Proceedings of 10th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2021 Month: October Volume: 10

Budget Pacing Optimization for Effective Online Ad Campaign using Quantum Accelerator Models

Pradip Bhandari [a], Subarna Shakya [b], Nanda Bikram Adhikari [c]

[a, b, c] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, Tribhuvan University, Nepal*
Corresponding Email: [a] 075mscsk013.pradip@pcampus.edu.np, [c] adhikari@ioe.edu.np

Abstract

The problem with the advertiser is not able to properly utilize or spend the allocated budget in the allocated hours. Current industry standards for probabilistic throttling and bid modification are designed to execute in the Central Processing Unit (CPU) for better results. Optimization and data parallelism is not performed efficiently in the CPU so this research needs some better solution. Probabilistic throttling and bid modification can be implemented in Quantum Processing Unit (QPU) but it does not optimize the budget timing as this is not an optimization problem. This paper shows modified edge-weighted bipartite matching formulated as Quadratic Unconstrained Binary Optimization (QUBO) problem and implemented in QPU units as an optimization problem with large data set in near real-time. This is both processing and data parallelism hungry so it is more suitable for QPU when compared with CPU and Graphics Processing Unit (GPU). Quantum accelerators are processing units specially designed using quantum phenomena like superposition, entanglement, annealing. This research has explored the untouched opportunities in the quantum computing field. This research has been able to show the increase in results significantly by the use of specific approaches suitable for QPU units. Implementing QPU has increased budget timing synchronization from 11th hour to 21st hour which is 41% increase. The processing time is about 8.1% better in QPU as compared to CPU and 21.3% better as compared to GPU and this also leads to a decrease in latency by 2.48% as compared to CPU and 24.41% as compared to GPU.

Keywords

Quantum Accelerators, Optimization, Budget pacing, Online advertisement

1. Introduction

Online advertising appears on websites. It is a multibillion-dollar sector that is rapidly expanding. It typically refers to marketing and advertising that makes use of internet traffic to promote products. It is less expensive than other traditional media such as television, radio, and newspapers. It also covers a wide spectrum of geographical and demographics.

Demand Side Platforms (DSP) enable advertisers to collect impressions from different publishers, which are generally targeted to specific users based on user behaviour, actions, location, demography, or previous online activity. Supply Side platform (SSP) evaluates advertisers, sets bidding parameters and places the ad content. It connects multiple ad networks, ad exchange and DSPs to sell inventory. Publisher own websites or have the right to place and rotate ads on them so that visitors could see advertisers' offers [1].

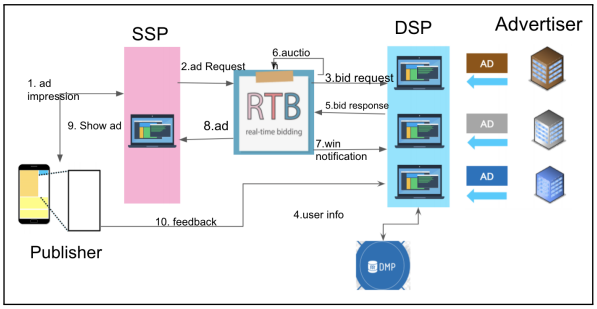

Figure 1: System Diagram showing Real Time Bidding

Pricing models like Cost Per Mille (CPM), Cost Per Click (CPC), and Cost Per Action (CPA) are used to charge the commission from the advertisers to the benefit of publishers. Advertiser pay for showing their offers to the users that are more likely to buy their products.

Real Time Bidding (RTB) is an automated auction where ad impressions are sold and bought and transactions take place [2] as shown in figure 1. Budget pacing helps to optimize advertisement campaigns to control the rate at which campaigns spend. It enables advertisers to reach a wider range of audience and also prevents premature campaigns.

Advertisers can buy ad inventory in bulk rather than one impression at a time through ad networks. Ad-exchanges, on the other hand, are vast pools of impressions and open markets where publishers provide their inventory and media buyers directly bid on ad impressions. Budget pacing aids in the optimization of advertising programs by controlling the rate at which they spend. It allows advertisers to reach a broader audience while also avoiding early marketing.

