

Название проекта: Оптимизация параметров торговых стратегий

Тип проекта: Практический(командный)

Выполнили: Ивановский Дмитрий, Мусин Камиль 295 группа

Была поставлена задача разработки и оптимизации алгоритма принятия решений о покупке/продаже ценных бумаг с использованием генетического алгоритма.

Задача разбивается на несколько независимых частей:

1. Сбор исторических данных. Дата, цена, объём
2. Формализация правил принятия решений, в виде алгоритма, зависящего от числовых параметров, алгоритм состоит из следующих частей:
 1. Правила входа в сделку
 2. Правила выхода из сделки с прибылью
 3. Правила выхода из сделки с убытком
3. Алгоритмизация целевой функции, равной итоговой прибыли, полученной после запуска алгоритма на исторических данных
4. Оптимизация параметров в целях максимизации функции прибыли



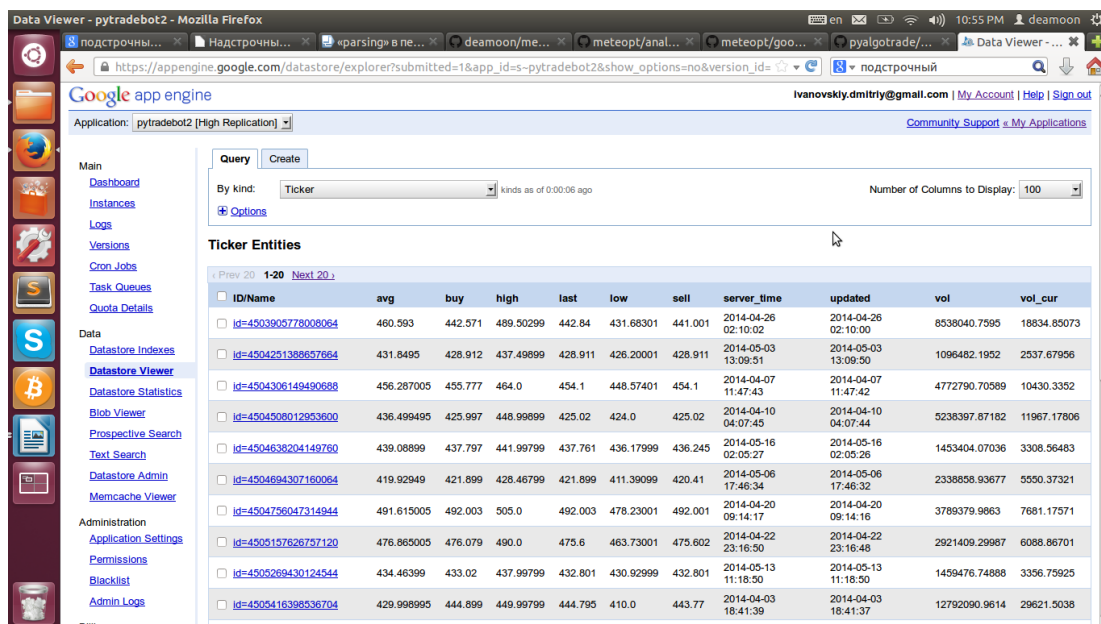
рис 1. Данные с биржи в формате “японских свечек”

Данные:

В качестве источника данных была выбрана электронная биржа btc-e [2], торгующая криптовалютами (Bitcoin, Litecoin и т.д.). Выбор был сделан в силу ряда причин:

- Актуальность торгов криптовалютами
- Торги проводятся круглосуточно и без перерывов
- Низкие требования для доступа к торгам, достаточно регистрации на сайте
- Небольшой размер минимальной сделки
- Простой интерфейс взаимодействия с сервером биржи (HTTP-запросы)
- Большие колебания цены
- Кол-во активных участников(>3000) и более 1000 роботов

Данные о торгах предоставляются в виде JSON и могут быть получены обращением на страницу [3]. Для их сбора был написан скрипт на Python, исходный код доступен по адресу: https://github.com/deamoon/meteopt/blob/master/google_app/page.py. Он запущен на сервере Google App Engine и собирает каждые 5-минут статистику торгов, сохраняя данные в базе. Доступ к базе осуществляется из своего аккаунта.



