

Research Article

A Generalization Belief Propagation Decoding Algorithm for Polar Codes Based on Particle Swarm Optimization

Yingxian Zhang,¹ Aijun Liu,¹ Xiaofei Pan,¹ Shi He,² and Chao Gong¹

¹ Key Laboratory of Military Satellite Communications, College of Communications Engineering, PLA University of Science and Technology, Nanjing 210007, China

² Unit 75706, PLA, Guangzhou 510000, China

Correspondence should be addressed to Yingxian Zhang; jenus.xyz@gmail.com

Received 31 December 2013; Accepted 21 April 2014; Published 13 May 2014

Academic Editor: Balaji Raghavan

Copyright © 2014 Yingxian Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a generalization belief propagation (BP) decoding algorithm based on particle swarm optimization (PSO) to improve the performance of the polar codes. Through the analysis of the existing BP decoding algorithm, we first introduce a probability modifying factor to each node of the BP decoder, so as to enhance the error correcting capacity of the decoding. Then, we generalize the BP decoding algorithm based on these modifying factors and drive the probability update equations for the proposed decoding. Based on the new probability update equations, we show the intrinsic relationship of the existing decoding algorithms. Finally, in order to achieve the best performance, we formulate an optimization problem to find the optimal probability modifying factors for the proposed decoding algorithm. Furthermore, a method based on the modified PSO algorithm is also introduced to solve that optimization problem. Numerical results show that the proposed generalization BP decoding algorithm achieves better performance than that of the existing BP decoding, which suggests the effectiveness of the proposed decoding algorithm.

1. Introduction

Since the ability of achieving Shannon capacity and their low encoding and decoding complexity, the polar codes have received much attention in the research field of error-correcting coding recently [1–17]. While compared to some existing coding schemes, such as low-density parity-check (LDPC) and turbo codes, the performance of the polar codes in the finite length regime is disappointing [2, 3]. Hence, motivated by this observation, researchers have proposed many decoding algorithms to improve the performance of the polar codes [4–17].

Based on the successive-cancellation (SC) decoding algorithm proposed by Arıkan [1], authors of [4, 5] had introduced a list successive-cancellation (SCL) decoding algorithm with consideration of L SC decoding paths, where the results showed that performance of SCL was very close to that of maximum-likelihood (ML) decoding. Then, to decrease the time complexity of the SCL, another decoding algorithm

derived from SC called stack successive-cancellation (SCS) was proposed in [6]. Furthermore, it was proven in [7] that, with cyclic redundancy check (CRC) aided, SCL even outperformed more than some turbo codes. However, due to the serial processing nature of the SC, all the algorithms in [4–7] would suffer a low decoding throughput and high latency. Therefore, some improved versions of SC were further proposed with the explicit aim to increase throughput and reduce the latency without sacrificing error-rate performance, such as simplified successive-cancellation (SSC) [8], maximum-likelihood SSC (ML-SSC) [9], and repetition single parity check ML-SSC (RSM-SSC) [10, 11]. Besides from those SC based algorithms, researchers had also investigated some other algorithms more in parallel, one typical representation of which was the belief propagation (BP) [12] decoding algorithm. With the factor graph representation of polar codes [13], authors in [14, 15] showed that BP polar decoding has particular advantages with respect to the decoding throughput, while the performance was better than that of the

SC and some improved SC decoding. What is more, with the minimal stopping set optimized, results of [16, 17] had shown that the error floor performance of polar codes was superior to that of LDPC codes.

Indeed, all the decoding algorithms in [4–17] can improve the performance of polar codes to a certain degree. However, as the capacity achieving coding scheme, the results of those algorithms is still limited. Hence, in this paper, we propose a generalization belief propagation (BP) decoding algorithm based on particle swarm optimization (PSO) to improve the performance of the polar codes with the finite length. Based on the analysis results of the existing BP decoding algorithm, we first show that the error-correcting capacity of the BP decoding algorithm is important for performance of the decoding. Motivated by that observation, we introduce a probability modifying factor to each node of the BP decoder to enhance the reliability of the probability updated. Then, we generalize the BP decoding algorithm on the basis of these modifying factors and further drive the probability update equations for the proposed generalization BP decoding. Finally, in order to achieve the best performance, we formulate an optimization problem to find the optimal probability modifying factors for the proposed generalization BP decoding. Furthermore, a method based on the modified PSO algorithm is also introduced to solve the optimization problem. The main contributions of this paper can be summarized as follows.

- (i) A generalization belief propagation (BP) decoding algorithm based on probability modifying factor is introduced. Furthermore, as to improve the performance of the proposed decoding, a BP decoding algorithm optimization problem is formulated.
- (ii) A method based on the modified PSO algorithm is introduced to solve the optimization problem.

The finding of this paper suggests that with the probability modifying factor, the error-correcting capacity of the BP decoding can be enhanced, and the performance of the polar codes can be improved correspondingly, which is finally proven by the simulation results.

The remainder of this paper is organized as follows. In Section 2, we explain some notations and introduce some background on the polar codes. And in Section 3, the generalization belief propagation (BP) decoding algorithm based on the probability modifying factors is described in detail. In Section 4, we formulate and solve the optimization problem of finding the probability modifying factors. And Section 5 provides the simulation results for the complexity and bit error performance of the proposed decoding. Finally, we make some conclusions in Section 6.

2. Preliminary

2.1. Notations. In this paper, the blackboard bold letters, such as \mathbb{X} , denote the sets, and $|\mathbb{X}|$ denotes the number of elements in \mathbb{X} . The notation u_0^{N-1} denotes an N -dimensional vector $(u_0, u_1, \dots, u_{N-1})$, and u_i^j indicates a subvector $(u_i, u_{i+1}, \dots, u_{j-1}, u_j)$ of u_0^{N-1} , $0 \leq i, j \leq N - 1$.

When $i > j$, u_i^j is an empty vector. Further, given a vector set \cup , vector \vec{u}_i is the i th element of \cup .

