xSA: A Binary Cross-Entropy Simulated Annealing Polar Decoder

Ryan Seah, Huayi Zhou, Marwan Jalaleddine, Warren J. Gross
Department of Electrical and Computer Engineering, McGill University, Montréal, Québec, Canada
Emails: {ryan.seah, huayi.zhou, marwan.jalaleddine}@mail.mcgill.ca, warren.gross@mcgill.ca

***Abstract*—Polar decoders such as successive-cancellation and successive-cancellation list decoders are limited by their sequential nature, which leads to a linear increase in latency with the codeword length. Heuristic based decoders such as quantum annealing have been proposed to overcome this limitation. However, these decoders have shown poor performance when decoding polar codes with more than eight bits.**

In this paper, we developed new meta-heuristic based polar decoder, called xSA, which uses a new receiver constraint modeled by the binary cross-entropy function. We also propose a method to determine the weights used in a quadratic unconstrained binary optimization (QUBO) function.

The polar code is assumed to have been sent across an AWGN channel and we conducted our experiments and simulations using PyQUBO and dwave-neal. Our results show that xSA is able to decode codes of length 16 and 32 with a near-ML FER performance, presenting itself as a promising alternative to traditional polar decoders for real world applications and next generation cellular communications.

I. Introduction

Polar codes are capacity-achieving error-correcting codes with low-complexity encoding and decoding algorithms [1]. They have been incorporated into the development of the 5G wireless communication standard [2].

Polar codes achieve their superior performance by polarizing the channels, i.e., transforming a set of independent and identically distributed (i.i.d.) channels into a set of highly reliable and highly unreliable channels. This polar transformation enables the use of simple and efficient decoding algorithms, such as the successive cancellation (SC) decoder and successive cancellation list (SCL) decoder [3], [4]. However, as SC and SCL decoders make decisions sequentially, their latency grows linearly with the codeword length. As an alternative paradigm for next generation communications, optimization-based meta-heuristic decoders (MHD) may offer a high decoding-performance and lower latency solution. MHD can be implemented with generalized hardware that solves optimization problems expressed in a standard form with code-specific constraints, as shown in Fig. 1.

Recently, a meta-heuristic polar decoder using quantum annealing (QA) combined with classical methods, called the Hybrid Polar Decoder (HyPD) was proposed in [5]. QA is a quantum technique used to solve optimization problems, which leverages quantum mechanics to search for the optimal solution among numerous possibilities efficiently.

The QA process involves encoding the problem into a Hamiltonian, which is a mathematical representation of the energy levels of a quantum system. The quantum system is then initialized in a superposition of all possible states and allowed to evolve through time.

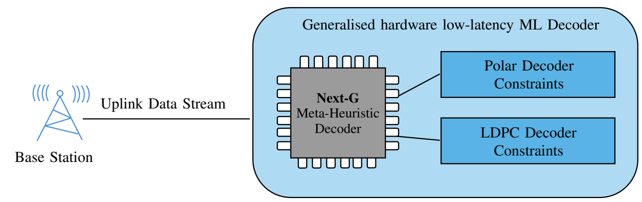


Fig. 1. A next generation (Next-G) meta-heuristic decoder architecture.

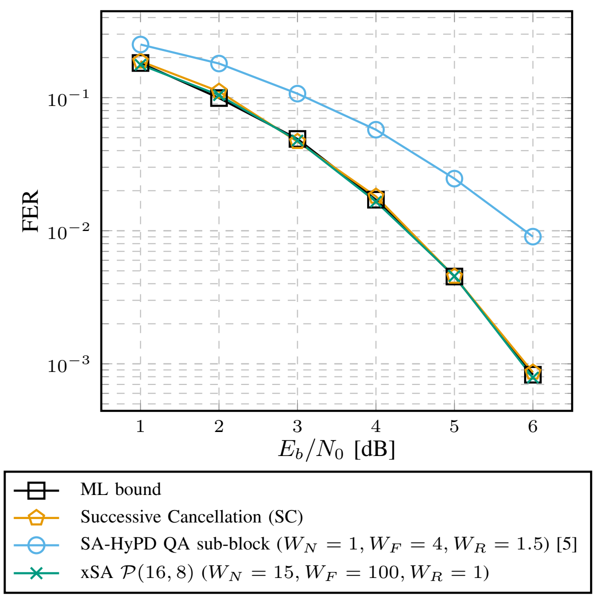


Fig. 2. xSA FER performance for a polar code of length $N = 16$ and rate $R = 0.5$, using `num_reads` $= 300$.

