www.czasopisma.pan.pl PAN www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2022, VOL. 68, NO. 1, PP. 109-114
Manuscript received September 23, 2021; revised January, 2022. DOI: 10.24425/ijet.2022.139857

Optimizing the Bit-flipping Method for Decoding Low-density Parity-check Codes in Wireless Networks by Using the Artificial Spider Algorithm

Ali Jasim Ghaffoori and Wameedh Riyadh Abdul-Adheem

***Abstract*—In this paper, the performance of Low-Density Parity-Check (LDPC) codes is improved, which leads to reduce the complexity of hard-decision Bit-Flipping (BF) decoding by utilizing the Artificial Spider Algorithm (ASA). The ASA is used to solve the optimization problem of decoding thresholds. Two decoding thresholds are used to flip multiple bits in each round of iteration to reduce the probability of errors and accelerate decoding convergence speed while improving decoding performance. These errors occur every time the bits are flipped. Then, the BF algorithm with a low-complexity optimizer only requires real number operations before iteration and logical operations in each iteration. The ASA is better than the optimized decoding scheme that uses the Particle Swarm Optimization (PSO) algorithm. The proposed scheme can improve the performance of wireless network applications with good proficiency and results. Simulation results show that the ASA-based algorithm for solving highly nonlinear unconstrained problems exhibits fast decoding convergence speed and excellent decoding performance. Thus, it is suitable for applications in broadband wireless networks.**

***Keywords*—a low-density parity-check (LDPC); hard-decision Bit-Flipping (BF); particle swarm optimization (PSO); artificial spider algorithm (ASA)**

I. Introduction

THE low-density parity-check (LDPC) code is a good method for approaching the Shannon limit [1]; it has been widely used in many fields, such as deep-space communications, optical communications, and disk storage [2], [3]. The performance of LDPC codes approaches the Shannon limit through iterative belief propagation decoding. In recent years, LDPC codes have become a popular topic in the field of coding due to their excellent performance [4]. Although the decoding performance of belief propagation algorithms is commendable, their implementation complexity is relatively high. Gallager [5] first introduced the concept of bit-flipping (BF) decoding. This process involves repeatedly passing binary messages between two sets of nodes, namely variable nodes (VN) and check nodes (CN) [6]; it uses reliable channel information as input values. CN adopts exclusive OR operation. The BF has a very low complexity decoder but low-resolution and its error correction capability is poor. Although its performance is slightly poor, the BF algorithm only involves logical operations; thus, it is simple, fast, and easy to implement. To improve the performance of the BF algorithm, a weighted BF (WBF) algorithm [8] and various improved algorithms [9] have been developed. However, decoding complexity is considerably increased. In real-time communication systems, reducing decoding complexity is frequently necessary and decoding speed must be increased as much as possible while ensuring decoding performance [10]. ASA is an optimization approach proposed by Behrouz et al. [11]; it is inspired by the natural hunting behavior of spiders. The application of ASA to BF exhibits good capacity, efficiency, and durability.

Compared with other known evolutionary methods, such as the honey bee algorithm, particle swarm optimization (PSO) algorithm, and artificial bee colony algorithm, ASA exhibits better performance and effectiveness. Its response is more accurate, and it can be used to find the global optimal value and reduce the costs of various measurements and tests [11].

Considering the time-varying nature of wireless communication channels, error correction codes should be compatible with rates. ASA exhibits extremely low complexity and excellent decoding performance; thus, it is highly suitable for designing high-throughput hardware decoders in wireless networks.

II. BF AND WBF ALGORITHMS

Cellular communication systems, including broadband wireless networks, require reliable and high-speed transmission for gigabit ethernet, data storage, and long-haul optical channels. Three critical issues must be addressed to design high-performance and high-speed LDPC codes for these systems. A binary regular LDPC code with a code length of N and an information length of K must be considered [12]. Its check matrix H is a sparse matrix, i.e., $H = [h_{mn}]$ of M and N, and each column of H has a value of $\gamma = 1$ and $\rho = 1$. The set of bits participating in the m-th check equation (i.e., the set of 1 in the m-th row of H) is denoted as $(m) = (n{:}h_{mn} = 1)$, and the set of bits participating in the n-th check equation (i.e., the set of rows with 1 in the n-th column of H) is denoted as $M(n) = \{m{:}h_{mn} = 1\}$.

