

Моделирование тестовых сценариев поведения участников биржевой торговли

Елена Герасимова
 ОПМИ НИУ ВШЭ
 ул. Мясницкая, дом 20
 Москва, Россия, 101000
 elenagerasimova@gmail.com

Ростислав Яворский
 ФКН НИУ ВШЭ
 ул. Мясницкая, дом 20
 Москва, Россия, 101000
 ryavorsky@hse.ru

В данной работе рассматривается тестирование информационных систем на примере программ биржевой торговли. В работе описана имитационная мультиагентная модель биржи и описаны макро состояния, упрощающие анализ системы. Созданную модель можно использовать и дополнять для исследования биржевой торговли и моделирования различных стратегий участников и ситуаций на рынке, а также для тестирования реальных биржевых систем. Построен граф состояний системы из двух игроков и на его основе получены сценарии для тестирования реальных торговых систем, а также выведены правила оценки количества возможных перемещений системы по состояниям и построения сценариев для системы из большего числа участников.

Ключевые слова: мультиагентное моделирование, сценарный анализ, тестирование распределенных систем, тестирование на основе моделей, граф макро-состояний, системы биржевой торговли

I. ВВЕДЕНИЕ

A. Предмет исследования

Современная фондовая биржа работает и управляется онлайн, и, по сути, является электронной торговой площадкой. Сейчас биржа - это компьютерная программа, которая собирает заявки от участников и исполняет их. Неправильная работа этой программы может привести к значительным убыткам не только участников биржевой торговли, но и, например, компании, чьи акции торгуются на бирже и даже людей, которые не участвуют в торговле. Нас интересует тестирование подобных систем с целью выявления ошибок или неправильной работы системы.

B. Возможные подходы к задаче тестирования ИС

Есть несколько способов тестирования информационных систем:

- Динамический анализ (полное тестирование) – анализ программы посредством запуска. У полного тестирования есть ряд достоинств: тестируем саму систему и подбираем для нее входные данные и сразу видим результат, так как есть возможность посмотреть на действия реальной системы и сразу выявить ошибку. Однако есть сложности с подбором полного

тестового комплекта и не всегда есть возможность тестировать на реальной системе. Она может быть слишком большой или недоступной для тестирования. Другая важная проблема – необходимо придумать тестовый набор, который бы являлся достаточно полным.

- Статический анализ – работа с кодом с целью выявления ошибок и потенциальных узких мест программы без непосредственного выполнения. Современные компиляторы проводят некоторый синтаксический анализ кода перед запуском и выводят потенциальные ошибки пользователю. Есть более сложные методы анализа кода, такие как использование формальных методов или специальных программ для поиска потенциальных угроз.
- Качественный анализ системы – для некоторых систем можно провести аналитическую работу, чтобы понять и вывести общие правила, если они есть. Этот метод позволяет подбирать те же тестовые наборы, но не перебором, например, а используя правила вывода состояний системы и так далее. Это возможно не для всех систем.

В данной работе мы сосредоточимся на динамическом анализе, а именно, на том, как подобрать для программы биржевой торговли набор тестовых сценариев, чтобы удостовериться, что она работает корректно. Для этого используется имитационная модели системы, которая проще, чем реальная биржа, однако сохраняет базовые свойства и не содержит критических упрощений. Подробно подход тестирования с помощью моделей описан, например, в [1-2].

C. Цель работы

Целью данного исследования является разработка методов и алгоритмов автоматизированного построения тестовых сценариев для использования в интеграционном, системном и приемочном тестировании биржевых систем.

В качестве критерия полноты набора тестовых сценариев рассматривается покрытие графа макро-состояний тестируемой системы.

Основной задачей данной работы являются исследование структуры графа макро-состояний и разработка методов его автоматизированного построения (см. [1] и [2]).

II. ФОНДОВАЯ БИРЖА И СУЩЕСТВУЮЩИЕ ИМИТАЦИОННЫЕ МОДЕЛИ

A. Фондовая биржа

Фондовая биржа – это организованный и регулируемый финансовый рынок, где продаются и покупаются ценные бумаги. Ценная бумага является документом, дающим право на владение каким-либо активом (деньги, акции, имущество, какая-либо продукция и другие ресурсы). Ценные бумаги бывают основные и производные. К основным ценным бумагам относятся акции, облигации, сертификаты, векселя и т.д. Производные ценные бумаги или деривативы – это контракты, обязывающие участников соглашения купить или продать определенный актив в будущем по определенной цене. К производным финансовым инструментам относятся опционы, деривативы, фьючерсы, свопционы и другие. Как правило, деривативы используются не с целью покупки или продажи актива, а являются объектами спекуляции на рынках и инструментами для снижения рисков в будущем. Участниками биржевой торговли являются трейдеры (те, кто торгует в собственных интересах), брокеры (представляют интересы своих клиентов), инвесторы, компании (размещают свои акции на рынке). Трейдеры и брокеры могут выступать в разных ролях в зависимости от своих целей: они могут спекулировать на разнице цен, стремиться сократить убытки или риски, совершать инвестиции и многое другое. Вне зависимости от этого, базовым действием для трейдеров на бирже являются выставление ордеров на покупку или продажу ценных бумаг. Основным инструментом работы трейдеров – ордер, который является заявкой на продажу или покупку определенной ценной бумаги. Ордер содержит наименование ценной бумаги, которая является предметом покупки/продажи, и количество, которое трейдер хочет приобрести или продать. Результатом исполнения ордера является продажа или покупка ценной бумаги. Ордер может содержать цену продажи/покупки. Если ордер не содержит цену, в случае его исполнения сделка происходит по рыночной цене. Рыночная цена бывает двух типов: Ask – рыночная цена спроса, Bid – рыночная цена продажи. Соответственно, в зависимости от типа заявки (на продажу или на покупку), используется соответствующая цена.

