CRAFT Quantum Arbitrage on Cryptocurrency

June 29, 2023

Abstract

Since 1960s, quantum computing has always interested physicists and engineers, which accelerated the development of both theory and hardware. Especially in past decade, several successful prototypes of quantum machines built by industry convinced people that we can be benefit from such amazing technology, not only in long term, even in short term. Finance might be the first industry sector that tried to explore the potential of quantum computing, since it has lots of interesting but computationally hard problems to be solved, many of which require researchers to adopt computational techniques borrowed from mathematics and science. For example, the application of stochastic analysis to financial derivatives pricing, machine learning to fraud detection, and portfolio optimization are all such problems that are computationally expensive in classical computers. Thus, finance industry decided to embrace quantum computing due to its powerful capability in scientific computations.

1 Introduction

To state introduction of quantum computing......

2 Arbitrage on Cryptocurrency

There is no doubt that cryptocurrencies, one kind of digitally or virtually existent currencies, gain massive attention of investors in recent years. Their high volatility and high liquidity both imply that the crypto market can easily deviate from the market equilibrium. As we know, an **arbitrage opportunity** occurs when the market lacks of equilibrium, therefore it would be interesting to find out whether we can detect such arbitrage opportunities quickly and efficiently in crypto market.

2.1 Optimal Arbitrage

In classical currency market, a currency exchange rate is typically given as a pair consisting of a **bid price** and **ask price**. Suppose that you have a base currency (e.g. Australian Dollar) and a quote currency (e.g. US Dollar). If **ask price** is \$0.65 US Dollar, which means in order to buy 1 unit of Australian Dollar, you need to pay \$0.65. If **bid price** is \$0.64, which means if you sell 1 unit of Australian Dollar, you will get paid \$0.64 US Dollar. Let r_{ij} denote the given currency exchange rate for converting currency i to currency j, and we want to find a currency chain $(i \to j \to k \to \cdots \to p \to q \to i)$ such that $r_{ij} r_{jk} \cdots r_{pq} r_{qi} > 1$, and this chain admits a currency arbitrage opportunity. Let's see one example presented in [1],

Table 1 Example For Currency Arbitrage
EXCHANGE RATES ON JUNE 1, 2005

-	**EUR**	**USD**	**GBP**	**CHF**
EUR	1	1.2395	0.6829	1.5443
USD	0.80658	1	0.55109	1.2449
GBP	1.4637	1.8141	1	2.2584
CHF	0.64725	0.80295	0.44250	1

in which the currency chain $(EUR \to CHF \to GBP \to EUR)$ is a typical arbitrage opportunity since the product of their currency exchange rates satisfy the inequality that $r_{14} r_{43} r_{31} = 1.5443 \cdot 0.4425 \cdot 1.4637 \approx 1.00022 > 1$. However, once we define such arbitrage detection problem on a directed graph (E, V), where "edges" & "vertices" represents currency exchange rates and currency kinds respectively, it can be solved by classical Bellman-Ford's algorithm with time complexity $O(|E||V|)$. Therefore, our interest would be the problem of finding **optimal arbitrage**, which means we aim to find the most profitable arbitrage opportunity among currency market, which is a **NP-hard** problem [1].

Soon and Ye [1] formulated an optimal arbitrage detection problem of classical currency in the graph structure as a binary linear programming problem. Inspired by their work, 1QBit published one white-paper [2] discussing the same problem but it reformulated the problem as a quadratic unconstrained binary optimization (QUBO) and solved it using quantum annealer (D-Wave machine).

In our work, we are going to extend their methodologies to the crypto market, whose data source is from API provided by Kaiko. QUBO formulation will continue to be adopted, since QUBO can be solved, not only by Quantum annealing (on D-wave machines), but by QAOA algorithm (on IBM machines). Besides, two candidates representing the classical optimization techniques, the binary linear programming formulation in [1] and AlphaQUBO provided by AWS marketplace, will serve as the benchmarks to be compared with our results in quantum machines.