The screenshot shows the Google App Engine Datastore Viewer interface. The browser address bar displays the URL: `https://appengine.google.com/datastore/explorer?submitted=1&app_id=s-pytradebot2&show_options=no&version_id=`. The page title is "Data Viewer - pytradebot2 - Mozilla Firefox". The interface includes a sidebar with navigation links such as "Main", "Data", "Administration", and "Billing". The main content area shows a "Query" section with a "By kind:" dropdown set to "Ticker" and a "Number of Columns to Display:" set to "100". Below this, a table titled "Ticker Entities" displays data for various tickers. The table has columns for "ID/Name", "avg", "buy", "high", "last", "low", "sell", "server_time", "updated", "vol", and "vol_cur". The data is sorted by "server_time" in descending order.

ID/Name	avg	buy	high	last	low	sell	server_time	updated	vol	vol_cur
id=4503905778008064	460.593	442.571	489.50299	442.84	431.68301	441.001	2014-04-26 02:10:02	2014-04-26 02:10:00	8538040.7595	18834.85073
id=4504251386657664	431.8495	428.912	437.49899	428.911	426.20001	428.911	2014-05-03 13:09:51	2014-05-03 13:09:50	1096482.1932	2537.67956
id=4504306149490688	456.287005	455.777	464.0	454.1	448.57401	454.1	2014-04-07 11:47:43	2014-04-07 11:47:42	4772790.70589	10430.3352
id=4504508012953600	436.499495	425.997	448.99899	425.02	424.0	425.02	2014-04-10 04:07:45	2014-04-10 04:07:44	5238397.87182	11967.17806
id=4504638204149760	439.08899	437.797	441.99799	437.761	436.17999	436.245	2014-05-16 02:05:27	2014-05-16 02:05:26	1453404.07036	3308.56483
id=4504694307160064	419.92949	421.899	428.46799	421.899	411.39099	420.41	2014-05-06 17:46:34	2014-05-06 17:46:32	2338858.93677	5550.37321
id=4504756047314944	491.615005	492.003	505.0	492.003	478.23001	492.001	2014-04-20 09:14:17	2014-04-20 09:14:16	3789379.9863	7681.17571
id=4505157626757120	476.865005	476.079	490.0	475.6	463.73001	475.602	2014-04-22 23:16:50	2014-04-22 23:16:49	2921409.29987	6088.86701
id=4505269430124544	434.46399	433.02	437.99799	432.801	430.92999	432.801	2014-05-13 11:18:50	2014-05-13 11:18:50	1459476.74888	3356.75925
id=4505416398536704	429.998995	444.899	449.99799	444.795	410.0	443.77	2014-04-03 18:41:37	2014-04-03 18:41:37	12792090.9614	29621.5038

Рис 2. Аккаунт Google App Engine, раздел Datastore Viewer, данные по торгам

Правила принятия решений:

В общем задача отыскания правил выгодной торговли очень сложна. В целом способы решения можно разделить на фундаментальный анализ и технический. В фундаментальном анализе объектом исследования становится стоимость компании и производственных показатели её деятельности. Технический анализ рассматривает изменение цены в прошлом и на основе этого пытается предсказать изменения будущие. В техническом анализе применяются такие продвинутые технологии как байесовские сети, машинное обучение, нейронные сети. Мы же в своем проекте ограничились техническими индикаторами, которые предоставляют правила торговли.

Технический индикатор – это инструмент анализа биржевых данных, который пытается воспользоваться моментом, когда текущая цена не соответствует цене справедливой, а значит в скором времени, цена вернется к справедливой. Это очень простой способ предсказания, но как показала практика он вполне рабочий, использование технических индикаторов подробно описано в книге[1], пример законченной стратегии описан в статье Original Turtles[2].

