Convolutional Decoders Based on Quantum Annealing

Ilyas Rysbekov, B.Eng

Submitted in fulfilment of the requirements for the degree of

Master of Science in Electrical and Computer Engineering

School of Engineering and Digital Sciences

Department of Electrical and Computer Engineering

53 Kabanbay Batyr Avenue,

Nur-Sultan, Kazakhstan, 010000

Supervisor: Refik Kizilirmak, PhD

Co-supervisor: Farzad Salmasi, PhD

April, 2021

Abstract

This research proposes a new method of decoding convolutional codes using quantum computers. The proposed method obtains maximum likelihood (ML) estimate of the transmitted codeword using quantum annealing (QA). The performance of the proposed method is assessed by its error performance and compared with the conventional Viterbi decoder on classical computers. The results verify the feasibility of QA for decoding convolutional codes. Furthermore, the execution time of both classical and quantum computers for decoding are compared and discussed.

Acknowledgements

Firstly, I would like to thank my supervisor professor Refik Caglar Kizilirmak, for all those hours spent explaining me every detail of our master thesis. I would never have accomplished it without him. Secondly, I would like to thank my friends and family and especially my roommate Rinat Khalimov for their support during the whole project.

Contents

List of Abbreviations and Symbols

QA - Quantum Annealing

QC - Quantum Computing

SQUID - Superconducting QUantum Interference Device

C-RAN - Cloud Radio Access Network

BS - Base Station

BER - Bit Error Rate

QUBO - Quadratic Unconstraint Binary Optimization

SNR - Signal to Noise Ratio

SA - Simulated Annealing

QPU - Quantum Processing Unit

API - Application Programming Interface

CPU - Central Processing Unit

List of Figures

1 Introduction

Our generation is on the verge of quantum computations transforming many fields of science. The quantum computing is not only about changing the physical components of a computer, but changing the whole definition of computation itself [1]. Superconducting QUantum Interference Device (SQUID) is the main building block inside quantum computers. Term interference express electrons which has a wave behavior inside quantum waves [2]. Despite its rapid growth, quantum computers still have restriction on physical layout, maximum number of connections which could be physically merged equals to 6 per spin [3].

Convolutional codes are type of error-correcting codes which enable reliable data transmission over communication channels [4]. They are useful for correcting errors in transmission links and they are usually implemented to deal with errors provided by attackers or a format conversion [5]. Furthermore, due to noise appearance in wired or wireless channels errors could be transferred to a receiver [4]. Convolutional codes are such type of codes where encoders have memory, it means the output bits are determined by logic operations on bits in a stream and a small number of previous bits. For decoding convolutional codes, the most popular algorithm in maximum-likelihood sense was developed by Viterbi [4].

In this work, we implement decoding of convolutional codes on a quantum computer as an alternative and powerful approach to existing techniques that are extensively based on classical computers. Recently, there is increasing interest in quantum computing and its applications to wireless communication systems. In [6] authors implement multiple input multiple output (MIMO) signal detection using

quantum computers. In [7], non-orthogonal multipile access (NOMA) signal processing algorithms are demonstrated using quantum computer. In [8], decoding of LDPC codes were decoded on a quantum computer.

1.1 Research objectives

The main idea of this master thesis is to implement a convolutional decoder on quantum computers that are based on quantum annealing. Implementation of quantum annealing is performed on D-Wave 2000Q. We further compare the performance of the proposed decoder on quantum computer with traditional Viterbi decoders on classical computers in terms of their error performances and execution times.

Another aim of this work to contribute to the interdisciplinary field of computational methods for digital communication systems, especially using quantum computers. Nowadays, classical computers still outperform its quantum counterparts in terms of execution time. However, with rapid developments in the field, we can expect robust efficiency increase in quantum computing in a few years.

Potential usage of quantum computers in wireless communication systems is in C-RANs (Cloud Radio Access Networks). C-RAN is a cellular network architecture where each small area is covered by base stations (BS) of which computational tasks are outsourced to a central station. BSs of C-RANs are simple RF transceivers that consume less power than traditional BSs and one central station performs all the necessary baseband and network layer processing in a cost effective way [9]. For high data rate applications, due to the need of high computational power, quantum

computers can play an important role in C-RANs.