Естественно, рыночные цены спроса и предложения отличаются, потому что продавцы всегда стремятся продать товар дороже, в то время как покупатели хотят купить дешевле. Есть несколько типов заявок в зависимости от цены, но в целом, это относится к возможности покупки строго по указанной цене или меньшей/больше в зависимости от того, тип на покупку или продажу.

Сама биржа является торговой площадкой, где трейдеры выставляют ордера. Задача биржи – обеспечение

и проведение сделок. Это включает в себя нахождение соответствующих друг другу сделок, которые могут быть выполнены (например, ордер на продажу и ордер на покупку по одной цене), проведение сделки, обновление счетов и балансов трейдеров, удаление ордеров. Биржа также может устанавливать и собирать комиссионные сборы за услуги.

Раньше биржа была местом, где требовалось физическое присутствие участников и существовали жесты, которые участники использовали, чтобы обозначить свои действия. Сделки обговаривались и заключались устно. Современная биржа оперирует онлайн, торги проводятся на электронной площадке с помощью специальных программ. Таким образом, современная биржа – это распределенная информационная система, которая принимает и исполняет заявки участников биржевой торговли. Каждый участник использует специальную программу для работы на бирже и управления своим балансом и ценными бумагами.

Использование компьютерных программ для торговли привело к появлению алгоритмического трейдинга, когда торгуют программы по заданным алгоритмам. Это, в свою очередь, привело к увеличению частоты торговли и появлению понятия высокочастотной алгоритмической торговли. Алгоритмическая торговля используется повсеместно и крупными институциональными игроками и независимыми трейдерами, причем чаще всего с целью спекуляции. Это привело к громадному увеличению числа сделок за день и уменьшению времени осуществления сделок. Исследования Deutsche Bank в США [3] показывают, что в среднем ордер держится на рынке 22 секунды, при этом торговля осуществляется за доли секунд.

Тем не менее, несмотря на произошедшие изменения, механизмы работы и стратегии трейдеров остались прежними, только происходят значительно быстрее и автоматизировано.

B. Мультиагентные системы

Мультиагентная система – это система, состоящая из некоторого множества агентов, взаимодействующих друг с другом. Основными свойствами такой системы являются:

- Автономность агентов - агенты независимы.
- Ограниченность представления – агенты имеют ограниченные знания о мире.
- Децентрализация – агенты не могут контролировать систему.

Мультиагентные системы обычно имеют большое число агентов с простыми недетерминированными стратегиями поведения и простыми взаимодействиями между агентами. Здесь есть принципиальная проблема – глобальная динамика системы и тот факт, что системы может перемещаться из одного состояния в другое быстро и иногда непредсказуемо.

Мультиагентный подход активно развивается в течение последнего десятилетия и представляет интерес, так как мультиагентное моделирование дает возможность исследовать динамику системы как результат активных взаимодействий агентов в течение некоторого периода. Более того, такое моделирование очень удобно в случаях, когда эксперименты с реальной системой невозможны, а эксперименты с моделью можно повторять много раз с изменением параметров и данных.

Наиболее подходящим и одним из самых распространенных методов моделирования мультиагентных систем является имитационное моделирование. Имитационно моделирование состоит в том, чтобы построить модель реальной системы и процессы в ней так, как они происходят и в реальной жизни. Данный подход используют, когда нет возможности построить аналитическую модель реальной системы или нет возможности для экспериментов с реальным объектом в динамике. Необходимо отметить, что модель может являться относительно простой конструкцией реальности, так и достаточно сложной в зависимости от возможностей и целей моделирования.

III. ОБЗОР СУЩЕСТВУЮЩИХ МОДЕЛЕЙ БИРЖЕВОЙ ТОРГОВЛИ

Исследования финансовых рынков с помощью имитационного мультиагентного моделирования начались в 1980-х и продолжаются до сих пор в различных формах. Все исследования содержат примерно одинаковую структуру искусственной биржи: мир, биржа как торговая площадка и агенты. Мир – это вся моделируемая реальность, где находятся биржа и агенты и все события происходят. В ряде моделей мир – это система, которая содержит агентов и биржу, однако более сложные модели содержат других участников мира или позволяют модели использовать информацию из внешнего мира, получать новости и так далее. Моделируемая биржа – это, как и в реальной жизни, площадка, где трейдеры могут выставлять свои ордера на покупку/продажу ценных бумаг и где происходят сделки. Широко распространено, что модели содержат только трейдеров как участников, потому что это позволяет понять и проанализировать влияние трейдеров (или различных типов трейдеров) на рынок и ценообразование, поведение игроков и исследовать микроструктуру рынка. Нас интересует моделирование биржевой торговли и структура рынка, поэтому мы будем рассматривать существующие имитационные модели биржи как мультиагентной системы.

В статье «Development and testing of an artificial stock market» [4] авторы представляют модель *Genoa Artificial Stock Market*. В этой модели есть два класса: агенты и маркет-мейкер (торговая площадка). На искусственном рынке торгуется один тип акции с заданной начальной ценой. Агенты независимы друг от друга и имеют ограниченный бюджет и количество акций в начале торговли. Начальные параметры каждого трейдера устанавливаются вначале случайным образом. В начале каждого хода игроки размещают свои заявки, учитывая свои ограничения, маркет-мейкер рассчитывает цену как пересечение кривых спроса и предложения и

устанавливает цену. Все заявки на продажу и на покупку, удовлетворяющие цене, исполняются, остальные заявки отбрасываются. Первая цена устанавливается внешне. При этом маркет-мейкером обладает неограниченным запасом ресурсов, чтобы удовлетворить все заявки по установленной цене. Начинается следующий ход. Снова происходит все то же самое: игроки размещают заявки, маркет-мейкер вычисляет цену и исполняет заявки по новой цене. В начале торговли каждому игроку присваивается тип кластера, к которому он принадлежит – оптимист или пессимист. Сначала распределение игроков по кластерам 50 на 50, затем в начале каждого хода генерируется некоторая вероятность попасть в кластер для игроков. Смысл кластеров в том, что в начале хода игроки получают сигнал – пессимисты получают сигнал продавать как можно больше, оптимисты покупать как можно больше. Это сделано, чтобы был баланс заявок и сохранялся положительный баланс у трейдеров и работа продолжалась. В итоге, авторы исследовали рынок с 500 игроками и 10 000 шагов и динамику изменения цен на нем, а также распределение доходов трейдеров на заданных ценах.

