

# Отчёт по практике

Курилова Олега, группа 204

# Общий план презентации

1) Статьи и задачи

2) Моя задача

3) Биржа

4) Итог

# Статьи и задачи

Мною были прочитаны 3 статьи и решены некоторые задачи из сборника тигров.

Статьи:

1) 2 статьи про задачу максимального разреза в графе.

2) Сборник задач с неожиданными ответами.

В одной из статей был описан алгоритм, который предлагалось реализовать, но так его уже заняли ребята, то я решил придумать сам себе новые цели на практику.

# Моя задача.

## Введение

Из сборника ТиГров первый раздел мне показался самым интересным, в плане неожиданности ответов. Потом я стал думать, а как можно поменять данные задачи, чтобы они были сложнее и интереснее в решении. В частности, я решил изменить условие первой задачи, которое по моему мнению должно было изменить ответ. Я разрешил Усаме раз в 100 ночей передвигаться на  $\pm 2$  шахты. Мне показалось, что такое нововведение способно спасти нашего персонажа от бесконечных отрядов спецназа.

В более общем случае задача звучит как есть  $n$  шахт и раз в  $n$  ночей усаме можно передвигаться  $\pm 2$  шахты.

# Моя задача.

## Точное решение на стандартную задачу

Первый вопрос, который возникает после решения обычной задачи, а можно ли быстрее. Напомню, что обычный ответ получается по четности за  $2n$ ,  $n$  - количество пещер. Немного подумав, можно понять, что можно не смотреть крайние шахты, так как противник из них может бежать только в 2 стороны, поэтому ответ можно получить за  $2(n - 2)$ , то есть  $2n - 4$ . Но финальный ли это результат?

Программно было установлено, что ответ положительный и меньше нельзя.

На скрине ответ и конец стратегии для  $n = 26$ , как можно увидеть, решение алгоритма 1 в 1 повторяет нашу стратегию. (Шахты нумеруются с 0)

```
010101010101010101010111111-- Время: 19 -- Шахта для удара 19
1010101010101010101010111111-- Время: 20 -- Шахта для удара 20
0101010101010101010101011111-- Время: 21 -- Шахта для удара 21
1010101010101010101010101111-- Время: 22 -- Шахта для удара 22
0101010101010101010101010111-- Время: 23 -- Шахта для удара 23
1010101010101010101010101010-- Время: 24 -- Шахта для удара 24
0101010101010101010101010100-- Время: 25 -- Шахта для удара 24
1010101010101010101010101000-- Время: 0 -- Шахта для удара 23
0101010101010101010101010000-- Время: 1 -- Шахта для удара 22
1010101010101010101010100000-- Время: 2 -- Шахта для удара 21
0101010101010101010101000000-- Время: 3 -- Шахта для удара 20
1010101010101010101010000000-- Время: 4 -- Шахта для удара 19
0101010101010101010100000000-- Время: 5 -- Шахта для удара 18
1010101010101010101000000000-- Время: 6 -- Шахта для удара 17
0101010101010101000000000000-- Время: 7 -- Шахта для удара 16
1010101010101010000000000000-- Время: 8 -- Шахта для удара 15
0101010101010000000000000000-- Время: 9 -- Шахта для удара 14
1010101010100000000000000000-- Время: 10 -- Шахта для удара 13
0101010101000000000000000000-- Время: 11 -- Шахта для удара 12
1010101010100000000000000000-- Время: 12 -- Шахта для удара 11
0101010101000000000000000000-- Время: 13 -- Шахта для удара 10
1010101010000000000000000000-- Время: 14 -- Шахта для удара 9
0101010100000000000000000000-- Время: 15 -- Шахта для удара 8
1010101000000000000000000000-- Время: 16 -- Шахта для удара 7
0101010000000000000000000000-- Время: 17 -- Шахта для удара 6
1010100000000000000000000000-- Время: 18 -- Шахта для удара 5
0101000000000000000000000000-- Время: 19 -- Шахта для удара 4
1010000000000000000000000000-- Время: 20 -- Шахта для удара 3
0100000000000000000000000000-- Время: 21 -- Шахта для удара 2
0000000000000000000000000000-- Время: 22 -- Шахта для удара 1
Length of way: 48
Bush win!!!
```

# Моя задача.

## Улучшение

Также вычислительно был получен ответ для моей модифицированной задачи.