1.2 Thesis structure

This thesis is structured as follows: Chapter 2 introduces error correcting codes and convolutional encoders. Chapter 3 contains preliminaries on quantum annealing and convolutional decoders, traditional decoder and Viterbi algorithm. Chapter 4 represents convolutional decoders using quantum annealing. Chapter 5 presents the performance comparision between Viterbi decoder and QA, in terms of bit-error-rate (BER) and executation time. In Chapter 6, we conclude the thesis with discussion and future works. In Appendix A, contains result after QA.

2 Error Correcting Codes

In this Chapter, we review main concepts in error correcting codes, their working principles, convolutional encoders and Viterbi decoder.

2.1 Convolutional encoders

Convolutional coding is frequently used as coding method based on the output code bits determined by logic operations on the current and previous bits in the stream, instead of using block of bits. Each new bit enters the register and previous bits shifts to the right and oldest bits are removed from register when it becomes full (see Fig. 2.1) [10].

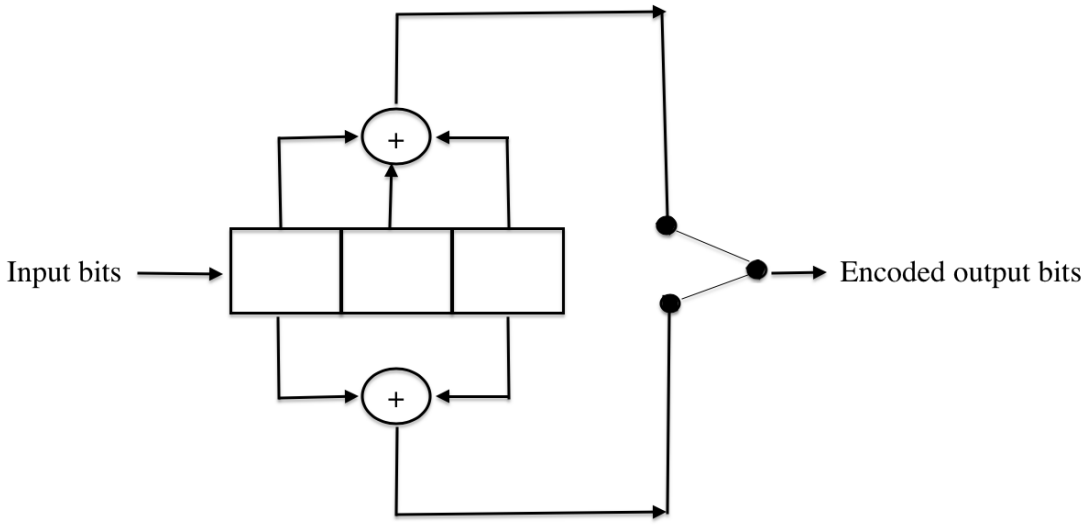

Figure 2.1: Example of convolutional encoder

Various methods for representing convolutional encoders exist, e.g., graphical, tree diagram and trellis diagram. In majority cases trellis diagram is used [11]. Trellis diagram provides better encoder description by its repetitive working prin-

ciple [13]. Trellis's nodes are characterized as encoder states (see Fig. 2, yellow lines indicate discarded paths in the forward and backward recursions [12].). All the states of trellis diagram is represented on a vertical axis and each transition from one state to another denoted on the diagram [13]. Number of transitions from each state equals to k, where k stands for the size of encoder's memory and $k + 1$ is called constraint length of the encoder [13]. Convolutional encoders could be described as set of n generator polynomials, where one node for each of the n mod2 adders [13].