Вторая известная модель [3] – модель Santa Fe института Santa Fe Institute. Она была создана между 1980 – 1990 годами и широко использовалась, как авторами модели, так и их последователями. Модель состоит из трейдеров и маркет-мейкера. Цену назначает маркет-мейкер. Трейдеры имеют возможность разместить заявку на продажу или покупку одной акции, при этом на рынке почти всегда перевес одной из групп – больше желающих купить или наоборот продать. Так как в итоге трейдеры из меньшей группы остаются с невыполненными заявками, на следующем шаге цена назначается с учетом разницы в спросе и предложении.

Позднее в модель добавили второй актив. В итоге на рынке продавались два актива – рисковый и безрисковый. Безрисковый актив имел установленный процент возврата инвестиции, рисковый же актив был очень выгодным только с некоторой вероятностью. Впоследствии модель не раз дорабатывалась, и добавлялись новые расчеты для агентов.

Помимо базового устройства рынка модели многие модели используют различные стратегии для трейдеров и исследуют влияние этих стратегий на рынок и ценообразование. Конечно, типы стратегий в моделях зависят от конкретных целей исследования, однако, можно выделить общие методы построения стратегий трейдеров в моделях рынка.

Во-первых, стратегии технического и фундаментального анализа. Это самый распространенный метод построения стратегий. Он состоит в том, что агенты анализируют информацию и делают свои предсказания относительно истинной цены акции. Трейдеры, использующие фундаментальный анализ, считают, что у любой акции есть своя фундаментальная цена и их стратегия, соответственно, состоит в том, чтобы покупать, если цена ниже фундаментальной и продавать, если выше. Фундаментальную цену агенты могут вычислять

несколькими способами (например, используя информацию о компании и ее финансовых и операционных показателях), или такая цена может задаваться агентам внешне. Технический анализ состоит из вычисления будущей цены на основе предыдущих цен и их изменений, а также из нахождения определенных закономерностей в тренде. В техническом анализе используются временные ряды и статистические методы. Это деление трейдеров на две группы похоже на реальную жизнь и многие модели в том или ином виде используют эти стратегии.

Во-вторых, модель [7] предоставляет агентам возможность выбора предиктивной модели. Агенты имеют некоторую базу моделей для предсказания цен и будущих заработков и на каждом шаге могут выбрать какую они будут использовать сейчас. Также доступна общая публичная база моделей, как масс-медиа, которую агенты могут тоже использовать. Более того, все агенты обладают разным доступом: у кого-то есть своя база моделей и есть доступ к общей, у кого-то нет доступа к общей, кому-то доступна только общая база.

В-третьих, большинство моделей включает в себя иррациональных трейдеров как «шум» или как неопытных трейдеров в системе.

Наконец, авторы моделей стараются сделать модель более разнообразной и придают агентам уникальные свойства и характеристики такие как, склонность к риску, наличие информации, функция полезности и другие. Имея такие характеристики, можно исследовать их влияние на рынок или адаптировать рынок к различным условиям.

Завершая обзор методов построения стратегий трейдеров важно отметить, что в реальной жизни трейдеры не всегда придерживаются какой-то одной стратегии поведения и принятия решений и в этом смысле модели упрощают реальность. Однако данный метод позволяет исследовать конкретные стратегии и получать значительные результаты.

IV. РАЗРАБОТАННАЯ МОДЕЛЬ РЫНКА

Существующие модели рынка ценных бумаг достаточно сложные для использования и, кажется, что пока нет универсального инструмента для моделирования рынка. Для целей нашей работы мы решили создать модель рынка ценных бумаг. Разработанная модель состоит из трех классов: биржа, агент, ордер. Агентом является трейдер, который имеет деньги и акции и может выставлять заявки на продажу или покупку акций. Биржа представляет собой торговую площадку, которая собирает заявки участников и исполняет те, которые можно выполнить. Невыполненные заявки остаются на бирже до выполнения.

Модель устроена так, что трейдеры и биржа ходят по очереди. Сначала один из агентов делает ход – размещает или не размещает в зависимости от наличия ресурсов заявку на бирже. Следующий ход за биржей – она смотрит на последнюю размещенную заявку и пытается ее выполнить. Если произошло исполнения заявки, заявка

удаляется с биржи, и обновляются данные трейдера – изменяется денежный баланс и количество акции.

В нашей модели используются лимитные заявки, то есть те, которые дают возможность купить (продать) ценную бумагу по цене не более (не менее, в случае продажи) указанной в заявке. Это дает большую гибкость модели и более того, это делает ее более реалистичной. Если трейдер подал заявку на покупку по цене 25, конечно для него будет выгодно, если заявка выполнится по меньшей цене. В данном случае, 25 – это лимит, верхняя граница цены, по которой он согласен купить.

