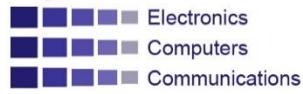


Department of Electrical Engineering



Arbitrage system for cryptocurrency trading

Performed by:

Ameer Saleh & Muhammad Abu Yonis

Supervised by:

Mr. Mark Levin

Winter 2018/2019

Table of Contents

Introduction	3
Related work	4
Why arbitrage	6
Triangular Arbitrage	9
Cross exchange Arbitrage	14
Simulation results	21
Cross exchange simulation data analysis	24
Project challenges	27
Arbitrage market challenges and risks	28
Fees	29
Some math	30
Future development	32
CCXT	33
References	34

Introduction

"ARBITRAGE IS BUYING A SECURITY IN ONE MARKET AND SELLING IT IN ANOTHER MARKET AT A HIGHER PRICE, PROFITING FROM THE TEMPORARY DIFFERENCE IN PRICES. THIS IS CONSIDERED RISK-FREE PROFIT FOR THE INVESTOR/TRADER." Investopedia

Cryptocurrency is still a new and inefficient market. Several cryptocurrency exchanges exist around the world and the bid/ask prices they propose can be briefly different from an exchange to another. The purpose of Arbitrage is to automatically profit from these temporary price differences while being market-neutral.

Cryptocurrency should not cost varying amounts across each exchange, especially consider that all the exchanges can be accessed from one's computer. Arbitrage simply brings the exchanges together to an average price. As market grows, the gap between exchanges will narrow, as the rate at which people arbitrage increases. The current volume can certainly help an individual make a significant amount of profit, but it is not yet worthwhile for large financial firms to engage in cryptocurrency arbitrage directly.

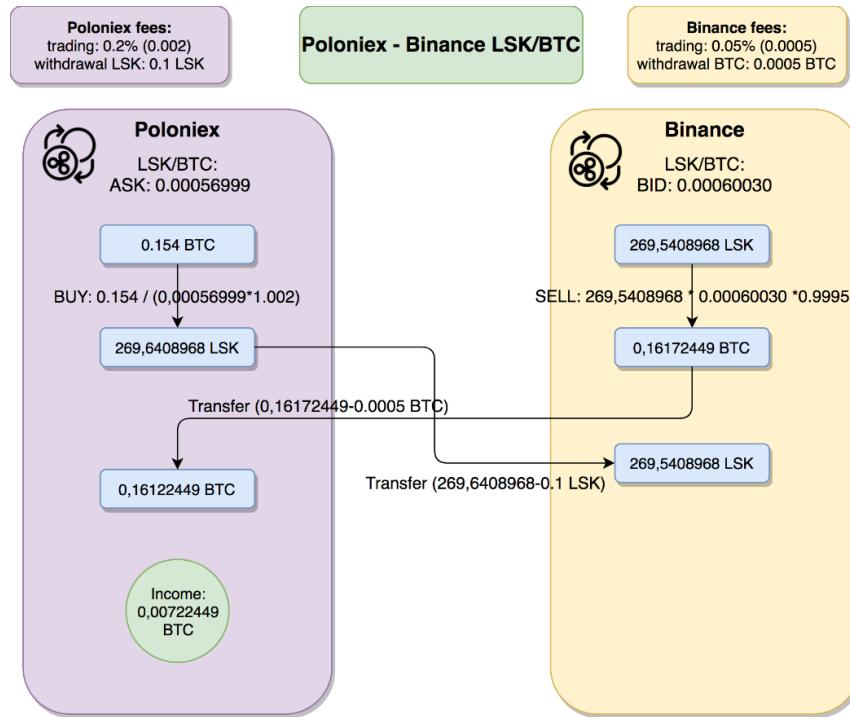
In this project we build a program that finds trading opportunities between two different cryptocurrency exchanges and can perform automatic low-risk trades.



Related work

Parallel “Loop” arbitrage

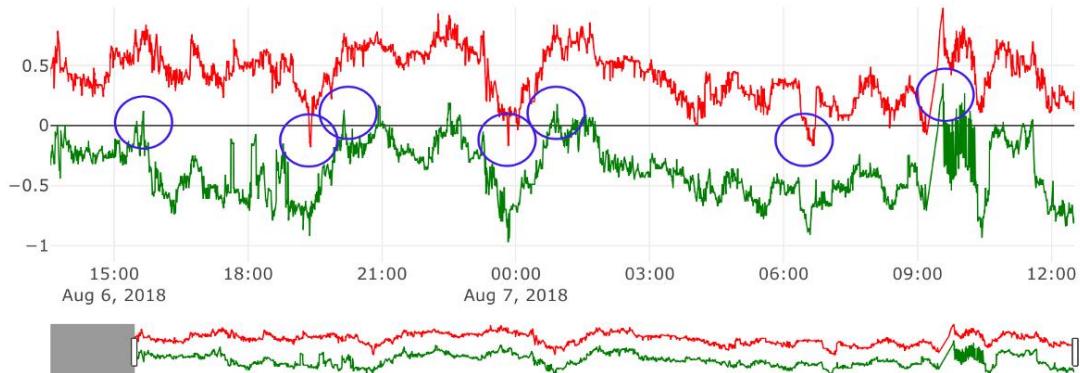
In parallel arbitrage, we hold our assets at different exchanges, and aim to arbitrage to increase our total balance of each currency. In this method we execute two transactions simultaneously, as described in the following graph:



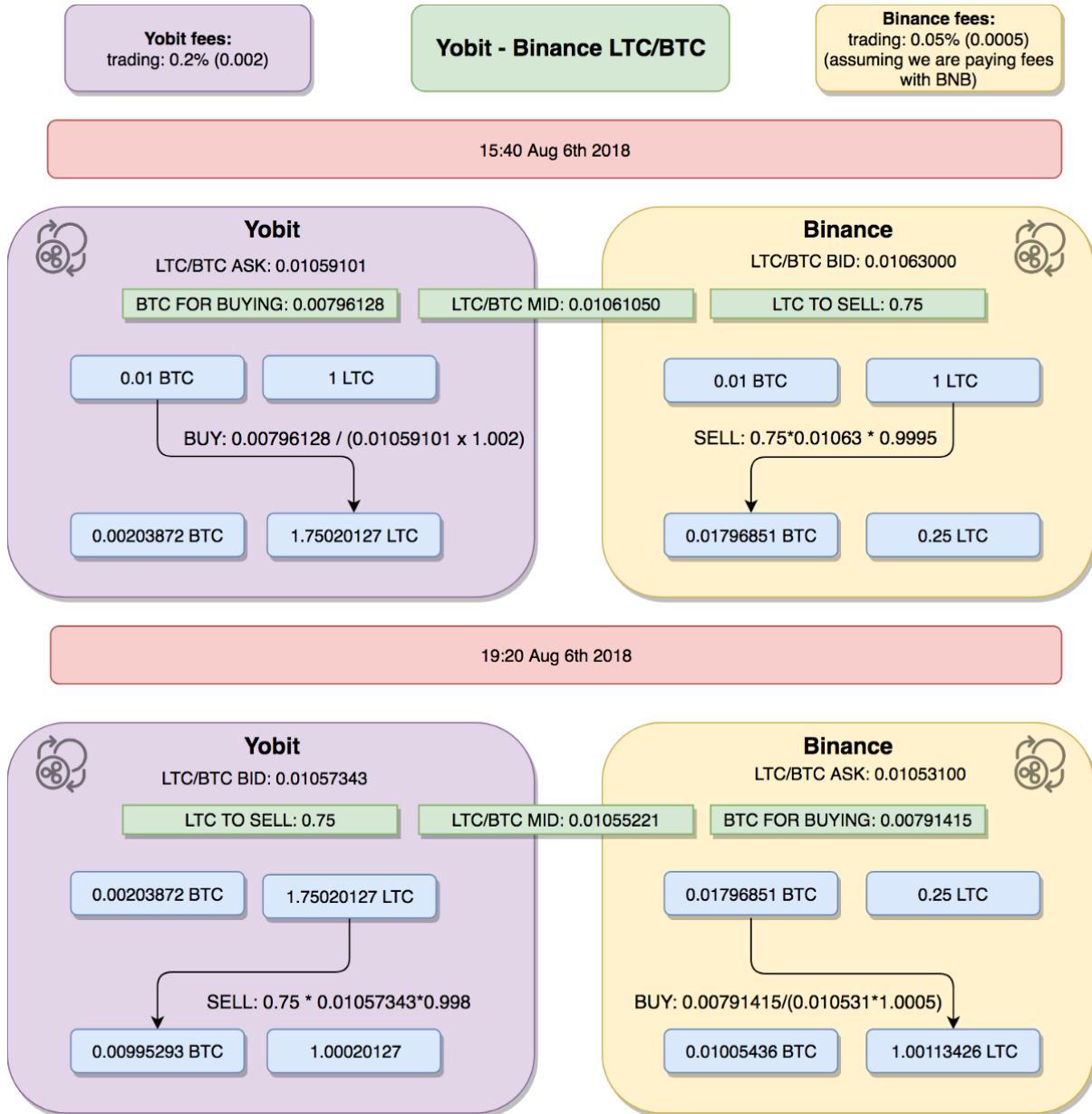
This method accomplishes our mission with low risk since the arbitrage opportunity is caught immediately, the downside is that we have to have the right coins in the right exchanges and our assets will be split among many accounts in different exchanges.

Oscillator

This method is like the Parallel arbitrage but doesn't transfer assets to rebalance, instead it tries to do counter trades to rebalance without transfer.



green line represents the **gain %** in the direction from YoBit to Binance (buy on YoBit, sell on Binance). **Red** line represents the **loss** value for the trades that would be made in the opposite direction (buy on Binance, sell on YoBit).



We can see that while holding BTC and LTC at both YoBit and Binance, we performed two profitable arbitrages, without transferring assets and succeeded in acquiring a profit, while this method saves the transfer fees of the parallel method, it still requires the assets being spread among different exchanges.