A. J. Ghaffoori is with Department of Electrical Power Techniques Engineering , AL_Ma'moon University College, Baghdad, Iraq (email: ali_jk1972@yahoo.com).

W. R. Abdul-Adheem is with Department of Electrical Power Techniques Engineering , AL_Ma'moon University College, Baghdad, Iraq (email: Wameedh.r.abduladheem@almamonuc.edu.iq).

Assume that the binary code word is $c = [c_0, c_1, ..., c_{N-1}]$. After binary phase-shift keying modulation, the sequence $x = [x_0, x_1, \ldots, x_{N-1}]$ is obtained, where $x_n = 1-2c_n$. x enters the mean value of zero, and the variance is $\sigma^2 = N0/2$. An additive white Gaussian noise (AWGN) channel is obtained after the channel output sequence $r = [r_0, r_1, n..., r_{N-1}]$, where $r_n = x_n + v_n$, and v_n is AWGN. In accordance with the received sequence r, the binary hard decision sequence $z = [z_0, z_1, \ldots, z_{N-1}]$. If $r_n > 0$, then $z_n = 0$; if rn = 0, then $z_n = 1$.

As shown in Figure 1, in each iteration of the BF algorithm, the value of the syndrome is first calculated according to the hard decision sequence of the previous round. If all syndromes are 0, the iteration is stopped and the decoding is successful, otherwise the correction involved in each bit is calculated. The number f_n of the check equation with sub-1, flip the bit z_n whose f_n is greater than a certain preset threshold T and then enter the next round of iterations until the decoding is successful or the maximum number of iterations is reached. The size of the threshold T can be set appropriately to achieve the best decoding performance.

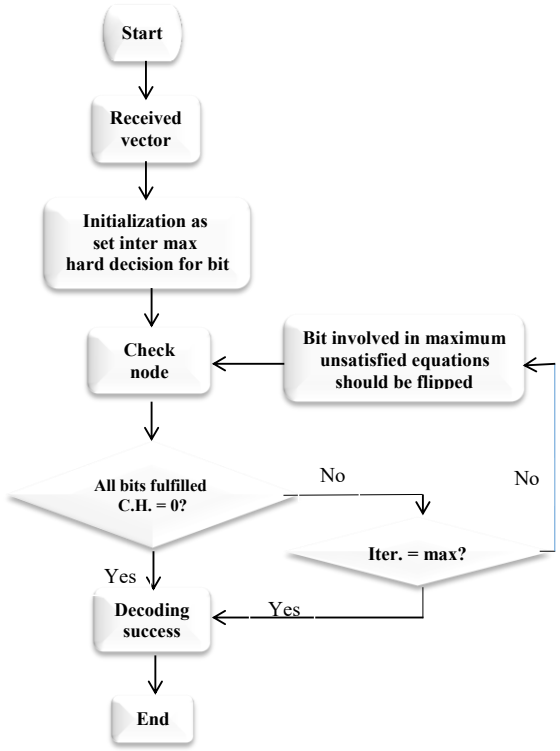

Fig. 1. Flowchart for BF.

The syndrome $s = [s_0, s_1, \ldots, s_{M-1}]$ is calculated through the hard decision z as follows:

$$s_m = \sum_{n=0}^{N-1} z_n \, h_{mn} \bmod 2, m = 0, 1, \ldots, M-1 . \quad (1)$$

If s = 0, then stop the iteration and display the decoding success; otherwise, proceed to Step (2).

1) For each bit z_n in the hard decision sequence, the syndrome that it participates in is 1. The number of check equations f_n is as follows:

$$f_n = \sum_{m=0}^{M-1} s_m \, h_{mn} , n = 0, 1, \ldots, N-1 . \quad (2)$$

If $f_n > T$, then flip z_n and make a new hard decision z.