Оказывается, что Буш, несмотря на то, что ему отрезали возможность пользоваться четностью, все же может выиграть Усаму.

Причем мы можем понять общую стратегию Буша, а именно отсекал треугольниками. На следующем кадре я покажу стратегию Буша для 20 шахт.

Для начала нужно ещё рассказать, что есть что на моем скрине.

Сначала идут позиции, где игрок мог оказаться, используя +2 за свои  $n$  ходов, потом идут позиции, где игрок мог оказаться, не используя +2 за последние количество ходов  $\% n$ . Далее идёт время по модулю  $n$  и в какую шахту мы бьем

# Моя задача. Скрины

```
11111111--11111111-- Время: 1 -- Шахта для удара 0
11111111--01111111-- Время: 2 -- Шахта для удара 1
11111111--10111111-- Время: 3 -- Шахта для удара 2
11111111--01011111-- Время: 4 -- Шахта для удара 3
11111111--10101111-- Время: 5 -- Шахта для удара 4
11111111--01010111-- Время: 6 -- Шахта для удара 5
01111111--00101111-- Время: 7 -- Шахта для удара 1
10111111--00001111-- Время: 0 -- Шахта для удара 2
11111111--01011111-- Время: 1 -- Шахта для удара 3
11111111--10101111-- Время: 2 -- Шахта для удара 4
11111111--01010111-- Время: 3 -- Шахта для удара 5
11111111--10101010-- Время: 4 -- Шахта для удара 6
11111110--01010100-- Время: 5 -- Шахта для удара 6
11111101--10101000-- Время: 6 -- Шахта для удара 5
11111010--01010000-- Время: 7 -- Шахта для удара 4
11110101--10100000-- Время: 0 -- Шахта для удара 3
11111111--11101010-- Время: 1 -- Шахта для удара 2
11111111--01010101-- Время: 2 -- Шахта для удара 1
01111111--00101010-- Время: 3 -- Шахта для удара 1
10111111--00010101-- Время: 4 -- Шахта для удара 2
01011111--00001010-- Время: 5 -- Шахта для удара 3
10101111--00000101-- Время: 6 -- Шахта для удара 4
01010111--00000010-- Время: 7 -- Шахта для удара 5
10101010--00000000-- Время: 0 -- Шахта для удара 6
11111110--01010100-- Время: 1 -- Шахта для удара 6
11111101--10101000-- Время: 2 -- Шахта для удара 5
11111010--01010000-- Время: 3 -- Шахта для удара 4
11110101--10100000-- Время: 4 -- Шахта для удара 3
11101010--01000000-- Время: 5 -- Шахта для удара 2
01010101--00000000-- Время: 6 -- Шахта для удара 1
00101010--00000000-- Время: 7 -- Шахта для удара 1
01111111--00101010-- Время: 1 -- Шахта для удара 3
00101111--00000101-- Время: 2 -- Шахта для удара 4
01010111--00000010-- Время: 3 -- Шахта для удара 5
10101010--00000000-- Время: 4 -- Шахта для удара 6
01010100--00000000-- Время: 5 -- Шахта для удара 6
10101000--00000000-- Время: 6 -- Шахта для удара 5
01010000--00000000-- Время: 7 -- Шахта для удара 4
10100000--00000000-- Время: 0 -- Шахта для удара 3
01100000--01000000-- Время: 1 -- Шахта для удара 2
01010000--00000000-- Время: 2 -- Шахта для удара 1
10100000--00000000-- Время: 3 -- Шахта для удара 3
01000000--00000000-- Время: 4 -- Шахта для удара 2
00000000--00000000-- Время: 5 -- Шахта для удара 1
Length of way: 45
Bush win!!!

Process finished with exit code 0
```

# Моя задача.

## Как это все реализовано?

Всю игру можно записать в некотором графе состояний возможных местонахождений Усамы в какой-то момент времени по модулю  $n$ , в котором ребрами будут выступать проверки Буша. То есть запустив в таком графе от стартовой вершины BFS, можно проверить достижимость стартовой вершины до состояния, где Усама нигде не может оказаться. Всего состояний порядка  $(2^n)^2 \cdot n = 4^n \cdot n$ , что достаточно много для  $n > 15$ . Почему такое число:  $n$  – время, которое осталось до обнуления возможности использования +2, а 2 раза по  $2^n$  кодируют возможность Усамы находится в той или иной шахте с использованием буста или нет. Для оптимизации алгоритма было решено подсчитывать ребра в режиме онлайн, ведь, вероятно, мы не пробежимся прям по всем ребрам в графе, а только по какой-то его части. Также было принято решение не учитывать ребра, которые не меняют состояние или ведут в вершины, где мы уже было (то есть с точно таким же временем и возможностью вершин), ведь это означает, что мы ничего нового так не откроем.