The matrices in this paper are denoted by bold letters. The subscript of a matrix indicates its size, for example, $\mathbf{A}_{N \times M}$ represents an $N \times M$ matrix \mathbf{A} . Specifically, the square matrices are written as \mathbf{A}_N , size of which is $N \times N$, and \mathbf{A}_N^{-1} is the inverse of \mathbf{A}_N . Furthermore, the Kronecker product of two matrices \mathbf{A} and \mathbf{B} is written as $\mathbf{A} \otimes \mathbf{B}$, and the n th Kronecker power of \mathbf{A} is $\mathbf{A}^{\otimes n}$.

During the procedure of the encoding and decoding, we denote the intermediate node by $v_I(i, j)$ or $v_O(i, j)$, $0 \leq i \leq n$, $0 \leq j \leq N - 1$, where $N = 2^n$ is the code length. Besides, we also indicate the probability messages of the intermediate node $v_I(i, j)$ as $p_{I(i, j)}$, where the probability of $v_I(i, j)$ being equal to 0 or 1 is $p_{I(i, j)}(0)$ or $p_{I(i, j)}(1)$. Similarly, the probability messages of $v_O(i, j)$ are $p_{O(i, j)}(0)$ and $p_{O(i, j)}(1)$.

Throughout this paper, “ \oplus ” denotes the modulo-two sum, and “ $\sum_{i=0}^M \oplus x_i$ ” means “ $x_0 \oplus x_1 \oplus \dots \oplus x_M$ ”. While the operators “ \otimes ” and “ \odot ” are defined as $p_x \otimes p_y = p_{(x \oplus y)}$ and $p_x \odot p_y = p_x * p_y$, respectively, $x, y \in \{0, 1\}$.

2.2. Background of Polar Codes. A polar coding scheme can be uniquely defined by three parameters: block-length $N = 2^n$, code rate $R = K/N$, and an information set $\mathbb{I} \subset \mathbb{N} = \{0, 1, \dots, N - 1\}$, where $K = |\mathbb{I}|$. With these three parameters, a source binary vector u_0^{N-1} consisting of K information bits and $N - K$ frozen bits can be mapped a codeword x_0^{N-1} by a linear matrix $\mathbf{G}_N = \mathbf{B}_N \mathbf{F}_2^{\otimes n}$, where $\mathbf{F}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, \mathbf{B}_N is a bit-reversal permutation matrix defined in [1], and $x_0^{N-1} = u_0^{N-1} \mathbf{G}_N$.

In practice, the construction procedure of a polar code can be divided into $n = \log_2 N$ stages, as shown in Figure 1(a), where the circle nodes in the leftmost column are the input nodes of encoder, values of which are equal to binary source vector, that is, $v_I(0, i) = u_i$, and the circle nodes in the rightmost column are the output nodes of encoder, $v_O(n - 1, i) = x_i$. It is also noticed that, each encoding stage has $N/2$, so-called processing units (PU) with a generation matrix of \mathbf{F}_2 , and each PU has two input and two output variable nodes, as shown in Figure 1(b). During the process of polar encoding, based on \mathbf{F}_2 , we have

$$\begin{aligned} v_O(i, 2j) &= v_I(i, 2j) \oplus v_I(i, 2j + 1) \\ v_O(i, 2j + 1) &= v_I(i, 2j + 1), \end{aligned} \quad (1)$$

where $v_I(i, 2j)$ and $v_I(i, 2j + 1)$ are the input nodes of the j th PU in the i th stage, $0 \leq i \leq n - 1$, $0 \leq j \leq N/2 - 1$, and similarly, $v_O(i, 2j)$ and $v_O(i, 2j + 1)$ are the output nodes. After the procedure of the polar encoding, all the bits in the codeword x_0^{N-1} are passed to the N -channels, which consisted of N independent channels of W , with a transition probability of $W(y_i | x_i)$, where y_i is i th element of the received vector y_0^{N-1} . Then, the decoder of the receiver will output the estimated codeword \hat{x}_0^{N-1} and the estimated source binary vector \hat{u}_0^{N-1} with different decoding algorithms [4–17].