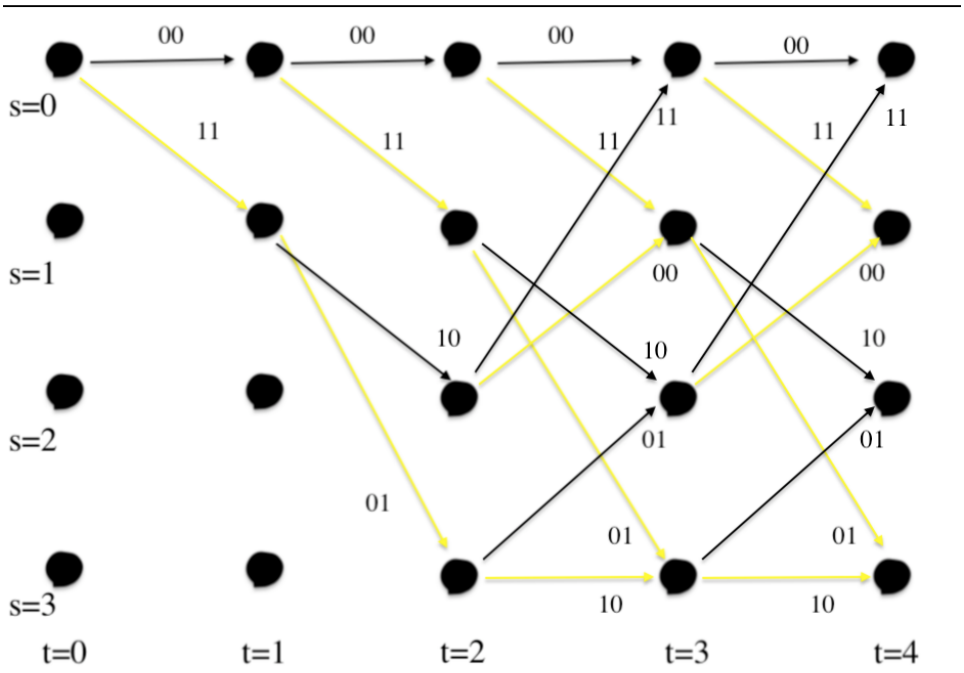

Figure 2.2: Example of trellis diagram

2.2 Viterbi decoder

There are many convolutional decoder algorithms, but Viterbi algorithm is the optimal and traditionally convolutional decoders performed on Viterbi algo-

rithm [14]. In the encoder, bits interpreted as input to a shift register of length k. The complexity of the Viterbi algorithm based on the constraint length of the encoder [15]. To reduce this complexity, we propose a convolutional decoder based on quantum annealing.

Main drawback of the Viterbi's algorithm is that for decoding single binary data, decoder performs O(2^k) iterations. It searches the most likely sequences the message could match and uses this information to decide what was the initial message [16]. Viterbi algorithm estimates a message as a sequence instead of estimating each message as an individual sample from the signal. Moreover, Viterbi algorithm provides a level of correlation between each sample from every signal [17].

3 Preliminaries on Quantum Annealing

In this chapter we give the preliminaries of quantum annealing, reason why our task should be reformulated as a QUBO problem, quantum tunneling and quantum annealing's constraints.

3.1 Quantum Annealing

Quantum annealing (QA) is a method to find the global minimum of a given objective function over a given set of candidate solutions (candidate states), by a process using quantum fluctuations. QA could be called as an extension of simulated annealing (SA). SA is a general solution of meta-heuristic method which was applied to solve combinatorial optimization problems [18]. Initial state of annealing process has more likelihood to move to anywhere in space rather than later moves. Each transition depends on random value and moving further to another value (also random) [18]. All these operations depend on the temperature parameter.

QA has similar algorithm as SA, however, SA uses thermal fluctuations, while QA uses quantum fluctuations to solve the similar task [19]. QA calculates energy changing at a point in space for really short time lapses for the reason that it uses uncertainty principle [20]. QA is based on extension of classical annealing and this approach is robust to avoid local minimum [21]. This effect is obtained by "tunnel effect" and it allows us to take advantage of wave-matter quality [21].

Tunnel effect helps to find global minimum going through barriers directly to the global minimum (Fig. 3.1). [22].