Заметим, что технические индикаторы по своей сути опираются на гипотезу о рациональном или логичном рынке, а значит в моменты “кризисные”, когда преобладают панические настроения, будут показывать плохие результаты. Многие разработчики профессиональных стратегий, опасаясь убытков, отключают автоматическую торговлю в период нестабильности рынка.

С точки зрения математики, технические индикаторы, которые используются в экспериментах – это свёртка. *Свёртка функций — операция в функциональном анализе, показывающая «схожесть» одной функции с отражённой и сдвинутой копией другой.* Физики называют это фильтром низких частот, который используется в обработке сигналов. В работе рассмотрены 3 вида свёрток, которые еще называют взвешенное скользящее среднее.

Общий случай

В общем случае, взвешенные скользящие средние вычисляются по формуле:

$$WWMA_t = \sum_{i=0}^{n-1} w_{t-i} \cdot p_{t-i},$$

(WWMA 1)

где $WWMA_t$ — значение взвешенного скользящего среднего в точке t ; n — количество значений исходной функции для расчёта скользящего среднего; w_{t-i} — нормированный вес (весовой коэффициент) $t - i$ -го значения исходной функции;

p_{t-i} — значение исходной функции в момент времени, отдалённый от текущего на i интервалов.

Нормирование весовых коэффициентов означает, что:

$$\sum_{i=0}^{n-1} w_{t-i} = 1.$$

Указанную выше формулу с произвольными значениями весовых коэффициентов можно переписать в виде:

$$WWMA_t = \frac{\sum_{i=0}^{n-1} W_{t-i} \cdot p_{t-i}}{\sum_{i=0}^{n-1} W_{t-i}},$$

(WWMA 2)

где $WWMA_t$ — значение взвешенного скользящего среднего в точке t , n — количество значений исходной функции для расчёта скользящего среднего, W_{t-i} — вес (весовой коэффициент) $t-i$ -го значения исходной функции, p_{t-i} — значение исходной функции в момент времени, отдалённый от текущего на i интервалов.

Весовые коэффициенты в формулах (WWMA 1) и (WWMA 2) соотносятся как:

$$w_{t-i} = \frac{W_{t-i}}{\sum_{i=0}^{n-1} W_{t-i}}.$$

Зачастую, в качестве веса используют либо 1 (для простого скользящего среднего — SMA), либо формальные ряды, например, арифметическая прогрессия (WMA) или экспоненциальная функция (EMA). Но в качестве весового коэффициента могут выступать и значения связанного временного ряда. Например, для взвешивания биржевых цен по объёмам сделки (VMA) в качестве p_{t-i} следует рассматривать цену сделки по инструменту, а в качестве $W_{t-i} = V_{t-i}$ — объём в момент времени $t-i$:

$$VMA_t = \frac{\sum_{i=0}^{n-1} V_{t-i} \cdot p_{t-i}}{\sum_{i=0}^{n-1} V_{t-i}}.$$

Генерация сигналов от технических индикаторов

Скользящее среднее является удобным генератором сигнала на вход в сделку и на выход из сделки. После расчета значений скользящего среднего, необходимо сравнить как это значение соотносится со значением цены. Моменты пересечения графика цены и графика скользящего среднего являются потенциальными моментами совершения сделок.

Однако все средние зависят от некоторых параметров, например длина окна усреднения. Заранее сказать при каких параметрах алгоритм будет генерировать самые выгодные сигналы нельзя, зато их можно найти методами оптимизации.

Далее будут рассмотрены примеры генерации сигналов на основе пересечения.

Технический индикатор SMA (простое скользящее среднее)

Численно равно среднему арифметическому значений исходной функции за установленный период, этот период является параметром алгоритма. Если текущая цена превышает значение скользящего среднего за прошедший период, то бумага недооценена и велика вероятность, что в скором времени её цена увеличится, значит нужно покупать. Если же текущая цена ниже среднего, то пора продавать, пока цена не упала. Помимо обычного скользящего среднего часто применяется экспоненциальное или линейное.