2) Steps (1) and (2) are repeated until the decoding is successful or the maximum number of iterations is reached and the decoding failure is displayed. The BF algorithm is a simple hard decision algorithm. It only requires logical operations, and its implementation is extremely simple. However, its performance is slightly poor. The channel output is introduced into the BF algorithm. The soft information of symbol credibility can be effective in improving the performance of the BF algorithm, but it increases complexity.

The WBF algorithm is an improved BF algorithm based on soft information. In each round of iteration, the flip of each symbol bit is calculated in accordance with the credibility of the check equation, i.e., function E_n, and the largest bit of E_n is flipped.

The steps of the WBF algorithm are as follows.

1) The reliability L_m of the m-th check equation is calculated.

$$L_m = \min_{n \in N(m)} \{ |r_n| \} , m = 0, 1, \ldots, M-1 \quad (3)$$

The syndrome s from Equation (1) is calculated. If s = 0, then the iteration is stopped and the decoding success is displayed; otherwise, proceed to Step (3).

3) For each hard decision bit z_n, its flip function E_n is calculated in accordance with the credibility of the check equation in which the bit participates.

$$E_n = \sum_{m \in M(n)} (2s_m - 1) \, L_m , n = 0, 1, \ldots, N-1 \quad (4)$$

The largest bit z_n of E_n is flipped to obtain a new hard decision sequence z.

4) Steps (2) and (3) are repeated until the decoding is successful, or the maximum number of iterations is reached and the decoding failure is displayed.

From Equation (3), the WBF algorithm M(ρ−1) requires real number operations to calculate the credibility of the check equation. Moreover, I (N−1 + ργ) times real numbers are required to calculate the flip function and perform bit flip operation, where I is the number of iterations, in the WBF algorithm. That is, the WBF algorithm has achieved performance improvements compared with the BF algorithm at the expense of complexity.

III. OPTIMIZATION ALGORITHMS

A) ASA

ASA is a new method proposed by Behrouz et al. [11]. It simulates the behavior of hunting spiders, and its function is to limit the primary network to the allowable search range. All the nodes of the spiders web is calculated as the target function value. The best knot is equivalent to a captured insect around the node and an artificial spider can notice one, two, or as many captured insects as possible based on the choices. Then, an artificial spider weaves a small web around the center of its target to find the exact location of insects (i.e., prey. The small mesh nodes are calculated and the better node represents the most accurate location of an insect.

The next step is to weave artificial spiders at small intervals of time throughout the research area, allowing new insects (i.e., nodes with better target functional values) to enter the spider web. In addition to the insects caught in the previous stage, a spider also sends a small net to catch new insects and determine which is the best hunting option. As long as a spider weaves its web around its chosen victim, the optimum point is always reached. This pressure continues, and the spider web is transmitted. Spiders will not continue weaving in certain parts

of the entire primary web where no insects are caught after many iterations which are saving energy and accelerating the search for ideal insects.

In addition, the captured insects are secured as fast as possible in the subsequent repetition. If the position of the insects is not changed, then the small net will become smaller to cover the space between the nodes more accurately, finally catching the chosen insects.

The steps of the proposed ASA are described as follows.

(1) The population points of each iteration that are regarded as primary are stored in a matrix called C.

(2) The composition of the sample population is between zero and one in accordance with the number of departments.

(3) The search range, number of elites, and initial area radius are determined and defined.

(4) The elites in the previous iteration are stored in a matrix called D.

(5) New populations are formed in the domain in matrices C and D.

(6) The objective function value is calculated and sorted.

(7) A search is conducted locally for the elites in this iteration and for the global elites.

(8) The elites in this iteration and the global elites are stored.

(9) If the elite in this iteration is better than the global elite, then it is replaced.

(10) The radius of neighbors will shrink if the global elite does not change.

(11) If the neighboring radius reaches the minimum value, then it will remain constant.

(12) In accordance with the coordinates of the elite, the range is reduced after a considerable number of iterations.

(13) When the stop criterion is satisfied, return to Step (4).

The flowchart of ASA is depicted in Figure 2.