В результате таких оптимизаций мы получили достаточно неплохое ускорение по времени, что позволяет считать ответ для  $n = 30$  за 22 секунды.



# Моя задача.

## Выводы

Как видим, Буш применяет технологию смывающей волны.

Пробегаёт около половины шахт, потом возвращается назад и так несколько раз.

По всему видимому, так он заставляет Усаму использовать +2 и оказываться только в первом столбике. Как видим, через 40% игры у Буша это получается и он начинает уже гнать Усаму по столбику +2, иногда гоняя его со столбика, где Усама ещё может использовать буст.

Программно было проверено для всех  $n \leq 30$ . Везде один и тот же принцип, просто очень сильно увеличивается время загонов.

Также прикрепляю [ссылку](#) на код

# Биржа.

## Общая идея и игроки

После того, как у меня получилось численно решить задачу, то мне захотелось смоделировать что-то более серьезное, чем простую игру.

Первое, что пришло в голову, сделать упрощенную версию биржи и посмотреть на общее поведение агентов.

Сначала я реализовал почти случайное поведение агентов с 2мя параметрами, развивающимися с помощью генетического алгоритма.

После я подключил линейную регрессию к агентам, что позволило убрать случайность и получить более красивые результаты

Если захочется поиграть с моделью, то оставляю [ссылку](#) на код

# Биржа .

## Игроки

Каждый агент содержит в себе коэффициент риска и паники.

Риск отвечает за желание агента покупать акции, паника за продавать.

В первой версии игры выбиралось какое-то число от -паника до +риск и если  $> 0$ , то покупали число / риск от акций, которые мы могли позволить себе купить.

Аналогично с паникой.

Во второй версии была реализована линейная регрессия, позволяющая убрать случайность. Теперь каждый агент старается предсказывать цену акции в будущем, и в зависимости от того, насколько % она поднимется или упадет, покупает или продает акции. Потом это число умножается на коэффициент риска или паники в зависимости от вердикта и уже это число акций мы покупаем(или продаем).

В качестве параметров я брал тренд акции. Количество дней до дивидендов, количество дней до новых акций, коэффициенты тренда, который складывался исходя из max/min/mean цены за акции в течении дня.

Задумка состояла в том, чтобы найти некие типы удачных игроков и посмотреть на их стратегии

# Биржа .

## Генетический алгоритм

Что вообще такое генетический алгоритм?

Это протокол, взятый из природы, благодаря которому можно выбрать наилучшие стратегии агентов из общей массы в формате постоянной конкурентности.

Как он работает: Для начала мы берем пачку случайных агентов и запускаем игру среди них. Далее подсчитываем результат и устраиваем кроссинговер, то есть исходя из результата агента, передаем часть его характеристик какой-то части следующего поколения агентов. Далее части нового поколения агентов случайным образом меняем гены, как бы производя мутацию. Потом все крутим ещё раз и так до тех пор, пока лучшие агенты примерно не будут меняться. Также в нашей ситуации я решил добавить в каждую игру часть случайных агентов, чтобы видеть, что агенты дети в целом играют лучше, чем если бы они совсем не обучались.

Именно в нашей игре я перебрасываю от агентов его детям риск, панику и линейную регрессию. Это можно интерпретировать как гены от родителей и мудрость учителя.

# Биржа . Рынок

Теперь посмотрим на то, как работает сам биржевой рынок.

Игра длится  $m$  дней, где каждый день происходят следующие события:

С некой долей вероятности добавляется новый случайный агент (чтобы он добавлял акции и капитал), если день кратен  $p$ , то выпускаем новые акции и если он кратен  $q$ , то выплачиваем акционерам дивиденды. Затем случайным образом меняем на  $(-200; 200)$  цену акции и запускаем круг.