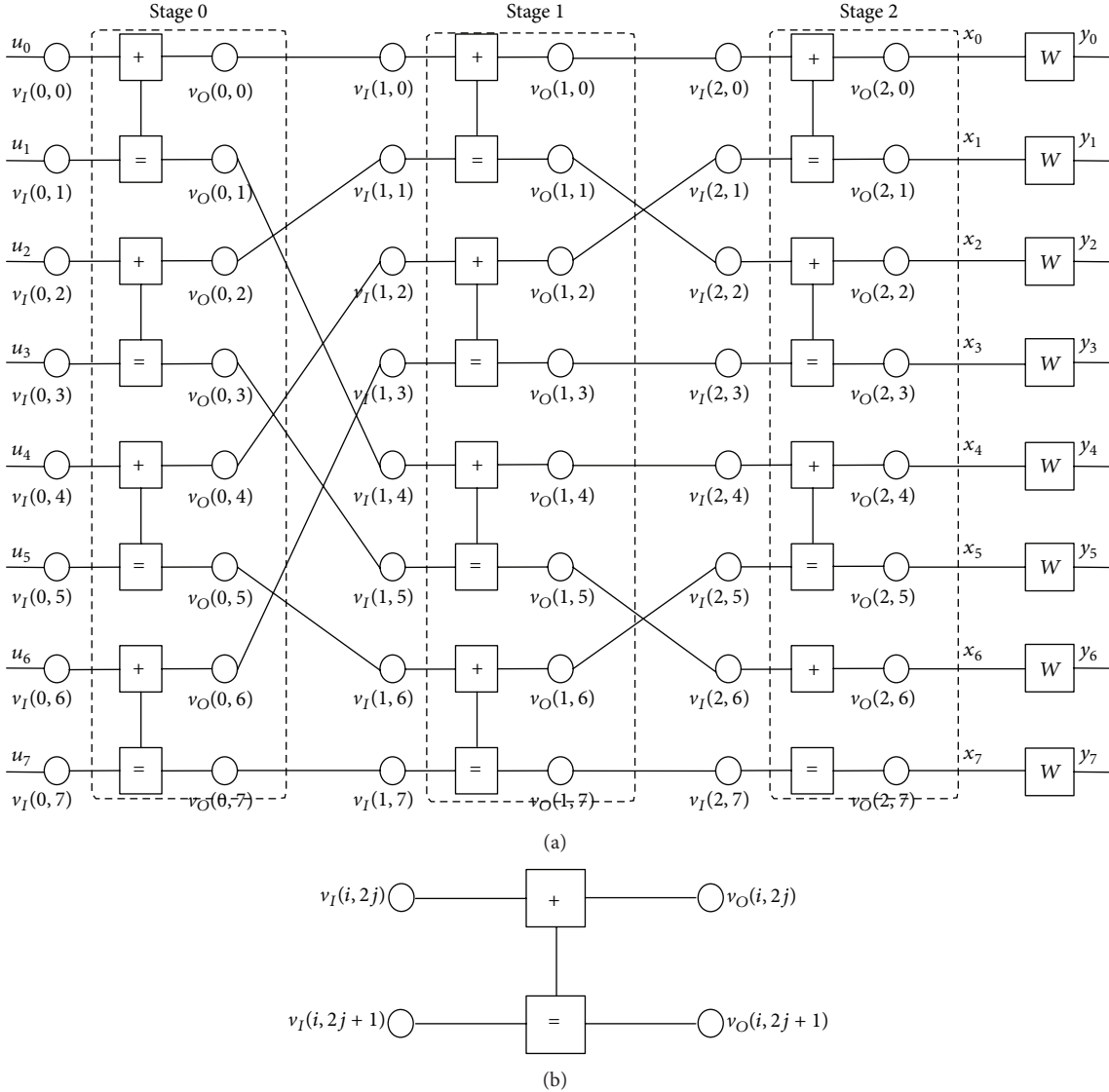


FIGURE 1: (a) Construction procedure of polar codes with $N = 8$. (b) Diagram of PU in code construction.

3. BP Decoding for Polar Codes

In this section, we will generalize the BP decoding algorithm for polar codes with the analysis of the existing polar BP decoding.

3.1. Existing BP Decoding. The basic BP process unit (BP-PU) of polar codes is shown in Figure 2, where $p_{IL(i,2j)}^t$ and $p_{IR(i,2j+1)}^t$ are the right-to-left probability messages passed to $v_I(i, 2j)$ and $v_I(i, 2j + 1)$; t is the iteration number. $p_{IR(i,2j)}^{t-1}$ and $p_{IR(i,2j+1)}^{t-1}$ are the left-to-right probability messages passed from the nodes. $p_{OL(i,2j)}^{t-1}$ and $p_{OL(i,2j+1)}^{t-1}$ are the right-to-left probability messages passed from $v_O(i, 2j)$ and $v_O(i, 2j + 1)$ and $p_{OR(i,2j)}^t$ and $p_{OR(i,2j+1)}^t$ are the left-to-right probability messages passed to the nodes. According to the formula of total probability, there has

$$p_{IL(i,j)}^t(0) + p_{IL(i,j)}^t(1) = 1,$$

$$p_{IR(i,j)}^t(0) + p_{IR(i,j)}^t(1) = 1,$$

$$p_{OL(i,j)}^t(0) + p_{OL(i,j)}^t(1) = 1,$$

$$p_{OR(i,j)}^t(0) + p_{OR(i,j)}^t(1) = 1.$$

(2)

Furthermore, based on the transition probability of the channel W , that is, $W(y_i | x_i)$, and the received vector y_0^{N-1} , we have

$$p_{OL(n-1,i)}^0(0) = W(y_i | x_i = 0) \frac{P(x_i = 0)}{P(y_i)}$$

$$p_{OL(n-1,i)}^0(1) = W(y_i | x_i = 1) \frac{P(x_i = 1)}{P(y_i)},$$

(3)

where $P(x_i)$ and $P(y_i)$ are the probability distribution functions of coded bit x_i and received signal sample y_i . In the BP

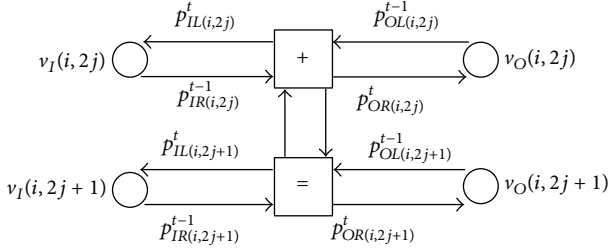


FIGURE 2: The BP decoding algorithm for polar codes is based on the factor graph representation of the codes [13]. In the algorithm, each node $v_I(i, j)$ or $v_O(i, j)$ is associated with two types of probability messages: left-to-right probability messages and right-to-left probability messages. Both of messages are propagated and updated iteratively between adjacent nodes in the procedure of BP. And the messages updated schedule is the same as that of [14]; that is, messages will firstly propagate from the rightmost nodes to the leftmost nodes. After arriving at the leftmost nodes, the course will be reversed, and messages propagation will move toward the rightmost nodes. This procedure makes one *round iteration* of BP. Diagram of BP-PU.

decoding algorithm [13–17], the messages updated equations are given by

$$\begin{aligned}
 P_{IL}^t(i, 2j) &= P_{OL}^{t-1}(i, 2j) \otimes \left(P_{OL}^{t-1}(i, 2j+1) \odot P_{IR}^{t-1}(i, 2j+1) \right), \\
 P_{IL}^t(i, 2j+1) &= \left(P_{IR}^{t-1}(i, 2j) \otimes P_{OL}^{t-1}(i, 2j) \right) \odot P_{OL}^{t-1}(i, 2j+1), \\
 P_{OR}^t(i, 2j) &= P_{IR}^{t-1}(i, 2j) \otimes \left(P_{IR}^{t-1}(i, 2j+1) \odot P_{OL}^{t-1}(i, 2j+1) \right), \\
 P_{OR}^t(i, 2j+1) &= \left(P_{IR}^{t-1}(i, 2j) \otimes P_{OL}^{t-1}(i, 2j) \right) \odot P_{IR}^{t-1}(i, 2j+1).
 \end{aligned} \quad (4)$$