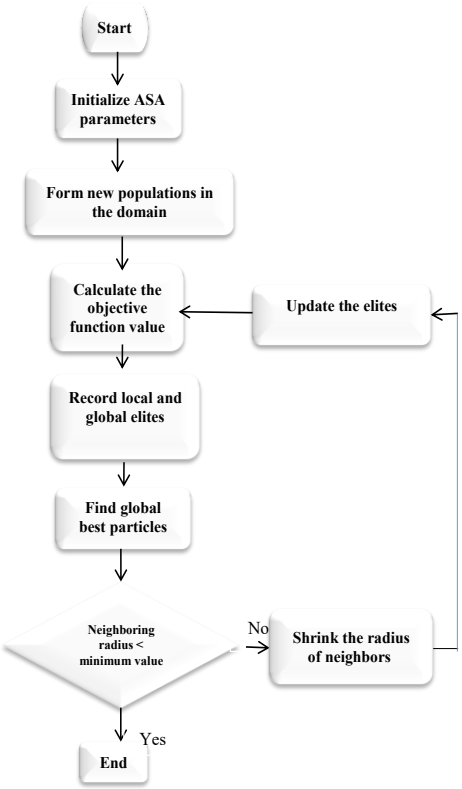

Fig. 2. Flowchart of ASA.

B) PSO

The PSO algorithm was proposed by American electrical engineer Russel Eberhart and social psychologist James Kennedy [13] in 1995; it is based on the behavior of a foraging flock of birds. Each bird does not know where food is located. Over time, however, the birds, which are initially in random positions, learn from other members of the group and share information. Individuals continue to accumulate experience in finding food and spontaneously organize a community. Gradually, individuals move toward only one goal: food. Each bird can use certain experiences and information to estimate how valuable its current position is for finding food. Each bird can remember the best position it has found, which is called the local maximum.

In addition, each bird can remember the best position that all individuals in the flock have found is called the global optimum. In entire flock the foraging center of the birds moves toward the global optimum. Through the constant movement of the bird flock's foraging position.

The process of PSO is summarized as follows [14].

Step 1: The particle swarm is randomly initialized within the scope of initialization, including random position and velocity.

Step 2: The fitness value of each particle is calculated.

Step 3: The historical optimal position of an individual particle is updated.

Step 4: The historical optimal position of the particle swarm is updated.

Step 5: The velocity and position of the particles are updated.

Step 6: If the termination conditions are not met, then proceed to Step 2.

The flowchart of the PSO algorithm is presented in Figure 3.

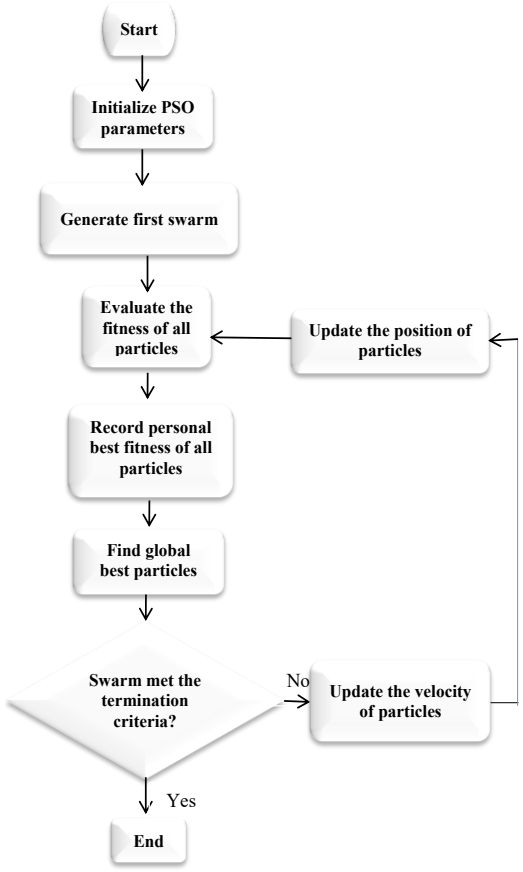

Fig. 3. Flowchart of the standard PSO algorithm.