Круг - круговое движение всех агентов, у каждого из них есть возможность купить/продать акции (уже обговаривал). Желание купить, продать влияет на цену акции. Если он хочет продать акции, то количество акций на рынке увеличивается, а если хочет купить, то уменьшается. Также количество акций не может стать  $< 0$ , ведь в жизни так не бывает.

Также я обновляю линейную модель, отвечающую за тренд акций.

Я упрощаю модель полагая, что есть какие-то внешние регуляторы, которые всегда готовы купить акции и готовые не опустить акции ниже какой-то цены

# Биржа .

## Линейная регрессия

Для подсчета тренда и предсказания цены нужна была с одной стороны быстро обучаемая, но в то же время достаточно точная линейная регрессия.

Пользоваться методами из линейной алгебры было бы непозволительно долго, потому что мне пришлось бы каждый раз считать пусть не очень большие, но матричные операции. Поэтому я решил воспользоваться методом градиентного спуска с небольшими ухищрениями:

- 1) Хранить импульс, который позволяет нам маневрировать от временных выбросов в пару дней и смягчать от них тренд. Чтобы одна точка не позволила нам ничего порушить.
- 2) Подбирать коэффициент обучения под каждую модель отдельно и динамически. Этого можно достичь, если сделать коэфф.обучения функцией, зависящей от последних ошибок модели. То есть из 1 и 2 получился оптимизатор ADAM
- 3) Сделать обучение по случайным бачам, а не по одной точке чтобы опять же уменьшить влияние выбросов.
- 4) В бач обязательно пропихивать последние 4 результата, а остальные выбирать случайно из всей истории. Это позволяет делать больший акцент на последние дни, но все ещё не ломаясь об короткие выбросы

Благодаря таким улучшениям с одной стороны мы относительно быстро перестраиваемся, а с другой стороны не настолько сильно, чтобы зависеть от каждого выброса + мы достаточно быстро обновляемся. Быстрее, чем если бы пошли через аналитический метод с онлайн-добавлением измерений(строчек).

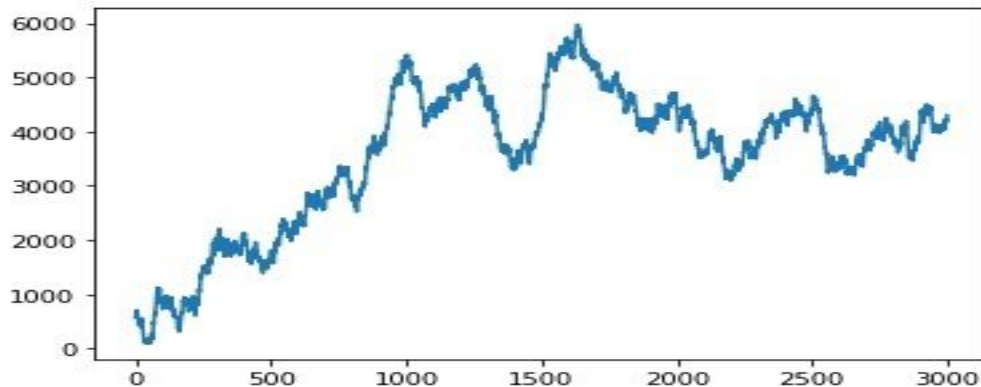
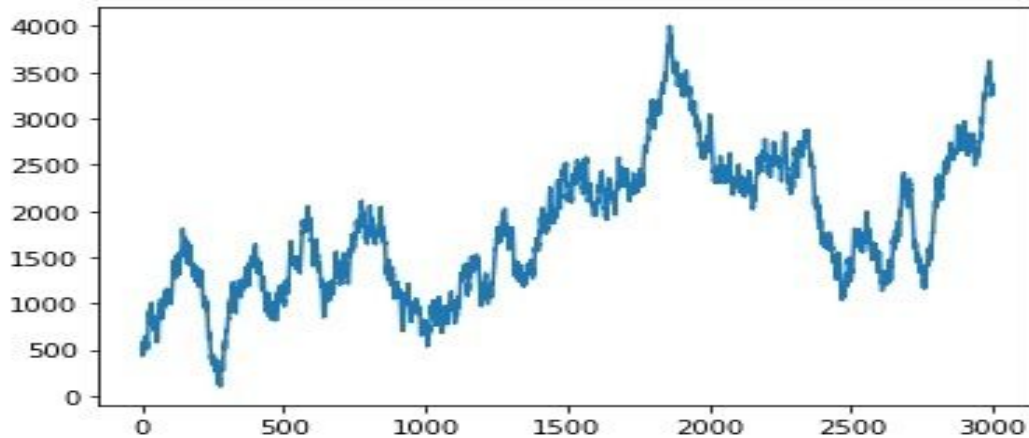
# Биржа.