The system settles into its ground state, which corresponds to the optimal solution. The search process is aided by quantum tunneling, which allows the system to overcome energy barriers and reach lower energy levels.

While HyPD has shown that it is possible to use meta-heuristic optimizers to build a polar code decoder on a multi-path Rayleigh fading channel, the decoder is limited to codes of length $N = 8$; with longer codes showing poor performance. This has been attributed to the accumulation of QA integrated control error (ICE) [5] [6].

Another meta-heuristic optimization method is simulated annealing (SA). SA is a probabilistic technique that approximates the global optimum of a given function. SA mimics the process of annealing in metallurgy, where a material is heated and then slowly cooled to increase the size of its crystals and reduce their defects. Simulated annealing starts with a high temperature and slowly cools down to a low temperature. At each temperature, the algorithm generates a random neighbor of the current solution and accepts it if it is better than the current solution. Otherwise, it accepts it with a probability that decreases as the temperature decreases. This process is repeated until the temperature reaches a sufficiently low value.

We used SA to extend the work of [5] to develop a polar decoder for a codeword of length $N = 16$ at $R = 0.5$ over an Additive White Gaussian Noise (AWGN) channel. We observe the same poor performance suggested in [5], as shown in Fig 2. Instead of attributing this to accumulated errors in QA, we suggest that the poor performance is mainly caused by the rigidity of the distance metric function used in the receiver constraint, which prevents the optimizer from escaping a local minimum. Other works such as [7] have also proposed the use of QA to decode LDPC codes, but both works use the same distance metric function. Hence, in this paper we use a binary cross entropy loss function [8] to develop a cross-entropy SA polar decoder, which we call xSA, and were able to achieve near-maximum likelihood (near-ML) performance for polar codes of length $N = 16$ with rate $R = 0.5$. We were able to further demonstrate that xSA was able to decode codewords of length $N = 32$ at rates $R \in \{0.375, 0.5, 0.8125\}$. In addition, we propose a method to determine the weights required for our optimization function to further prevent our overall optimization function from being stuck in a local minimum.

In summary, this paper makes the following contributions:

- We developed a SA polar decoder for $N = 16$ with rate $R = 0.5$ with near ML performance and aim to extend it to decode polar codes of length $N = 32$ for rates $R \in \{0.375, 0.5, 0.8125\}$.
- We propose replacing the distance metric loss function with a binary cross-entropy loss function.
- We implemented a method for rough approximation of weights tuning for a SA polar decoder optimization function.

II. Preliminaries

A. Polar codes

A polar code $\mathcal{P}(N, K)$ with length N and dimension K can be constructed by taking a message word $\mathbf{u} = [u_0, u_1, \ldots, u_{N-1}]$, containing K *information bits* and a set of $N - K$ *frozen bits* and applying a linear transformation $\mathbf{x} = \mathbf{u}\mathbf{G}^{\otimes n}$, where $\mathbf{x} = [x_0, x_1, \ldots, x_{N-1}]$ is a codeword, $\mathbf{G}^{\otimes n}$ is the n-th Kronecker power of the polarizing matrix $\mathbf{G} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, and $n = \log_2 N$. The location of the frozen bits (assumed here to have value 0) are known to both the encoder and decoder. The rate of the code is $R = K/N$.

For example, the encoding process $\mathbf{x} = \mathbf{u}\mathbf{G}^{\otimes 3}$ for an input vector $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7]$ with length $N = 8$ can be seen in Fig. 3. The codeword $\mathbf{x}$ is then modulated and sent over a channel.