Важно, что когда биржа ищет подходящую пару выставленной заявке, поиск идет по цене, а не по количеству. В случае подходящей цены, но разного количества, заявка может исполниться не полностью. Например, биржа получила заявку от некоего агента на покупку 100 акций по цене 10 и нашла заявку на продажу 50 акций по цене 8 от другого агента. Цена соответствует лимитной заявке на покупку и даже в два раза дешевле, чем агент заявлял в заявке. Тогда биржа исполнит эти заявки, только заявку на продажу удалит, так как она было полностью выполнена, а заявку на покупку обновит. Можно сказать, выставит на продажу новую заявку с обновленными данными.

В базовой реализации мы позволяем агентам указывать в заявке цену, потому что ценообразование не влияет на цели нашего исследования.

Для программной реализации модели был выбран язык C++, так как это объектно-ориентированный язык, и это подходит для нашего мультиагентного моделирования. В реализованной модели, как было сказано ранее, три класса.

Первый класс – «Агент». Этот класс описывает абстрактного агента на рынке. Считаем, что агент имеет номер (id), баланс денежных средств (balance), количество акции (stock1) и историю своих ордеров (order_history). История содержит только активные заявки, исполненные заявки удаляются. В истории храним не сами заявки, а их номер (id).

Так выглядит класс «Агент» в программной реализации модели, если на рынке торгуется только одна ценная бумага (stock1):

```
struct Agent{
    unsigned id;
    unsigned balance;
    unsigned stock1;
    vector<int> order_history;
};
```

Заявка в модели имеет 5 полей, которые содержат информацию о номере заявки (id), типе заявки () – на продажу или покупку, количестве акций (number), цене за одну акцию(price) и номере агента (id_agent), который выставил эту заявку.

В качестве начальных параметров мы задаем, что денежный баланс агентов определяется случайным образом от 1000 до 15000. Количество акций у каждого от 100 до 1000.

Так выглядит класс, описывающий заявку в модели:

```
struct Order{
    unsigned id;
    unsigned type; //0 - buy, 1 - sell
    unsigned number;
    unsigned price;
    unsigned id_agent;
};
```

Заявки генерируется случайным образом для агентов в заданных интервалах, чтобы был баланс, и чтобы не было слишком большого разброса. При этом заявки удовлетворяют бюджетным ограничениям трейдера.

Чтобы модель была приближенной к реальности, заявки на продажу выставляют по более высокой цене, чем заявки на покупку. Это обусловлено тем, что реальные люди всегда хотят купить дешевле, а продать дороже. Трейдеры также, при этом они еще могут пытаться заработать на разнице цен между покупкой и продажей.

Биржа в модели, как и в реальной жизни, исполняет несколько функций:

- Хранит выставленные заявки
- Исполняет подходящие заявки
- Обновляет данные участников торговли после совершения сделки
- Обновляет данные о выставленных заявках на бирже и у участников.

Соответственно, состояние биржи состоит из двух векторов, которые хранят заявки – один на продажу, другой на покупку. Когда трейдер выкладывает новую заявку, она добавляет ее в нужный вектор и пытается найти в другом подходящую заявку. Если заявка может быть исполнена, она исполняется, и происходит обновление счетов трейдеров, которые участвовали в сделке.

Таким образом, в модели по очереди ходят игрок и биржа. За один ход могут произойти следующие действия: один из игроков выставляет заявку, происходит исполнение двух заявок – сделка между двумя игроками. Это базовые шаги системы.

V. ТЕСТИРОВАНИЕ СИСТЕМЫ

A. Сценарии для тестирования биржевой площадки

У нас есть модель биржи, и мы хотим получить тестовые наборы для тестирования реальной биржи. Важно понимать, что эти тестовые сценарии – не наборы

входных данных, так как биржевая программа не получает ничего на вход. Смысл тестового сценария в том, что он отражает последовательность корректных переходов имитационной модели и значит, та же последовательность должна произойти и на реальной бирже. Другими словами, реальная биржа должна перемещаться по состояниям так же, как и модель. Для этого посмотрим в каких состояния система может находиться и какие вообще могут быть состояния, так как нам важно, чтобы биржа перемещалась по состояниям корректно. Нам важен именно сценарий перемещения, а не конечный результат, потому что у модели есть начальные вершины, но нет конечных, так как торговля может продолжаться сколько угодно и закончится когда угодно по желанию участников.

Базовых действий на бирже всего 2: либо один из игроков разместил заявку, либо произошла сделка по ранее размещенным заявкам. При этом состоянии системы можно описать как совокупность состояний каждого из ее участников. Состояние каждого участника, если на рынке торгуется m типов ценных бумаг, можно описать фиксированным набором целочисленных переменных (x рублей на счету; s_1 акций типа 1, ..., s_m акций типа m). При этом, хорошо бы добавить количество заявок на продажу и на покупку и состояние по каждой заявке, чтобы можно было отслеживать какая именно заявка выполнялась и правильное списание с баланса и обновление данных. Если детально описывать состояние биржи в каждый момент, если количество трейдеров N , то состояние биржи это совокупность состояний всех ее участников. Так как минимальное изменение денежного баланса участников возможно на единицу и акций так же на единицу, то мы получаем, что возможно огромное количество состояний и переходов из состояния в состояние в такой системе.