A quantum computer manipulates quantum states of matter and employs quantum effects like superposition and entanglement to speed up specific processes. Classical computers will be unable to imitate the behavior of programmable quantum computers as quantum computing progresses. Quantum computing relies on atomic state called "qubits" which denotes both zero and one simultaneously at the same time. Quantum computing is in memory computation devices. Quantum accelerators are co-processor link to the big architecture that performs specific kernels tasks. Usually there are two types of quantum accelerators as supplementary co-processors. The initial one uses quantum gates and another one uses quantum annealing. The traditional host processor preserves the overall control over the total system and assigns the execution of certain tasks to the handy accelerators. In the world of quantum, there are two principle disputes. The first one points about the enough and good qubits for any experimental quantum chip. The present players in the quantum field are working with different quantum technologies like ion traps, majorana's, NV-centers, semi-conducting and superconducting qubits and even Graphene. These edge technologies are trying to survive the status of the qubits that bear from decoherence and that brings in errors when executing some kind of quantum gate tasks [3] .

A quantum processing unit (QPU), also referred to as a quantum chip, is a physical (fabricated) chip that contains a number of interconnected qubits. It is the foundational component of a full quantum computer, which includes the housing environment for the QPU, the control electronics, and many other components.

A qubit uses the quantum mechanical phenomena of superposition to achieve a linear combination of two states. Superposition, simultaneously off and on, allows quantum algorithms to process information in a fraction of the time that it would take even the fastest classical systems to solve certain problems. Entanglement allows two or more qubits in a single state. Changing the state of an entangled qubit will change the state of the paired qubit immediately. Therefore, entanglement improves the processing speed of quantum computers. Quantum annealing starts from a quantum-mechanical superposition of all possible states with equal weights. Quantum annealers are a type of adiabatic quantum computer that provides a hardware implementation for finding the minimum energy configuration of Hamiltonians whose ground states represent optimum solutions of the original problems of interest. It is less affected by noise than gate model quantum computing.

Hence, the purpose of this research are:

1. To optimize the budget pacing using different ad campaign metrics.
2. To develop an approach to process budget pacing problems using quantum accelerators.

2. Literature Review

In 2009, the term "real-time bidding" was coined. Advertisers used to buy a large number of impressions for the same per-unit price even when their worth differed. It is currently a billion-dollar industry, with a value of $304 billion in 2019 and a projected value of $980 billion by 2025. It has increased by double-digit percentages in recent years.

2.1 Research Gap

There have been numerous implementations that have used greedy algorithms. However, no other new budget pacing approaches have been implemented. Edge-Weighted Online Bipartite Matching is a popular algorithm that may be used to solve an optimization problem with a budget pacing change. Both data and processor parallelism are required for optimization issues. QPUs are more suited to optimization challenges. Budget pacing based on optimization problems is not implemented. There is

also a research gap in the implementation of edge-weighted online bipartite matching utilizing Quadratic Unconstrained Binary Optimization (QUBO)[4].

2.2 Budget pacing

The optimization related to online advertising campaign are tunned via Clicked Through Rate (CTR) and action rate or bid landscape. But it lack the smooth delivery constraint. This problem somehow deals with adaptive targeting for RTB [5]. Many linear programming model are already in practices like keyword matching, online bidding etc. Deepak Agarwa from linkedin has also shown an evenly distributed advertising budget where their proposed algorithm helps to improve advertiser experience and provides better revenue to LinkedIn. N.B, M.F [6] has also done case study in frequency capping in Online Advertising. This all leads to maximize revenue in online ad auctions and it has gained lots of attention[7]. But this doesn't smooth out the budget pacing for allocated time period.