2.2 Methodology

Let us define QUBO problem for optimal arbitrage detection on cryptocurrency. Suppose that we have m cryptocurrencies and let r_{ij} denote the given currency exchange rate for converting cryptocurrency i to cyptocurrency j, for $i, j = 1, 2, \cdots, m, j \neq i$, which can be understood as a directed graph (E, V) with $(|E|, |V|) = (m(m-1), m)$. Besides, we let binary variables $x_{ij} \in \{0, 1\}$ be the decision variables. Therefore the optimization objective is to maximize the product

$$\max \prod_{j \neq i} r_{ij}^{x_{ij}} \quad \forall\ (i, j) \in E \tag{1}$$

Or equivalently,

$$\max \sum_{(i,j) \in E} c_{ij} x_{ij}, where\ c_{ij} = \log r_{ij}. \tag{2}$$

which is subject to two constraints that

- the flow conservation constraint: $\sum_{j,(i,j) \in E} x_{ij} = \sum_{j,(j,i) \in E} x_{ji}$ for all $i \in V$
- forbidden to pass through a node twice:$\sum_{j,(i,j) \in E} x_{ij} \leq 1$ for all $i \in V$

where the first constraint means the inflow must be equal to the outflow at each vertex and the second constraint does NOT allow a cycle existing in our arbitrage chain. By rewriting two constraints as penalty terms, the above optimization can be reformulated as a QUBO [2], that is,

$$\max \sum_{(i,j) \in E} c_{ij} x_{ij} + \lambda \sum_i \Big(\sum_{j,(i,j) \in E} x_{ij} - \sum_{j,(j,i) \in E} x_{ji} \Big)^2 + \mu \sum_i \Big(\sum_{j,(i,j) \in E} x_{ij} - 1 \Big) \tag{3}$$

2.2.1 Quantum Algorithm I: Forward quantum annealing

One way to solve a QUBO is to implement forward quantum annealing algorithm [3] on D-Wave machine, the principle behind quantum annealing is that we start from an "easy-to-prepare" ground state of initial Hamiltonian $\mathcal{H}_0 = \sum_{(i,j)\in E} \sigma_x^{ij}$, in which σ_x^{ij} is the Pauli X-operator (or Quantum NOT gate), to the unknown low-energy subspace of the states of problem Hamiltonian $\mathcal{H}_F$. In particular, the problem Hamiltonian for the optimal arbitrage detection QUBO would be

$$\mathcal{H}_F = \sum_{(i,j)\in E} c_{ij}\sigma_z^{ij} + \lambda \sum_i \Big(\sum_{j,(i,j)\in E} \sigma_z^{ij} - \sum_{j,(j,i)\in E} \sigma_z^{ji} \Big)^2 + \mu \sum_i \Big(\sum_{j,(i,j)\in E} \sigma_z^{ij} - 1 \Big) \tag{4}$$

where σ_z^{ij} are Pauli Z-operator. Then, the whole procedure for forward quantum annealing procedure can be described as attempting to derive the evolution of the time-dependent Hamiltonian on time interval $[0, \tau]$:

$$\mathcal{H}(t) = A(t)\mathcal{H}_0 + B(t)\mathcal{H}_F \tag{5}$$

To be more specific, once we choose the time horizon parameter τ, the time-dependent coefficients will satisfy that $\lim_{t\to\tau} A(t) = 0$ and $\lim_{t\to\tau} B(t) = 1$. For example, we can choose $A(t) = 1 - \frac{t}{\tau}$ and $B(t) = \frac{t}{\tau}$.