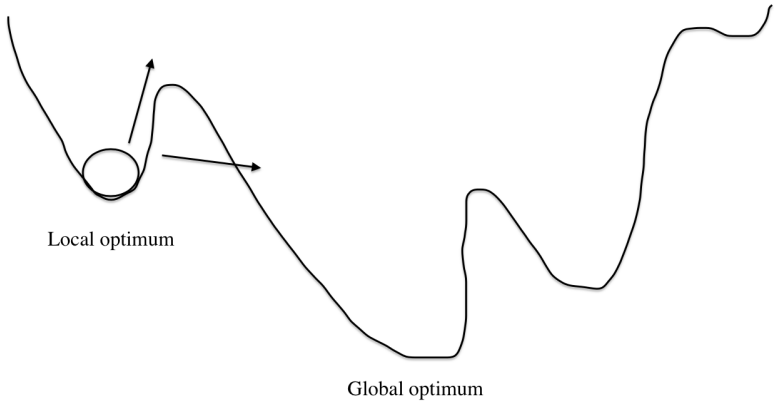

Figure 3.1: Quantum annealing searches global minimum

In general QA is structured as sum of potential and quantum kinetic energies [22]. It maps the optimization problem into a quantum system and the optimized objective functions could be placed into a potential field [22]. That potential field introduces us controllable field of quantum fluctuations [22]. The evolution of quantum machines could be described by time-dependent Schrodinger equations as [22]

$$ih\frac{d\Psi}{dt} = H(t)|\Psi(t) > \tag{1}$$

$$H(t) = H_{pot} + H_{kin}(t) \tag{2}$$

where H_{pot} represents potential energy and H_{kin}(t) represents time-dependent kinetic energy. However, larger energy levels correspond to larger quantum fluctua-

tions [22].

When QA process is finished, qubits which represent the solution are expected to be at the lowest energy state of the solution [23]. Fig 3.2 illustrates a case with 2 qubit, the lowest energy sequence '10' becomes the solution.

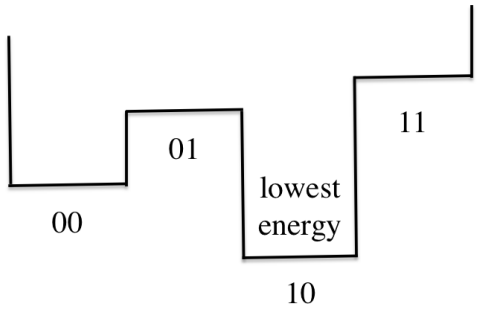

Figure 3.2: Lowest energy state gives the best solution

3.2 Quantum Annealer's limitations

One of the challenges in future wireless communication systems is increased computationally demanding tasks that base stations should perform. In this work, one particular quantum computation area, quantum annealing, will be considered to address one of the most commonly performed tasks in digital communication systems that is decoding convolutional codes. QA machines do not implement quantum gates as in most other quantum computers.

As compared to its quantum gate based counterparts, there are a few advantages of quantum computers that are based on QA. First of all, it does not require profound knowledge in quantum physics in order to be programmed. Second, there are readily available QA machines that are open to public use by D-WAVE through

its own cloud platform Leap or through Amazon AWS. One challenge today with the quantum computers is that providing real time data input from classical computers to quantum computers and performing computations creates several delays [7]. Another constraint is limited number of qubits in nowadays annealers.

4 Convolutional Decoders using Quantum Annealing

In order to decode convolutional codes, we propose finding the maximum likelihood (ML) estimate of the transmitted codeword using QA. Quantum annealer D-WAVE 2000Q that is available on Amazon AWS with 2048 qubits was used.

The Quantum Processing Unit (QPU) can be expressed as a set of functions which minimize Ising functions having only physical version of quantum annealing [1]. For using D-WAVE 2000Q quantum machine, the objective function should be reformulated as a quadratic unconstraint binary optimization (QUBO) problem or an equivalent Ising function [18] as

$$\hat{q_1}, \hat{q_2}, ...\hat{q_N} =_{q_1,q_2,...q_K} \sum_{k\leq\ell}^{N} Q_{k\ell} q_k q_\ell \tag{3}$$

where N stands for number of qubits, Q_{kl} and Q_{ll} stand for quadratic linear coefficients.

Decoding convolutional codes with ML with additive white Gaussian noise can be expressed as

$$C^* = \underset{C}{argmin} ||r - C||^2 \tag{4}$$

where C^* stands for the estimated codeword from the received bit stream with length N, C is transmitted codeword $[c_1, c_2, ..., c_N]$ where each entry c_n is either +1 or -1, r is received bits that are corrupted by additive white Gaussian noise.

QUBO model in (3) is valid for binary variables, i.e., q_n is either 1 or 0 [24]. A simple transform of $c_n = 2q_n - 1$, together with $q_k = q_k^2 = q_k q_k$, can be applied to (4) in order to obtain the linear and quadratic coefficients, Q_{kl} and Q_{ll}, of the objective function in (3).