Based on (4), when the decoding reaches maximum iteration number (mlter), BP decoder output the decoded vector $\hat{\mathbf{u}} = (\hat{u}_0, \dots, \hat{u}_{N-1})$, each elements of which depended on

$$\hat{u}_i = \begin{cases} 0, & P_{IL(0,i)}^{\text{mlter}}(0) > P_{IL(0,i)}^{\text{mlter}}(1) \\ 1, & \text{otherwise.} \end{cases} \quad (5)$$

3.2. Analysis of BP Decoder. It is noticed from [13–17], based on the BP decoding, the performance of polar codes can be improved to a certain degree. However, as the capacity achieving coding scheme, the results of those algorithms is still disappointing. Hence, we cannot help wondering why the performance of the polar codes with finite length is inferior to that of LDPC codes [16, 17], and how we can improve it. To answer the questions, we need to make a further analysis of the existing BP decoder.

In fact, it is noticed from the factor graph of polar codes in [13] that the degree of the check or variable nodes in the polar BP decoder is 2 or 3. While the average degree of the LDPC codes is usually greater than 3 [12, 18], which means that the error-correcting capacity of the polar BP decoding will be weakened. Hence, the performance of the polar codes is inferior to that of LDPC codes with the same length [18]. As shown in Figure 3, two Tanner graphs with different

degree distributions are given. We assume that there exists an error variable node v_1 in a certain iteration. Compared to the check nodes c_0 and c_1 in Figure 3(a), the influence of error probability messages from v_1 to c_0 and c_1 in Figure 3(a) is weak, because with a greater degree number, the proportion of the error probability messages in the input probability messages of c_0 and c_1 will be reduced. What is more, the convergence of BP decoding with a greater average degree will be quicker, which results in less computational iteration number, as shown in [12–18]. Therefore, to improve the performance of polar BP decoding, it is important to enhance the error-correcting capacity of the algorithm.

3.3. Generalization of BP Decoding. Motivated by aforementioned observation, in this subsection, we will introduce a generalization framework of BP decoder for polar codes. In order to improve the error-correcting capacity of polar BP decoding, twelve probability messages modifying factors are introduced to the nodes of the BP-PU shown in Figure 2. When the probability messages of a node of the BP-PU is updated, three probability messages modifying factors are added to other three nodes, respectively.

As illustrated in Figure 4, $\gamma_{L0}^t(i, j), \dots, \gamma_{L5}^t(i, j)$ and $\gamma_{R0}^t(i, j), \dots, \gamma_{R5}^t(i, j)$ are the probability messages modifying factors of the j th BP-PU in i th stage. In this case, each equation in (4) will be correspondingly added with three variables, for example, the first equation of (4) is added with $\gamma_{L0}^t(i, j), \gamma_{L1}^t(i, j)$, and $\gamma_{L2}^t(i, j)$. Besides, based on the new construction of BP-PU, we further define the result of each nodes' probability messages calculation as a function with respect to these three variables and the probability messages of other three nodes; hence, we can get the new messages update equations as (6). Consider

$$\begin{aligned}
 P_{IL}^t(i, 2j) &= f_{(i,j)}^0 \left(P_{OL}^{t-1}(i, 2j), P_{OL}^{t-1}(i, 2j+1), P_{IR}^{t-1}(i, 2j+1), \right. \\
 &\quad \left. \gamma_{L0}^t(i, j), \gamma_{L1}^t(i, j), \gamma_{L2}^t(i, j) \right), \\
 P_{IL}^t(i, 2j+1) &= f_{(i,j)}^1 \left(P_{OL}^{t-1}(i, 2j+1), P_{OL}^{t-1}(i, 2j), P_{IR}^{t-1}(i, 2j), \right. \\
 &\quad \left. \gamma_{L3}^t(i, j), \gamma_{L4}^t(i, j), \gamma_{L5}^t(i, j) \right), \\
 P_{OR}^t(i, 2j) &= f_{(i,j)}^2 \left(P_{IR}^{t-1}(i, 2j), P_{IR}^{t-1}(i, 2j+1), P_{OL}^{t-1}(i, 2j+1), \right. \\
 &\quad \left. \gamma_{R0}^t(i, j), \gamma_{R1}^t(i, j), \gamma_{R2}^t(i, j) \right), \\
 P_{OR}^t(i, 2j+1) &= f_{(i,j)}^3 \left(P_{IR}^{t-1}(i, 2j+1), P_{IR}^{t-1}(i, 2j), P_{OL}^{t-1}(i, 2j), \right. \\
 &\quad \left. h \gamma_{R3}^t(i, j), \gamma_{R4}^t(i, j), \gamma_{R5}^t(i, j) \right).
 \end{aligned} \quad (6)$$

In (6), $f_{(i,j)}^0(\varphi_0^5), f_{(i,j)}^1(\varphi_0^5), f_{(i,j)}^2(\varphi_0^5)$, and $f_{(i,j)}^3(\varphi_0^5)$ are the four probability messages update functions of the j th BP-PU in i th stage, and $\varphi_0, \dots, \varphi_5$ are six probability messages variables. Based on (6), we can easily get a new expression of

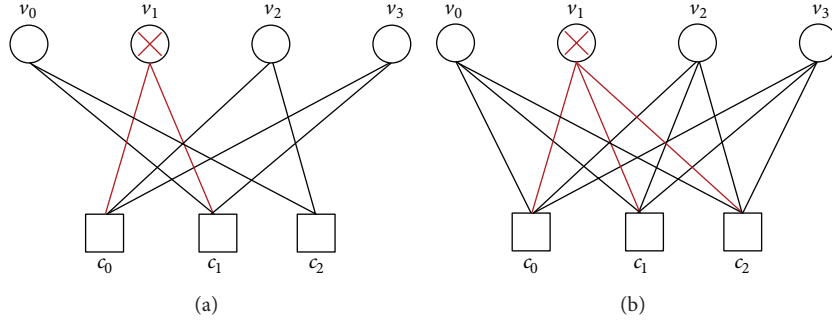


FIGURE 3: (a) Tanner graph with average degree less than 3. (b) Tanner graph with average degree more than 3.