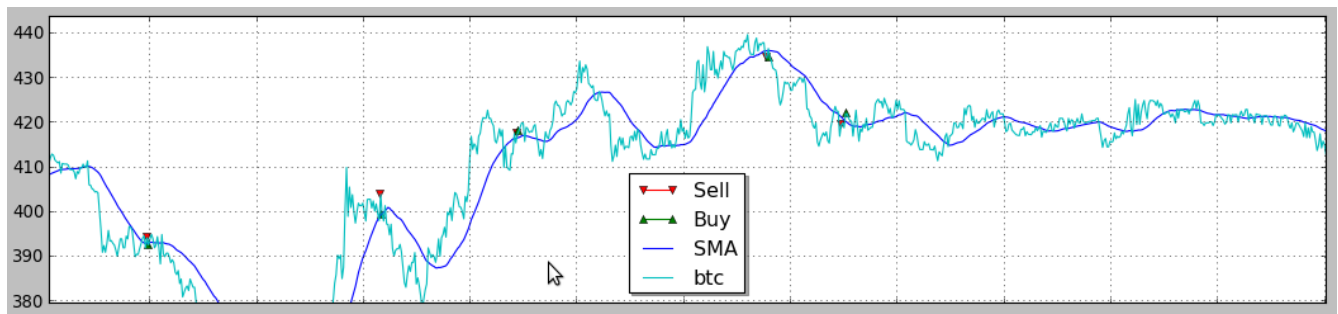


Рис 3. График SMA(синий) поверх графика цены(голубой)

Технический индикатор Bollinger Bands, Полосы Боллинджера

Состоит из трех кривых, две из которых образуют канал, ограничивающий движение цены валютной пары сверху и снизу, третья же, расположена между первыми двумя и находится на равном удалении от последних. Средняя полоса – SMA с периодом, который является параметром, нижняя полоса – SMA минус 2 стандартных отклонения, верхняя полоса – SMA плюс 2 стандартных отклонения.

За стандартное отклонения принят корень из выборочной дисперсии:

$$S_n^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 = \frac{1}{n} \sum_{i=1}^n X_i^2 - \left(\frac{1}{n} \sum_{i=1}^n X_i \right)^2$$

Сигналом покупки является пробой текущей цены верхней полосы снизу вверх, сигналом продажи является пробой нижней полосы сверху вниз.

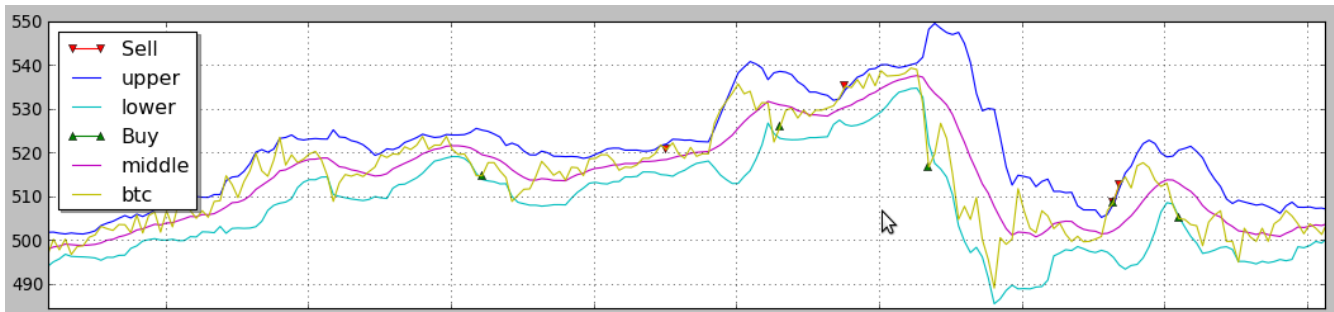


Рис 4. Полосы Болинджера, нижняя(зеленый), средняя(фиолетовый), верхняя(синий), график цены(желтый)

Технический индикатор Volume-weighted average price (VWAP)

Это взвешенное среднее за определенный период, являющееся параметром. С весами равными объему торгов в течение 5-минутного интервала. Правила генерации сигналов аналогичны индикатору SMA.