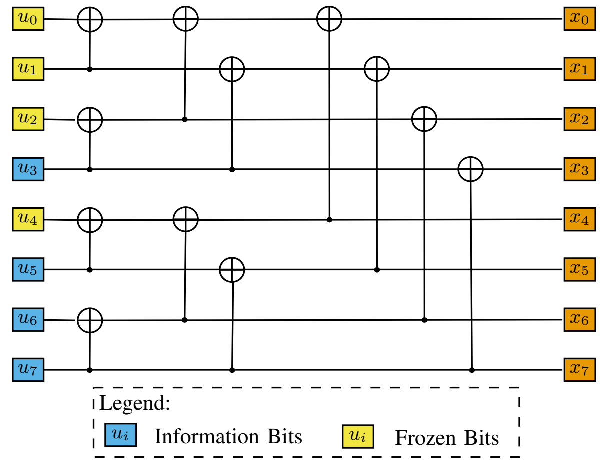


Fig. 3. Polar code ($N = 8$) $\mathbf{x} = \mathbf{u}\mathbf{G}^{\otimes 3}$ encoding process.

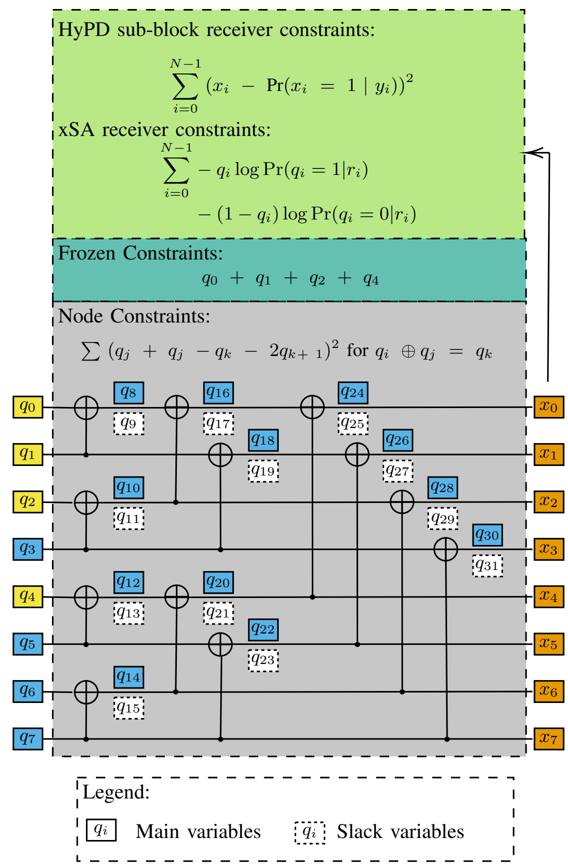


Fig. 4. Cost function design for xSA and HyPD sub-block $\mathcal{P}(8, 4)$.

When sent over an AWGN channel, noise $\mathbf{n} \sim N(\mathbf{0}, \sigma^2)$,, is added to the transmitted codeword and the receiver receives the signal $\mathbf{r} = [r_0, r_1, \ldots, r_{N-1}]$:

$$\mathbf{r} = \mathbf{x} + \mathbf{n}, \tag{1}$$

where σ^2 is the noise variance.

B. Optimization objective function

The objective function for our SA polar decoder takes the form of a Quadratic Unconstrained Binary Optimization (QUBO) function. A QUBO function can be described as follows:

$$H_{QUBO} = \sum_{i=0}^{T-1} \sum_{j=0}^{T-1} Q_{ij} q_i q_j, \tag{2}$$

where $q_i, q_j \in \{1, 0\}$. $Q_{ij} \in \mathbb{R}$ are coefficients used to penalize or reward a solution. T is the total number of variables in the optimization function.

The QUBO function is equivalent to the Ising model [9] and can solved using a quantum annealing (QA) device.

Take $\mathbf{q} = [q_0, q_1, \ldots, q_{T-1}]$ to be a particular solution, which corresponds to the input bits $\mathbf{u} = [u_0, u_1, \ldots, u_{N-1}]$; $\mathcal{F}$ to be the set of frozen bits; $\mathcal{T}$ to be the set of nodes in the polar code's binary tree and a_i to be *slack* variables used for calculation, $a_i \in \{0, 1\}$.