Для того, чтобы упростить состояния системы, а также анализировать не вообще все перемещения, которых много и не всегда конечно, а перемещения системы из классов состояний, которые наиболее общие и интересные, мы рассматриваем макро состояния. Интересно не как изменился баланс участника или сколько акций он продал, а какое действие по смыслу произошло, так как от этого и зависит изменение баланса и данных. Мы ввели состояния для агентов: 3 для покупки и 3 для продажи. С точки зрения продажи акций он может находиться в трех состояниях: 0 – нет акций; 1 – есть акций, нет заявок на продажу; 2 – есть акции, есть заявки на продажу. С точки зрения покупки акций он может находиться в трех состояниях: 0 – нет денег; 1 – есть деньги, нет заявок на продажу; 2 – есть деньги, есть заявки на продажу. Эти макро состояния – классы эквивалентности состояний. Состояние (0;1) содержит состояния, когда у агента 1 акция, 2, 3 и так далее. Мы свернули детальные состояния в макро состояния. Данные состояния были выбраны, так как они позволяют понять, какое действие произошло – ничего, агент разместил заявку, произошла сделка и определяют готовность агента к действию в том смысле, что мы можем понять, может ли он участвовать в торговле или только размещать заявки. Каждый агент одновременно находится в некотором состоянии на продажу и на покупку, то есть всего возможно 9 состояний для одного

агента. Будем записывать состояние агента в виде пары чисел $(a;b)$, где a – его макро состояние на покупку акций, b – макро состояние на продажу акций. При этом не все переходы между этими состояниями возможны. Например, из состояния $(0;0)$, когда у него нет денег и нет акций, агент не может никуда перейти вообще, кроме как в то же состояние. Таким же образом, например, из состояния $(0;1)$, в котором у него нет денег, но есть акции, он может переместиться только в то же самое состояние или в состояние $(0;2)$, в котором у него уже есть заявки на продажу акций и не может в другие. Этот переход перемещает агента в другой класс – когда он был в состоянии $(0;1)$, он не мог участвовать в торговле, так как у него было заявок, а по состоянию $(0;2)$ можно понять, что агент может торговать.

Важно отметить, что, например, переход из состояния $(1;2)$ в $(1;2)$ не всегда означает бездействие. Это может означать также, что агент разместил еще одну заявку на продажу акций, однако, на выбранном уровне абстракции это не очень важно, так как интересует не сколько у него заявок 10 или 11, а наличие ордеров у него вообще: нет или есть, и не важно сколько.

Понятно, что при таком определении макро состояний, главные базовые действия, которые могут совершать трейдеры – выставить ордера и торговать. Например, переход из состояния $(1;0)$ в состояние $(2;0)$ означает, что трейдер выставил заявку на покупку акций. А переход из состояния $(2;0)$ в состояние $(1;1)$ означает, что одна из его заявок на покупку исполнилась, и у него появились акции, при этом исполнившаяся заявка была его единственной заявкой.

Матрица возможных переходов из состояния в состояние состоит из 81 возможного ребра. Однако, не все переходы возможны, как было замечено ранее. Аналитически было получено, что всего возможно 29 переходов из 81. Эмпирические данные, полученные из многократного запуска программ, подтверждают это.

Ниже приведен граф возможных переходов для одного игрока. Считаем, что он не один на рынке и может вести торговлю, однако мы смотрим только на него. На графе отображены переходы, которые нас интересуют. Возможны еще переходы из вершины в саму себя (петли), которые не отображены, чтобы не зашумлять картинку.

Начальные вершины обозначены зелёным цветом, только с них игроки могут начинать участие в биржевой торговле. Это понятно, потому что игрок не может начать с состояния, когда у него есть заявки. Торговля начинается с того, что у агента есть деньги или акции, и он пришел на биржу. Соответственно, следующим его ходом может быть размещение заявки на продажу или покупку, в зависимости от наличия ресурсов.

Когда мы рассматриваем одного игрока, получаем, что он может перемещаться из одного состояния в другое в течение торговли, однако всего его количество возможных переходов за раз ограничено 27 ребрами. Это примерно треть от количества ребер в матрице, что значительно

облегчает понимание состояния игрока и тестирование такой системы.

Если рассматривать биржу с двумя участниками и аналогичными состояниями, то можно наблюдать уже за состоянием системы в целом. Будем считать, что состояние системы – это совокупность состояний каждого из участников в определенный момент. Тогда состояние системы из двух игроков будем обозначать как $(a;b)(a_1; b_1)$, где $(a;b)$ – состояние первого участника, а $(a_1; b_1)$ – состояние второго участника торговли.

Получается, что всего у такой системы 81 состояние, а если мы хотим посмотреть все возможные переходы из состояний в состоянии нам надо заполнить матрицу 81×81 , в которой 6561 ребро может быть максимум. Не имея модели и не зная, какие ребра в матрице существуют, мы были бы вынуждены тестировать торговую систему случайно, отслеживая корректное выполнение, но не зная, какие еще случаи возможны.

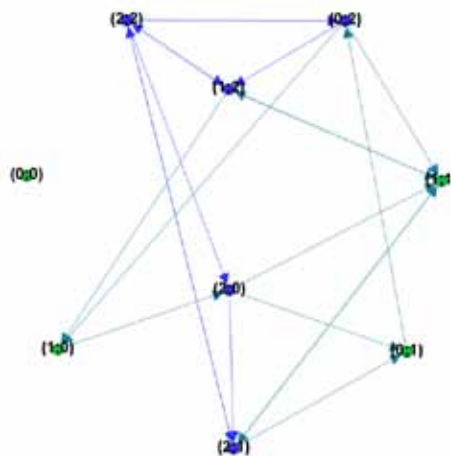


Fig. 1. Граф макро состояний одного трейдера

На самом деле, число ребер в этой матрице ограничено сверху числом $29 \times 29 = 841$. Это объясняется тем, что у нас есть матрица переходов для одного игрока и мы знаем, что всего для одного игрока возможно 29 переходов. Матрица для одного игрока учитывает наличие других игроков, как минимум одного, но рассматривает только одного. При этом эта оценка не является точной, потому что необходимо еще исключить несовместные состояния. Однако, необходимым, но не достаточным, условием существования ребра в матрице для двух агентов является существование соответствующих ребер в матрице для одного игрока. Это можно сформулировать как: для существования ребра из $(a_1; b_1)(c_1; d_1)$ в $(a_2; b_2)(c_2; d_2)$

матрице для двух игроков необходимо, но не достаточно, чтобы в матрице для одного игрока существовали ребра $(a_1; b_1) \rightarrow (a_2; b_2)$ и $(c_1; d_1) \rightarrow (c_2; d_2)$.