Why arbitrage

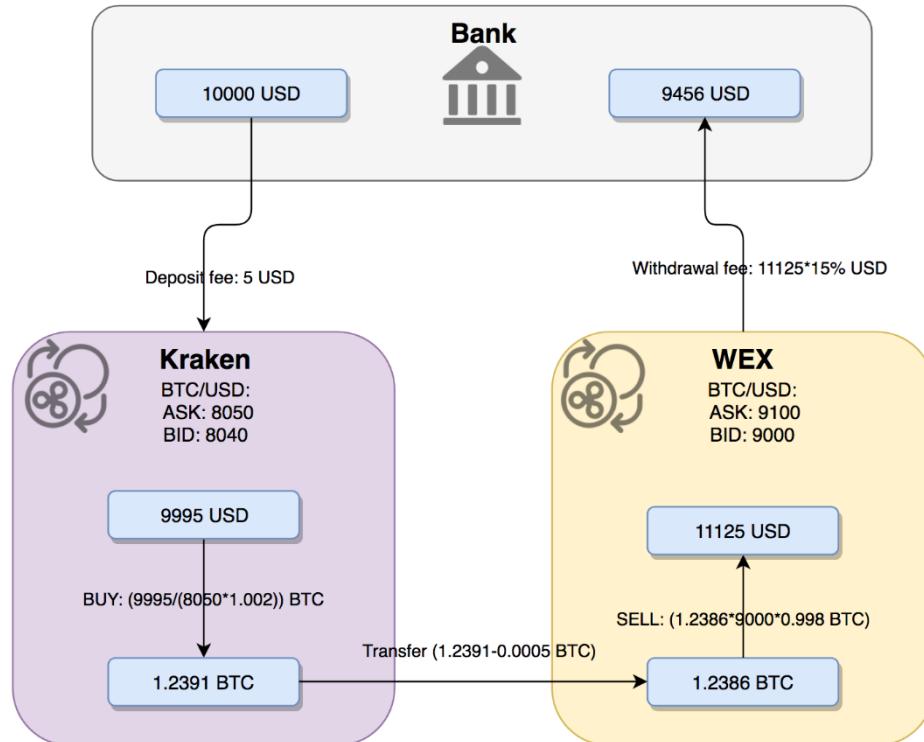
Looking at trade rates of BTC on different exchanges reveal a gold mine:

Bitcoin Markets

#	Source	Pair	Volume (24h)	Price
1	FCoin	BTC/USDT	** \$980,641,917	\$5,094.72
2	BitMEX	XBT/USD	** \$826,883,636	* \$5,086.50
3	CoinBene	BTC/USDT	\$407,185,001	\$5,097.64
4	OEX	BTC/USDT	\$402,974,326	\$5,097.55
5	BW.com	BTC/USDT	\$370,071,105	\$5,100.34
6	Negocie Coins	BTC/BRL	\$327,618,494	\$5,331.33
7	BitForex	BTC/USDT	\$297,227,147	\$5,097.94
8	Coinall	BTC/USDT	** \$287,099,561	\$5,097.12
9	FCoin	ETH/BTC	** \$270,222,272	\$5,099.77
10	OOOBTC	BTC/USDT	\$264,475,200	\$5,097.99
11	DigiFinex	BTC/USDT	\$228,289,187	\$5,094.16
12	Coinall	LTC/BTC	** \$227,202,352	\$5,094.46
13	OOOBTC	ETH/BTC	\$225,306,179	\$5,106.07
14	ZBG	BTC/USDT	** \$219,552,630	\$5,099.23

The prices are different, we could potentially buy BTC cheap and sell it for more!

Turns out the story is more complicated, let's look at an example:



- We've just spotted wonderful arbitrage trade opportunity — we can buy BTC for 8050 USD on Kraken and sell it for 9000 on Wex.
- We take our 10000 USD and we deposit them to Kraken, after paying the deposit fee we have 9995 USD.
- Then buying BTC on Kraken for all the USD we have, gives us 1.2391 BTC.
- After withdrawing this BTC from Kraken to WEX we have 1.2386 BTC as we had to pay Kraken withdrawal fee.
- And we are selling our BTC on Wex for USD. And we are rich! We have 11125 USD. 1125 USD more than when we started.
- But they money is still on WEX. We are checking the withdrawal fees and Those guys are calling 15% for the USD withdrawal! Instead of being richer, we are poorer. After withdrawing we have 9456 USD. We have just lost 544 USD.

In a more realistic approach

- Most of the arbitrage opportunities disappear after considering exchange fees and transaction fees.
- Most of the arbitrage opportunities appear between exchanges where wallets are on maintenance, which makes deposits and withdrawals impossible.
- Transaction times between exchanges cause arbitrage to become a lottery due to market volatility.

With this being said, it can be still profitable to arbitrage, especially with the crypto market still being in its infancy, and unlike the stock market, we don't see huge corporations taking over and killing arbitrage opportunities.

And in a different note, arbitrage is always positive as it stabilizes the markets and sets a real stable value to crypto assets.

- correct profit calculations considering:
 - taker fees at the purchase exchange
 - transaction fees
 - taker fees at the sale exchange.
- Maintenance of wallets.
- Time of transactions and estimated changing value in prices during this time.

We will discuss two types of arbitrage: Triangular in-exchange arbitrage, and cross chain arbitrage.

Triangular Arbitrage:

Triangular Arbitrage is the process of trading three currencies to take advantage of a price difference among the three exchange rates in order to make a profit.

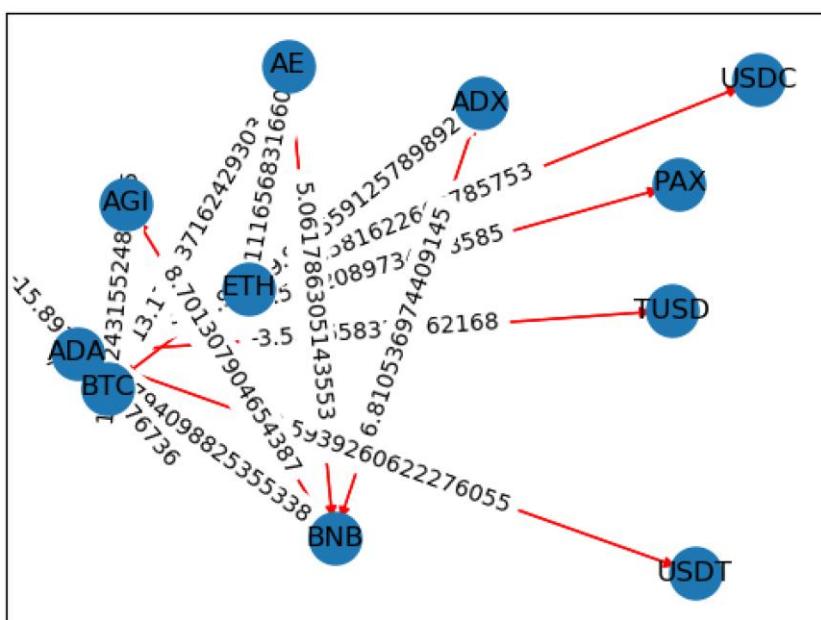
General idea:

- Start with deposit of cryptocurrency 1 on an exchange
- Sell cryptocurrency 1 and buy cryptocurrency 2
- Repeat steps 2 and 3
- Sell cryptocurrency 2 for cryptocurrency 1
- Withdraw the profit

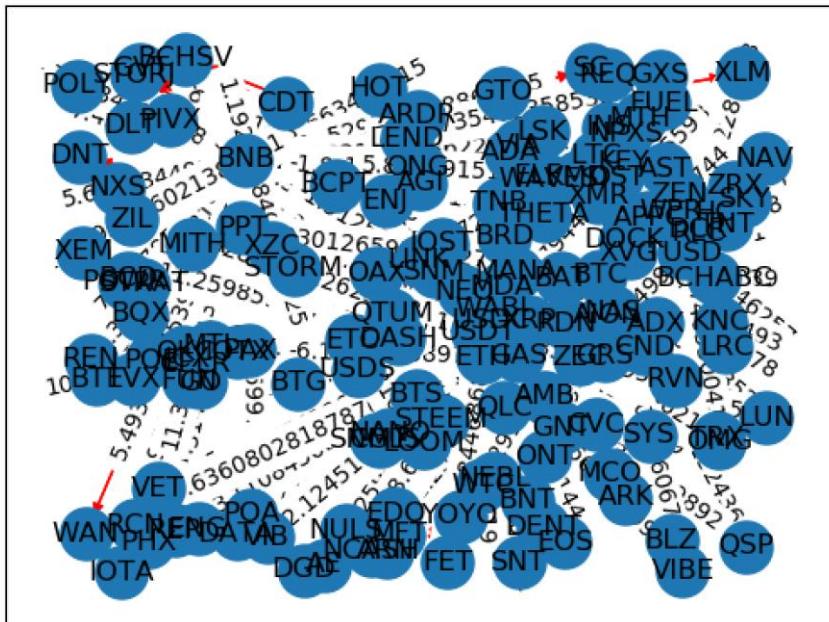
Our Implementation

Generic Bellman Ford graph: Build big graph where nodes are currencies and edges are trading rates. The idea was to run bellman ford from some vertex and look for profitable cycles, the advantage of this method is we can find profitable paths of different length, but the disadvantage is the runtime complexity that doesn't allow us to find opportunities in computable time.

With only 15 currencies:



With all currencies:



Code:

```

import time
import networkx as nx
import matplotlib.pyplot as plt
import ccxt
from ArbitrageExchange import ArbitrageExchange
import math

binance = ccxt.binance()
binanceArb = ArbitrageExchange(binance)

# build weighted graph to find triangular arbitrage opportunities
g = nx.DiGraph()
Pairs = binanceArb.symbols
tickers = binance.fetch_tickers()
for pair in Pairs:
    marketData = binance.market(pair)
    w = tickers[pair]['ask']
    if w <= 0:
        continue
    w_inv = (-1) * math.log(1.001 / w, 2.0)
    w = (-1) * math.log(w * 1.001, 2.0)

    g.add_edge(marketData['baseId'], marketData['quoteId'], weight=w)
    g.add_edge(marketData['quoteId'], marketData['baseId'], weight=w_inv)

# print marketData['baseId'] + " " +marketData['base'] + " " +
# marketData['quoteId'] + " " + marketData['quote']

tickers = binance.fetch_tickers()
for pair in Pairs:
    marketData = binance.market(pair)
    w = tickers[pair]['ask']
    if w <= 0:
        continue
    w_inv = (-1) * math.log(1.001 / w, 2.0)
    w = (-1) * math.log(w * 1.001, 2.0)

```

```

g[marketData['baseId']][marketData['quoteId']]['weight'] = w
g[marketData['quoteId']][marketData['baseId']]['weight'] = w_inv

pos = nx.random_layout(g)
nx.draw_networkx_nodes(g, pos, cmap=plt.get_cmap('jet'), node_size=500)
nx.draw_networkx_labels(g, pos)
nx.draw_networkx_edges(g, pos, edge_color='r', arrows=True)
labels = nx.get_edge_attributes(g, 'weight')
nx.draw_networkx_edge_labels(g, pos, edge_labels=labels)

plt.show()

```

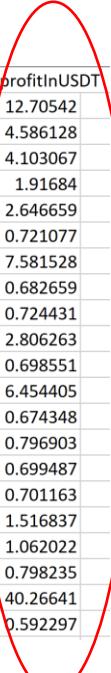
Triangular matching algorithm

The system tries to find triangular profitable trades in the same exchange, mostly to stabilize one currency in two markets making a profit.

We chose to perform a simulation on one of the most dominant cryptocurrency exchanges, Binance, which offers a wide variety of cryptocurrency trading pairs.