2.3 Quantum Accelerator

The basic idea of computing devices based on quantum phenomena was first thought in the 1970's and early 1980's by physicists and computer scientists such as Charles H. Bennet of the IBM Thomas J. Watson Research Centre, Paul A. December, 2005, first quantum byte is set forth to have been created by scientists at the Institute for Quantum Optics and Quantum Information and the University of Innsbruck in Austria. From 2007-2012 many groundbreaking algorithms and systems were developed and tested like rise of D-Wave, Q-bits stored, 28 q-bit annealing system. D-Wave Systems Inc. on 2017 announced general commercial availability of its D-Wave 2000Q quantum computer which is based on quantum annealer, which it claims has 2000 qubits. 2019 oct, google claim the quantum supremacy. In the same month IBM 53 qubits system goes online [8].

K. Bertels, I. Ashraf, QCA laboratory, Delft University of Technology, Netherlands Quantum Force, Netherlands published a paper called Towards Full-Stack Quantum Accelerators and near future promising approaches are presented.The first initiative from the quantum accelerator community which involves the full stack integration of the different layers that are needed to build the quantum accelerator. They also proposed a system for Quantum Accelerators to integrate with current systems. They also suggest it will be very hard to predict what the performance based improvement will be of any quantum computing device, but that it will be much higher than any current existing computational technology [9]. However, whether it will be 5, 10, 50, 100, or even more times faster will depend on the complexity of the quantum application and how the qubits will be generated. Before the full-integration effects become fully obvious and verified, more research will be required for at least 10 to 15 years. According to those articles, the focus of this research is on integrating quantum accelerators into cloud-based systems that are easily available to all industry-grade systems[10].

3. Methodology

This research design consists of different steps as in figure 2. Greedy based algorithms is implemented in CPU and QUBO based algorithm is implemented in QPU because of their nature.

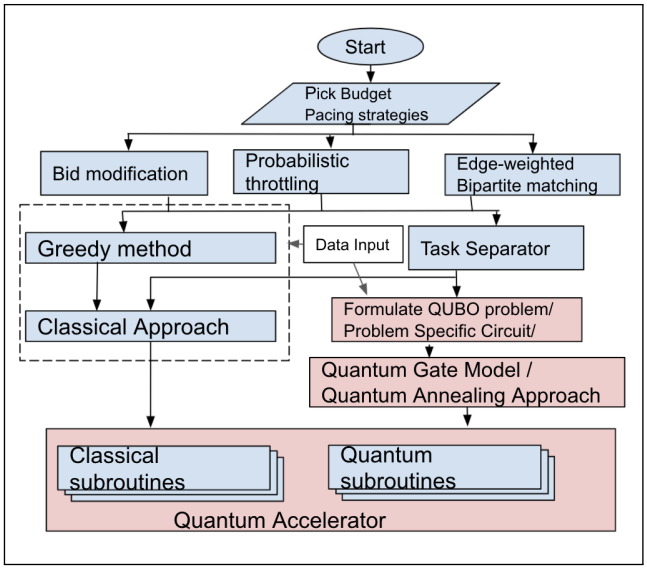

Figure 2: Budget pacing implementation architecture

Bid modification and probabilistic throttling are more processor intensive, but edge-weighted bipartite matching in optimization problems is both CPU and data parallelism intensive, making it particularly suitable to QPU. The CPU excels at processing parallelism, while the GPU excels at data parallelism. First this research makes a budget pacing strategy using different techniques like greedy or Edge weighted bipartite as in figure 2. The problem will be

analysed and divided for solving using CMOS and QPU. This is done by task separator. After that the problem will be executed using different languages(Q sharp and Python) and compiled and will run in respective hardware. For QPU, the problem will be converted to into QUBO problem, which can be further computed using Annealing based QPU. Here this research utilizes 4 qubits in d-wave 2000 system.

3.1 Budget pacing methods

There are different budget pacing methods like even pacing, traffic based pacing. These are the simple unscientific methods. Besides this more scientific approach are a) Probabilistic Throttling and b) Bid modification. These methods are widely used by DSP's for proper budget Pacing. Both methods are similar only the placement of modules varies. A new Approach is to use an edge-weighted bipartite matching for DSPs.

Edge Weighted bipartite matching algorithm.