In the flock of birds foraging model, the each individual can be as a particle regarded and the flock of birds can be as a group of particles regarded. An individual particle is composed of three vectors. Suppose that in a D-dimensional target search space with $\vec{x}_i = (x_{i1}, x_{i2}, ..., x_{iD})$. Then the best position that it has individually found as $\vec{p}_i = (p_{i1}, p_{i2}, ..., p_{iD})$ and its velocity as $\vec{v}_i = (v_{i1}, v_{i2}, ..., v_{iD})$.

Particles are originally initialized in a manner uniform random throughout the search space and the velocity are randomly initialized. The particles are move throughout the search space in according with a fairly simple set of update equations. The algorithm updates the entire swarm at each time step by updating the position of each particle in every dimension and the velocity by using the following equations [15]:

$$v_{id} = v_{id} + c\,\epsilon_1 (p_{id} - x_{id}) + c\epsilon_2 (p_{gd} - x_{id}), \quad (5)$$

$$x_{id} = x_{id} + v_{id}, \quad (6)$$

Where $c = 2.0$ is a constant value, and ϵ_1 and ϵ_2 are independent random numbers. For each individual dimension, $d = 1$ to D is uniquely generated at every update and in the best position found by any neighbor of the particle.

The maximum value of particle velocity (V_{max}) is a parameter that prevents the system from entering an explosive state by limiting the velocity of all the particles. Thus, particle position increases rapidly and approaches infinity.

The PSO algorithm is summarized as follows.

for each time step t, do,
for each particle i in the swarm, do,
update position $\vec{x}_t$ by using Equations (1) and (2),
calculate particle fitness $f(\vec{x}_t)$,
update $\vec{p}_i$, $\vec{p}_g$,
end for,
end fo,r

IV. PROPOSED OPTIMAL BF ALGORITHM

For the AWGN channel, consider the real number sequence $r = [r_0, r_1, ..., r_{N-1}]$ of the channel output. For each output symbol r_n, its absolute value $| r_n |$ can be used as the reliability measure of the hard decision z_n because the absolute value of the log-likelihood ratio associated with this hard decision $| \ln(\rho(r_n | c_n = 1)/\rho (r_n | c_n = 0)) |$ is proportional to $| r_n |$.$| r_n |$ the greater the value of z_n, the higher the reliability of the hard decision, the smaller the possibility of error, the smaller the value of $| r_n |$, the lower the reliability of the hard decision of z_n, and the greater the possibility of error. Therefore, an improved BF algorithm based on two thresholds is proposed in accordance with a preset real number. The bits in the hard decision sequence are divided into two groups. The first group is the bit with the channel output amplitude, i.e., the bit group with the higher error probability. The first group is the bit whose channel output amplitude with greater error probability. The second group is composed of bits whose channel output amplitude with less error probability. In the iterative process, the improved BF algorithm uses two decoding thresholds T_1 and T_2 to determine hard decisions whether to flip the features in the sequence. In order to better correct the bits with greater error probability in the first group, a smaller threshold T_1 is used to determine whether to flip, and at the same time, to avoid falsely flipping the more reliable bits in the first group, use a larger threshold T_2 to determine whether to bits flip.

The steps of the improved BF algorithm are as follows.

1) The bits in the hard decision sequence are divided into two groups in accordance with the amplitude of the channel output sequence.

2) The syndrome s from Equation (1) is calculated. If $s = 0$, then stop the iteration and display the decoding success; otherwise, proceed to Step (3).

3) For each bit z_n in the hard decision sequence, the number f_n of the check equation with syndrome 1 is calculated using Equation (2). If f_n is greater than the threshold that corresponds to the bit, then z_n is flipped.

4) Steps (2) and (3) are repeated until the decoding is successful or the maximum number of iterations is reached and the decoding fails.

In summary, the improved BF algorithm only requires N real number comparison operations before iteration, and the iteration process is complicated. The degree is the same as that in the standard BF algorithm, and it only involves logical operations. Multiple bits can be flipped in each iteration, and thus, the improved BF algorithm has faster convergence speed. Moreover, the use of two decoding thresholds can reduce errors when flipping bits in each iteration, therefore, the improved BF algorithm exhibits excellent decoding performance. The quantity of real number operations is extremely small; hence, the improved algorithm's decoding delay is very short and its decoding speed is very fast.