Однако знание верхней границы не облегчает задачу поиска возможных переходов.

Построим граф для системы из двух игроков. Так выглядит граф переходов системы из двух игроков. Матрица смежности была сгенерирована многочисленными запусками программы и аналитически. Как и прежде, зеленые вершины – начальные. Всего их 16.

В такой системе возможно всего 349 переходов из потенциальных 6561. Это сильно облегчает задачу тестирования, когда мы знаем, что всего может реализоваться 349 переходов, которые мы знаем, и, значит, реальная система должна вести себя также. В ней должны реализовываться те же переходы.

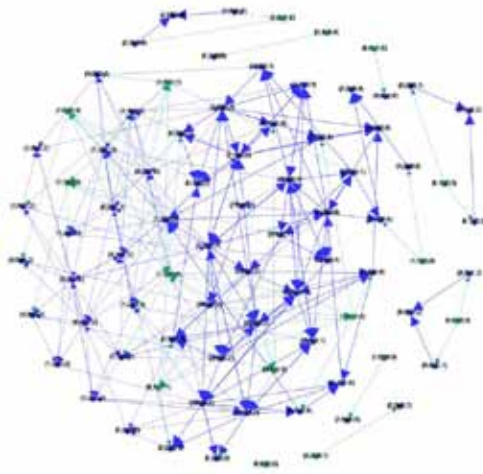


Fig. 2. Граф макро состояний для системы из двух трейдеров

Видно, что у графа одна большая компонента связности и 7 маленьких. Маленькие компоненты связности, состоящие из 1 - 4 вершин, отражают ситуацию, когда агенты не могут торговать. У них либо у обоих нет денег, либо нет акций и единственное действие, которое им доступно – это размещать заявки на покупку или продажу, которые не будут выполнены, потому что у них идентичные ресурсы. Большая компонента связности отражает взаимодействия агентов, когда у них есть возможность торговать. В этой компоненте 9 начальных вершин и 330 возможных ребер для перемещения.

Как было сказано ранее, на бирже нет конечного состояния, поэтому агенты могут торговать, а система перемещаться из состояния в состояние сколько бесконечно.

Если мы хотим тестировать систему из двух игроков, нам достаточно этого графа, потому что по нему мы можем строить сценарии для тестирования. Сценарий представляет собой последовательность допустимых переходов системы из вершины в вершину. Смысл сценария в том, что он покрывает некоторый набор допустимых переходов и, значит, подходит для тестирования реальной системы. То, что работает на модели, должно работать и на реальной торговой площадке. Для тестирования системы необходимо, чтобы набор сценариев для тестирования был исчерпывающим, только так мы сможем проверить корректность программы. Исчерпывающий набор тестовых сценариев – такой, который покрывает 100% возможных переходов системы, то есть существующих ребер в графе.

Известно, что задача построения точного графа макро состояний для мультиагентных систем общего вида неразрешима (это можно доказать, например, моделированием произвольной машины тьюринга M в виде системы простых агентов, и сведением задачи о достижимости определенных состояний M к вопросу о наличии ребра в графе макро состояний для соответствующей M мультиагентной модели).

С помощью случайной генерации сценариев построенной модели мы можем построить оценку снизу для этого графа (если какое-то ребро реализуется в случайно найденном сценарии, значит оно точно есть, но в обратную сторону не верно – если какое-то ребро не найдено в процессе случайных запусков модели это еще не значит, что переход невозможен).

С другой стороны, статический анализ модели позволяет построить верхнюю оценку для графа макро состояний, а именно, если для какого-то перехода доказано, что он невозможен, значит он невозможен. Если же не доказано, то это не значит, что перехода нет. См. подробнее [11] и [12].

В результате мы получили матрицу и соответствующий граф для системы из двух агентов, по которому можно строить сценарии для тестирования реальной торговой площадки. Мы построили тестовый набор сценариев, покрывающий все возможные ребра. Данные сценарии должны быть осуществимы и на самой системе торговли.

Очень важно так же, что, так как за один шаг на бирже может происходить только одно действие, а действий всего два – один из трейдеров выкладывает заявку или происходит сделка между двумя трейдерами, то матрица для системы из двух агентов является базовой. Когда агентов много и происходит сделка, то можно зафиксировать состояния всех агентов, кроме участвующих в сделке, потому что их состояния не изменятся, а для торгующих агентов важно, чтобы переход системы был корректным и возможным в матрице для двух. Если считать, что за один шаг может произойти, например, две сделки или больше, то пары, участвующие в сделке, должны перейти по существующему ребру в графе для системы из двух игроков, а остальные участники остаться в прежних положениях.

В. Качественный анализ модели

Когда мы вводили новые состояния для участников, мы, по сути, вводили более общие состояния, то есть мы объединяли состояния в классы эквивалентности. Новые состояния позволили нам строить меньшие по объему конечные графы для системы и исследовать возможные переходы и получать сценарии для тестирования. Однако, дальше нас интересует вопрос: как и можно ли исследовать систему из любого количества участников? Для системы из одного и двух агентов мы можем строить графы, пользуясь моделью и аналитической работой. Но как количество участников меняет возможные переходы в системе и есть ли какие-то правила, позволяющие оценить количество таких переходов, или может даже их получать?