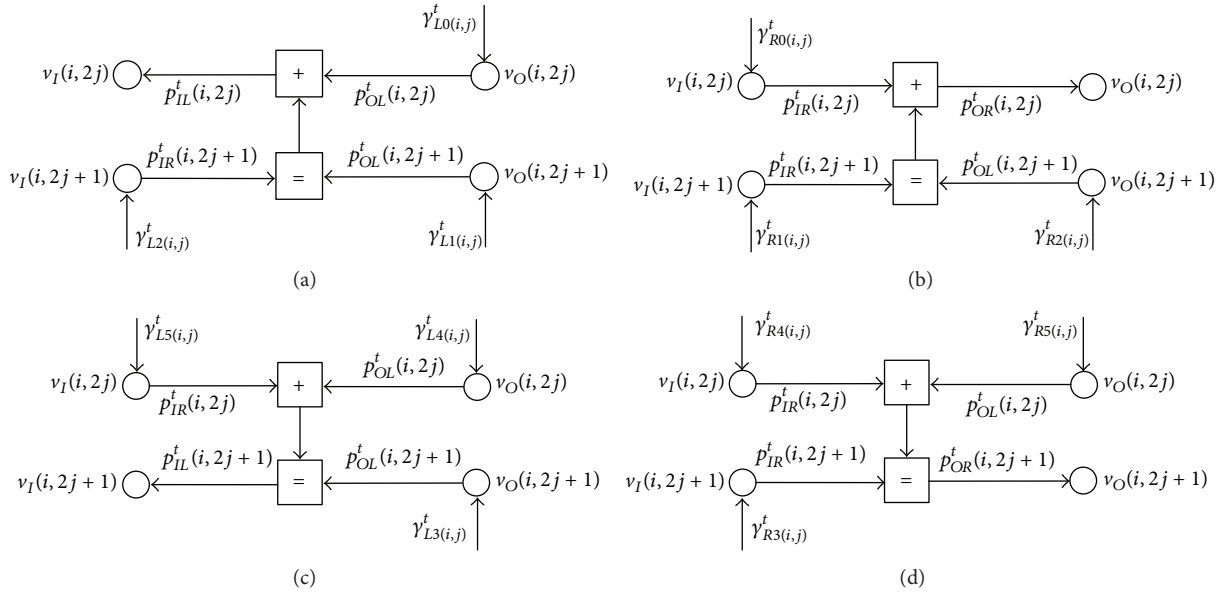


FIGURE 4: Diagrams of the new BP-PU.

(4) defined as the functions of $f_{(i,j)}^0(\varphi_0^5)$, $f_{(i,j)}^1(\varphi_0^5)$, $f_{(i,j)}^2(\varphi_0^5)$, and $f_{(i,j)}^3(\varphi_0^5)$; that is,

$$\begin{aligned}
 f_{(i,j)}^0(\varphi_0^5) &= \varphi_0 \otimes (\varphi_1 \otimes \varphi_2), \\
 f_{(i,j)}^1(\varphi_0^5) &= \varphi_0 \odot (\varphi_1 \otimes \varphi_2), \\
 f_{(i,j)}^2(\varphi_0^5) &= \varphi_0 \otimes (\varphi_1 \odot \varphi_2), \\
 f_{(i,j)}^3(\varphi_0^5) &= \varphi_0 \odot (\varphi_1 \otimes \varphi_2).
 \end{aligned} \tag{7}$$

That is to say, the existing BP decoding algorithm is a special case of the generalization BP decoding proposed in this work. Furthermore, with the four probability messages update functions of

$$f_{(i,j)}^0(\varphi_0^5) = \varphi_0 \otimes \varphi_1,$$

$$f_{(i,j)}^1(\varphi_0^5) = \varphi_0 \odot (\varphi_1 \otimes \varphi_2),$$

$$f_{(i,j)}^2(\varphi_0^5) = \varphi_0 \otimes \varphi_1,$$

$$f_{(i,j)}^3(\varphi_0^5) = \varphi_0;$$

(8)

we can also get the probability messages update equations of the SC based decoding algorithms as shown in [1, 4–11]. Similarly, it is concluded that the SC based decoding algorithms are also the special cases of our generalization BP decoding, which has shown the intrinsic relationship of the SC based decoding and BP decoding algorithms.

Therefore, we can conclude from (6)–(8) that, in order to improve the performance of the polar codes, it is needed to find the optimal functions of $f_{(i,j)}^0(\varphi_0^5)$, $f_{(i,j)}^1(\varphi_0^5)$, $f_{(i,j)}^2(\varphi_0^5)$ and $f_{(i,j)}^3(\varphi_0^5)$, and the optimal parameters of $\gamma_{L0(i,j)}^t, \dots, \gamma_{L5(i,j)}^t$ and $\gamma_{R0(i,j)}^t, \dots, \gamma_{R5(i,j)}^t$, which will be discussed in the next section.

4. BP Decoding Algorithm Optimization

Motivated by the aforementioned conclusion, we now consider how to optimize the generalization BP decoding algorithm so as to improve the performance of the polar codes.