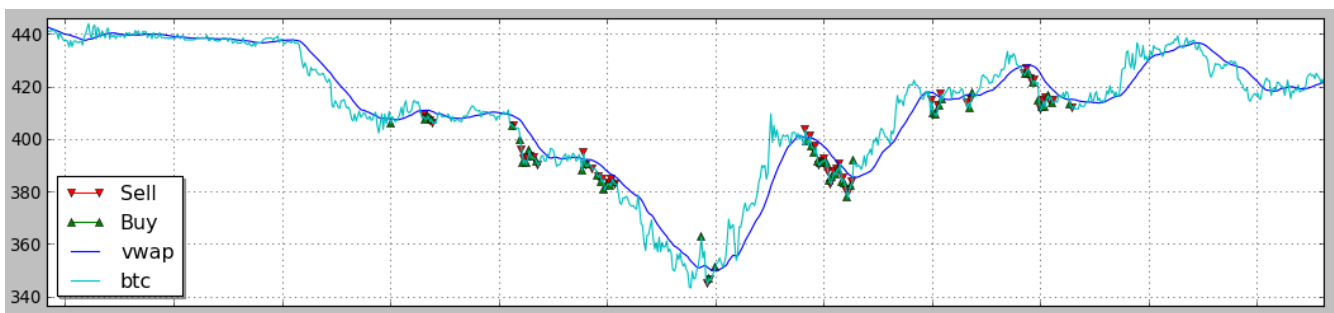


Рис 5. VWAP(синий), график цены(голубой)

Итоговая стратегия, или функция прибыли:

Итоговая стратегия является объединением 3-х технических индикаторов. В цикле по историческим данным на основе цены вычисляются новые значения средних. Проверяется их пересечения с графиком цены, на основе этого генерируются сигналы о покупке и продаже. Если в течение определенного кол-ва 5-минутных интервалов все индикаторы прислали сигнал на покупку, покупка совершается, аналогично с продажей. Это кол-во 5-минутных интервалов является параметром алгоритма.

Так же были проведены эксперименты по введению параметра stop-loss,

ограничивающего выход из сделки с убытком, если убыток от сделки превышает данный параметр, нужно выходить из сделки с убытком. И с параметром, ограничивающим частотность сделок, совершать новую сделку, только если абсолютное изменение цены с предыдущей сделки превысило этот параметр.

Итого 6 параметров:

1. Параметр окна SMA, целое число от 1 до 100
2. Параметр окна Боллинджера, целое число от 1 до 100
3. Параметр окна VWAP, целое число от 1 до 100
4. Кол-во 5-минуток, целое число от 1 до 10
5. Максимальный убыток сделки (stop-loss), вещественное число от 0 до 1 (не более 10% от стартового капитала)
6. Максимальное кол-во сделок, разрешенных совершить за день, целое число от 0 до 10

Напомню, что максимизируемая функция — это прибыль от работы алгоритма на определенном историческом промежутке. Вышеописанный алгоритм является реализацией функции прибыли, которую в свою очередь использует генетический оптимизатор.

Конечная реализация функции прибыли на языке Python доступна по адресу:

https://github.com/deamoon/meteopt/blob/master/analyze_data/optimizer_strategy.py

Оптимизация параметров:

Кол-во параметров для оптимизации в разных тестах было от 4 до 6, значения которых принадлежат целым числам на интервале от 5 до 100. То есть для полного перебора потребовалось бы перебрать 10^{12} значений, при том что вычисление функции прибыли обладает линейной сложностью от кол-ва данных и в среднем работает 4 секунды на месячных данных.