Runtime

First simulations were very encouraging revealing nice profits.



date	Base	pair	River	percProf%	wBase	wPair	wRiver	RiverBidQ	PairBidQty	BaseAskQt	maxTrade	profitInBT	profitInUSDT
14:35.2	NPXS/BTC	NPXS/ETH	ETH/BTC	2.78%	1.50E-07	4.58E-06	0.033864	3.49	7180926	1.37E+09	0.114301	0.003176	12.70542
14:35.2	TNB/BTC	TNB/ETH	ETH/BTC	1.29%	7.80E-07	2.35E-05	0.033864	3.49	114323	718686	0.089172	0.001147	4.586128
14:46.1	NPXS/BTC	NPXS/ETH	ETH/BTC	2.77%	1.50E-07	4.58E-06	0.03386	1.132	7180926	1.37E+09	0.037074	0.001026	4.103067
14:46.1	TNB/BTC	TNB/ETH	ETH/BTC	1.27%	7.80E-07	2.35E-05	0.03386	1.132	114323	718686	0.037621	0.000479	1.91684
14:57.2	NPXS/BTC	NPXS/ETH	ETH/BTC	2.78%	1.50E-07	4.58E-06	0.033864	0.727	7180926	1.37E+09	0.02381	0.000662	2.646659
14:57.2	TNB/BTC	TNB/ETH	ETH/BTC	1.29%	7.80E-07	2.35E-05	0.033864	0.727	17975	718686	0.014021	0.000118	0.721077
15:07.8	NPXS/BTC	NPXS/ETH	ETH/BTC	2.79%	1.50E-07	4.58E-06	0.033866	2.078	7180926	1.37E+09	0.068057	0.001895	7.581528
15:07.9	OST/BTC	OST/BNB	BNB/BTC	1.09%	6.70E-06	0.001881	0.003622	13.43	3414	2340	0.015678	0.000171	0.682659
15:07.9	TNB/BTC	TNB/ETH	ETH/BTC	1.29%	7.80E-07	2.35E-05	0.033866	2.078	17975	718686	0.014021	0.000131	0.724431
15:18.8	NPXS/BTC	NPXS/ETH	ETH/BTC	2.78%	1.50E-07	4.58E-06	0.033865	0.77	7180926	1.37E+09	0.025218	0.000702	2.806263
15:18.8	TNB/BTC	TNB/ETH	ETH/BTC	1.25%	7.80E-07	2.35E-05	0.033865	0.77	17975	718686	0.014021	0.000175	0.698551
15:29.5	NPXS/BTC	NPXS/ETH	ETH/BTC	2.78%	1.50E-07	4.58E-06	0.033865	1.771	7180926	1.37E+09	0.058002	0.001614	6.454405
15:29.5	TNB/BTC	TNB/ETH	ETH/BTC	1.20%	7.80E-07	2.35E-05	0.033865	1.771	17975	718686	0.014021	0.000139	0.674348
15:40.2	NCASH/BT	NCASH/ET	ETH/BTC	1.80%	4.40E-07	1.33E-05	0.03388	453.057	25200	858210	0.011088	0.000199	0.796903
15:40.3	TNB/BTC	TNB/ETH	ETH/BTC	1.25%	7.80E-07	2.35E-05	0.03388	453.057	17975	718686	0.014021	0.000175	0.699487
16:01.9	TNB/BTC	TNB/ETH	ETH/BTC	1.25%	7.80E-07	2.35E-05	0.033881	15.225	17975	718686	0.014021	0.000175	0.701163
16:13.2	BTT/BTC	BTT/BNB	BNB/BTC	2.82%	1.90E-07	5.42E-05	0.003626	3.84	80721	1.71E+09	0.013461	0.000379	1.516837
16:13.3	NCASH/BT	NCASH/ET	ETH/BTC	2.39%	4.40E-07	1.33E-05	0.033881	10.859	25200	925718	0.011088	0.000266	1.062022
16:13.3	NPXS/BTC	NPXS/ETH	ETH/BTC	1.80%	4.40E-07	1.33E-05	0.033881	10.859	7180926	1.37E+09	0.355644	0.010067	40.26641
16:13.3	OST/BTC	OST/BNB	BNB/BTC	1.08%	6.70E-06	0.001879	0.003626	3.84	11563	2340	0.013692	0.000148	0.592297

But the party ends when we take into account bids and asks in orderbooks and adding trading fees, the results would show constant negative cycles with a loss due to fees and difference between the ask and bid prices.

date	Base	pair	River	percProf%	wBase	wPair	wRiver	RiverBidQ	PairBidQty	BaseAskQty	maxTrade	profitInBT	profitInUS
49:28.6	ADA/BTC	ADA/BNB	BNB/BTC	99.10%	1.64E-05	0.0045	0.003629	1.21	39682.4	76773	0.004404	-3.97E-05	-0.20034
49:28.6	ADA/BTC	ADA/ETH	ETH/BTC	99.24%	1.64E-05	0.000508	0.032172	0.528	2192	76773	0.017014	-0.00013	-0.64922
49:28.6	ADX/BTC	ADX/BNB	BNB/BTC	99.13%	3.22E-05	0.00886	0.003629	1.21	1260	369	0.004403	-3.83E-05	-0.19317
49:28.6	ADX/BTC	ADX/ETH	ETH/BTC	99.17%	3.22E-05	0.001	0.032172	0.528	1115	369	0.011897	-9.85E-05	-0.49695
49:28.6	AE/BTC	AE/BNB	BNB/BTC	99.01%	0.000109	0.02989	0.003629	1.21	846.3	82.79	0.004408	-4.38E-05	-0.22083
49:28.6	AE/BTC	AE/ETH	ETH/BTC	99.08%	0.000109	0.003374	0.032172	0.528	522.23	82.79	0.009016	-8.29E-05	-0.41822
49:28.6	AGI/BTC	AGI/BNB	BNB/BTC	98.94%	8.75E-06	0.0024	0.003629	1.21	17157.7	99427	0.004411	-4.68E-05	-0.23593
49:28.6	AGI/BTC	AGI/ETH	ETH/BTC	99.04%	8.75E-06	0.000271	0.032172	0.528	9839	99427	0.017049	-0.00016	-0.82457
49:28.6	AION/BTC	AION/BNB	BNB/BTC	98.85%	3.62E-05	0.00992	0.003629	1.21	1144.5	2489.1	0.004416	-5.08E-05	-0.25644
49:28.6	AION/BTC	AION/ETH	ETH/BTC	99.12%	3.62E-05	0.001122	0.032172	0.528	2444.72	2489.1	0.017035	-0.00015	-0.75746
49:28.6	AMB/BTC	AMB/BNB	BNB/BTC	98.59%	1.11E-05	0.00302	0.003629	1.21	15368.7	1493	0.004427	-6.26E-05	-0.31592
49:28.6	AMB/BTC	AMB/ETH	ETH/BTC	98.92%	1.11E-05	0.000342	0.032172	0.528	1101	1493	0.012166	-0.00013	-0.66152
49:28.6	APPC/BTC	APPC/BNB	BNB/BTC	98.51%	1.53E-05	0.00417	0.003629	1.21	9199	142	0.002168	-3.24E-05	-0.16335
49:28.6	APPC/BTC	APPC/ETH	ETH/BTC	98.79%	1.53E-05	0.000472	0.032172	0.528	232	142	0.002168	-2.63E-05	-0.13273
49:28.6	ARDR/BTC	ARDR/BNE	BNB/BTC	99.10%	1.60E-05	0.00439	0.003629	1.21	3000	1768	0.004405	-3.98E-05	-0.20089
49:28.6	ARDR/BTC	ARDR/ETH	ETH/BTC	99.14%	1.60E-05	0.000495	0.032172	0.528	297	1768	0.004746	-4.07E-05	-0.20537
49:28.6	ARK/BTC	ARK/ETH	ETH/BTC	98.97%	0.000124	0.003825	0.032172	0.528	352	16.16	0.001997	-2.07E-05	-0.10423
49:28.6	ARN/BTC	ARN/ETH	ETH/BTC	98.93%	8.56E-05	0.002647	0.032172	0.528	56	19	0.001626	-1.74E-05	-0.088
49:28.6	AST/BTC	AST/ETH	ETH/BTC	99.11%	8.26E-06	0.000256	0.032172	0.528	3526	883	0.007294	-6.47E-05	-0.32635
49:28.6	BAT/BTC	BAT/BNB	BNB/BTC	99.22%	6.00E-05	0.01649	0.003629	1.21	548.7	516	0.004399	-3.43E-05	-0.1731
49:28.6	BAT/BTC	BAT/ETH	ETH/BTC	99.22%	6.00E-05	0.00186	0.032172	0.528	168	516	0.010072	-7.82E-05	-0.39415
49:28.6	BCD/BTC	BCD/ETH	ETH/BTC	98.83%	0.00021	0.00649	0.032172	0.528	299.97	3.264	0.000685	-8.01E-06	-0.0404
49:28.6	BCPT/BTC	BCPT/BNB	BNB/BTC	98.94%	1.10E-05	0.00302	0.003629	1.21	856	66	0.000727	-7.68E-06	-0.03873
49:28.6	BCPT/BTC	BCPT/ETH	ETH/BTC	99.08%	1.10E-05	0.000341	0.032172	0.528	922	66	0.000727	-6.70E-06	-0.03379

This is not surprising since it's much harder to find in exchange arbitrages, so we decided to shift our attention to cross exchange arbitrage where opportunities are more common.

Code

```

import ccxt
from ArbitrageExchange import ArbitrageExchange
import datetime
import time

'''configurations'''
exchanges = {}
min_profit = -1
min_profit_usd = -1
freq = 5
minInvestment = 0.00
baseCurr = 'BTC'
'''

def print_header():
    print ('date' + "," + 'Base' + "," + 'pair' + "," + 'River' + "," + 'percProf' +
+ "%"
        + "," + 'wBase' + "," + 'wPair' + "," + 'wRiver' + "," + 'RiverBidQty' +
        + "," + 'PairBidQty' + "," + 'BaseAskQty' + "," + 'maxTrade' + "," +
'profitInBTC' + ","
        + 'profitInUSDT')

def triangular_arbitrage_explorer():
    binance = ccxt.binance()

    binanceArb = ArbitrageExchange(binance)
    binanceCurrencies = binanceArb.currencies.keys()
    binancePairs = binanceArb.symbols
    while(1):
        binanceTickers = binance.fetch_tickers()
        s = baseCurr + "/USDT"