5 Results

In order to verify the the feasibility of decoding convolutional codes with QA, we compared it with brute force and Viterbi decoders that are implemented on CPU. In the experiments, convolutional encoder with generator polynomials [1 1 1] and [1 0 1] for rate 1/2 is used (see Fig. 5.1). A message binary sequence with length 3 is generated and obtained codewords are transmitted through an Additive White Gaussian Noise (AWGN) channel. At the receiver side, QA is implemented and ML estimate of transmitted codeword is obtained.

Proposed QA method was implemented on D-WAVE 2000Q provided by Amazon AWS using quantum annealing sampler API and exact solution (brute force) on local CPU. Both results from QA sampler and exact solution were compared. Monte Carlo simulations were run to obtain BER results on both local CPU and quantum machine with random Gaussian noise. We run more than 10000 trials with exact solver on CPU and 1000 trials with QA on QPU. At each trial random white Gaussian noise was generated and added to encoded messages.

D-WAVE's Python library "$dimod$" is used which has inbuilt function ($num.reads$) to define the number of states to read from the solver. It minimizes errors from quantum computer by averaging the solutions. In our work, $num.reads$ value was set at 1000. In the following, we discuss our findings in detail.

After code execution on D-Wave 2000Q we obtain results from Amazon AWS which are shown in Appendix. We have obtained such information as "solution-Counts", "shots": 1000, "createdAt": "2021-02-25T08:11:09.123Z", "endedAt": "2021-02-25T08:11:11.442Z", "status": "COMPLETED", "type": "QUBO", and

N	s1	s2	s3	s4	s5	s6	energy	num_oc
36	**0**	**0**	**1**	**1**	**1**	**1**	-4	1
27	**0**	**0**	**1**	**1**	**1**	**0**	-3	1
35	**0**	**0**	**0**	**1**	**1**	**1**	-3	1
37	**0**	**1**	**1**	**1**	**1**	**1**	-3	1
39	**0**	**0**	**1**	**1**	**0**	**1**	-3	1
43	**1**	**0**	**1**	**1**	**1**	**1**	-3	1
59	**0**	**0**	**1**	**0**	**1**	**1**	-3	1
4	**0**	**0**	**1**	**0**	**1**	**0**	-2	1
20	1	0	1	1	1	0	-2	1
24	0	0	1	1	0	0	-2	1
26	0	1	1	1	1	0	-2	1
28	0	0	0	1	1	0	-2	1
32	0	0	0	1	0	1	-2	1
34	0	1	0	1	1	1	-2	1
38	0	1	1	1	0	1	-2	1
40	1	0	1	1	0	1	-2	1
42	1	1	1	1	1	1	-2	1
44	1	0	0	1	1	1	-2	1
52	1	0	1	0	1	1	-2	1
56	0	0	1	0	0	1	-2	1
58	0	1	1	0	1	1	-2	1
60	0	0	0	0	1	1	-2	1
3	0	0	0	0	1	0	-1	1
5	0	1	1	0	1	0	-1	1
7	0	0	1	0	0	0	-1	1
11	1	0	1	0	1	0	-1	1
19	1	0	0	1	1	0	-1	1
21	1	1	1	1	1	0	-1	1
23	1	0	1	1	0	0	-1	1
25	0	1	1	1	0	0	-1	1
29	0	1	0	1	1	0	-1	1
31	0	0	0	1	0	0	-1	1
33	0	1	0	1	0	1	-1	1
41	1	1	1	1	0	1	-1	1
45	1	1	0	1	1	1	-1	1

N	s1	s2	s3	s4	s5	s6	energy	num_oc
47	1	0	0	1	0	1	-1	1
51	1	0	0	0	1	1	-1	1
53	1	1	1	0	1	1	-1	1
55	1	0	1	0	0	1	-1	1
57	0	1	1	0	0	1	-1	1
61	0	1	0	0	1	1	-1	1
63	0	0	0	0	0	1	-1	1
0	0	0	0	0	0	0	0	1
2	0	1	0	0	1	0	0	1
6	0	1	1	0	0	0	0	1
8	1	0	1	0	0	0	0	1
10	1	1	1	0	1	0	0	1
12	1	0	0	0	1	0	0	1
16	1	0	0	1	0	0	0	1
18	1	1	0	1	1	0	0	1
22	1	1	1	1	0	0	0	1
30	0	1	0	1	0	0	0	1
46	1	1	0	1	0	1	0	1
48	1	0	0	0	0	1	0	1
50	1	1	0	0	1	1	0	1
54	1	1	1	0	0	1	0	1
62	0	1	0	0	0	1	0	1
1	0	1	0	0	0	1	1	1
9	1	1	1	0	0	0	1	1
13	1	1	0	0	1	0	1	1
15	1	0	0	0	0	0	1	1
17	1	1	0	1	0	0	1	1
49	1	1	0	0	0	1	1	1
14	1	1	0	0	0	0	2	1