Algorithm 1 Edge-weighted bipartite matching algorithm

- **Input:** *Weight vectors, vertices, restriction metrics*
- *When online vertex $j \in R$ arrives:*
- *If random gain $<$ 0 and deterministic gain $<$ 0: Leave j unmatched*
- *If random gain deterministic gain : Assign j randomized*
- *If deterministic gain $>$ random gain : Greedily assign j*

This research work has mainly tried to contribute by manipulating input parameters like CPM, bid_request, vertices size in edge weighted bipartite matching in LHS users and RHS Advertisement element and formulating the same algorithm as a QUBO problem. And implementing it in the Amazon Web Services (AWS) quantum braket. This is new work and has lots of promising applications in other optimization problem fields also. This research work will lead to extract the performance matrix of QPU in above field of area. This algorithm has been modified and used as QUBO problem as in equation 1.

3.2 QUBO Problem formulation

Quadratic Unconstrained Binary Optimization is a popular pattern matching technique which is common in ML applications. QUBO is a special kind of problem to minimize a quadratic polynomial over binary variables. QUBO can be considered as a mathematical formulation for expressing and resolving a type of different optimization tasks which is combinational in nature[4]. Thus, formulate our edge-weighted bipartite matching for budget pacing to the QUBO problem.

The objective function is formulated as

$$Arg\ max\{E[\ \omega^T]\ x - \alpha x^T W x - \beta(Px-1)^2\} \quad (1)$$

here,

$E[\ \omega^T]\ x$: *maximize CTR*

$x^T W x$: *Low variation*

$\beta(Px-1)^2$: *Constraint*

$x \in \{0,1\}^{N_a x N_c}$: *Decision variable*

$\omega : \Omega \rightarrow R^{N_a x N_c}$: *Weight vector (CTR, CPC, CPM, CVR) for each edge*

$W \in R^{N_a x N_c}$: *Covariance matrix of w*

$N_a x N_c$: *Size of each vertices*

P : *Restriction matrix*

$\alpha, \beta \in R$: *Control parameters*

3.3 Classical Approach

Greedy methods are used in classical approaches for probabilistic throttling and bid modification. Greedy algorithms build a solution strategy that makes the best optimal choice at each small step with the goal of this eventually leading to a globally optimum solution. It produces an approximate which is within a reasonable limit.

Probabilistic Throttling: It is defined for each advertiser.

For each advertiser A, It is define as:

B_A: left-over budget for a time period

T_A: left-over maximum spend for rest of the time period

This paper define algorithm as:

Algorithm 2 Probabilistic Throttling algorithm

- *For each arriving request r:*
- *For each budget constrained advertiser A:*
- *Calculate probability,* $P[\,Heads] = B_A/T_A$.
- *If heads, participates in the auction*

3.4 Quantum accelerator and Budget pacing

Once QUBO is formulate, it is much faster to execute in a quantum accelerator than in a CMOS based system. Moreover optimization problems are more suited to Annealing based computers. QUBO binary objectives can be represented as graphs and these can be mapped to QPU. Also live data can be easily incorporated into coefficients in QUBO.

3.5 Greedy and Edge-weighted bipartite matching for CPU and QPU

For optimizing budget pacing the algorithm takens are different for CPU and QPU it is because, the standard QBUO solver developed by IBM takes time in range of seconds to solve simple matrix problem where as same problem can be solved within the range of milliseconds in QPU. So the comparison between same QUBO problem in CPU and GPU are not justifiable therefore greedy and edge-weighted bipartite matching are two best algorithm on respective processing unit that can be comparable and have significant contribution.

3.6 Data Collection and Analysis

Advertising industry is going very rapidly. Monetization digital platform creates RTB. RTB generates data per each auction. RTB consists of auction_init, bid_request, bid_response, bid_fail, bid_won, set_targeting etc data. Each section consists of DSP, SSP attributes. Data is collected from a website that has 1000+ realtime users and generates around 7,70,000 rows ads data per day.