```

```

USDTBase = binanceTickers[s]['ask']
for pair in binancePairs:
    marketData = binance.market(pair)
    Base = str(marketData['baseId']) + "/" + baseCurr
    River = str(marketData['quoteId']) + "/" + baseCurr
    if Base in binancePairs and River in binancePairs:
        wBase = binanceTickers[Base]['ask']
        wPair = binanceTickers[pair]['bid']
        wRiver = binanceTickers[River]['bid']
        if wBase == 0:
            continue
        profit = 1 / wBase * wPair * wRiver * 0.998 ** 3
        if profit > 1 + min_profit:
            RiverBidQty = float(binanceTickers[River]['info']['bidQty'])
            PairBidQty = float(binanceTickers[pair]['info']['bidQty'])
            BaseAskQty = float(binanceTickers[Base]['info']['askQty'])
            maxTrade = min(min(RiverBidQty, PairBidQty * wPair) / wPair,
                           BaseAskQty) * wBase
            if maxTrade > minInvestment:
                profitInBase = maxTrade * (profit - 1)
                profitInUSDT = profitInBase * USDTBase
                if profitInUSDT > min_profit_usd:
                    percProf = str((profit) * 100)
                    print ((str(datetime.datetime.now()) + "," + Base + ","
+ str(
                            pair) + "," + River + "," + percProf + "%"
                            + "," + str(wBase) + "," + str(wPair) + "," +
str(wRiver) + "," + str(RiverBidQty) +
                            "," + str(PairBidQty)) + "," + str(BaseAskQty) +
                            "," + str(maxTrade) + "," + str(
                                profitInBase) + ","
                                + str(profitInUSDT))
                    time.sleep(freq)

def main():
    print_header()
    triangular_arbitrage_explorer()

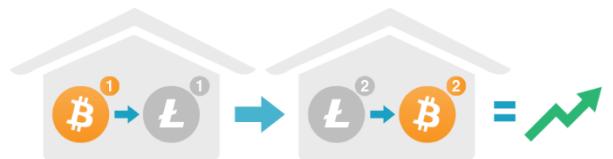
if __name__ == "__main__":
    main()

```

Cross Exchange Arbitrage:

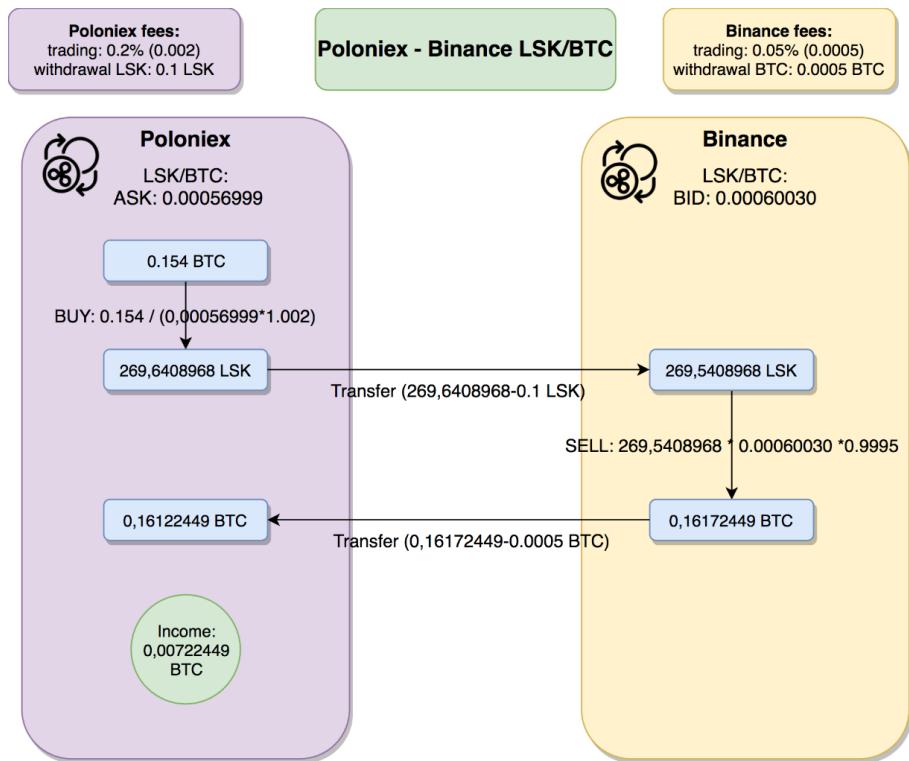
Our strategy to trade coin C between exchanges E1 and E2 implies buying C at E1, transferring C coins to E2 and selling them there, then return the assets to E1. We want to perform all these steps as quickly as we can because we don't want somebody taking advantage of us, so we are going to execute each trade immediately, where immediately means buying at the lowest price somebody is willing to sell, i.e. lowest ask value, and selling at the highest price somebody is willing to buy, i.e. highest bid value. So, let's define the steps:

- At E1, set a buy market order for C.
- Transfer the C coins to E2.
- At E2, set a sell market order for C.
- Withdraw from E2 and deposit into E1



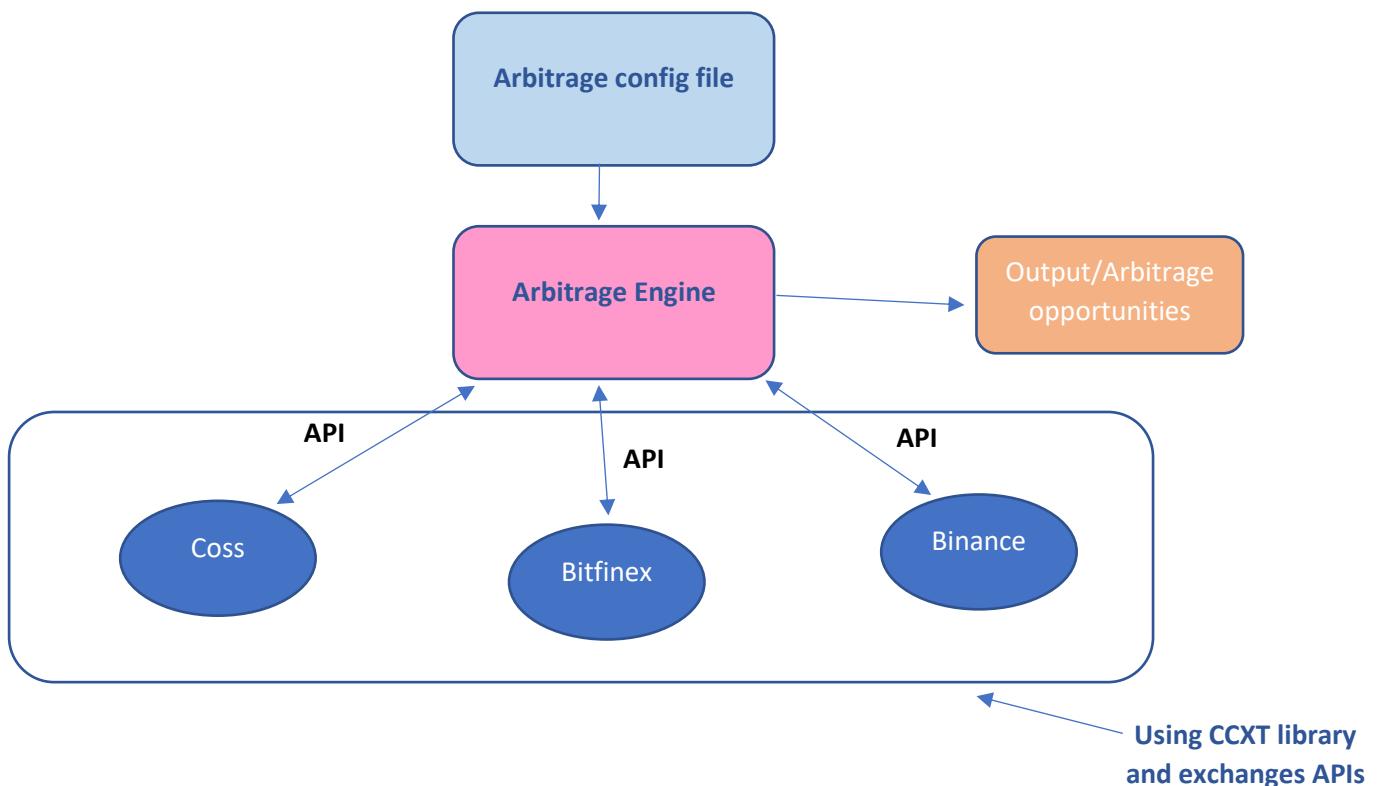
Why do these arbitrage opportunities exist?

1. Liquidity: difference in the trading volumes at different exchanges, meaning difference in supply and demand, affects the prices. On the big exchanges prices fluctuate less than on the smaller or new ones.
2. Geography: something can happen in the morning in Europe, that influences prices, the most of people in US are still sleeping, hence the price difference.
3. Listings: price difference when a crypto coin gets listed in one of the major exchanges.



Our Implementation

Code architecture:



Real life example

Let's look at a real-life example of an opportunity found by our system:

```
estimated initial profit: 7.035397579376168%
trading: DASH srcExchange: Coss dstExchange: Binance
srcValue: 0.637830619999999ETH invested amount of altcoin: 0.938
dstValue: 0.6634330299828799ETH after selling amount of altcoin: 0.926124
final Amount: 0.6634330299828799 initial Amount: 0.637830619999999 profit in ETH: 0.025602409982879992 %: 4.0139825809679675
```

Looking at the Dash bid book at Binance:

0.72781	1.422	1.03494582
0.72759	0.717	0.52168203
0.72755	12.764	9.28644820

And Dash ask book at Coss:

Price (ETH)	Amount (DASH)	Total (ETH)
0.70100	0.316	0.22151599
0.68000	0.986	0.67047999
0.67999	0.938	0.63783060

We can see the big price difference and sufficient volume to make a profitable trade, The system took care of taking fees into consideration.

Arbitrage configuration module:

The configuration module accepts new exchanges and coins adding exchange to the arbitrage directory:

```
exchanges_list = ["Binance", "Bitfinex", "Hitbtc", "Huobipro", "Kraken", "Kucoin", "Livecoin", "Okex",
| "Poloniex", "Southxchange", "Coss", "Bittrex"]
```

you will need to add the exchange to the dictionary exchangesDir in arbitrage_cfg.py, with the following fields:

- api: the api key of your account if there exist one.
- secret: the secret key of your account if there exist one.
- tradingFees: the taker fee of this exchange.
- withdrawalFees: the withdrawal fees of the currencies that you can withdraw from this website.
- depositFees: the deposit fees of the currencies in this exchange.

```
exchangesDir = {
    'Binance': {...},
    'Bitfinex': {...},
    'Kraken': {'api': 'K6htEIVd2WiVtYcnsMEFkqbB8+vTydoRZ6g4aoH83rSHC9BdMrWtujIG',
               'secret': 'bNqB59gDd3QpUqcYrb/3XMyHzV0p4IYbsY93157agU7YiQ8QNiUBVTZRmxOXHd9N8jt58AxGUKyW5X+1h18JQg==',
               'tradingFees': 0.0026,
               'withdrawalFees': {'BTC': 0.0005, 'ETH': 0.005, 'XRP': 0.02, 'XLM': 2e-05, 'LTC': 0.001,
                                  'DOGE': 2,
                                  'ZEC': 0.0001, 'REP': 0.01, 'ETC': 0.005, 'MLN': 0.003, 'XMR': 0.0001,
                                  'DASH': 0.005, 'GNO': 0.005, 'EOS': 0.05, 'BCH': 0.0001, 'XTZ': 0.2},
               'depositFees': None,
               'minWithdrawals': None},
    'Coss': {...},
    'Kucoin': {...}, // Kucoin entry is highlighted
    'Livecoin': {...},
    'Bittrex': {...},
    'Hitbtc': {...},
    'Huobipro': {...},
    'Southxchange': {...},
}
```