In HyPD [5], the objective function comprises three constraints expressed as penalties added to the QUBO function, namely the *Node* constraint (C_N), the *Frozen* constraint (C_F), and the *Receiver* constraint (C_R). The *Node* constraint ensures that the decoded codeword is a polar code. The *Frozen* constraint ensures that frozen bits conform to the set of frozen bits in $\mathcal{F}$. The *Receiver*(C_R) constraint ensures that decoded code word does not deviate too much from the received information. The optimization function is:

$$\begin{aligned} f(\mathbf{q}) = W_N \sum_{T \in \mathcal{T}} C_N(T) + W_F \sum_{\mathbf{q}_i \in \mathcal{F}} C_F(q_i) \\ + W_R \sum_{q_i \in \mathbf{q}} C_R(q_i), \end{aligned} \tag{3}$$

where the optimization problem is to find the values $\mathbf{q}$ that minimize $f(\mathbf{q})$:

$$\arg\min_{\mathbf{q}} \{f(\mathbf{q})\}. \tag{4}$$

The weights W_N, W_F and W_R determine the importance of each constraint in the optimization problem. The weights in HyPD [5] are set to $W_N = 1$, $W_F = 4$ and $W_R = 2 - R$. The overall cost function design can be seen in Fig. 4.

1) Encoding constraint: This constraint models the XOR operations of the Polar encoder. Take ϵ_T as the set of all XOR operations performed at node T. We can then define the constraint as follows,

$$C_N = \sum_{(q_i, q_j) \in \epsilon_T} (q_i + q_j - q_k - 2q_{k+1})^2 \tag{5}$$

where q_k, q_{k+1} are slack variables and each slack variable is only introduced once. Since $C_N(T)$ is the sum-of-squares its minimum energy (i.e., $C_N(T) = 0$). In addition, the sum $q_i + q_j$ must be equal to the sum $q_k + 2q_{k+1}$. As all the variables are binary, this models the XOR operation of q_i and q_j, $q_k = q_i \oplus q_j$.

2) Frozen constraint: As the frozen bits are always zero, we can define the frozen constraint as follows,

$$C_F(q_i) = q_i \tag{6}$$

C_F is minimum when all the frozen bits position in a solution, $\mathbf{q}$ is zero.

3) Receiver constraint: For AWGN with noise variance σ^2 we can compute the probability that received information, $r_i = 1$ as:

$$\Pr(q_i = 1 | r_i) = \frac{1}{1 + \exp\left(\frac{2r_i}{\sigma^2}\right)} \tag{7}$$

In HyPD, the receiver constraint is defined using a distance metric as follows,

$$C_R(q_i) = \left(q_i - \Pr(q_i = 1|r_i)\right)^2, \tag{8}$$

where C_R is minimized when $q_i \in \{0, 1\}$ has a greater probability of being the corresponding bit at the encoder. Hence, this constraint ensures that the solution does not deviate too much from the received information.

III. Methodology

A. Distance metric vs Binary Cross Entropy

In our work, we introduce a new receiver constraint and replace the distance metric, defined in Equation (8), with a binary cross entropy function as follows,

$$\begin{aligned} C_R(q_i) = &-q_i \log \Pr(q_i = 1|r_i) \\ &- (1 - q_i) \log \Pr(q_i = 0|r_i) \end{aligned} \tag{9}$$

where the probability function is defined in Equation (7). The new optimization cost function design can be seen in Fig. 4.

Similar to the distance metric, the binary cross entropy function is minimized when $q_i \in \{0, 1\}$ has a greater probability of being the corresponding bit at the encoder. In Fig. 5, the binary cross entropy function and distance metric is first normalized for a received symbol $r_i \in [-2, 2]$ and $\sigma^2 = 0.5$. Next, the normalized cost, $\hat{C_R}$ for $r_i \in [-2, 0]$ is plotted as it is symmetrical about the origin.

From Fig. 5, we can observe that when sufficient noise is added to the transmitted symbol, r_i, the distance metric would introduce a higher cost compared to the binary cross-entropy function.