Table 5.1: Energy levels of codewords

"timing" is one of the most important obtained information. Main reason is that one of the objectives of this research is to compare code execution time on classical and quantum computers.

From Table 5.1, we see an output of one realization of a decoding process with quantum annealing where all 64 possible solutions with 6 bits are represented. The valid codewords in our code book are shown in bold and the one with lowest energy level is the best solution. If energy level is the same for two codewords, our algorithm selects one which is placed higher.

5.1 Error Performance Results

In Fig. 5.1, we compare the bit error rate (BER) results of decoding with Viterbi algorithm on CPU, Brute Force on CPU and QA on QPU. We also present the results of uncoded transmission as a reference. Local CPU results has the following characteristics: Macbook Pro 2.6 GHz Intel core i5 processor and Python v3.7.5.

Viterbi algorithm returns ML estimate of the transmitted data. It follows semi-brute force approach and it is optimal [22]. QA, on the other hand, provides optimal solution through tunnel effect with potential reduced executation time [22]. It is observed in Fig. 5.1 that the error performance of a received with QA gives the same performance of Viterbi decoder which validates the feasiblilty of such systems.

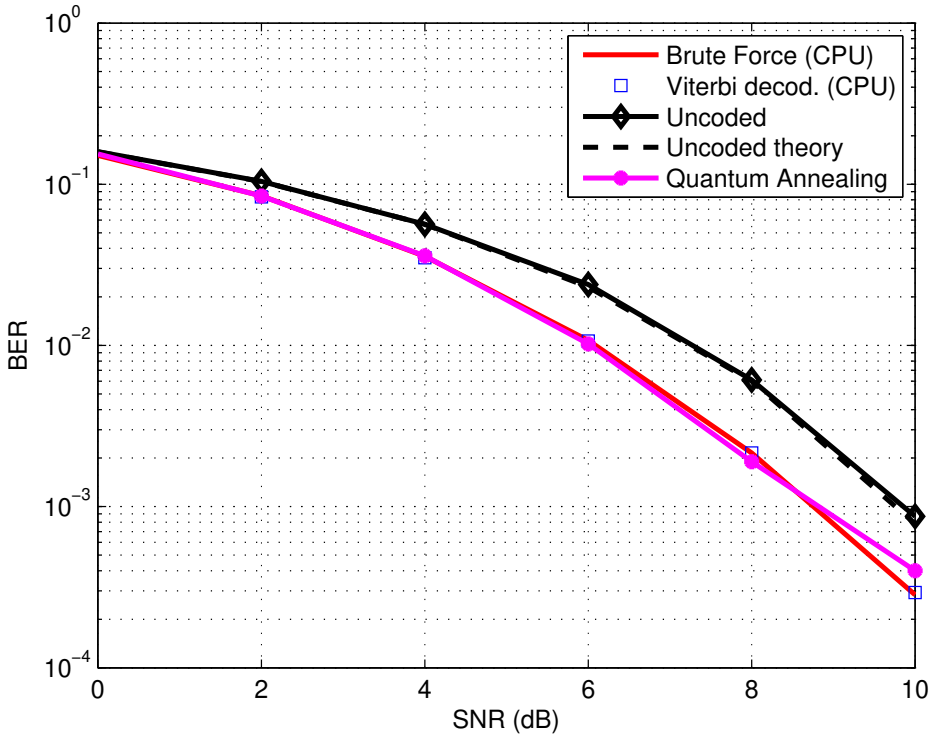

Figure 5.1: BER Comparison of Viterbi, Brute Force and QA methods.