Adding a currency to use in arbitraging:

you need to add the currency market symbol in the coins list in arbitrage_cfg.py file:

```
coins = ['AGI', 'AION', 'AMB', 'APPC', 'ARDR', 'ARK', 'ARN', 'AST', 'BAT', 'BCC', 'BCD', 'BCH',
         'BCN', 'BCPT', 'BLZ', 'BNB', 'BNT', 'BQX', 'BRD', 'BSV', 'BTC', 'BTG', 'BTS', 'BTT', 'CDT', 'CELR',
         'CHAT', 'CLOAK', 'CMT', 'CND', 'CVC', 'DASH', 'DATA', 'DCR', 'DENT', 'DGD', 'DLT', 'DNT', 'DOCK', 'EDO',
         'ELF', 'ENG', 'ENJ', 'EOS', 'ETC', 'ETH', 'EVX', 'FET', 'FUEL', 'FUN', 'GAS', 'GNT', 'GO', 'GRS', 'GTO',
         'GVT', 'GXS', 'HC', 'HOT', 'HSR', 'ICN', 'ICX', 'INS', 'IOT', 'IOTA', 'IOTX', 'KEY', 'KMD', 'KNC',
         'LEND', 'LINK', 'LOOM', 'LRC', 'LSK', 'LTC', 'LUN', 'MANA', 'MCO', 'MDA', 'MFT', 'MITH', 'MOD', 'MTH',
         'MTL', 'NANO', 'NAS', 'NAV', 'NCASH', 'NEBL', 'NEO', 'NPXS', 'NULS', 'NXS', 'OAX', 'OMG', 'ONG', 'ONT',
         'OST', 'PAX', 'PHX', 'PTVX', 'POA', 'POE', 'POLY', 'POWR', 'PPT', 'QKC', 'QLC', 'QSP', 'QTUM', 'RCN',
         'RDN', 'REN', 'REP', 'REQ', 'RLC', 'RPX', 'RVN', 'SALT', 'SC', 'SKY', 'SNGLS', 'SNM', 'SNT', 'STEEM',
         'STORJ', 'STORM', 'STRAT', 'SUB', 'SYS', 'THETA', 'TNB', 'TNT', 'TRIG', 'TRX', 'TUSD', 'USDC', 'USDS',
         'USDT', 'VEN', 'VET', 'VIA', 'VIB', 'VIBE', 'WABI', 'WAN', 'WAVES', 'WINGS', 'WPR', 'WTC', 'XEM', 'XLM',
         'XMR', 'XRP', 'XVG', 'XZC', 'YOYOW', 'ZEC', 'ZEN', 'ZIL', 'ZRX']
```

Base currency, min, max invest and profit configuration:

```
.....  
baseCurrencies = {'ETH': {'min_invest': 0.5, 'max_invest': 15, 'step': 0.5},  
                  'BTC': {'min_invest': 0.03, 'max_invest': 0.5, 'step': 0.01},  
                  'XMR': {'min_invest': 3, 'max_invest': 25, 'step': 1},  
                  'EOS': {'min_invest': 20, 'max_invest': 400, 'step': 5},  
                  'XRP': {'min_invest': 400, 'max_invest': 7000, 'step': 30}}  
min_profit_per = 0  
limitThreshold_per = 7
```

the base currencies are the ones that the client already have in the exchanges and want to use the arbitrage to get more.

- Min_invest: is the minimum amount of the specified currency that you would want the system to try and find opportunity for.
- Max_invest: is the maximum amount that the client would want to invest.
- Step: while searching for the optimal amount to invest the system would try invest amount from min invest to max , step is the amount that the system wil, add to reach max.
- Min_profit_per: the minimum profit that the client want the system to do arbitrage on.
- limitThreshold_per: the profit percentage that the system sends alert on to check validity, can be too good to be true.

The Engine:

Start by fetching all the tickers from exchanges to estimate arbitrage opportunities, we fetch the tickers from the exchange by using the API libraries with ccxt.

```
# fetching all tickers from specified exchanges  
tickers = {}  
for e in exchanges:  
    try:  
        tickers[e] = clients[e].fetch_tickers()  
    except:  
        print("couldn't fetch ticker from: " + str(e))  
        exchanges.remove(e)  
  
print("fetched tickers successfully....")
```

Then, the system searches for arbitrage opportunities. when it predicts one, the system fetches the order book, and checks if this opportunity is real, taking into consideration the fees and all the requests from the client like min_invest, min_profit, max_invest ...

```
try:
    wask = tickers[src][pair]['ask']
    wbid = tickers[dst][pair]['bid']
    profit = ((wbid / wask) - 1) * 100
except:
    continue
if profit > 0:
    try:
        srcBook = clients[src].fetch_order_book(pair)
        dstBook = clients[dst].fetch_order_book(pair)
    except:
        continue
```

Optimizing the profit:

The engine starts by trying to find opportunities with min invest, and it keeps increasing the investment (the system increases the investment according to the configurations that was stated b by the client) until it reaches the max profit percentage, when it gets to the maximum profit it checks that it still profitable trade after taking out the fees.

How we see this in the code:

Optimizing the amount to invest in the source exchange:

```
curr_profit = 0
while possible_profit:
    while srcValue < step_invest and sizeI > i:
        amount = min(srcBook['asks'][i][1],
                     (baseCurrData['max_invest'] - srcValue) / srcBook['asks'][i][0])
        srcValue = srcValue + amount * srcBook['asks'][i][0]
        srcAmount += amount
        i += 1
```

To maximize the profit safely, we do order book matching at the source and destination exchanges.

```

while srcAmount > dstAmount and sizeJ > j:
    amount = min(dstBook['bids'][j][1], (srcAmount - dstAmount))
    dstValue = dstValue + amount * dstBook['bids'][j][0]
    dstAmount += amount
    j += 1

```

example of taking fees into considerations after getting to max profit:

```

while srcAmount > dstAmount and sizeJ > j:
    amount = min(dstBook['bids'][j][1], (srcAmount - dstAmount))
    dstValue = dstValue + amount * dstBook['bids'][j][0]
    dstAmount += amount
    j += 1

dstValue *= (1 - exchangesDir[dst]['tradingFees'])
try:
    dstValue -= exchangesDir[dst]['withdrawalFees'][baseCurr]
except:
    continue
profitInBase = dstValue - srcValue
profitInPer = (dstValue / srcValue - 1) * 100

```

Exceptions:

Every API transaction is gated by try and except block, because we frequently lose connection to the server and need to reconnect in the next iteration.

Also there some currencies that exist only in some markets so when we tried accessing the currency information in an exchange, we gate it in try and except block as well.

In case of failure the system moved to the next opportunity.

Simulation results

Over 24H period our system found **3242** cross exchange arbitrage opportunities.

The following table shows the discovered profitable opportunities' details. The profit is calculated including orderbook matching and trading and transfer fees.

time	Estimated profit%	Base	Intermed iate	srcExcha nge	dstExcha nge	initialBaseA mount	initialAmo untOfAlt... CoinAtDst	finalBaseAm ount	amountOfAlt CoinAtDst	ProfitInBase	TotalProfit%
2019-04-12 12:57:40	2.3925906869	ETH	IOST	Binance	Okex	1.06149738	13667	1.0706172698	13652.333	0.0091198898	0.8591532985
2019-04-12 12:58:05	7.6133472551	ETH	DASH	Coss	Binance	0.61539095	0.905	0.6428060488	0.89319	0.0274150988	4.4549076959
2019-04-12 12:58:06	7.4095207282	ETH	DASH	Coss	Hitbtc	0.61539095	0.905	0.6414780276	0.89319	0.0260870776	4.2391064662
2019-04-12 12:58:07	7.2706951573	ETH	DASH	Coss	Kucoin	0.61539095	0.905	0.6392316377	0.89319	0.0238406877	3.8740718686
2019-04-12 12:58:08	7.6486418918	ETH	DASH	Coss	Okex	0.61539095	0.905	0.6419425072	0.89319	0.0265515572	4.3145836261
2019-04-12 12:58:09	7.5435212283	ETH	DASH	Coss	Bittrex	0.61539095	0.905	0.6455436862	0.89319	0.0301527362	4.8997691985
2019-04-12 12:58:25	2.380952381	BTC	IOST	Binance	Okex	0.043848	17400	0.0442772612	17381.6	0.0004292612	0.9789755702
2019-04-12 12:58:28	4.4371245883	BTC	STORJ	Binance	Coss	0.05176386	1003	0.0527456803	998.497	0.0009818203	1.8967293243
2019-04-12 12:58:29	2.380952381	BTC	IOST	Binance	Bittrex	0.09047808	35904	0.0918057649	35867.096	0.0013276849	1.4674105715
2019-04-12 12:59:10	2.380952381	BTC	IOST	Kucoin	Okex	0.0655419041	26008.6921	0.0664323205	25981.683408	0.0008904164	1.3585452056
2019-04-12 12:59:12	3.5169210352	BTC	GVT	Kucoin	Coss	0.0451911218	59.974946	0.0453297764	59.764971054	0.0001386546	0.306818256
2019-04-12 12:59:12	2.380952381	BTC	IOST	Kucoin	Bittrex	0.0655419041	26008.6921	0.0663651613	25981.683408	0.0008232572	1.2560777014
2019-04-12 13:00:53	2.3895169579	ETH	IOST	Binance	Okex	2.45554064	31546	2.4706648307	31513.454	0.0151241907	0.6159210085
2019-04-12 13:01:20	7.6133472551	ETH	DASH	Coss	Binance	0.61539095	0.905	0.6423131735	0.89319	0.0269222235	4.374816279
2019-04-12 13:01:21	7.4096677892	ETH	DASH	Coss	Hitbtc	0.61539095	0.905	0.6408023436	0.89319	0.02541113936	4.1293089555
2019-04-12 13:01:22	7.2368711305	ETH	DASH	Coss	Kucoin	0.61539095	0.905	0.6404754515	0.89319	0.0250845015	4.076189538
2019-04-12 13:01:23	7.5015809056	ETH	DASH	Coss	Okex	0.61539095	0.905	0.6410842774	0.89319	0.0256933274	4.1751227319
2019-04-12 13:01:24	7.5435212283	ETH	DASH	Coss	Bittrex	0.61539095	0.905	0.6455436862	0.89319	0.0301527362	4.8997691985
2019-04-12 13:01:42	2.371541502	BTC	IOST	Binance	Okex	0.5	197628.4...	0.5003401046	197429.83004	0.0003401046	0.068020912
2019-04-12 13:01:45	4.4776119403	BTC	STORJ	Binance	Coss	0.05176386	1003	0.0527456803	998.497	0.0009818203	1.8967293243
2019-04-12 13:01:46	2.766798419	BTC	IOST	Binance	Bittrex	0.5	197628.4...	0.5026931	197429.83004	0.0026931	0.5386200093
2019-04-12 13:02:33	3.1746031746	BTC	IOST	Kucoin	Bittrex	0.2506668494	98687.736	0.2515702795	98588.048264	0.0009034301	0.3604106791
2019-04-12 13:04:19	2.4521761458	ETH	IOST	Binance	Okex	0.54110183	6947	0.5456772825	6939.053	0.0045754525	0.8455806782
2019-04-12 13:04:31	1.3977128335	ETH	IOST	Kucoin	Okex	0.9279762623	11791.3121	0.9299210866	11778.520788	0.0019448243	0.2095769497
2019-04-12 13:04:43	7.582464448	ETH	DASH	Coss	Binance	0.61539095	0.905	0.6425433154	0.89319	0.0271523654	4.4122139573
2019-04-12 13:04:44	7.3934910808	ETH	DASH	Coss	Hitbtc	0.61539095	0.905	0.641381756	0.89319	0.025990806	4.2234624937
2019-04-12 13:04:44	7.2059883234	ETH	DASH	Coss	Kucoin	0.61539095	0.905	0.6398151519	0.89319	0.0244242019	3.9688919511
2019-04-12 13:04:45	7.5015809056	ETH	DASH	Coss	Okex	0.61539095	0.905	0.6419425072	0.89319	0.0265515572	4.3145836261