Given an example shown in Fig. 6, for a certain $q_i = 0$, the symbol, $x_i = +1$ is transmitted and during transmission, a noise of, $n_i = -1.5$, is added to the received symbol, $r_i = -0.5$, which corresponds to a bit flip. In case 1, the distance metric and binary cross-entropy would both view $q_i = 1$ as the more optimum solution. However, in case 2, in order to select the right solution of $q_i = 0$, the distance metric would introduce a higher cost of 0.78 compared to the binary cross-entropy function, 0.27. Thus, it is less probable for the MHD to explore a sub-optimum solution of $q_i = 0$ when the constraint is formulated using the distance metric compared to binary cross-entropy function. This limits the MHD search space for a correct solution and causes the MHD to be stuck in a local minimum with the wrong solution. Hence, from our analysis, we can conclude that the distance metric is not

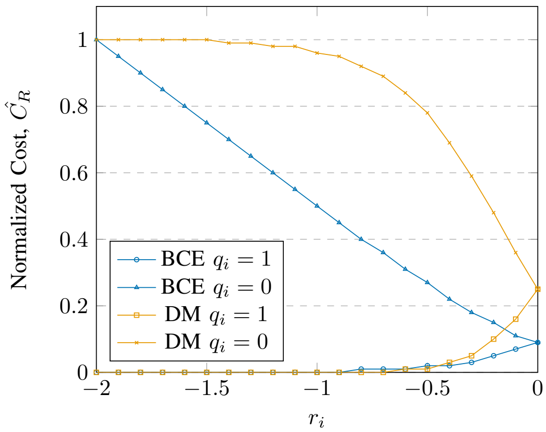


Fig. 5. Comparison of the cost introduced by distance metric (DM) vs binary cross-entropy (BCE) for $r_i \in [-2, 0]$ normalized in the range $r_i \in [-2, 2]$ and $\sigma^2 = 0.5$. The symbol mapping here corresponds to $1 \rightarrow -1$ and $0 \rightarrow +1$

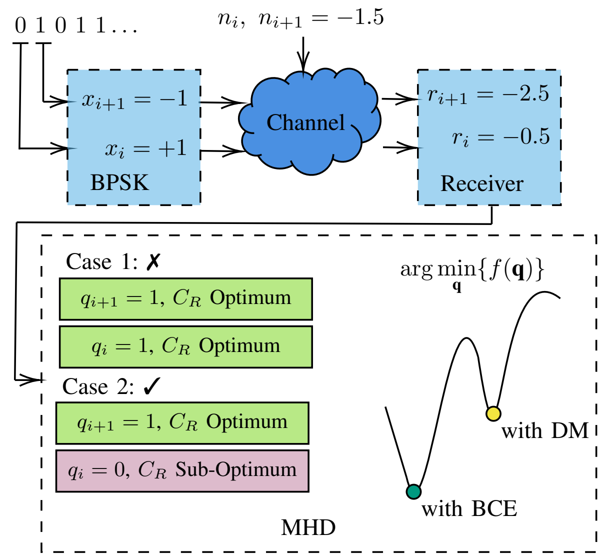


Fig. 6. Illustration of meta-heuristic decoder(MHD) decision process. Case 1 represents a wrong solution, while case 2 represents the correct solution. In case 1 the receiver constraint cost is lower but by compromising the receiver constraint cost slightly we can find a lower overall cost.

suitable for the receiver constraint and likely accounts for HyPD poor performance.

B. Coarse parameter tuning

When developing our SA decoder, we realized that depending on the number of variables, it is possible for a constraint to dominate other constraints of higher weight. This results in certain important constraints to be continuously violated.

For example, take a codeword of $N = 8$, and we would like the *receiver* constraint to have a higher importance in the decoding process compared to the *encoding* constraint. We can then set $W_N = 1.5$ and $W_R = 2$. However, as the encoding constraint has at least 14 variables compared to the 8 variables in *receiver* constraint, with the highest cost of the *encoding* constraint and *receiver* constraint to be 21 and 16 respectively. Thus, the QA deoder would place more important to the *encoding* constraint instead of the *receiver* constraint.

To prevent constraints from dominating each other. We proposed that the initial weights to be set as follows,

$$W_N = \frac{1}{\max \sum_{T \in \mathcal{T}} C_N(T)} \tag{10}$$

$$W_F = \frac{1}{\max \sum_{\mathbf{q}_i \in \mathcal{F}} C_F(q_i)} \tag{11}$$

$$W_R = \frac{1}{\max \sum_{q_i \in \mathbf{q}} C_R(q_i)} \tag{12}$$

Further fine-tuning can be done later, to increase the importance of each constraint.