## Результаты первой версии

На графике изображены цены за акцию в зависимости от дня.

Пойдя таким путем через 40 игр можно увидеть, что цена на акции изменяется случайно (данные выгружены в обработ в питоне).

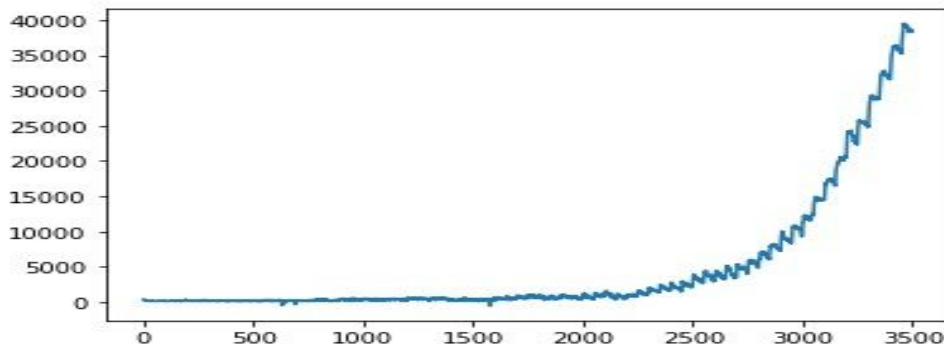
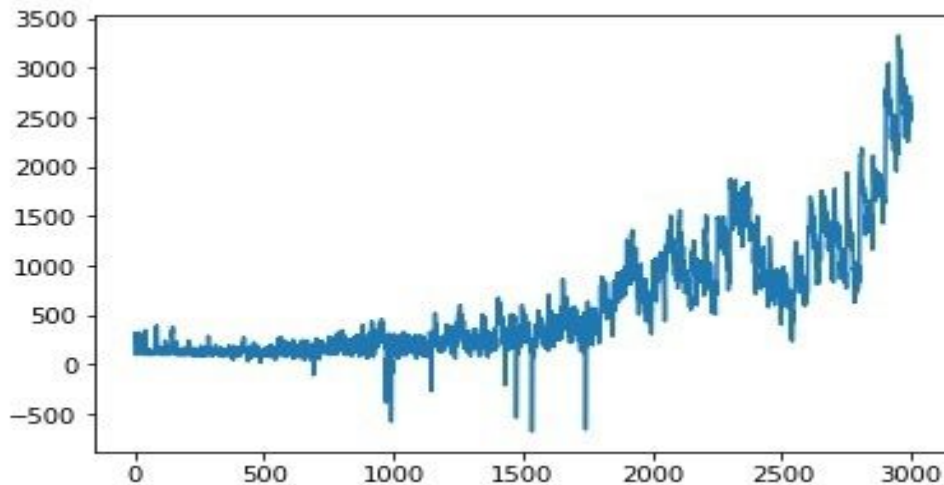
Интерес тут представляют агенты - обычно первые места занимают люди, которые ходят первыми на круге. Видимо, они сильнее всего могут влиять на рынок и покупать новые акции. А желание продавать акции у них не очень большое, в отличие от почти максимального желания покупать (что-то около 5 к 900). То есть в такой версии выгодна стратегия максимальной покупки



# Биржа.

## Результаты первой версии

Интереснее выглядит ситуация, если ограничить максимально возможное количество купли/продажи акций и вместо  $(-200; 200)$  сделать  $(-220; 180)$  на ежедневном изменении цены, к примеру, одной тысячей акций. Там не так сильно выражено преимущество позиций агента в круге, а купля/продажа различаются, но не так сильно (что-то около 800 к 500). При этом в среднем половина агентов выигрывает, а половина проигрывает. Интерес тут вызывает результат генетического алгоритма. На первом скрине цена за акции через 1 игру, на втором через 40 игр. Как видим, боты коллективно научились сначала закупаться акциями, а после накручивать на них цены. То есть они выработали некоторую стратегию, от которой тебе может стать сильно лучше, если ты войдешь в первую половину игроков (их score намного больше, чем на первой игре)