Основным препятствием для применения большинства методов оптимизации является по сути произвольный характер функции цены от времени. Цена зависит от многих параметров, как эндогенных, таких как спрос, и предложении на бирже в текущий момент, так и от экзогенных, к примеру политические заявления, или

природные катаклизмы.

Поэтому приходится прибегнуть к методам, основанным на статистических данных, и дающий лишь вероятностную сходимость к оптимальному ответу. Одним из таких методов является генетический алгоритм, который мы для данной задачи.

Генетический алгоритм — это принцип выбора аргументов, построенные по аналогии с биологическим процессом эволюции. Роль генов играют параметры, а роль организмов наборы параметров. Для получения новых наборов данных используются два основных принципа: скрещивания и мутации. Где скрещивание — это объединение параметров двух «родителей», а мутации — это некоторое случайная функция над параметрами. Принцип скрещивания устроен таким образом, что более успешные особи (наборы параметров) имеет больше шансов на «репродукцию»

Стандартный генетический алгоритм состоит из следующих стадий:

1. Генерация начальной популяции
2. Вычисление целевой функции
3. Селекция
4. Генерация нового поколения
5. Мутация

Выбор этого метода был обусловлен следующими причинами:

- Простой принцип работы
- Возможность легко модифицировать ключевые элементы алгоритма, для улучшения сходимости в конкретной задаче
- Возможность распараллелить вычисления

В результате получилось достичь очень хороших результатов: быстрая сходимость (10 поколений) к результатам близким к оптимальным

Детали реализации генетического алгоритма:

1. *Создание начальной популяции:* для создания начальной популяции был выбрано генерирование наборов случайных параметров для оптимизированной функции. Так как данный алгоритм оптимизации предполагается использовать оптимизации широкого спектра различных торговых алгоритмов, было принято решение параметры выбирать с равномерным распределением из допустимого диапазона. В связи с этим для любой оптимизируемой стратегии требуется указывать допустимые интервалы для параметров. Также возможно задавать начальную популяцию непосредственно в виде параметров оптимизатора.
2. *Отбор:* Отбор осуществляется за счет введения вероятности участия, в репродукции. Для этого вводится дискретная случайная величина принимающая значения от 1 до N (где N — размер популяции), при этом значение i принимается с вероятностью p_i , где p_i — вероятность i -ой особи участвовать в репродукции. Данная случайная величина используется для выбора N пар особей. Каждая пара создает одну особь для следующего поколения.
3. *Вычисление целевой функции:* данная стадия является самой время затратной, поэтому было принято решение, распараллелить вычисления на данной стадии. Распараллеливание возможно, так как вычисления функции прибыли от различных наборов параметров не зависят друг от друга.
4. *Принцип присвоения вероятностей:* Вероятности присваиваются пропорционально значению целевой функции (полученной прибыли на конец рассматриваемого периода). При проведении опытов было замечено, что лучшие результаты по сходимости достигается, когда вероятности задаются пропорционально квадрату целевой функции, вместо первой степени. Это обусловлено тем, что при этом условии, особи с более высоким значением целевой функции поощряются больше, и соответственно, в следующем поколении оказывается, больше особей являющихся их потомками.
5. *Скрещивание:* при создании особи нового поколения, применяется скрещивание параметров от родителей. В данной работе потомок для каждого параметра имеет одинаковый шанс иметь параметр от любого из потомков. Другими словами с вероятностью 50% выбирается параметр от первого потомка, и с 50% от второго. Возможны также варианты реализации, когда более успешные (в смысле значения целевой функции) особи из пары имеют больший шанс передать свой параметр потомству. Но данные методы в совокупности с уже примененной приоритизацией успешных особей на уровне отбора, не дают хороших результатов, так проявляется явное доминирование конкретных «генотипов» из начальной популяции над

другими, несмотря на то что они являются оптимальными (порой даже далеки от оптимальных).