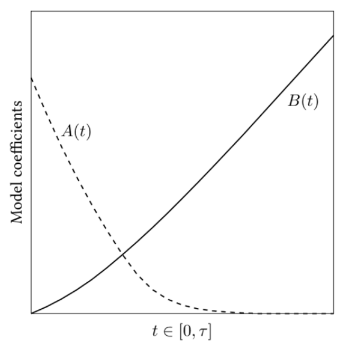

Figure 1: The variation of coefficients $A(t)$ and $B(t)$ graph in [3]

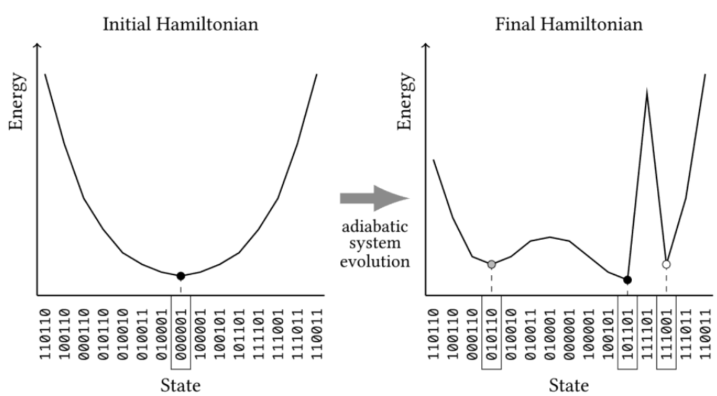

Figure 2: The procedure of quantum annealing algorithm in [3]

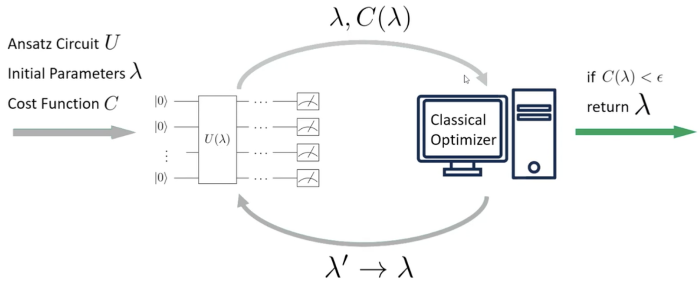

Figure 3: QAOA Quantum Algorithm

2.2.2 Quantum Algorithm II: QAOA

We can state the QAOA algorithm as following procedure:

- Input: The initial Hamiltonian $\mathcal{H}_0$ and the final Hamiltonian $\mathcal{H}_F$.
- Step 1: A parameterised quantum state $|\psi(\beta,\gamma)\rangle$ is created by applying the operators $\mathcal{H}_0$ and $\mathcal{H}_F$ for $K = m(m-1)$ rounds, where

  $$|\psi(\beta,\gamma)\rangle = \prod_{i=1}^{K} e^{-j\beta_i \mathcal{H}_0} e^{-j\gamma_i \mathcal{H}_F} (H^{\otimes n}|0\rangle^{\otimes n})$$

  in which H is the Hardamard Gate.
- Step 2: A computational basis (z-basis) measurement is performed is performed on the obtained state, and this returns a candidate solution for QUBO optimization. Then, we can repeat this "state preparation-measurement" procedure, and the expected value of cost function f (we need to choose this function) over the candidate solution samples is given by $\langle f \rangle = \langle \psi(\beta,\gamma)|\mathcal{H}_F|\psi(\beta,\gamma)\rangle$
- Step 3: The above steps will be repeated for updated parameters β and γ with the classical optimization procedure (e.g. descent-type algorithms) until we reached the minimization of the expected value of the cost function $C(\beta,\gamma)$.
- Output: The best candidate solution to orginal QUBO optimization problem.

Here the initial Hamiltonian $\mathcal{H}_0$ and the final Hamiltonian $\mathcal{H}_F$ are the same as the ones we mentioned in quantum annealing part. If we denote the parameter $\lambda = (\beta,\gamma)$, the procedure of QAOA algorithm can be illuminated by figure 3 above.