4.1. Messages Update Functions. It is noticed from (4) and (6)–(8) that, in order to enhance the reliability of messages propagation, it is important to determine the appropriate messages update functions for the generalization BP decoding algorithm. However, in general, it is difficult to find the optimal messages update functions; therefore, in this work, the messages update functions are introduced by the analysis of the construction of polar encoding. And based on (1), we have

$$\begin{aligned}
 v_I(i, 2j) &= v_O(i, 2j) \oplus v_O(i, 2j+1) \\
 &= v_O(i, 2j) \oplus v_I(i, 2j+1), \\
 v_I(i, 2j+1) &= v_O(i, 2j+1) = v_O(i, 2j) \oplus v_I(i, 2j), \\
 v_O(i, 2j) &= v_I(i, 2j) \oplus v_I(i, 2j+1) \\
 &= v_I(i, 2j) \oplus v_O(i, 2j+1), \\
 v_O(i, 2j+1) &= v_I(i, 2j+1) = v_I(i, 2j) \oplus v_O(i, 2j).
 \end{aligned} \tag{9}$$

Therefore, based on the error coding theory and (9), we can get the messages update functions as

$$\begin{aligned}
 f_{(i,j)}^0(\varphi_0^5) &= ((\varphi_0 \odot \varphi_3) \otimes (\varphi_1 \odot \varphi_4)) \\
 &\quad \odot ((\varphi_0 \odot \varphi_3) \otimes (\varphi_2 \odot \varphi_5)), \\
 f_{(i,j)}^1(\varphi_0^5) &= (\varphi_0 \odot \varphi_3) \odot ((\varphi_1 \odot \varphi_4) \otimes (\varphi_2 \odot \varphi_5)), \\
 f_{(i,j)}^2(\varphi_0^5) &= ((\varphi_0 \odot \varphi_3) \otimes (\varphi_1 \odot \varphi_4)) \\
 &\quad \odot ((\varphi_0 \odot \varphi_3) \otimes (\varphi_2 \odot \varphi_5)), \\
 f_{(i,j)}^3(\varphi_0^5) &= (\varphi_0 \odot \varphi_3) \odot ((\varphi_1 \odot \varphi_4) \otimes (\varphi_2 \odot \varphi_5)).
 \end{aligned} \tag{10}$$

That is to say, with the probability messages modifying factors, the new probability messages update equations will be written as

$$\begin{aligned}
 P_{IL(i,2j)}^t &= \left((P_{OL(i,2j)}^{t-1} \odot \gamma_{L0(i,j)}^t) \otimes (P_{OL(i,2j+1)}^{t-1} \odot \gamma_{L1(i,j)}^t) \right) \\
 &\quad \odot \left((P_{OL(i,2j)}^{t-1} \odot \gamma_{L0(i,j)}^t) \right) \\
 &\quad \otimes \left(P_{IR(i,2j+1)}^{t-1} \odot \gamma_{L2(i,j)}^t \right), \\
 P_{IL(i,2j+1)}^t &= \left(P_{OL(i,2j+1)}^{t-1} \odot \gamma_{L3(i,j)}^t \right) \\
 &\quad \odot \left((P_{OL(i,2j)}^{t-1} \odot \gamma_{L4(i,j)}^t) \right) \\
 &\quad \otimes \left(P_{IR(i,2j)}^{t-1} \odot \gamma_{L5(i,j)}^t \right),
 \end{aligned}$$

$$\begin{aligned}
 P_{OR(i,2j)}^t &= \left((P_{IR(i,2j)}^{t-1} \odot \gamma_{R0(i,j)}^t) \otimes (P_{IR(i,2j+1)}^{t-1} \odot \gamma_{R1(i,j)}^t) \right) \\
 &\quad \odot \left((P_{IR(i,2j)}^{t-1} \odot \gamma_{R0(i,j)}^t) \right) \\
 &\quad \otimes \left(P_{OL(i,2j+1)}^{t-1} \odot \gamma_{R2(i,j)}^t \right), \\
 P_{OR(i,2j+1)}^t &= \left(P_{IR(i,2j+1)}^{t-1} \odot \gamma_{R3(i,j)}^t \right) \\
 &\quad \odot \left((P_{IR(i,2j)}^{t-1} \odot \gamma_{R4(i,j)}^t) \right) \\
 &\quad \otimes \left(P_{OL(i,2j)}^{t-1} \odot \gamma_{R5(i,j)}^t \right).
 \end{aligned} \tag{11}$$

Next, our task is to determine the optimal probability messages modifying factors, that is, $\gamma_{L0(i,j)}^t, \dots, \gamma_{L5(i,j)}^t$ and $\gamma_{R0(i,j)}^t, \dots, \gamma_{R5(i,j)}^t$, so as to achieve the best performance.

4.2. Probability Modifying Factors Optimization. In practice, due to the input of frozen bits [1], there exists some so-called frozen nodes in the decoder, and values of these nodes are determined and independent of the decoding algorithm. Hence, if the decoding is correct, the probability messages of a frozen node $v_I(i, j)$ or $v_O(i, j)$ (the determined value of frozen nodes is 0) must satisfy the condition of

$$\begin{aligned}
 P_{IL(i,j)}^t(0) &> P_{IL(i,j)}^t(1) \\
 P_{IR(i,j)}^t(0) &> P_{IR(i,j)}^t(1) \\
 &\text{or} \\
 P_{OL(i,j)}^t(0) &> P_{OL(i,j)}^t(1) \\
 P_{OR(i,j)}^t(0) &> P_{OR(i,j)}^t(1).
 \end{aligned} \tag{12}$$

While in practice, due to the noise of received signal, there may exist some frozen nodes unsatisfied the reliability condition. Therefore, in order to improve the accuracy of decoding, it is needed to find the optimal probability messages modifying factors such that all the frozen nodes could always satisfy the conditions of (12) during the procedure of BP decoding.