Data size for three different tables are not equal this is due to, for one single request, 9 bidder bids and sends the response. Out of total bid response all will not be able to serve the ads.

The auction_init contains status, bidder request data. This data will be further used in bid request, bid response and win bid to analysis the auction process.

Table 1: Data size for bid request, bid response and bid won

Type	Row Count
bid_request	27,19,852
bid_response	72,38,632
bid_won	55,04,406

Bid request will contain bids information for each bidder. Bid response will contain the response send by DSP for bid request. It will have the information like CPM, ttl, time-to-response etc. This research will utilize this data to get the performance matrix like CTR, CPC, CPM etc. Based on this matrix the edge weight will be calculated.

3.7 Implementation

Some of the computing intensive problems can be solved in heterogeneous models consisting of CPU, GPU and QPU working parallel. The optimal task per each processing unit should be defined based on the availability of processing by each unit. This can be done based on the Task separator, which will be able to separate tasks for each processing unit. For a budget packing algorithm, This research will formulate it as a QUBO problem and it will be executed in QPU. Also classical approaches will be executed in CMOS based units. First it generate the bid request, this is generated randomly using geoip maxmind. Geoip maxmind simulates the ip address of different locations. Currently there are 3 requests every 15 seconds. It is because the minimum ad refresh supported by any demand partner is 15 seconds. This request goes to respective processing units CPU, GPU and QPU. CPU and GPU runs greedy algorithms whereas QPU runs edge-weighted online bipartite matching algorithms. High performance EC2 instances are used for CPU and GPU whereas the D-Wave 2000Q system from AWS quantum bracket is used for QPU processing. The output is sent to a data logger instance via webhook(api) hosted in ec2. Different parameters like processing time, latency, throughput, CPM, qubo output etc are measured in log in data logger. Further the output is visualized in graphs and charts.

3.8 Evaluation Metrics

For proper evaluation the research work will be compared with classical approaches.

1. Time: Should run as fast as possible

2. Qubit Footprint: Should as few qubits as possible

3. Accuracy: Should get the objective function value as close to the global maxima(CTR) as possible

4. RESULT AND ANALYSIS

The Histogram analysis for input data and output results are as shown in figure 3 and figure 4

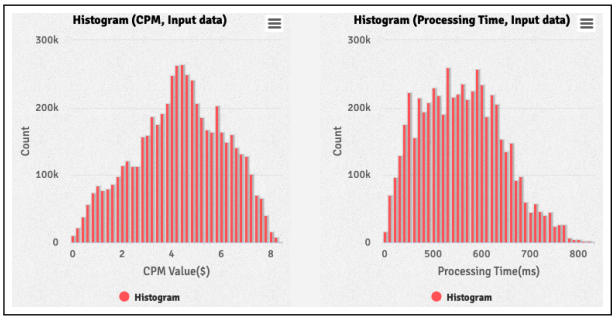

Figure 3: Input data histogram analysis

It tells that input data has symmetrical CPM and somewhat skewed right processing time.

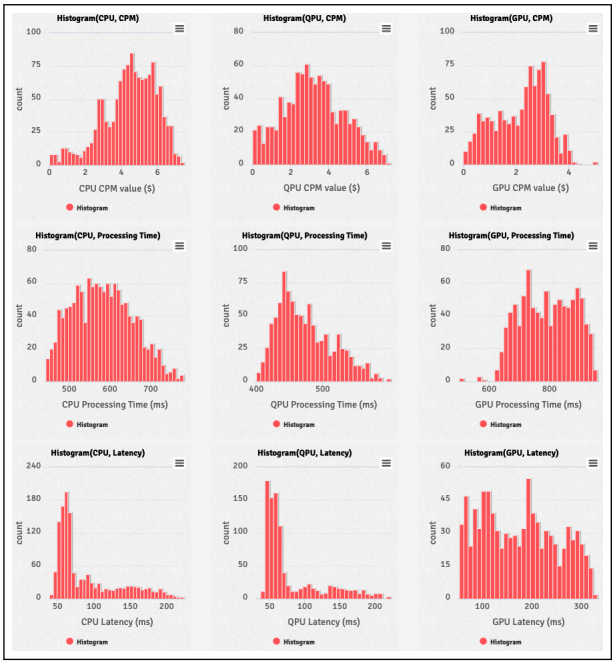

Figure 4: Output data histogram analysis

The output histogram shows that the cmp, processing time and latency are better in QPU.