V. SIMULATION RESULTS

The improved BF algorithm for LDPC codes is based on error checking. The algorithm uses a method that involves setting decision thresholds when decoding. This method reduces the number of iterations in the decoding process along with decoding complexity. The simulation results show that LDPC codes for ASA have fewer iterations and exhibit better performance than other BF algorithms.

Figures 4 and 5 present the LDPC codes (255, 175) with a column weight of 16. When the signal-to-noise ratio (SNR) is 4 dB and the number of iterations I = 500, the different values of T1 that correspond to the decoding performance of T2 are set as 12 and 16, respectively. Through the simulation results presented in Figure 4 and the comparison between the PSO and ASA results, T1 = 8.3 and 8.15, respectively. When T1 is determined, the best threshold for the second group can also be determined and described by simulating the effect of the T2 value on the decoding performance illustrated in Figure 5. Therefore, T2 = 12.3 and T2 = 11.9, respectively.

The maximum iteration numbers of PSO and ASA are approaching 360 and 290 for the best T1 and T2 values for each iteration, respectively. The final threshold approaches close to the base after these iteration numbers. ASA is effective at finding the minimum threshold of the base matrix of LDPC codes during the given maximum number of iterations. Consequently, the best thresholds can be determined.

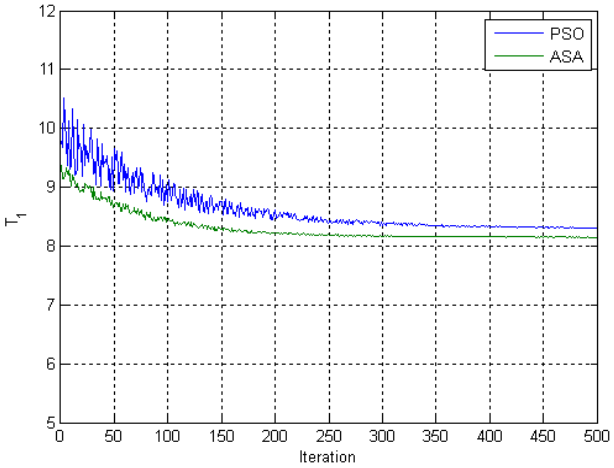

Fig. 4. Decoding thresholds T1 of LDPC codes under varying numbers of iterations

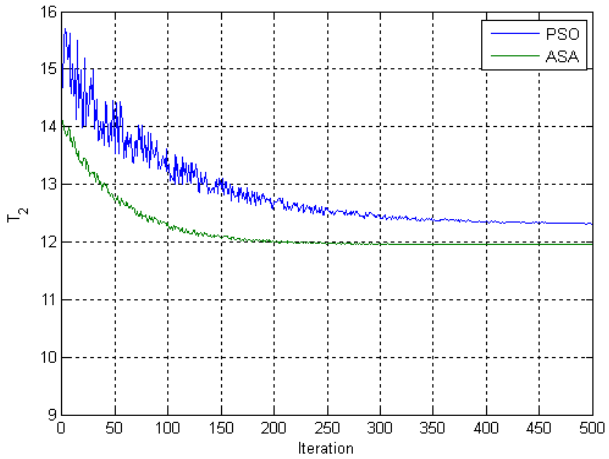

Fig. 5. Decoding thresholds T2 of LDPC codes under varying numbers of iterations

The modified LDPC scheme can change the information for threshold decoding and is expected to provide better performance in the presence of the bit error rate (BER) along with the iteration numbers shown in Figure 6.