Этот вопрос интересен, потому что у нас есть матрица для одного участника, которая подразумевает наличие других участников. Система из двух участников является расширением системы из одного участника и это означает, что если мы хотим построить без модели граф для двух агентов, мы должны добавить второго участника, то есть подняться на уровень выше. Мы понимаем, что когда мы добавляем второго агента, классы состояний в матрице состояний для одного агента раскрываются на состояния из двух участников. Например, ребро $(0;1) \rightarrow (0;2)$ раскрывается в 81 ребро для двух участников, так как для каждого участника возможно всего 9 состояний. При этом, если два класса были соединены ребром в случае одного агента, то какие-то из состояний получившиеся из этих классов добавлением второго участника, тоже будут связаны. Однако, мы не знаем какие.

Возникает вопрос: как переходить на уровни выше, когда участников больше, и как оценивать число переходов в системах из n участников.

У нас есть матрица для одного игрока, в которой 29 ребер. Мы хотим перейти к системе из двух игроков, и мы знаем оценку сверху для количества ребер равную $29 \times 29 = 841$. Разобьем все ребра на три группы: когда ничего не произошло (9 ребер), когда агент разместил заявку (6 ребер), когда произошла сделка (17 ребер). В первой и третьей группе есть пересечения из 3 ребер, поэтому в сумме их получается 32. Попробуем понять, сколько и какие ребра получаются из этих ребер, если мы добавляем второго участника. Возможные случаи:

- Первый агент не совершал действий, и второй тоже бездействовал. Всего таких переходов $9 \times 9 = 81$. Матрица для двух игроков имеет размер 81×81 , и эти состояния являются ее диагональю.
- Первый агент не совершал действий, второй разместил заявку (или наоборот). В таком случае в 6 ребер первого участника добавляется состояние второго участника, состояние которого не меняется. Число таких состояний $= 6 \times 9 = 54$. С учетом перестановки игроков получаем 108.
- Произошла сделка между агентами. В данной ситуации мы знаем, как изменилось состояние одного участника и все сводится к подбору соответствующих параметров для второго

участника. Например, есть ребро $(0;2) \rightarrow (1;2)$. Мы должны найти параметры второго участника в переходе $(0; 2) (c_1; d_1) \rightarrow (1; 2) (c_2; d_2)$. Это значит, что заявка на продажу акций у второго выполнена и у него появились деньги. Какие могли быть состояния у второго игрока? Он совершил переход $(c_1; d_1) \rightarrow (c_2; d_2)$. Так как у первого купили, значит у второго была заявка на покупку, то есть $c_1 = 2$, а про d_1 мы ничего сказать не можем, оно могло быть любое. На этом шаге мы уже понимаем, что ребро выглядит так $(0;2) (2; d_1) \rightarrow (1; 2) (c_2; d_2)$. Параметры c_2 и d_2 зависят от параметров c_1 и d_1 . $C_1 = 2$ могло перейти в c_2 равное 0, 1 и 2, потому что это могла быть его единственная заявка, за которую он заплатил все деньги, либо у него могли остаться деньги, либо у него остались и деньги и заявки. Единственное ограничение на d_1, d_2 в данном случае – оно не может быть равно 0, потому что если агент купил, то у него точно должны появиться акции. Мы получаем 9 ребер.

$$\begin{aligned} (0;2) (2; 0) &\rightarrow (1;2) (0; 1) | (1;2) (1; 1) | (1;2) (2; 1) \\ (0;2) (2; 1) &\rightarrow (1;2) (0; 1) | (1;2) (1; 1) | (1;2) (2; 1) \\ (0;2) (2; 2) &\rightarrow (1;2) (0; 2) | (1;2) (1; 2) | (1;2) (2; 2) \end{aligned}$$

Исходное ребро переходит в 9 ребер для двух агентов, плюс с учетом перестановки получаем 18 ребер.

Это будет верно для 16 ребер из 17. Последнее ребро $(2;2) \rightarrow (2;2)$, так как может произойти и продажа, и покупка у обоих агентов переходит не в 18 ребер, а в 30. Всего таких ребер 318.

Из этих расчетов в матрице для двух участников число ребер равно $81 + 108 + 16 \times 18 + 30 = 507$. Однако, мы знаем, что действительно в этой матрице 349 ребер. Это произошло потому что, при таком подборе есть пересечения и некоторые ребра мы посчитали несколько раз. Тем не менее, мы получили более точную оценку числа ребер, а самое главное – теперь есть правила вывода и понятно как конструировать ребра для двух участников из ребер для одного участника. По сути, даже не нужна модель для того, чтобы построить матрицу, потому что есть правила вывода переходов.

Теперь мы хотим перейти от матрицы для двух игроков к матрице для трех игроков. Для этого мы точно так же разобьем на три группы ребра матрицы для двух агентов – когда ничего не произошло, и система перешла в то же состояние, один из участников выложил заявку, произошла сделка между двумя участниками.

- Все три агента не совершали действий – $9 \times 9 \times 9 = 729$ ребер для этих переходов, вся диагональ в матрице.
- Один выложил заявку, два других ничего не сделали. Таких ребер $6 \times 9 \times 9 = 486$ ребер. Учитывая перестановки, получаем $486 \times 3 = 1458$ ребер.

- Происходит торговля между двумя. Это когда двое торгуют, а третий остается в том же состоянии. Таких ребер $160 \times 9 = 1440$ ребер. Мы умножили 160, а не 318, так как у нас есть точная матрица для двух игроков и она уже не содержит совпадения, поэтому можно использовать не оценку 318, а точное число. Плюс мы должны учесть возможные перестановки игроков, их всего три, и получим $1440 \times 3 = 4320$ ребер.

Получаем, всего для трех игроков количество ребер = $729 + 1458 + 4320 = 6507$. Сама матрица для трех игроков содержит 729 вершин и между ними может быть максимум 531 441 ребро. А в действительности их всего 6507. Причем, нам не потребовалась модель, многочисленные запуски, чтобы заполнить эту матрицу и аналитическая проверка.