6. *Мутации*: в качестве мутации использовались следующие два принципа:

- принцип случайного изменения генов: любой параметр новой особи при создании имеет небольшой шанс измениться на случайное значение из диапазона (оптимальные значения зависят от размера популяции, для популяции размера 100 оптимальный шанс около 1%)
- принцип добавочной популяции: в каждую новую популяцию добавляется некоторая кол-во особей с произвольными значениями параметров. В данном проекте размер популяции всегда остается постоянным, и поэтому в каждой популяции 15% наиболее неуспешных особей заменяется на новые сгенерированные случайным образом.

Сам алгоритм был реализован на языке Python (код можно найти в репозитории по ссылке:

https://github.com/deamoon/meteopt/blob/master/analyze_data/genetic_optimizer/genetic.py)

Данный язык программирования был выбран в виду существования мощных инструментов для работы с http-серверами, для сбора и обработки информации, а также по причине простоты и наглядности получаемого исходного кода.

Итоговая торговая стратегия:

Результатом оптимизации является набор параметров, который при подстановки в функцию прибыли образует торговую стратегию. Подобная стратегия может быть запущена в режиме реальных торгов или опробована на других исторических данных.

Подобно ходу проекта, разработка настоящей стратегии начинается с реализации функции прибыли, которая зависит от параметров, эти параметры оптимизируются на последних исторических данных, затем пригодность стратегии проверяется в тестовом режиме, когда алгоритм совершает сделки с виртуальным балансом. И только когда во всех этих этапах были показаны хорошие результаты и пригодная стабильность, алгоритм запускают на биржу.

Заметим, что в наиболее продвинутых стратегиях параметры не являются константами и обновляются тоже по вполне формализованным правилам, по мере изменения ситуации на рынке.

Пример исторического тестирования:

Пример исторического тестирования изображен на рис. 6. Был выбран исторический период 31 марта — 12 мая, за который были накоплены данные цены от времени. Этот период отличается достаточная стабильность на концах и сильные колебания посередине

Алгоритм как раз смог зафиксировать прибыль 20 апреля, когда курс стабилизировался. Также впоследствии вошел ещё в несколько сделок в начале мая, уже не столь успешных. Во всех экспериментах стартовый баланс был равен 10 долларам. На этих данных алгоритм закончил с балансом 10.8