The system discovered opportunities with huge profit of ~21%:

Base	Intermediate	srcExchange	dstExchange	initialBaseAmount	initialAmountOf AltCoin	finalBaseAmount	amountOfAltCoinAt Dst	ProfitInBase	TotalProfit%
BTC	THETA	Binance	Livecoin	0.03061424	1350	0.0373044481	1348.55	0.0066902081	21.853255673
BTC	THETA	Binance	Livecoin	0.03013187	1328	0.0366978503	1326.572	0.0065659803	21.790815786
BTC	THETA	Binance	Livecoin	0.03110711	1371	0.0378633082	1369.529	0.0067561982	21.719144747
BTC	THETA	Binance	Livecoin	0.03376067	1488	0.0396661335	1486.412	0.0059054635	17.492139497
BTC	THETA	Binance	Livecoin	0.03466974	1554	0.0371878162	1552.346	0.0025180762	7.2630375687

This raises an alert, and with further checking, all deposits and withdrawals of THETA coin on livecoin exchange are offline!

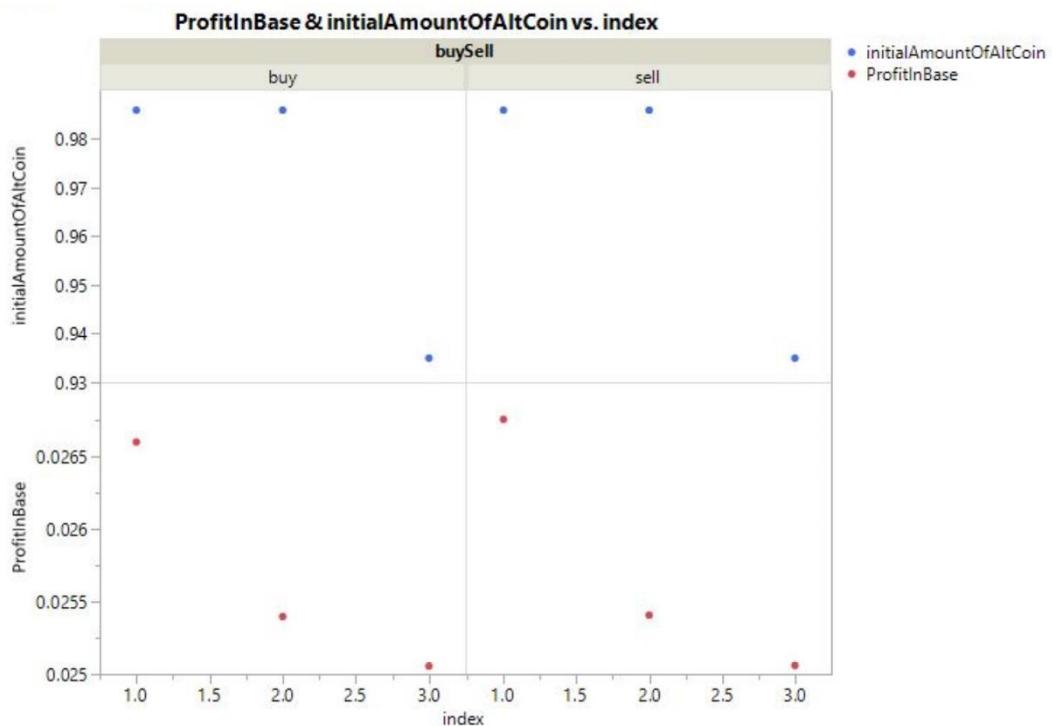


Profitable opportunities:

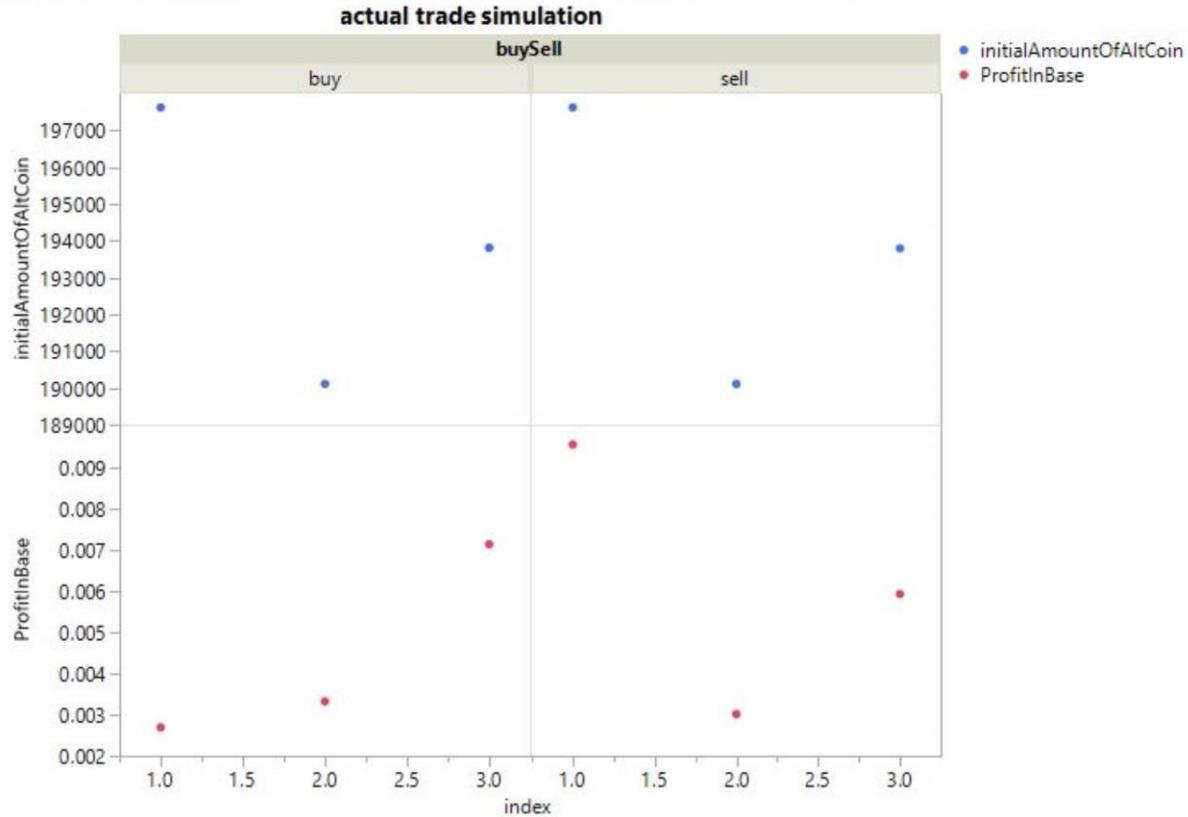
We chose 3 arbitrages that our system would have performed at different times

time	Estimated profit%	Base	Intermediate	srcExchange	dstExchange	initialBaseAmount	initialAmountOfAlt...	finalBaseAmount	amountOfAltCoinAt Dst	ProfitInBase	TotalProfit%
2019-04-13 04:02:22	6.8602941176	ETH	DASH	Coss	Binance	0.67048	0.986	0.6970793993	0.974028	0.0265993993	3.9672174104
2019-04-13 04:06:25	6.8617647059	ETH	DASH	Coss	Binance	0.67048	0.986	0.6972350879	0.974028	0.0267550879	3.9904378847
2019-04-13 06:16:04	6.2348941176	ETH	DASH	Coss	Bittrex	0.67048	0.986	0.6958769449	0.974028	0.0253969449	3.7878750896
2019-04-13 06:23:55	6.2363691176	ETH	DASH	Coss	Bittrex	0.67048	0.986	0.6958868357	0.974028	0.0254068357	3.789350274
2019-04-13 10:27:16	6.4992426359	ETH	DASH	Coss	Bittrex	0.63579065	0.935	0.6608457449	0.92313	0.0250550949	3.9407775068
2019-04-13 10:33:59	6.5001338255	ETH	DASH	Coss	Bittrex	0.63579065	0.935	0.660850421	0.92313	0.025059771	3.9415129774

And we double checked after 3-4 minutes to see if the same opportunity still exists, this is the time we assume it takes to complete these transactions. The results were encouraging, showing consistent profit of 0.025 ETH per transaction.



Here we can see different arbitrage opportunity found at different times and currencies/exchanges with the same positive consistent profit.