5.2 Execution time comparison

In Table 5.2, execution time for each method is compared. Signal detection with QA takes 249611 usec excluding internet connection to the remote server but QPU access time (T) is included. However, access time includes both sampling time T_s and programming time T_p [25]. Most of the programming time is spent for data preparation before the sampling. Sampling time splits into anneal time per sample which is actual computation time for each trial [25].

Method	Time
Viterbi decoder	233usec
Quantum annealing	(QA)
Access time (T)	249611 usec
Sampling time (T_s)	238940 usec
Programming time (T_p)	10671 usec
Anneal Time per Sample (T_a)	20 usec

Table 5.2: Execution time for each method

Although anneal time is just 20 usec, the data preparation overhead still makes the QA machines slower than classical computing with Viterbi decoder. As it can be seen from Table 5.2, majority of time goes to connection to the server and programming time T_p.

6 Conclusion

In this work, we demonstrated that convolutional decoders could be executed on quantum computers using QA. First, we reformulated our problem as a QUBO problem to be able to run it on quantum computer. Amazon's cloud quantum computer D-Wave 2000Q was used to demonstrate successful decoding with QA. The results were compared with Viterbi decoder on classical computer. Results showed the potential use of quantum annealing in wireless communications, especially in C-RANs.

6.1 Future works

This research could be extended by different convolutional encoders with varying contraint leghth as well as can be applied to Low Density Parity Check (LDPC) codes. LDPC codes are frequently used in many wireless communications fields as 5G [26]. In this research, we studied convolutional codes by reformulating the decoding problem into QUBO and Ising models, however, the same methodology could be applied to LDPC codes (e.g., an earlier study can be found in [8]). LDPC codes have a better efficiency than convolutional codes when there are no practical delay constraints. In LDPC decoding the whole codeword must be sent to the receiver side and only after this step codeword could be processed. Thus, long codewords has much more latency to perform decoding. Nevertheless, convolutional codes could continuous decoding with a substantially low latency [26].

Another step forward could be implementing this project with Advantage QPU on Amazon AWS service which has 5000 available qubits and it could enhance

the results with longer bit sequences and varying code rates.

References

[1] D-Wave., Inc. (n.d.). Reformulating a problem. Retrieved March 18, 2021, from https://docs.dwavesys.com/docs/latest/chandbook3.html

[2] D-Wave System Inc. (2011). Introduction to D-Wave Quantum Hardware. Retrieved October 09, 2020, from https:/www.dwavesys.com/tutorials/background-r ading-series/introduction-dwave- quantum-hardware

[3] J. Cai, B. Macready, A. Roy, "A practical heuristic for finding graph minors", https://arxiv.org/abs/1406.2741

[4] Hamming, R. W. (1950). Error detecting and error correcting codes. The Bell system technical journal, 29(2), 147-160.

[5] Wayner, P. (2009). Disappearing cryptography: information hiding: steganography and watermarking. Morgan Kaufmann.

[6] Kim, M., Venturelli, D., Jamieson, K. (2019). Leveraging quantum annealing for large mimo processing in centralized radio access networks. In Proceedings of the ACM Special Interest Group on Data Communication (pp. 241-255).

[7] Kizilirmak, R. C. (2020, July). Quantum Annealing Approach to NOMA Signal Detection. In 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP) (pp. 1-5). IEEE.

[8] Ide, N., Asayama, T., Ueno, H., Ohzeki, M. (2020). Maximum-Likelihood Channel Decoding with Quantum Annealing Machine. arXiv preprint arXiv:2007.08689.

[9] Kim, M., Venturelli, D., Jamieson, K. (2019). Leveraging quantum annealing for large MIMO processing in centralized radio access networks. In Proceedings of the ACM Special Interest Group on Data Communication (pp. 241-255).

[10] Bensky, A. (2019). Short-range wireless communication. Newnes.

[11] Forney, G. (1970). Convolutional codes I: Algebraic structure. IEEE Transactions on Information Theory, 16(6), 720-738.

[12] Hu, S., Kröll, H., Huang, Q., Rusek, F. (2017). Optimal channel shortener design for reduced-state soft-output Viterbi equalizer in single-carrier systems. IEEE Transactions on Communications, 65(6), 2568-2582.

[13] Grami, A. (2015). Introduction to digital communications. Academic Press.