IV. Experimental results

In our experiments, the polar codes are constructed by Gaussian approximation [10] at 5dB. The polar codes are then assumed to have been sent across an AWGN channel decoded using a polar code decoder. The polar decoders are implemented on an Intel E5-2683 processor and both xSA and HyPD are implemented in python with PyQUBO and dwave-neal libraries.

PyQUBO is a python library that allows the user to formulate a QUBO problem and solve it using a variety of solvers. Dwave-neal [11] is a SA python library that is used to approximate Boltzmann sampling or meta-heuristic optimization. Within each iteration of the SA, each variable in the *Ising* model is updated once in a fixed order per point in a sequence determined by the Metropolis-Hastings update.

The ML bounds are generated using sphere decoding [12] and the SCL decoder is implemented according to [13].

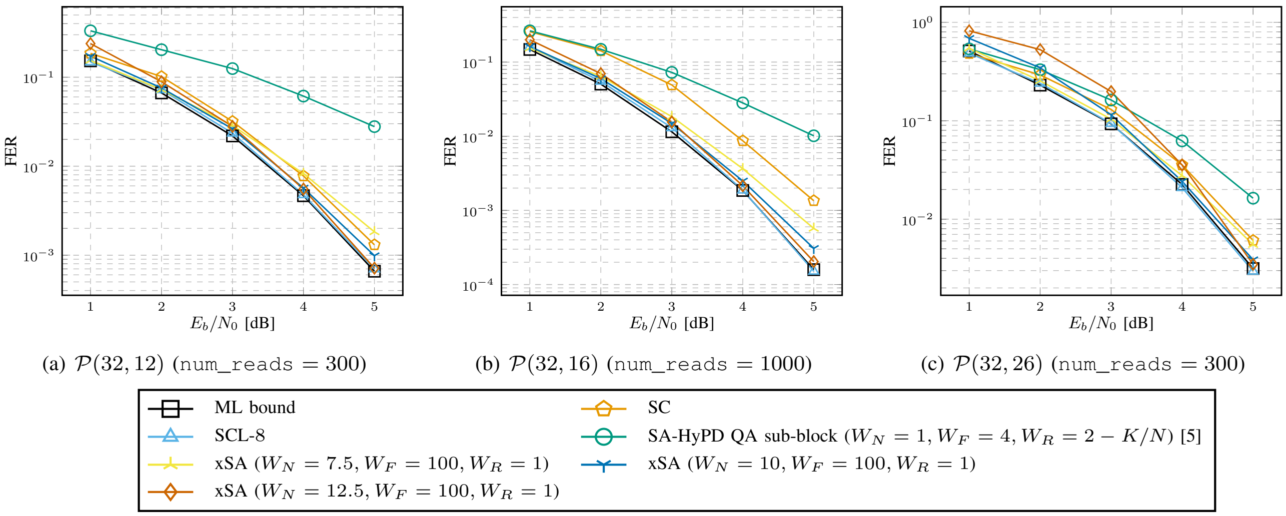
Fig. 2 shows the FER performance of polar code $\mathcal{P}(16, 8)$. `num_reads` represents the number of iterations of the SA algorithm. At $\mathcal{P}(16, 8)$ xSA is able to achieve near-ML performance, while HyPD SA decoder is unable to achieve the ML bound.

We then extended the length from $N = 16$ to $N = 32$ and compared the performance of xSA and HyPD. Fig. 7 shows the FER for $\mathcal{P}(32, 12)$, $\mathcal{P}(32, 16)$ and $\mathcal{P}(32, 26)$. At $\mathcal{P}(32, 12)$, as shown in Fig. 7(a), xSA was able to achieve near-ML performance. xSA was able to stay close to the ML bound at $\mathcal{P}(32, 16)$ but its performance degrades at $\mathcal{P}(32, 26)$, as shown in Fig. 7(c). This is likely due to the lack of frozen bits for error correction, making it difficult for the meta-heuristic optimizers to solve the problems correctly.

However, for $\mathcal{P}(64, 32)$ we notice that xSA was not able to achieve ML performance. This is likely due to the higher number of variables (448), compared to 192 variables for $\mathcal{P}(32, 16)$, resulting in higher accumulated error. More work needs to be done to ensure that MHD can be directed to explore the right search space.