3 Experimental Results

The numerical experiment for two quantum algorithm plan to be implemented on D-Wave machine and IBM machine respectively, both of which will be compared with the results generated by classical optimization method: (i) binary linear programming method and (ii) AlphaQUBO algorithm that would run on the regular computers. The optimal solutions and CPU time for each algorithms are presented as below:

Table 1 Optimal Arbitrage Detection of Traversal Algorithm, CPLEX and Quantum Annealing (D-WAVE) for past 30 days (4 Cryptos: ETH, BTC, BCH, ILT)

Date	Traversal	CPU Time	CPLEX	CPU Time	D-WAVE	CPU Time	Return Ratio
2023-03-08	$(0,2,1,0)$	0.0001287s	$(0,0,0,0)$	0.0697956s	No	0.3108678s	0.898529
2023-03-09	$(0,2,1,0)$	0.0001014s	$(0,0,0,0)$	0.0643461s	No	0.2624893s	0.899191
2023-03-10	$(0,2,1,0)$	0.0001159s	$(0,0,0,0)$	0.0408667s	No	0.2261098s	0.899078
2023-03-11	$(1,0,2,1)$	0.0001027s	$(0,0,0,0)$	0.0452669s	No	0.2266676s	0.900312
2023-03-12	$(0,2,1,0)$	0.0001615s	$(0,0,0,0)$	0.0517139s	No	0.2340727s	0.898315
2023-03-13	$(0,2,1,0)$	0.0001063s	$(0,0,0,0)$	0.0421877s	No	0.2760107s	0.898999
2023-03-14	$(0,2,1,0)$	0.0001034s	$(0,0,0,0)$	0.0378162s	No	0.2462074s	0.898134
2023-03-15	$(0,3,1,0)$	0.0001026s	$(0,0,0,0)$	0.1210791s	No	0.2485968s	0.898296
2023-03-16	$(0,2,1,0)$	0.0001125s	$(0,0,0,0)$	0.0835017s	No	0.2300412s	0.898366
2023-03-17	$(0,2,1,0)$	0.0001626s	$(0,0,0,0)$	0.0761472s	No	0.2240705s	0.898858
2023-03-18	$(0,3,1,0)$	9.9621e-5s	$(0,0,0,0)$	0.0736194s	No	0.2583420s	0.898168
2023-03-19	$(0,3,1,0)$	9.9813e-5s	$(0,0,0,0)$	0.0499179s	No	0.2435133s	0.898320
2023-03-20	$(0,2,1,0)$	9.9302e-5s	$(0,0,0,0)$	0.0805395s	No	0.2945682s	0.898615
2023-03-21	$(0,2,1,0)$	0.0001002s	$(0,0,0,0)$	0.0712949s	No	0.2326663s	0.898717
2023-03-22	$(0,3,1,0)$	0.0002852s	$(0,0,0,0)$	0.0977219s	No	0.2570551s	0.905242
2023-03-23	$(0,2,1,0)$	0.0001135s	$(0,0,0,0)$	0.0547286s	No	0.2187551s	0.898658
2023-03-24	$(0,3,1,0)$	0.0001004s	$(0,0,0,0)$	0.0400997s	No	0.2278457s	0.899137
2023-03-25	$(2,1,0,2)$	9.7608e-5s	$(0,0,0,0)$	0.0564897s	No	0.2648345s	0.899430
2023-03-26	$(2,1,0,2)$	9.4794e-5s	$(0,0,0,0)$	0.0704001s	No	0.2463384s	0.898888
2023-03-27	$(0,2,1,0)$	9.3460e-5s	$(0,0,0,0)$	0.0884236s	No	0.2384909s	0.899055
2023-03-28	$(0,2,1,0)$	9.5154e-5s	$(0,0,0,0)$	0.0676328s	No	0.2531631s	0.898476
2023-03-29	$(0,3,1,0)$	9.4774e-5s	$(0,0,0,0)$	0.1428454s	No	0.2271842s	0.898314
2023-03-30	$(0,2,1,0)$	9.9051e-5s	$(0,0,0,0)$	0.1181347s	No	0.2279079s	0.898314
2023-03-31	$(0,2,1,0)$	9.2597e-5s	$(0,0,0,0)$	0.0653288s	No	0.2699792s	0.899179
2023-04-01	$(2,1,0,2)$	9.2115e-5s	$(0,0,0,0)$	0.0730868s	No	0.2296044s	0.898835
2023-04-02	$(0,2,1,0)$	0.0001022s	$(0,0,0,0)$	0.0749110s	No	0.2363743s	0.899031
2023-04-03	$(1,0,2,1)$	9.1404e-5s	$(0,0,0,0)$	0.0665275s	No	0.2685084s	0.899494
2023-04-04	$(0,2,1,0)$	9.2033e-5s	$(0,0,0,0)$	0.0377184s	No	0.3467756s	0.898513
2023-04-05	$(1,0,2,1)$	9.1435e-5s	$(0,0,0,0)$	0.0391821s	No	0.2211354s	0.898421
2023-04-06	$(0,2,1,0)$	9.6175e-5s	$(0,0,0,0)$	0.1717798s	No	0.2577657s	0.898555
2023-04-07	$(0,2,1,0)$	9.2200e-5s	$(0,0,0,0)$	0.0362277s	No	0.2556961s	0.898342