[14] Berber, S. M., Kecman, V. (2004, July). Convolutional decoders based on artificial neural networks. In 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541) (Vol. 2, pp. 1551-1556). IEEE.

[15] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE transactions on Information Theory, 13(2), 260-269.

[16] Davis, J., Lin, A., Njuguna Njoroge, A. T. (2002). Viterbi Algorithms as a Stream Application. signal, 1, 2.

[17] Forney, G. D. (1973). The viterbi algorithm. Proceedings of the IEEE, 61(3), 268-278.

[18] Randall, M., McMahon, G., Sugden, S. (2002). A simulated annealing approach to communication network design. Journal of Combinatorial Optimization, 6(1), 55-65.

[19] de la Fuente Ruiz, A. (2014). Quantum annealing. arXiv, arXiv-1404.

[20] Sengupta, R., Sengupta, D., Kamra, A. K., Pandey, D. ARTIFICIAL INTELLIGENCE AND QUANTUM COMPUTING FOR A SMARTER WIRELESS NETWORK. ARTIFICIAL INTELLIGENCE, 7(19), 2020.

[21] Finnila, A. B., Gomez, M. A., Sebenik, C., Stenson, C., Doll, J. D. (1994). Quantum annealing: a new method for minimizing multidimensional functions. Chemical physics letters, 219(5-6), 343-348.

[22] Cao, Y., Zhao, Y., Dai, F. (2019, July). Node Localization in Wireless Sensor Networks Based on Quantum Annealing Algorithm and Edge Computing. In 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 564-568). IEEE.

[23] Rieffel, E. G., Polak, W. H. (2011). Quantum computing: A gentle introduction. MIT Press.

[24] Cruz-Santos, W., Venegas-Andraca, S. E., Lanzagorta, M. (2019). A QUBo formulation of Minimum Multicut problem instances in trees for D-Wave Quantum Annealers. Scientific reports, 9(1), 1-12.

[25] "Breakdown of QPU access time." https://docs.dwavesys.com/docs/latest/c˙timing˙2.html, 2020.

[26] Maiya, S. V., Costello, D. J., Fuja, T. E. (2012). Low latency coding: Convolutional codes vs. LDPC codes. IEEE Transactions on Communications, 60(5), 1215-1225.

7 Appendices

7.1 Appendix A

"braketSchemaHeader": "name": "braket.task_result.annealing_task_result", "version": "1" ,

"solutions": [[1, 1, 1, 0, 1, 1]],

"solutionCounts": [1000],

"values": [-5.0],

"variableCount": 2048, "taskMetadata": "braketSchemaHeader": "name": "braket.task_result.task_metadata", "version": "1" ,

"id": "arn:aws:braket:us-west-2:573790626417:quantum-task/16a6a0ea-6069-4044-b3cb-97beccbce776", "shots": 1000, "deviceId": "arn:aws:braket:::device/qpu/d-wave/DW_2000Q_6",

"deviceParameters": "braketSchemaHeader":

"name": "braket.device_schema.dwave.dwave_device_parameters",

"version": "1" ,

"providerLevelParameters": "braketSchemaHeader": "name":

"braket.device_schema.dwave.dwave _provider_level_parameters", "version": "1"

, "createdAt": "2021-02-25T08:11:09.123Z",

"endedAt": "2021-02-25T08:11:11.442Z",

"status": "COMPLETED" ,

"additionalMetadata": "action": "braketSchemaHeader":

"name": "braket.ir.annealing. problem", "version": "1" , "type": "QUBO", "linear": "1955": -1.0, "1827": 1.0, "813": -1.0, "341": -1.0, "678": -1.0, "1968": -1.0 , "quadratic": , "dwaveMetadata": "braketSchemaHeader":

"name": "braket.task_result.dwave_metadata", "version": "1" , "activeVariables": [341, 678, 813, 1827, 1955, 1968], "timing": "qpuSamplingTime": 238940, "qpuAnnealTimePerSample": 20, "qpuAccessTime": 249611,

"qpuAccessOverheadTime": 2645, "qpuReadoutTimePerSample": 198,

"qpuProgrammingTime": 10671, "qpuDelayTimePerSample": 21, "postProcessingOverheadTime": 608, "totalPostProcessingTime": 608