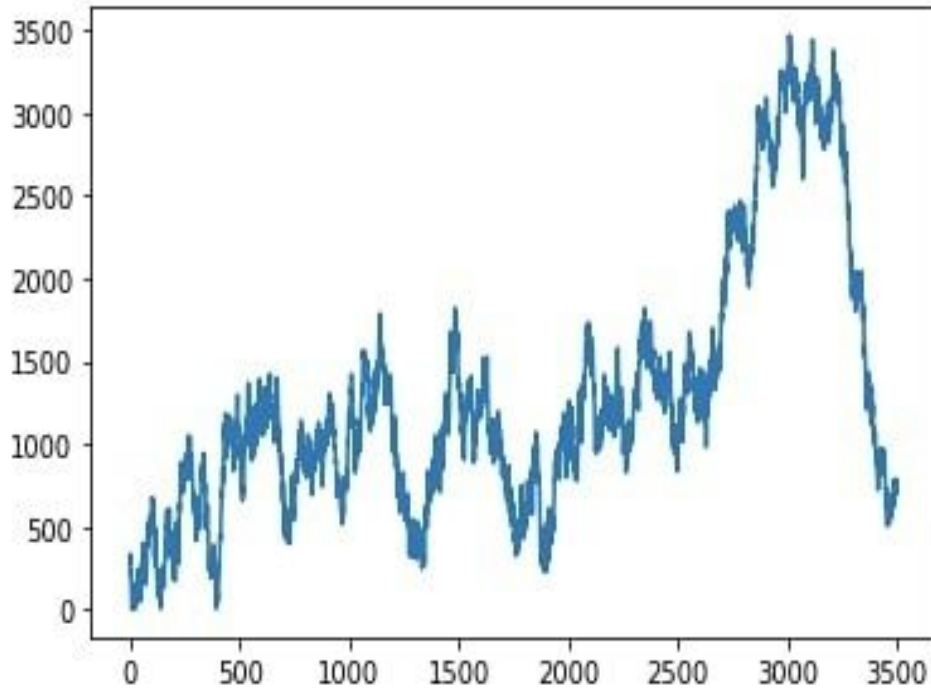
# Биржа .

## Результаты, вторая версия

После добавления линейной регрессии ситуация становится уже более реалистичной.

Потратив пару дней под обучение модели, через 40 раундов получил такой график цены акции.

За первые 2.5к дней мы наблюдаем баланс акций между 200 и 1700, игроки спокойно балансировали балансировали от дня ко дню, бегая от стенки к стенке. По к 2.7к дням появились олигархи, которые решили массово скупать акции, вследствие чего цена на акции резко взлетела, а затем пошли их продавать, из-за чего цена начала обваливаться и наступил своего рода кризис на бирже. А дальше история закончилась, потому как я считал для 3.5 дней



# Биржа .

## Анализ поведения игроков .

Итак, посмотрев на результаты игроков, можно узнать, что чем больше человек паникует и продает акции, тем меньше у него шансов попасть в топ рейтинга (корреляция позиции с паникой  $0.91$ ). С риском ситуация немного другая, там корреляция составляет  $-0.45$ , а значит чем больше риска, тем выше шанс подняться, но в то же время логично возрастает шанс все проиграть.

Какой из этого мы можем сделать вывод. В бирже произошло то, что часто случается и в настоящем мире – паникеры проигрывают, а зарабатывают те, кто трезво оценивает шансы, покупая акции и редко их продавая, играя в стратегии – купи и держи. В целом, это частично отражает одну из почти случайных стратегий первой версии, но только теперь тут уже более обосновано.

# Итог

- За прошедшую практику я прочитал несколько статей прорешал задачи по теории игр, узнав при этом некоторые новые методы для решения задач и, главное, получил удовольствие от решения задач и познания чего-то нового.
- Затем смоделировал решение одной из игр с усложненными правилами и получил точное оптимальное решение игры. Эта относительно нетрудная модель натолкнула на мысль, что я могу моделировать и более сложные процессы, чтобы смотреть на оптимальное поведение агентов в каких-то средах.
- Смоделировал биржу, получив результаты, схожие с реальными. В ходе моделирования я познакомился с реализацией генетического алгоритма и применением методов из мат.анализа для оптимизации линейных моделей.