The results show that the budget is well spent using edge-weighted bipartite matching. The budget was set to $20 dollars for 24 hours and using QPU it was fully utilized in the 21th hour whereas using CPU it was finished in the 10th hour and only $14.34 was spent using GPU.

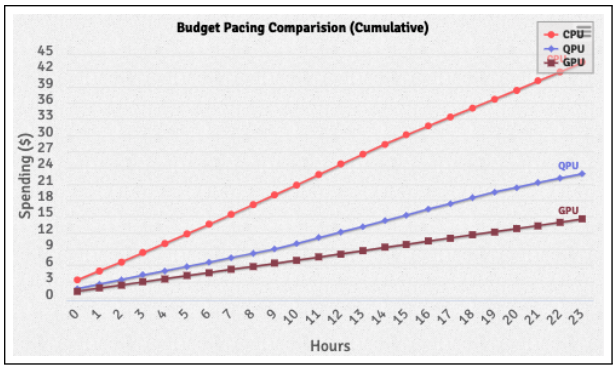

Figure 5: Cumulative budget pacing for 24 hour time period comparison

It is also shown in table 2. It allows QPU to reach wider users and avoid premature campaigns.

Table 2: Hourly budget distribution (Budget pacing comparison)

Hours	CPU($)	QPU($)	GPU($)
0	1.67	0.82	0.57
1	3.37	1.63	1.17
2	5.07	2.43	1.74
3	6.76	3.22	2.34
4	8.47	4.06	2.9
5	10.22	4.88	3.46
6	11.99	5.74	4.03
7	13.74	6.55	4.61
8	15.56	7.43	5.18
9	17.43	8.26	5.75
10	19.27	9.3	6.33
11	21.13	10.37	6.9
12	22.98	11.43	7.47
13	24.89	12.52	8.05
14	26.75	13.52	8.64
15	28.53	14.54	9.2
16	30.22	15.59	9.77
17	31.92	16.64	10.36
18	33.62	17.67	10.94
19	35.31	18.72	11.52
20	36.96	19.56	12.12
21	38.62	20.44	12.68
22	40.25	21.33	13.26
23	41.94	22.19	13.85

Processing time seems to be better in QPU as compared to CPU and GPU units. QPU processing time ranges from 679 ms to 401 ms whereas CPU processing time ranges from 782 ms to 439 ms and GPU ranges from 1087 ms to 485 ms. The processing

time in QPU is low as compared to CPU and QPU as this paper have implemented heterogeneous architectures so the input variables are processed by CPU and GPU and core QUBO processing was done in QPU.

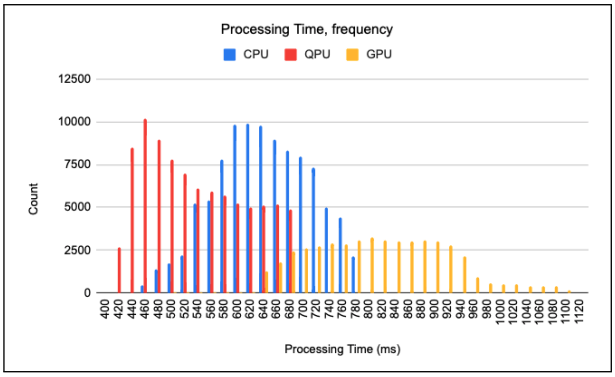

Figure 6: Histogram of CPU, GPU and QPU outputs against processing time with defined request intervals

Latency is the measure of time before core processing, it includes time after request received in server and before it feed to processing core algorithm. For QPU it includes network latency as it is heterogeneous whereas for cpu and gpu it is in a single instance so it doesn't include network latency. Since there is less work for QPU before feeding data to QUBO it seems to have lower mean latency. Overall latency for CPU and QPU are very near to each other.