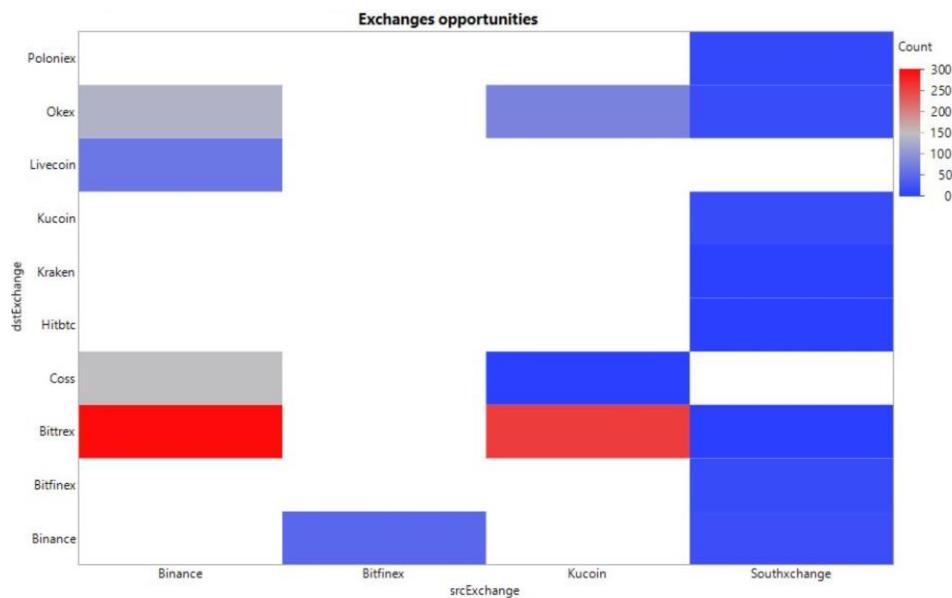
buy →

sell ←

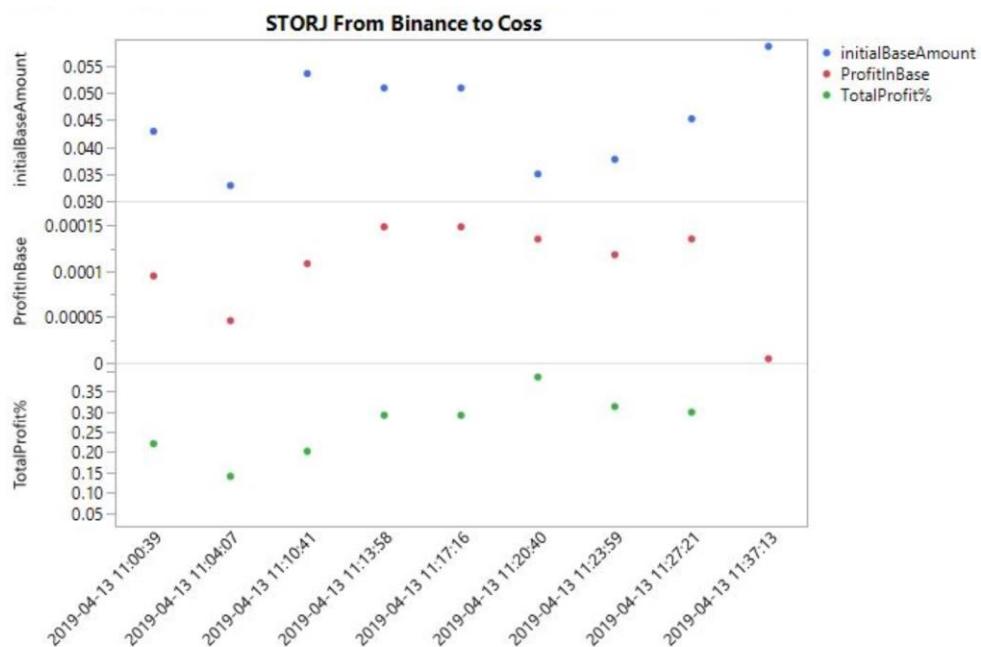
time	Estimated profit%	Base	Intermediate	srcExchange	dstExchange	initial Bas...	initialAmountOf AltCoin	finalBaseAmount	amountOfAltCoinAtDst	ProfitInBase	TotalProfit%
2019-04-12 13:01:46	2.766798419	BTC	IOST	Binance	Bittrex	0.5	197628.4585	0.5026931	197429.830...	0.0026931	0.5386200093
2019-04-12 13:05:09	2.371541502	BTC	IOST	Binance	Bittrex	0.5	197628.4585	0.5095649017	197429.830...	0.0095649017	1.9129803306
2019-04-12 14:58:57	1.1406844106	BTC	IOST	Binance	Bittrex	0.5	190114.06844	0.5033241486	189922.954...	0.0033241486	0.6648297178
2019-04-12 15:02:21	1.5209125475	BTC	IOST	Binance	Bittrex	0.5	190114.06844	0.5030149951	189922.954...	0.0030149951	0.6029990205
2019-04-12 17:00:39	6.2256809339	BTC	IOST	Binance	Bittrex	0.5	193813.37984	0.5071417583	193618.566...	0.0071417583	1.4283516674
2019-04-12 17:06:51	5.8139534884	BTC	IOST	Binance	Bittrex	0.5	193798.44961	0.5059331374	193603.651...	0.0059331374	1.1866274853

Cross Exchange arbitrage data analysis:

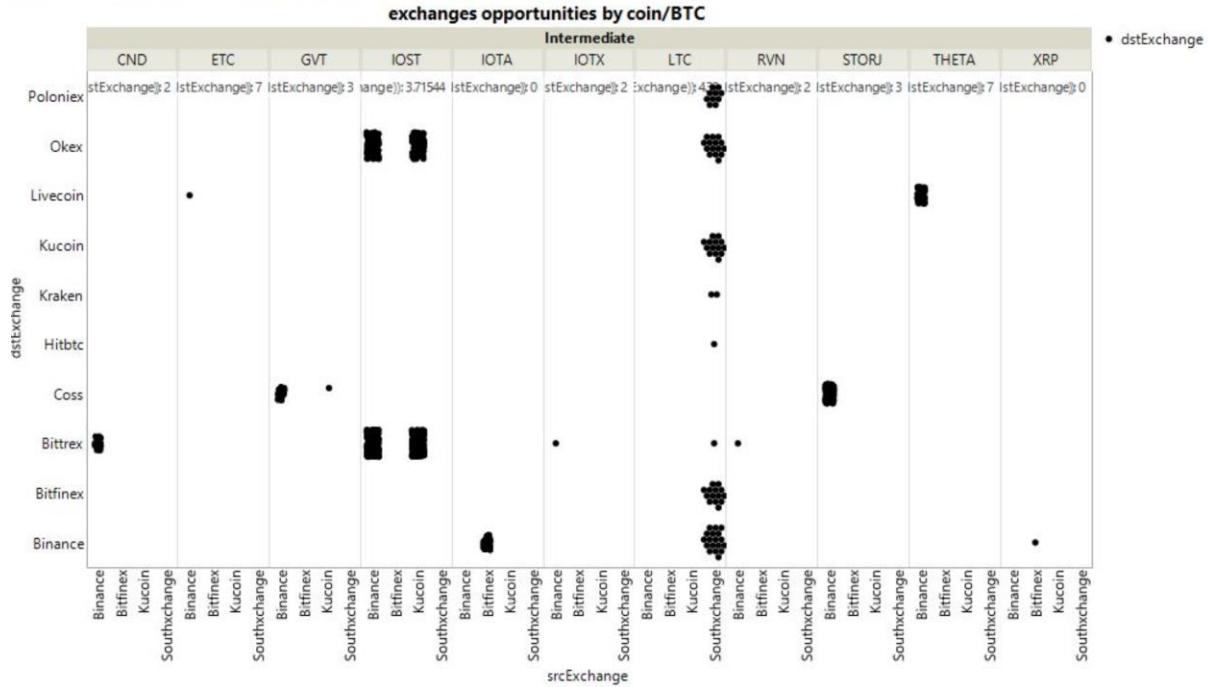
1. In the following graph we see the distribution of opportunities across exchanges. For example, we can see that most opportunities appeared from Binance to Bittrex, while no opportunities appeared between Kucoin and Hitbtc. This is of course dependent on the exchanges we used and the currencies we chose to work with, updating our system with new exchanges and currencies will improve our understanding of the nature of the arbitrage opportunities.



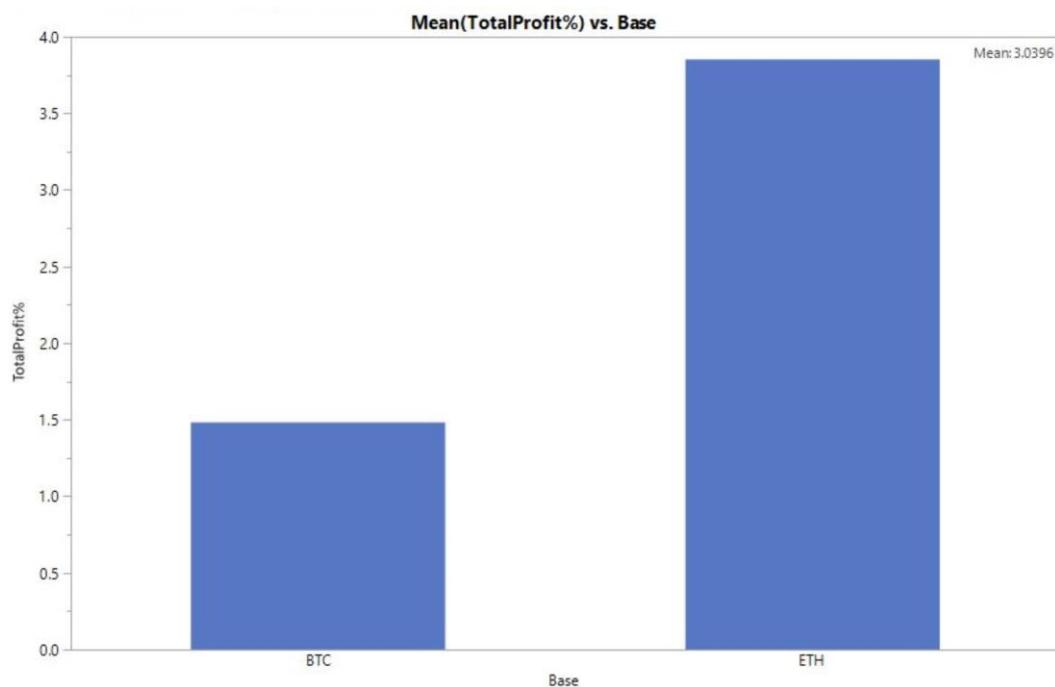
2. In the following graph we monitor an arbitrage opportunity for 37 minutes and see the fluctuations maintaining profit the whole time



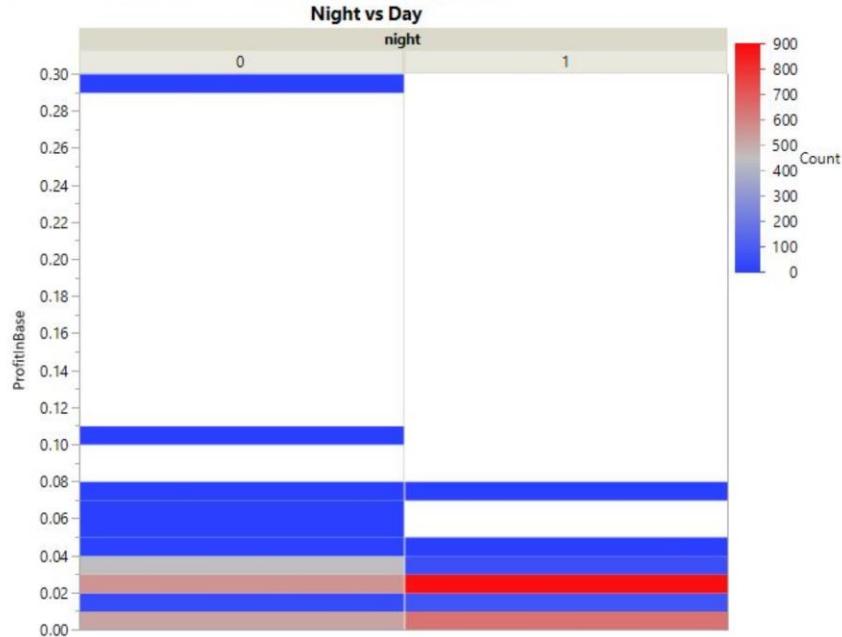
3. The following is a more detailed graph showing the redundancy of arbitrage opportunities by currency, for example we can see that LTC opportunities appear from southExchange to almost all other exchanges, which means that the ask price there is lower than most other exchanges.