Table 2 Optimal Arbitrage Detection of Traversal Algorithm, CPLEX and Quantum Annealing (D-WAVE) for past 30 days (4 Traditional Currencies: CHF, EUR, GBP, USD)

Date	Traversal	CPU Time	CPLEX	CPU Time	D-WAVE	CPU Time	Return Ratio
2023-03-08	$(0, 1, 3, 0)$	0.0001252s	$(0, 1, 3, 0)$	0.0860032s	Yes	0.1719861s	1.0000532
2023-03-09	$(1, 2, 3, 1)$	9.6498e-5s	$(1, 3, 1)$	0.0541646s	No	0.1922679s	0.9999295
2023-03-10	$(0, 3, 1, 0)$	9.4752e-5s	$(0, 3, 1, 0)$	0.0400733s	Yes	0.1591266s	1.0000064
2023-03-13	$(0, 3, 1, 0)$	9.3219e-5s	$(0, 0, 0, 0)$	0.0410824s	No	0.1900348s	0.9999530
2023-03-14	$(0, 2, 3, 0)$	9.3184e-5s	$(0, 3, 0)$	0.0986906s	No	0.1968059s	0.9999862
2023-03-15	$(0, 3, 1, 0)$	0.0001002	$(2, 3, 2)$	0.1164716s	No	0.2107165s	0.9999752
2023-03-16	$(0, 3, 1, 0)$	9.4137e-5s	$(0, 3, 1, 0)$	0.1088298s	Yes	0.1596797s	1.0001705
2023-03-17	$(0, 3, 1, 0)$	9.6505e-5s	$(0, 3, 0)$	0.0886378s	No	0.2038638s	0.9999762
2023-03-20	$(1, 2, 3, 1)$	9.2511e-5s	$(2, 3, 2)$	0.0858606s	No	0.1906189s	0.9999200
2023-03-21	$(1, 2, 3, 1)$	9.1975e-5s	$(1, 2, 3, 1)$	0.0798848s	Yes	0.2079759s	1.0000683
2023-03-22	$(0, 3, 1, 0)$	9.3123e-5s	$(1, 3, 1)$	0.0522734s	No	0.2453234s	0.9999762
2023-03-23	$(0, 3, 1, 0)$	9.2806e-5s	$(2, 3, 2)$	0.0872442s	No	0.2497159s	0.9999867
2023-03-24	$(0, 3, 1, 0)$	9.2988e-5s	$(0, 3, 1, 0)$	0.0418490s	Yes	0.2138448s	1.0001565
2023-03-27	$(0, 2, 3, 0)$	9.3071e-5s	$(0, 2, 3, 0)$	0.0411740s	Yes	0.1779079s	1.0008026
2023-03-28	$(0, 3, 1, 0)$	9.2709e-5s	$(2, 3, 2)$	0.0711443s	No	0.2235346s	0.9999608
2023-03-29	$(0, 3, 2, 0)$	9.7499e-5s	$(2, 3, 2)$	0.0764389s	No	0.1794511s	0.9999248
2023-03-30	$(0, 3, 1, 0)$	9.6654e-5s	$(0, 3, 1, 0)$	0.1734708s	Yes	0.1551328s	1.0001164
2023-03-31	$(0, 3, 1, 0)$	9.2066e-5s	$(0, 3, 0)$	0.1472245s	No	0.1904517s	0.9999910
2023-04-03	$(0, 3, 1, 0)$	9.5160e-5s	$(0, 3, 0)$	0.0963806s	No	0.1709726s	0.9999022
2023-04-04	$(1, 3, 2, 1)$	9.4697e-5s	$(1, 3, 2, 1)$	0.0876040s	Yes	0.1713712s	1.0002145
2023-04-05	$(0, 3, 2, 0)$	9.3415e-5s	$(0, 3, 2, 0)$	0.0673915s	Yes	0.1621442s	1.0003018
2023-04-06	$(0, 3, 1, 0)$	9.1579e-5s	$(2, 3, 2)$	0.0412772s	No	0.2080176s	0.9999946