V. Conclusion

In this paper, we proposed a method to determine the weights required for a meta-heuristic decoder that minimizes the QUBO function and an alternative meta-heuristic polar



(a) $\mathcal{P}(32, 12)$ (`num_reads` $= 300$) (b) $\mathcal{P}(32, 16)$ (`num_reads` $= 1000$) (c) $\mathcal{P}(32, 26)$ (`num_reads` $= 300$)

Fig. 7. The FER of decoding polar code length $N = 32$ on different code rates and decoders.

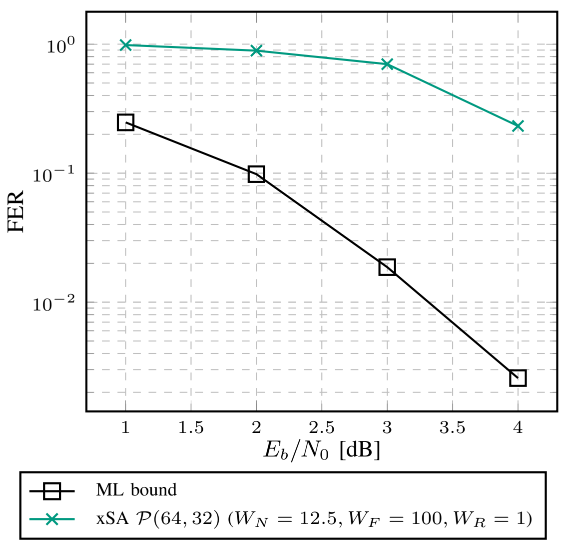


Fig. 8. xSA FER performance for $\mathcal{P}(64, 32)$, using `num_reads` $= 300$.

decoder, xSA. xSA introduces a binary cross-entropy receiver constraint. We have shown that xSA is able to decode polar codes with $N = 16$ and $N = 32$ and achieve near-ML performance, outperforming other works such as HyPD and SC decoders on an AWGN channel.

While we note that QA and SA are not the same, we believe that the comparison with HyPD is still valid as both methods use meta-heuristic optimization. The optimal solution should not be impacted by the choice of the solver. Also, xSA can be easily extended to be solved using QA as it's objective function is in QUBO form.

Finally, future advancements in quantum technology can make it a reality for meta-heuristic decoders to be used in real-world applications and next generation cellular wireless traffic channels.

Our future work includes reducing the number of variables required for meta-heuristic based polar decoders, to help reduce accumulated errors. This may allow us to decode polar codes at higher code length.

References

[1] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[2] 3GPP, "Multiplexing and channel coding," *Technical Specification (TS) 38.212, 3rd Generation Partnership Project (3GPP)*, vol. 6, 2018.

[3] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE transactions on information theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[4] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast list decoders for polar codes," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 2, pp. 318–328, 2015.

[5] S. Kasi, J. Kaewell, and K. Jamieson, "The design and implementation of a hybrid classical-quantum annealing polar decoder," in *IEEE Global Communications Conference*. IEEE, 2022, pp. 5819–5825.

[6] D-Wave, "Technical description of the d-wave quantum processing unit,," *D-Wave System User Manual*, p. 09–1109A–O, 2019.

[7] S. Kasi and K. Jamieson, "Towards quantum belief propagation for ldpc decoding in wireless networks," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020, pp. 1–14.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[9] D-Wave, "Problem formulation guide whitepaper,," *D-Wave Problem Formulation Guide*, 2022.

[10] P. Trifonov, "Efficient design and decoding of polar codes," *IEEE transactions on communications*, vol. 60, no. 11, pp. 3221–3227, 2012.

[11] D-Wave, "Dwave-neal," *D-Wave Neal Documentation*, 2022.

[12] H. Zhou, W. J. Gross, Z. Zhang, X. You, and C. Zhang, "Efficient sphere polar decoding via synchronous determination," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6777–6781, 2020.

[13] Y. Shen, C. Zhang, J. Yang, S. Zhang, and X. You, "Low-latency software successive cancellation list polar decoder using stage-located copy," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*. IEEE, 2016, pp. 84–88.