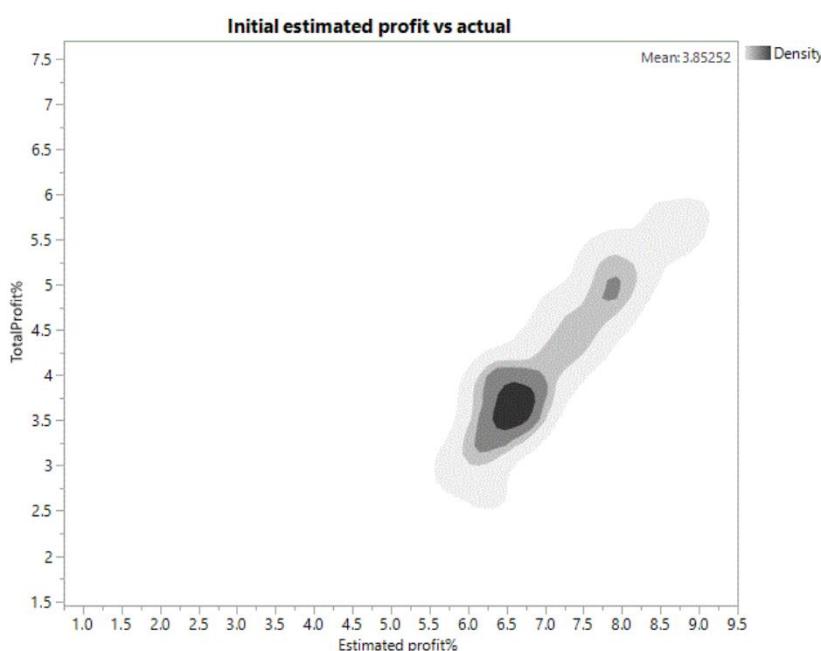
4. In the following graph we see a comparison of the profitability of arbitrages starting with ETH vs BTC, It's clear that ETH arbitrages yield higher profit percentage, and transfer times of ETH and the withdrawal/miner's fee are a fraction of bitcoin's.



5. In the following graph we can see the number of opportunities and profit for day or night, where day is 12 hours from 8:00 to 20:00 and night is 20:00 to 8:00 the following day in Jerusalem time. For example we can see that most realistic-profit opportunities are found at night (the red triangle).



6. Our system works in a way that it first estimates profit quickly in shallow orderbook checking, and then performs orderbook matching and realizes the real profit opportunities, mainly affected by trading fees and depth of orderbook, in the following graph we can see that estimated profits of less than 6.5% are not profitable, also estimates of more than 7.5% are too good and end up being rejected due to high transfer fees of blocked withdrawals.



Project challenges:

Here is a list of the main challenges we faced during this project:

- Exchanges structure and the different currencies, how the order book works (bids and asks), exploring the different markets and understanding volume and fees.
- APIs, how to connect and communicate with many different cryptocurrency exchanges from our systems, we had to create trading accounts on select exchanges and connect to them using an API key and using the connection to extract rates date and read orderbooks from our system.
- Maintaining connections with exchanges, some exchanges go offline from time to time or have limits on the number of requests, and we didn't want one disconnection to take our whole system down.
- Some exchanges lock currencies deposit/withdrawals at unpredictable times, causing differences in prices and seemingly good arbitrage opportunities to appear that can't be executed, we try to reduce this by alerting the trader from too good to be true arbitrage opportunities.
- Some exchanges have different symbols for the same underlying currency or worse, same symbol for different currencies, especially in a market with over 2500 currencies, this gets tricky, the solution is to use unified identification numbers for each currency regardless of symbol, we can do this through the ccxt library and it solves our issue.
- In in-exchange arbitrage we tried building graphs and running bellman ford on them to find profitable cycles but didn't proceed in this direction because of performance issues: It was very time consuming to iterate over all cycles, and we figured that triangular arbitrage is the most profitable in-exchange arbitrage and not profitable in general so we moved our focus toward cross exchange arbitrage.

Cryptocurrency arbitrage market challenges and risks

1. *Volume*

Volume is low. Really low. Clear signals that are profitable are often constrained by very small volume.

2. *Order Books*

Because volume is low, order books are thin. This means that the ‘depth,’ or available interest at different prices is quite thin, which makes understanding the full scope of an opportunity much more complicated. For example, there may be a great trade that can net 5%, but only with a maximum of USD\$3,000. The next opportunity might net 3% with an opportunity size of \$5,000. And so on.

3. *Network Transfer Speeds*

Network transfer speeds are VERY unpredictable. Bitcoin (BTC) transfers between wallets can be anywhere from 10 minutes to 8 hours. Network health and bandwidth is the key factor here. BTC has different considerations than ETH, and so on. It also might take time to validate deposits and withdrawals, adding to that wallets maintenance and possibility of servers being down at times, for that reason it’s preferable to do arbitrage on big established and reliable exchanges, although this option can be less profitable as many arbitrage bots are likely running on them.

4. *Exchange Reliability*

The exchanges are reasonably reliable. However, they are not even close to the level of an E-Trade or Bloomberg Terminal. Some of the exchanges have been shut down by governments. Some of the exchanges scale way too fast, like Kraken. Sometimes trades are not executed because of server overload. Some of the exchanges’ API endpoints are quite unreliable at times.

.

Fees

There are two major sources of fees at the exchanges:

- **Trading fee:** These fees are payed on every trade, usually a fixed percentage of the transaction ranging from 0.01% to 0.25% depending on the exchange and the trade volume.
Most exchanges have different trading fee rate for taker's and maker's fees. A trade gets the **taker fee** if the trade order is matched immediately against an order already on the order book, which is **removing liquidity**.
A trade gets the **maker fee** if the trade order is not matched immediately against an order already on the order book, which is **adding liquidity**.
- **Cryptocurrency deposit/withdrawal fees:** Usually, deposit of a cryptocoin is free, but if an exchange needs to create a new address for your chosen coin, then they will charge blockchain (or network fee). Withdrawals fee are depending on the crypto coin. Also some exchanges don't charge withdrawal fee, for example Coinbase Pro.

Fees on blockchain or network fees

Every crypto coin is connected to a blockchain. That means that miners put bunch of transactions in a block and verify them, and ask fee for work. This fee is called blockchain fee or network fee. Some exchanges don't take fee for the deposit or the withdrawal of a cryptocurrency. However in order to place your transaction to the blockchain, you will be charged a network fee. For example, you would place your freshly bought Bitcoin from Coinbase to your wallet or offline storage. You'd have to pay a fee for the miner who creates the block holding that transaction.

Some math

Step (0) Some symbols to denote amounts of various things -

- x is the amount of BTC used to buy ETH
- y is the amount of ETH bought
- p is the price in BTC at the 'buy low' exchange
- q is the price in BTC at the 'sell high' exchange
- f is the fee % at the 'buy low' exchange
- g is the fee % at the 'sell high' exchange
- δ is the amount of BTC gained by the arbitrage

Step (1) buy ETH using BTC, taking into account the fees incurred

$$(x/p)(1-f)=y$$

Step (2) the same amount of ETH is used on the second exchange to sell into BTC, with fees

$$yq(1-g)=x+\delta$$

The total balance (across all exchanges) of ETH needs to be kept at a fixed value leaving the profits in BTC while maintaining liquidity of ETH for future arbitrage opportunities. The profit needs to be positive, so start by isolating profit and the ETH amount to make calculations easier -

$$y = (x/p)(1-f) = (x+\delta)/[q(1-g)]$$

$$(x+\delta)/x = (q/p)(1-f)(1-g)$$

$$\delta = x[(q/p)(1-f)(1-g) - 1]$$

Lets test this equation using values from a previous example -

$$\delta = 1 * [(0.076485/0.076396)*0.9975*0.998 - 1] = -0.00333525$$

The result is the same number of lost satoshis as before, which suggests that the math is most likely correct.

Tip: To quickly check when an arbitrage opportunity has profit in it, you can use this simple inequality -

$$q(1-f)(1-g) > p \text{ means } \delta > 0$$

Future Development

- Check profitability of more than 3 pairs in-exchange arbitrage.
- Compare profitability of different arbitrage approaches discussed in related work.
- Enhance integrity and performance of the arbitrage system.
 - Handle network exceptions and log network errors.
 - Increase performance by making the code parallel, preferably work with `async.io`.
- Add and constantly update fee rates, new coins and exchanges.
- Build a probabilistic model to decide on the best available arbitrages and reward transactions (source exchange – currency – destination exchange) to be preferable in the future.

CCXT

<https://github.com/ccxt/ccxt>

A JavaScript / Python / PHP library for cryptocurrency trading and e-commerce with support for many bitcoin/ether/altcoin exchange markets and merchant APIs.

The **CCXT** library is used to connect and trade with cryptocurrency exchanges and payment processing services worldwide. It provides quick access to market data for storage, analysis, visualization, indicator development, algorithmic trading, strategy backtesting, bot programming, and related software engineering.

It is intended to be used by **coders, developers, technically-skilled traders, data-scientists and financial analysts** for building trading algorithms.

Current feature list:

- support for many cryptocurrency exchanges
- fully implemented public and private APIs
- optional normalized data for cross-exchange analytics and arbitrage
- an out of the box unified API that is extremely easy to integrate
- works in Node 7.6+, Python 2 and 3, PHP 5.4+, and web browsers

ccxt library currently supports 137 exchanges, full list and manual, examples are in github

Certified Cryptocurrency Exchanges

logo	id	name	ver	doc	certified
	binance	Binance	*	API	CCXT certified
	bitfinex	Bitfinex	1	API	CCXT certified
	bittrex	Bittrex	1.1	API	CCXT certified
	coss	COSS	1	API	CCXT certified
	kraken	Kraken	0	API	CCXT certified
	kucoin	KuCoin	2	API	CCXT certified
	theocean	The Ocean	1	API	CCXT certified
	upbit	Upbit	1	API	CCXT certified

References:

- **Arbitrage Trading Systems for Cryptocurrencies. Design Principles and Server Architecture.** Cristian PĂUNA
- **Constructing Cointegrated Cryptocurrency Portfolios for Statistical Arbitrage,** Tim Leung, Hung Nguyen
- **Trading and Arbitrage in Cryptocurrency Markets,** Igor Makarov and Antoinette Schoar

Useful links:

- ccxt library, manual and examples: <https://github.com/ccxt/ccxt>
- info about exchanges and trading volume: <https://coinmarketcap.com/rankings/exchanges/>
- <https://medium.com/coinmonks/cryptocurrency-arbitrage-strategies-part-i-20e9dd327919>
- <https://medium.com/coinmonks/arbitrage-strategies-part-ii-oscillator-f48e6ebec507>
- <https://blog.coinfabrik.com/an-efficient-algorithm-to-exploit-arbitrage-opportunities-in-crypto-markets/>
- <https://steemit.com/arbitrage/@kesor/the-math-behind-cross-exchange-arbitrage-trading>
- <https://www.tokenspread.com/>