Table 2 Optimal Arbitrage Detection of Traversal Algorithm, CPLEX and Quantum Annealing (D-WAVE) for past 10 days (15 Traditional Currencies: USD, EUR, JPY, GBP, AUD, CHF, CNY, SEK, NZD, SGD, HKD, NOK, KRW, INR, BRL)

Date	Traversal	CPU Time	CPLEX	D-WAVE	CPU Time	Return Ratio
2023-01-22	$(0, 8, 5, 9, 0)$	0.04011772s	$(0, 8, 5, 9, 0)$	Yes	971.827233s	1.0155983
2023-01-23	$(0, 12, 2, 8, 0)$	0.03402206s	$(0, 12, 2, 8, 0)$	Yes	945.061185s	1.0110963
2023-01-24	$(0, 10, 1, 5, 0)$	0.03416212s	$(0, 10, 1, 5, 0)$	Yes	982.363128s	1.0001362
2023-01-25	$(2, 0, 4, 3, 2)$	0.03397694s	$(2, 0, 4, 3, 2)$	Yes	988.265913s	1.0491754
2023-01-26	$(0, 14, 2, 6, 0)$	0.03720594s	$(0, 14, 2, 6, 0)$	Yes	958.768311s	1.0235012
2023-01-27	$(0, 14, 2, 3, 0)$	0.04743787s	$(0, 14, 2, 3, 0)$	Yes	954.744443s	1.0374157
2023-01-28	$(0, 2, 10, 8, 0)$	0.03590983s	$(0, 2, 10, 8, 0)$	Yes	987.612633s	9.9916388
2023-01-29	$(0, 2, 12, 6, 0)$	0.03490840s	$(0, 2, 12, 6, 0)$	Yes	948.449277s	183.2894011
2023-01-30	$(0, 8, 4, 11, 0)$	0.03624582s	$(0, 8, 4, 11, 0)$	Yes	967.444419s	1.0789395
2023-01-31	$(10, 2, 13, 0, 10)$	0.03528416s	$(10, 2, 13, 0, 10)$	Yes	966.203324s	1.0009738

References

[1] Wanmei Soon and Heng-Qing Ye. *Currency Arbitrage Detection Using a Binary Integer Programming Model.* International Journal of Mathematical Education in Science and Technology, 42(3):369–376, 2011.

[2] Gili Rosenberg. *Finding Optimal Arbitrage Opportunities Using a Quantum Annealer.* 1QBit Wihte Paper, 1QB Information Technologies, 2016

[3] Antoine Jacquier and Oleksiy Kondratyev *Quantum machine Learning and Optimisation in Finance* Packet Publishing Led, October 2022

[4] Dylan A. Herman, Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev *A Survey of Quantum Computing for Finance* arXiv preprint arXiv:2201.02773.