To achieve above goal, we introduce a pair parameters to each node in the decoder, which are called *left and right reliability degrees*. And for a node $v_I(i, j)$ (similar to $v_O(i, j)$), the left and right reliability degrees are denoted as $\zeta_{IL(i,j)}^t$ and $\zeta_{IR(i,j)}^t$, respectively. Where if $v_I(i, j)$ is a frozen node, we can get $\zeta_{IL(i,j)}^t$ and $\zeta_{IR(i,j)}^t$ by

$$\begin{aligned}
 \zeta_{IL(i,j)}^t &= \frac{P_{IL(i,j)}^t(0)}{P_{IL(i,j)}^t(1)}, \\
 \zeta_{IR(i,j)}^t &= \frac{P_{IR(i,j)}^t(0)}{P_{IR(i,j)}^t(1)};
 \end{aligned} \tag{13}$$

otherwise, the values of $\zeta_{IL(i,j)}^t$ and $\zeta_{IR(i,j)}^t$ will be given by

$$\zeta_{IL(i,j)}^t = \frac{P_{IL(i,j)}^t(0)}{P_{IL(i,j)}^t(1)}, \quad \zeta_{IR(i,j)}^t = \frac{P_{IR(i,j)}^t(0)}{P_{IR(i,j)}^t(1)},$$

$$\frac{P_{IL(i,j)}^t(0)}{P_{IL(i,j)}^t(1)} > 1 \quad (14)$$

$$\zeta_{IL(i,j)}^t = \frac{P_{IL(i,j)}^t(1)}{P_{IL(i,j)}^t(0)}, \quad \zeta_{IR(i,j)}^t = \frac{P_{IR(i,j)}^t(1)}{P_{IR(i,j)}^t(0)},$$

otherwise,

where $\zeta_{IL(i,j)}^t > 1$ and $\zeta_{IR(i,j)}^t > 1$.

The reliability degrees indicate the reliability of the node's decision value, and the larger reliability degree, the higher reliability of that value. Specially, when there is no noise in the channel, the reliability degrees of a node will approach to infinity. Based on previous observations, the problem of getting the optimal probability messages modifying factors can be formulated as an optimization problem to maximize the reliability degrees of all the node in decoder, while the target function is written as

$$\arg \max_{\substack{0 \leq \gamma_{L0(i,k)}^t, \dots, \gamma_{L5(i,k)}^t \leq 1 \\ 0 \leq \gamma_{R0(i,k)}^t, \dots, \gamma_{R5(i,k)}^t \leq 1}} \left\{ \sum_{i=0}^{\log_2 N} \sum_{j=0}^N (\zeta_{IL(i,j)}^t + \zeta_{IR(i,j)}^t) \right\}$$

$$\text{s.t.} \quad \gamma_{L0(i,k)}^t(0) + \gamma_{L0(i,k)}^t(1) = 1$$

$$\vdots$$

$$\gamma_{L5(i,k)}^t(0) + \gamma_{L5(i,k)}^t(1) = 1 \quad (15)$$

$$\gamma_{R0(i,k)}^t(0) + \gamma_{R0(i,k)}^t(1) = 1$$

$$\vdots$$

$$\gamma_{R5(i,k)}^t(0) + \gamma_{R5(i,k)}^t(1) = 1,$$

where $\zeta_{IL(i,j)}^t$ and $\zeta_{IR(i,j)}^t$ are decided by (11) and (13)-(14), and k is the integral value of $j/2$; that is, $k = \lfloor j/2 \rfloor$.

4.3. Problem Solving Based on Modified PSO Algorithm. Based on (15), we have formulated the problem to find the optimal probability modifying factors, and in this subsection, we will consider the solution of the problem so as to get the optimized BP decoding algorithm.

Among the different global optimizers, particle swarm optimization (PSO) has demonstrated its usefulness as an optimizer capable of finding global optimum. The idea of PSO is based on simulation of simplified social models such as birds flocking and fish schooling. Like other stochastic evolutionary algorithms, PSO is independent of mathematical characteristics of objective problems. However, unlike these algorithms, with PSO, each particle has its own unique

memory to "remember" its own best solution. Thus, each particle has its own "idea" of whether it has found the optimum. However, in other algorithms, previous knowledge of the problem will be destroyed once the population changes. PSO has attracted increasing attention because of its simple concept, easy implementation, and quick. These advantages allow it to be applied successfully in a variety of fields, mainly for unconstrained continuous optimization problems [19–26]. Therefore, in this work, we explore the PSO algorithm with certain modification based on particle fitness evaluation to solve the problem (15).

To present our modified PSO algorithm, we first introduce some preliminary definitions and concepts used in the operation of the PSO algorithm, which is described as follows.

(1) Parameters Definitions. We consider a swarm of L particles, $\{\bar{\mathbf{z}}_0, \dots, \bar{\mathbf{z}}_{L-1}\}$, where the position of $\bar{\mathbf{z}}_i = (z_{i0}, z_{i1}, \dots, z_{i(T-1)})$ represents a possible solution point of those probability modifying factors in the problem (15), $0 \leq i \leq L-1$, and T is the number of the probability modifying factors. Each element of $\bar{\mathbf{z}}_i$ is the value of a probability modifying factor; that is, $0 \leq z_{ij} \leq 1$, $0 \leq j \leq T-1$. Based on the work of [19], the position of particle $\bar{\mathbf{z}}_i$ can be updated by

$$\bar{\mathbf{z}}_i^{t+1} = \bar{\mathbf{z}}_i^t + \bar{\mathbf{w}}_i^{t+1}. \quad (16)$$

In (16), the pseudovelocity $\bar{\mathbf{w}}_i^{t+1}$ is calculated as follows:

$$\bar{\mathbf{w}}_i^{t+1} = \omega^t \bar{\mathbf{w}}_i^t + c_1 r_1 (\bar{\mathbf{p}}_i - \bar{\mathbf{z}}_i^t) + c_2 r_2 (\bar{\mathbf{p}}_g - \bar{\mathbf{z}}_i^t), \quad (17)$$

where ω^t is particle inertia, t indicates the iteration number, $\bar{\mathbf{p}}_i$ represents the best ever position of particle i at iteration t , and $\bar{\mathbf{p}}_g$ represents the global best position in the swarm at iteration t . r_1 and r_2 represent uniform random numbers between 0 and 1. It is proven in [19] that, when $c_1 = c_2 = 2$, the products, $c_1 r_1$ or $c_2 r_2$, have a mean of 1.