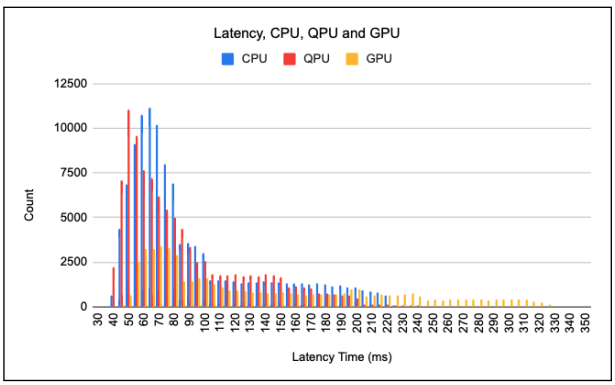

Figure 7: Histogram of CPU, GPU and QPU outputs against latency with defined request intervals

5. Conclusion

This research has developed a new way to optimize the budget and timing synchronization for budget pacing for display ads using QPU. From above it can be seen that using QPU the budget timing synchronous has increased to around 41% which is from 11th hour to 21st hour. And this indicated the processing time is about 8.1% better in QPU as compared to CPU and around 21.3% better than GPU. Similarly the latency seems to be 2.48% better in QPU as compared to CPU and around 24.41% better as compared to GPU.

This research work has mainly tried to contribute by manipulating CPM, bid_request, vertices size input parameters in edge weighted bipartite matching in LHS users and RHS Advertisement element and formulating the same algorithm as a QUBO problem. And implementing it in the AWS quantum braket. This is new work and has lots of promising applications in other optimization problem fields also. This research work will lead to extract the performance matrix of QPU in above field of area.

6. Limitation and Future Enhancement

QPU is limited to what different cloud provider provides. QPU has significant error. All cloud provider provides a way to repeat the same task and use the average result, which can be a performance barrier. Many real hardware provider like IBM, D-wave are gradually removing the abstract layer and providing more control to their hardware. So as more hardware are available, above approach can be implemented in a way so as to reduce the repeat task.

References

[1] Andy Letting. Five Steps to Understanding Programmatic. *Institute of Data & Marketing*, page 1, 2018.

[2] Francisco LupiáñezVillanueva Lucie Lechardoy, Alena Sokolyanskaya. Collection selection for managed distributed document databases. *Support to the Observatory for the Online Platform Economy*, pages 13–49, 2020.

[3] Thomas Hubregtsen, Christoph Segler, Josef Pichlmeier, Aritra Sarkar, Thomas Gabor, and Koen Bertels. *Integration and Evaluation of Quantum Accelerators for Data-Driven User Functions. arXiv:1912.06032v2. 2020.* 2020 21st International Symposium on Quality Electronic Design, 2021.

[4] M. Z. Alom, B. Van Essen, A. T. Moody, D. P. Widemann, and T. M. Taha. *Quadratic Unconstrained Binary Optimization (QUBO) on neuromorphic computing systemer*, 2018.

[5] M.Feldman N.Buch. Frequency capping in online advertising. 2014.

[6] Siyu Y. Deepak A., S.G kai. W. Budget pacing for targeted online advertisements at linkedin.

[7] Mengjuan Liu; Wei Yue; Lizhou Qiu; Jiaxing Li. An effective budget management framework for real-time bidding in online advertising. 2020.

[8] Frank Arute, Kunal Arya, Hartmut Neven, and John M. Quantum supremacy using a programmable superconducting processor. 2019.

[9] K. Bertels, A. Sarkar, A.A. Mouedenne, T. Hubregtsen, A. Yadav, A. Krol, and I. Ashraf. Quantum computer architecture: Towards full-stack quantum accelerators. 2019.

[10] https: //quantumexperience.ng.bluemix.net/qx/editor. Bm. quantum computer composer. 2018.