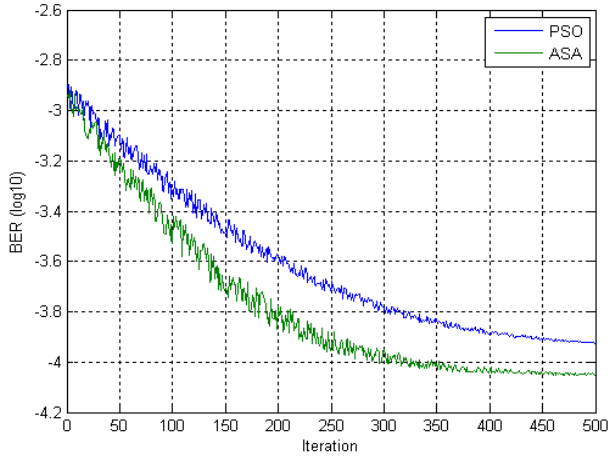

Fig. 6 BER of LDPC codes under varying numbers of iterations.

The error correction performance of LDPC codes (255, 175) under various BF algorithms is presented in Figure 7.

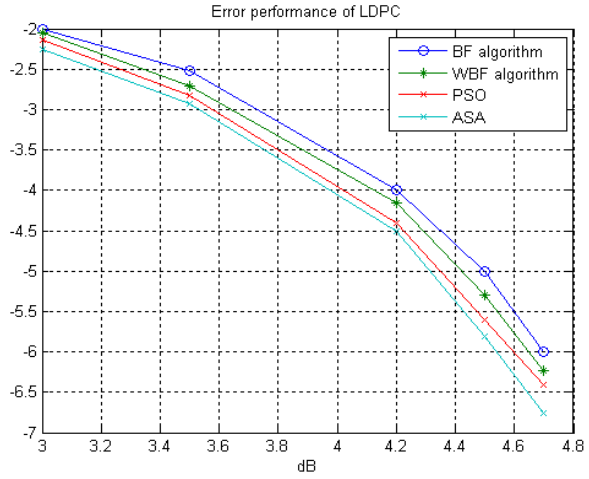

Fig. 7. Error performance of LDPC codes (255, 175) under BF algorithm, WBF algorithm, PSO, and ASA

When the BER is approximately 10−4, the PSO algorithm exhibits more improvement than the BF and WBF algorithms. Meanwhile, the improved ASA can obtain a coding gain of nearly 0.1 dB compared with PSO, i.e., an evident difference for all SNRs. Algorithm performance is slightly better, and the performance improvement of the decoding algorithm exhibits higher practical application values. From the previous analysis, decoding complexity is extremely low for ASA and its decoding delay is very short. The decoding convergence iteration times for PSO and ASA of the code are set as 360 and 290 for BERs of 1.122×10−4 and 8.766×10−5, respectively. ASA is highly suitable for communication systems with high real-time requirements, as shown in Table I.

TABLE I
NUMBER OF ITERATIONS REQUIRED FOR T1 AND T2

	Number of iterations	(T_1, T_2)	BER
PSO	360	(8.3, 12.3)	1.122×10^{-4}
ASA	290	(8.15, 11.9)	8.766×10^{-5}

VI. CONCLUSION

With the growth in diversity and the high-speed transmission rate of communication services in modern society, channel error correction coding and decoding technologies are playing an increasingly important role in digital communication. However, the generation of error code platforms limits the superior performance of LDPC code applications. Code algorithms play a crucial role in the error platform. The improvement of decoding algorithms is an important research direction toward reducing the error platform. The optimization of decoding algorithms plays an essential role in reducing the error platform to a certain extent and exploring better decoding algorithms.

Accurate evaluations are crucial for the search for thresholds of LDPC codes that are found or lie close to the Shannon limit. Such search will be considerably easier by using

optimization methods to accurately calculate the expected value of a message transmitted from a check node.

The search for thresholds is typically implemented by using intelligent algorithms due to the high dimensions of the solution. These algorithms generally require a pool of solutions, and a fitness function is applied to evaluate the goodness of the solutions. The best group is selected and generated.

In the current study, an optimization method (i.e., ASA) is applied to find the optimum degree via the developed capability improvement for the BF of LDPC codes. Compared with PSO, ASA is extremely close to the complex decoding algorithms for solving problems with high dimensions. To solve a continuous optimization problem and achieve highly rapid and accurate results, ASA exhibits good capability, robustness, and proficiency. Through the introduction of ASA, we achieve better trade-off between performance and complexity. We determine that ASA demonstrates considerably better performance than the WBF and the normalized PSO algorithms.