Таким же образом можно переходить к большему числу участников торговли, находить число переходов и даже, имея правила вывода для двух игроков, строить переходы. Это важно, потому что система из 10 игроков, например, содержит 9^{10} вершин и надо проверить 9^{20} ребер, что невозможно, и хранить такую матрицу тоже. Тем более, что ребер действительно будет немного и можно хранить их только их.

Более того, данный подход очень удобен при тестировании системы, так как есть понимание, как переходить от системы из двух игроков к системам с большим количеством игроков и создавать тестовые сценарии для биржевой площадки.

VI. ЗАКЛЮЧЕНИЕ

Результатом данной работы являются имитационная модель биржи на языке C++, сценарии для тестирования системы из двух участников, а так же правила выводов матриц состояний для большого количества участников из системы из двух агентов. Созданную модель можно использовать и дополнять для исследования биржевой торговли и моделирования различных стратегий участников и ситуаций на рынке, а так же для тестирования реальных биржевых систем. С помощью нее был получен граф состояний системы из двух игроков, а также выведены правила оценки количества переходов и построения сценариев для системы из большого числа участников, что позволяет строить матрицы состояний для данных систем, не используя запуски модели системы. Это облегчает работы с системами из большого числа участников, так как не требует хранения всей матрицы переходов, а только возможных. При этом не надо заполнять всю матрицу, которая может быть огромной.

Важно отметить, что предложенная модель обладает рядом ограничений:

- Рассматривается только торговля с участием маркет-мейкера, хотя на многих (а то и на большинстве) современных торговых площадках такая роль отсутствует (NASDAQ, PTC и т.д.).

- В работе рассматривается только один алгоритм сведения заявок – с упорядочиванием исключительно по цене, хотя существуют и другие (с учетом объема заявок и т.д.).
- Не рассматривается маргинальная торговля, при которой все имеющиеся у трейдера активы могут использоваться как гарантийное обеспечение новых сделок (т.е. трейдер может выставить заявку на покупку даже когда у него недостаточно средств, если при этом он имеет другой ликвидный актив).
- Отсутствует операция снятия трейдером активной заявки.
- Не рассматриваются различные типы заявок (например, рыночные с ограничением цены, заявки все-или-ничего).

Дальнейшими направлениями работы могут являться:

- Усложнение модели биржевой торговли – добавление комиссии биржи, других типов заявок, введение параллельной торговли на нескольких биржах.
- Усложнение модели взаимодействия между агентами.
- Интеграция данного подхода с другими исследованиями в области использования моделей для тестирования систем биржевой торговли, см. [13], [14].

Авторы выражают благодарность рецензентам за конструктивные замечания, которые послужили улучшению данного текста.

VII. ЛИТЕРАТУРА

- [1] Veanes, Margus, Colin Campbell, and Wolfram Schulte. Model-based software testing and analysis with C#. Vol. 1. Cambridge: Cambridge University Press, 2008.
- [2] Grieskamp, Wolfgang, et al. "Generating finite state machines from abstract state machines." ACM SIGSOFT Software Engineering Notes 27.4 (2002): 112-122.
- [3] High-frequency trading Better than its reputation?, Deutsche Bank research report. - 2011.
- [4] Marchesi M. Cincotti S., Focardi S., Raberto M. Development and testing of an artificial stockmarket. Proceedings of Modelli Dinamici in Economia e Finanza. – 2000
- [5] Arthur B. W., Lebaron B., Palmer R. The time series properties of an artificial stock market. Journal of Economic Dynamics and Control. –1999. – №23. – Pages 1487-1516.
- [6] Hull J.: Options, Futures & other Derivatives, 8th Global Edition. Pearson Education Limited. - 2012.
- [7] Kendall G., Su Y.: A multi-agent based simulated stock market - testing on different types of stocks. In: Congress

- on Evolutionary Computation, CEC 2003, pp. 2298-2305(2003)
- [8] Xiaorong Chen, Shozo Tokinaga. Multi-agent-based modeling of artificial stock markets by using the co-evolutionary GP approach. In: Journal of the Operations Research Society of Japan. – 2004. - Vol. 47. - No. 3.- pp.163-181
- [9] Wooldridge Michael. An Introduction to MultiAgent Systems. John Wiley & Sons. -2002.
- [10] Ческидов Б.М. : Модели рынков ценных бумаг. – СПб: Питер.- 2005
- [11] Gurevich, Yuri, and Rostislav Yavorskiy. "Observations on the Decidability of Transitions." Abstract State Machines 2004. Advances in Theory and Practice. Springer Berlin Heidelberg, 2004. 161-168.
- [12] Veanes, Margus, and Rostislav Yavorsky. "Combined Algorithm for Approximating a Finite State Abstraction of a Large System." (2003): 86-91.
- [13] Ольга Буянова, Алёна Булда, Алексей Зверев «Применение симуляторов рынка ценных бумаг для тестирования систем агрегации и распределения информации о котировках (Ticker Plant)» Инструменты и методы анализа программ Tools & Methods of Program Analysis TMPA-2013: материалы Международной науч.-практ. конф.- Кострома: Из-во Костромского гос. технол. ун-та, 2013. сс. 215-241
 См. <http://elib.spbstu.ru/dl/2/4154.pdf>
- [14] Наталья Прядкина, Антон Крюков «Использование МВТ-подхода для верификации систем мониторинга и контроля на фондовых биржах.» Инструменты и методы анализа программ Tools & Methods of Program Analysis TMPA-2013: материалы Международной науч.-практ. конф.- Кострома: Из-во Костромского гос. технол. ун-та, 2013. сс. 254-263