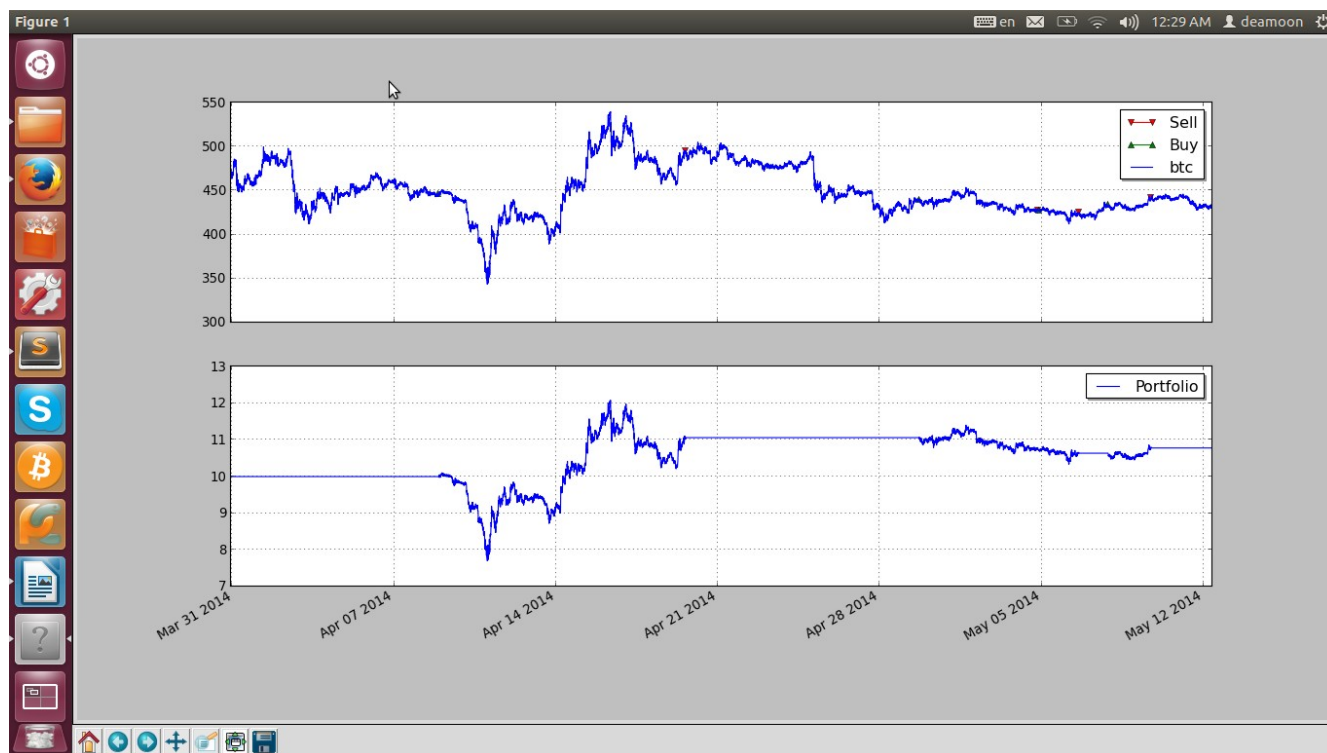


Рис 6. График цены(сверху) и баланс аккаунта в долларах(снизу)

Результаты:

- Были получены параметры, при которых алгоритм приносит прибыль, совершая в среднем 4-5 сделок в месяц
- Было доказана возможность использования генетических алгоритмов для оптимизации торговых стратегий
- Проверена применимость использования технических индикаторов для анализа ситуации рынка

Интересные факты:

1. Наиболее успешным размером окна для индикатора SMA оказался промежуток, в 2,5 часа, это означает, что именно 150 минут лучше всего характеризуют поведение цены в краткосрочной перспективе.
2. Использование индикатора VWAP приводило к негативным результатам, и в процессе оптимизации алгоритмом его сигналы перестали влиять на решение о купле продаже.
3. Размер комиссии очень сильно влияет на прибыльность конкретного набора параметров, т.е. не существует оптимальных параметров, которые зависят только от данных, и при этом не зависят от условий торговой площадки.
4. Не существует также, и алгоритма оптимального на всем промежутке времени, т. к. характер рынка может сильно меняться. К примеры на рассматриваемой нами валютной паре можно выделить два периода: высокие колебания (февраль-март), и стабилизация цены, сопровождающаяся отсутствием больших колебаний (апрель-май).
5. В процессе исследования, стало заметно, что участники торгов используют похожие алгоритмы в своей работе. Это косвенно подтверждается возвращением цены к среднему значению и информацией с сайта [4]

Планы на будущее:

- Добавить правила вычисления объема совершаемой сделки
- Рассмотреть другие целевые функции к примеру: отношения прибыли к риску или минимизация риска
- Проверить полученную стратегию в режиме реальных торгов

Ссылки:

1. <https://github.com/deamoon/meteopt> — репозиторий проекта
2. <https://btc-e.com> — биржа криптовалют
3. https://btc-e.com/api/2/btc_usd/trades — данные по сделкам
4. <https://cryptotrader.org> — облачная платформа для разработки торговых роботов

Литература:

1. “Энциклопедия торговых стратегий”, Джеффри Оуэн Кац, Донна Л. МакКормик
2. “The Original Turtle Trading Rules“, <http://www.tradingblox.com/originalturtles/>
3. Генетический алгоритм, просто о сложном, <http://habrahabr.ru/post/128704/>