ACKNOWLEDGEMENTS

We are very grateful to experts for their appropriate and constructive suggestions to improve this template.

REFERENCES

[1] I. B. Djordjevic, “LDPC-coded MIMO optical communication over the atmospheric turbulence channel using Q-ary pulse-position modulation,” Opt. Express, vol. 15, no. 16, p. 10026, 2007. https://doi.org/10.1364/OE.15.010026

[2] S. Y. Chung, G. David Forney, T. J. Richardson, and R. Urbanke, “On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit,” IEEE Commun. Lett., vol. 5, no. 2, pp. 58–60, Feb. 2001. https://doi.org/10.1109/4234.905935

[3] J. Meng, D. Zhao, H. Tian, and L. Zhang, “Sum of the magnitude for hard decision decoding algorithm based on loop update detection,” Sensors (Switzerland), vol. 18, no. 1, Jan. 2018. https://doi.org/10.3390/s18010236

[4] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” Electron. Lett., vol. 32, no. 18, p. 1645, 1996. https://doi.org/10.1049/el:19961141

5] R. G. Gallager, “Low-Density Parity-Check Codes,” IRE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, 1962. https://doi.org/10.1109/TIT.1962.1057683

[6] S. H. Kang and I. C. Park, “Loosely coupled memory-based edcoding architecture for low density parity check codes,” IEEE Trans. Circuits Syst. I Regul. Pap., vol. 53, no. 5, pp. 1045–1056, May 2006. https://doi.org/10.1109/TCSI.2005.862181

[7] N. Miladinovic and M. P. C. Fossorier, “Improved bit-flipping decoding of low-density parity-check codes,” IEEE Trans. Inf. Theory, vol. 51, no. 4, pp. 1594–1606, Apr. 2005. https://doi.org/10.1109/TIT.2005.844095

[8] S. Haddadi, M. Farhang, and M. Derakhtian, “Low-complexity decoding of LDPC codes using reduced-set WBF-based algorithms,” Eurasip J. Wirel. Commun. Netw., vol. 2020, no. 1, p. 180, Dec. 2020. https://doi.org/10.1186/s13638-020-01791-5

[9] Ali Jasim Ghaffoori & Wameedh Riyadh Abdul-Adheem, “Control of carbon nanotube cantilever vibrator for nano-antenna applications,” Cogent Engineering., vol. 6, Issue. 1, pp.1–12, 2019. https://doi.org/10.1080/23311916.2019.1710428

[10] Ali Jasim Ghaffoori, “PAPR REDUCTION IN OFDM SYSTEM USING ADAPTIVE HYBRID TECHNIQUE,” in IOP Conference Series: Materials Science and Engineering, ICSET 2019, vol. 518, Issue 5, pp. 1–7, 2019.

[11] B. Attaran, A. Ghanbarzadeh, and S. Moradi, “A novel evolutionary optimization algorithm inspired in the intelligent behaviour of the hunter spider,” Int. J. Comput. Math., 2020. https://doi.org/10.1080/00207160.2020.1775820

[12] Z. He, S. Roy, and P. Fortier, “Powerful LDPC codes for broadband wireless networks: High-performance code construction and high-speed encoder/decoder design,” in Conference Proceedings of the International Symposium on Signals, Systems and Electronics, 2007, pp. 173–176. https://doi.org/10.1109/ISSSE.2007.4294441

[13] J. Kennedy, J. Kennedy, and R. Eberhart, “Particle swarm optimization,” pp. 4--1942, 1995.

[14] A. Othman and H. Gabbar, “Enhanced Microgrid Dynamic Performance Using a Modulated Power Filter Based on Enhanced Bacterial Foraging Optimization,” Energies, vol. 10, no. 6, p. 776, Jun. 2017. https://doi.org/10.3390/en10060776

[15] D. Bratton and J. Kennedy, “Defining a standard for particle swarm optimization,” in Proceedings of the 2007 IEEE Swarm Intelligence Symposium, SIS 2007, 2007, pp. 120–127. https://doi.org/10.1109/SIS.2007.368035