time within its effective time window, then $n(t)$ can simply be computed by summing up the impact values of all events that are still effective. That is,

$$n(t) = N_i(\sum_{j: t^e_j \leq t \leq t^s_j} v_j),$$

(3)

where $t^e_j$, $t^s_j$, and $v_j$ are the start and the end of the effective time window and the impact value for event $j$ respectively (all of them are event parameters). $N_i(.)$ is a function for agent $i$ to map the cumulative impact value to the drifting factor in the price prediction. The last term of (1), $\sigma \epsilon(t)$, is the inherited
volatility associated with the underlying commodity where \( \epsilon(t) \) follows standard normal distribution \( N(0, 1) \) (white noise) and \( \sigma \) is a parameter governing volatility. In Algorithm 1, \( P_e(t) \) and \( P_f(t) \) are computed in line 4 and 5 respectively using Equations (2) and (1). Note that in line 5, the full detail on the \( n(t) \) computation is assumed to be handled by function GETEVENTIMPACT(). Past data is not used in our current implementation thus all occurrences of \( t \) are dropped in Algorithm 1.

There is a long list of reasons why such perfect prediction could not be obtained for all hedger agents. It could be that the agents have limited capabilities, both in terms of modeling or information acquisition. It could also be that the established hedge is not perfect (i.e., the desired maturity date and the commodity category don’t match futures contract exactly). To mimic such limitation, we assume that the realization of agent’s true prediction would follow a normal distribution with variance 1 and mean situated at the middle of the predicted price and the market price (we have to admit that the choice of the distribution and the parameters might be ad hoc in this case since we don’t have a good intuition about this). Since \( P_a(t) \) and \( P_b(t) \) are deemed as the market prices by producers and consumers respectively, the prices submitted by producers are normally distributed with mean \((P_f(t) + P_a(t))/2 \) (line 7), while the prices submitted by consumers are normally distributed with mean \((P_f(t) + P_b(t))/2 \) (line 10). Finally, the bid is submitted in line 8 and 11 respectively for producers and consumers. For producers, \( q \) needs to be negative since short hedges (sell bids) are required.

3.2 Speculator

While “mean-reverting” hedgers constitute the “fundamental” part of the simulated market, most of the market volatility, on the other hand, is generated by the speculator agents. In our simulation, we adopt the classical zero intelligence (ZI) strategy [10] in constructing our speculator agent. However, to prevent ZI agents from destroying the market trend (generated by producer and consumer agents), we limit the price range to be \([P_b(t) - \delta, P_a(t) + \delta]\), where \( P_b(t) \) and \( P_a(t) \) are bid and ask prices as defined previously, and \( \delta \) is an agent parameter, controlling how aggressive this ZI agent should be in creating volatility. In most of our simulations, we simply set \( \delta \) to be the bid increment.

After the price is randomly decided, the ZI agent will flip a fair coin and use it to decide whether it will take long or short positions. Since each ZI agent is granted the same trading limit as human traders, it will also randomly decide how much remaining position allowance it would devote to the new trade. Again, just like human traders, ZI agents are required to exit all positions at the end, and they are programmed to gradually exit their positions when the end draws near.

4 Validating Market Dynamics: The Event Study Approach

As described in Section 3, we take a constructive approach in generating market dynamics by constructing agents with different physical roles. One major issue that has been prevalent in all similar efforts is the validations of the generated agent-based simulations. For us this is particularly challenging, since the event we introduce to the simulation might be totally fictitious, and there is no real data we can use in validating the generated market dynamics. Therefore, to properly validate the market dynamics generated in our simulation, we will have to rely on the simulation output.

There are two types of validations we would like to perform. First, we would like to confirm the occurrence of the event. That is, the positive or negative price changes in response to the event should be significant enough to be detected. Second, we would like to make sure that the strength of the event is properly generated. In other words, we would like to affirm that the impact level specified by the event is properly realized. In economics and finance, the studies on how events affect market prices are quite common, and one of most popular choices is the event study approach. In the subsequent sub-sections, we will
provide a general introduction to this powerful methodology and apply necessary modification to it for our validation efforts.

4.1 Introduction

The modern event study approach, introduced by Fama et al. [11], is widely applied in economics and finance in measuring the effects of an event on the value of firms using financial data. As MacKinlay [12] puts it:

“The usefulness of such a study comes from the fact that, given rationality in the marketplace, the effects of an event will be reflected immediately in security prices.”

This feature is important since the impact of an event could be measured in a relatively shorter time periods of several weeks or days, as opposed to several months or even years if we use other more lagging indicators (e.g., production levels, revenues). This description also explains our choice in picking the event study as the tool in validating the event-based trading simulation we built, since our simulation progresses by letting market agents react to events.

The event study approach has already been applied in detecting a wide-variety of events, e.g., mergers and acquisitions, earning announcements (both examples are discussed in [12]), or even macroeconomic news [13]. In most applications, common equity price of the studied firm is used; however, the event study approach could also be applied to other type of securities with little modifications. An example of such application is in commodity derivatives. The impacts of various kinds of events on a wide-variety of commodity derivatives are studied extensively by applying event studies. Some example events include monthly governmental reports on supply and demand [14], public policy and legislation [15], and disease outbreak [16].

Roughly speaking, event studies try to statistically test for abnormal returns from the security prices within a predetermined time window. Due to practical limitations resulting from data (could be related to both security prices and events), many event study variants have been suggested. Binder [17] reviewed a wide variety of event study methodologies, and discussed some frequently encountered empirical issues.

4.2 Event Study Procedure

These past researches on event studies provide a sound analytical framework for us to analyze whether the artificial events generate consistent market dynamics in our simulation. Despite the difference in the statistical techniques applied, most event study methodologies have the following general procedures. To stay focused, we only include steps that are relevant to our analysis (complete coverage on the methodology can be found in [12]):

1. First, the events of interest are identified, and for each identified event, a time window surrounding that event is defined so that security price information could be collected. For event \( j \), let the beginning and the ending of the time window be \( \tau_{1j} \) and \( \tau_{2j} \) respectively.

2. Second, determine the firms to be included as data samples. The selection criteria usually involve data availability and the characteristics of the firm such as market capitalization and industry, so that an unbias set of samples could be constructed.

   In our case, since the target we are studying is the price of the commodity derivative itself, the concept of firms does not apply here. Alternatively, we will construct the sample set by executing multiple simulation instances for the same event series (this is conceptually identical to the reactions of multiple securities to the same event series).
3. To understand the impact of the event, we need a measure on the event-induced abnormal return (AR), which is simply the actual return minus the normal return over the event window. The normal return is the expected return when no event is introduced. In our study, the normal return is computed by finding the mean price of the commodity before the occurrence of any event.

To accommodate events with multiple periods, we define the cumulative abnormal return (CAR) for event $j$ as:

$$\text{CAR}(\tau_1^j, \tau_2^j) = \sum_{\tau=\tau_1^j}^{\tau_2^j} \text{AR}_\tau,$$

where $\text{AR}_\tau$ represents the abnormal return in time $\tau$.

With the computed CAR, various statistical tests could be administered, depending on our purposes. In our study, we are interested in testing the occurrence of an event and asserting that proper response strengths are generated. To detect the occurrence of event $j$, we define the null hypothesis to be no event occurrence, which can be tested by computing:

$$\theta = \frac{\overline{\text{CAR}}(\tau_1^j, \tau_2^j)}{\text{var}(\text{CAR}(\tau_1^j, \tau_2^j))^{1/2}}.$$

$\overline{\text{CAR}}(\tau_1^j, \tau_2^j)$ is the estimated CAR from the experiment instances. $\theta$ is the test statistics and should follow the standard normal distribution of $N(0, 1)$. To test the occurrence of a bullish or bearish event, we should define a one-tailed positive or negative alternative hypothesis (i.e., $H_1: \overline{\text{CAR}}(\tau_1^j, \tau_2^j) > 0$ for bullish events, $H_1: \overline{\text{CAR}}(\tau_1^j, \tau_2^j) < 0$ for bearish events). In either case, the rejection of the null hypothesis could lead us to the conclusion that a bullish or a bearish event has occurred.

To assert that impact levels from 1 to 5 indeed generate appropriate price dynamics in the market, we would like to establish that the events with higher impact levels indeed produce larger CAR. To establish this result statistically, we compare mean CAR for consecutive levels in pairs using $t$-tests, i.e., comparing levels 1 and 2, 2 and 3, and so on. The null hypothesis will be no difference in mean CAR. The alternative hypothesis is similarly defined to be one-tailed, stating that the mean CAR from the stronger event is greater than that of the weaker event, i.e., $H_1: \overline{\text{CAR}}(\tau_1^j, \tau_2^j) > \overline{\text{CAR}}(\tau_1^k, \tau_2^k)$, assuming that event $j$ is stronger than event $k$.

In the following two sections, we will demonstrate how to apply event studies in validating our commodity trading simulation.

5 Validating Event Occurrence

In this section, we follow the event study method described in the previous section, and apply the suggested statistical test. To simplify the simulation and avoid clustering effects from overlapping events, we create a special scenario with only one event. For the market agents, we include 12 producers, 13 consumers, and 2 ZI agents (to emulate human trader’s actions). Both producers and consumers are constructed following Algorithm 1. The length of a simulation day is defined to be 1 second, and the length of the simulation is just over 370 days. The event occurs in day 160 (which is known to all agents), and the event window is defined to be 20 days before the event occurrence and 20 days after the event occurrence. In other words, $\tau_1 = 140$ and $\tau_2 = 180$. For bullish, bearish, and no event scenarios, the impact levels are set to +5, -5, and 0 respectively. Sample price evolutions of these three scenarios are shown in Figure 3(a).

To collect enough sample data points, the same scenario is executed 15 times. Following the event study procedures described in Section 4.2, we test the null hypothesis for bullish, bearish, and no event cases.
Figure 3: (a) Sample price dynamics of the one-event scenarios. The event is announced in day 160. (b) The cumulative abnormal returns (CAR) for bullish, bearish, and normal (no event) scenarios.

One sample CAR from each case is plotted in Figure 3(b). Note that in Figure 3(b), we use -20 and 20 to represent 20 days before and after the event occurrence respectively.

For both the bullish and the bearish cases, the \( p \)-values \( \sim 0 \), implying that positive/negative abnormal returns are statistically significant. For the normal (no-event) case, \( p \)-value \( \sim 0.065 \), indicating that no significant abnormal return is detected during the event window.

6 Validating Event Strength

As described in Section 4.2, to establish that the impact parameter indeed causes market agents to generate consistent response strength, we should construct a similar one-event scenario as specified in the previous section and vary its impact level from 1 to 5. The setup of the experiment is almost identical to the one described in Section 5, except that we now have five scenarios, each with different impact level.

For each impact level, 20 to 60 samples are generated depending on how variable the samples are. For impact level 3, 4, and 5, the results are particularly noisy, thus we have executed more simulations. The experiment results are summarized in Table 1. From Table 1, we can see that the impact level of an event indeed dictates the strength of response from market agents.

<table>
<thead>
<tr>
<th>Level</th>
<th>Mean CAR</th>
<th>Std. dev. of CAR</th>
<th>Sample size</th>
<th>( p )-value against previous level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6.0424</td>
<td>3.0668</td>
<td>20</td>
<td>–</td>
</tr>
<tr>
<td>2</td>
<td>10.5619</td>
<td>2.8974</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>14.2055</td>
<td>4.6611</td>
<td>60</td>
<td>0.0008</td>
</tr>
<tr>
<td>4</td>
<td>16.6611</td>
<td>5.9619</td>
<td>60</td>
<td>0.0067</td>
</tr>
<tr>
<td>5</td>
<td>21.1381</td>
<td>7.8951</td>
<td>40</td>
<td>0.0009</td>
</tr>
</tbody>
</table>
7 Conclusions and Future Work

In this paper, we have presented our efforts in building an agent-based commodity trading simulation. Our framework is based on the constructive principle widely applied in the agent-based computational economics community, and the actors in our market are based on well-studied theoretical models. The fidelity of the market dynamics generated by the multiagent framework is statistically provable by using the event study method. The novelty of our approach lies in the combination of event-centric approach in generating market dynamics and the use of event studies in validating generated market dynamics.

Our aim in this research is not to create new pricing models for commodities; instead, we have focused on how to construct a highly fidel commodity trading simulator from the building blocks that are well-founded theoretically. The primary application of such framework is currently in training human traders, and as demonstrated by a number of deployments, such exercises are highly valuable for both participants and us [18]. For participants, the value of such simulation is apparent to them since they get to experience realistic trading. For us, the most valuable information we collected during these simulation sessions are the detail action logs from human traders. With this information, we might be able to identify what makes a good trader. On one hand, these analysis results could be used to improve the market agents employed in our simulation. On the other hand, these analyses could also help educators in designing better educational programs that could more effectively teach the art of trading. Ultimately, our platform might be used in benchmarking or diagnosing a trader’s skill. This is the line of future work that is very promising but relies heavily on the availability of a highly fidel and flexible trading simulation, which is the main issue we attempt to address in this paper.

We do notice that a large number of systems have already been developed for the purpose of simulating commodity trading (e.g., besides UMOO and FACTSim, which are mentioned earlier, also see [19] and [20]). However, to the best of our knowledge, none of them is fully flexible in terms of designing scenarios (e.g., deciding properties of the target commodity and the flow of events) and generating market dynamics (in our system it could be easily changed if new market agents are introduced). This flexibility, granted both by the event-based design and multiagent-based approach, is our contribution to the area of commodity trading simulation.

8 Acknowledgements

This work was supported in part by the International Enterprise Singapore and the International Trading Institute at Singapore Management University. The authors would like to acknowledge the dedicated effort by Chao-Chi Liu in building the simulation infrastructure. The authors would also like to thank Samuel Owens and Anil Shamdasani for developing the crude oil and the soybean trading scenarios.

References