(2) Fitness Evaluation. In general, with the parameter definitions aforementioned, we can get the global optimal solution of a optimization problem. However, in problem (15), the situation will be different, because during the procedure of the BP decoding for polar codes, a node's probability messages calculated on the basis of the probability modifying factors must satisfy the conditions of (12) or (14). In order to check these conditions during the decoding, we further define two functions as

$$g_1(\bar{\mathbf{z}}_i^t) = \frac{|\mathbb{V}_F^e|}{|\mathbb{V}_F|},$$

$$g_2(\bar{\mathbf{z}}_i^t) = \frac{|\mathbb{V}_I^e|}{|\mathbb{V}_I|}, \quad (18)$$

where \mathbb{V}_F is the frozen nodes set of the decoder, and \mathbb{V}_F^e is the set of frozen nodes whose probability messages based on $\bar{\mathbf{z}}_i^t$ do not satisfy the condition of (12). Similarly, \mathbb{V}_I is the nonfrozen nodes set of the decoder, and \mathbb{V}_I^e is the set of nonfrozen nodes whose left and right reliability degrees based on $\bar{\mathbf{z}}_i^t$ do not satisfy (14). Furthermore, we have $g_1(\bar{\mathbf{z}}_i^t), g_2(\bar{\mathbf{z}}_i^t) \in [0, 1]$.

Input:
 The stopping parameters ϵ_1 and ϵ_2 ;
 The frozen nodes set, \mathbb{V}_F ;
 The probability messages of the input nodes of the decoder, $\{W(y_0 | x_0), \dots, W(y_{N-1} | x_{N-1})\}$;

Output:
 The decoded source binary vector, \hat{u}_0^{N-1} ;

- (1) Initialize the parameters used in the PSO algorithm, including particles initial set $\{\bar{z}_0^0, \dots, \bar{z}_{L-1}^0\}$, the pseudovelocities $\{\bar{w}_0^0, \dots, \bar{w}_{L-1}^0\}$, particle inertia ω^0 , maximum iteration number and c_1, c_2 ;
- (2) **while** (Stopping criterion is not satisfied) **do**
- (3) Update particle pseudovelocity vector \bar{w}_i^{t+1} using (17);
- (4) Update particle position vector \bar{z}_i^{t+1} using (16);
- (5) Calculate the probability messages of each nodes in the decoder using (11);
- (6) **if** $g_1(\bar{z}_i^{t+1}) > \epsilon_1$ and $g_2(\bar{z}_i^{t+1}) > \epsilon_2$ **then**
- (7) Calculate the fitness value h_i^{t+1} using (19);
- (8) **if** $h_i^{t+1} \geq h_i^{best}$ **then**
- (9) $h_i^{best} = h_i^{t+1}, \bar{p}_i = \bar{z}_i^{t+1}$;
- (10) **end if**
- (11) **if** $h_i^{t+1} \geq h_g^{best}$ **then**
- (12) $h_g^{best} = h_i^{t+1}, \bar{p}_g = \bar{z}_i^{t+1}$;
- (13) **end if**
- (14) **end if**
- (15) **end while**
- (16) Output the decoded source binary vector by hard decision of the output nodes of the decoder;
- (17) **return** \mathbb{C} ;

ALGORITHM 1: BP decoding algorithm based on modified PSO optimization.

Based on (18), we can evaluate the fitness of a particle's current position so as to accelerate the finding of the the optimal solution. After the checking of conditions of (12) and (14), we further get the fitness value h_i^t of a particle by

$$h_i^t(\bar{z}_i^t) = \sum_{k=0}^{\log_2 N} \sum_{j=0}^N (\zeta_{IL(k,j)}^t + \zeta_{IR(k,j)}^t). \quad (19)$$

As the summarization of previous description, we now provide the whole procedure of the proposed BP decoding algorithm based on PSO optimization with the form of pseudocode, as shown in Algorithm 1.

It is noticed from Algorithm 1 that the steps 2–15 are the key of the algorithm. The main consideration of these steps is based on the following two observations. On the one hand, if there is no noise in the channel, the BP decoding of the polar codes will be errorless, which means that all the nodes in the decoder will satisfy the condition of (12); that is, $g_1(\bar{z}_i^t)$ and $g_2(\bar{z}_i^t)$ are both equal to 0. However, in practice, this case is impossible to happen due to the noise of channel; hence, we introduce two stopping parameters ϵ_1 and ϵ_2 for $g_1(\bar{z}_i^t)$ and $g_2(\bar{z}_i^t)$. Since the values of the frozen nodes are known to the decoder, we set ϵ_2 to 0. To ensure that the correction of the decoding, ϵ_1 should be as small as possible, such that when a particle \bar{z}_i^t is found to make $g_1(\bar{z}_i^{t+1}) \leq \epsilon_1$ and $g_2(\bar{z}_i^{t+1}) \leq \epsilon_2$ both true, which indicates that stopping criterion is satisfied, the loop of (2)–(15) can be stopped. On the other hand, when there exist multiparticles that make the stopping criterion satisfied, we can choose the one which can

generate maximum fitness value to be our final solution, as shown in steps (11)–(13).

5. Numerical Results

In this section, Monte Carlo simulation is provided to show the performance of the proposed decoding algorithm. In the simulation, the BPSK modulation and the additive white Gaussian noise (AWGN) channel are assumed. The code length is $N = 2^3 = 8$, code rate R is 0.5, and the index of the information bits is the same as [1].

As it is shown in Figure 5, the proposed decoding algorithm based on PSO achieves better bit error rate (BER, defined as the ratio of error decoding bits to the total decoding bits) performance than that of the existing BP decoding. Particularly, in the low region of signal to noise ratio (SNR), that is, E_b/N_0 (E_b is the average power of one bit and N_0 is the average value of the noise power spectrum density) the proposed algorithm provides a higher SNR advantage; for example, when the BER is 10^{-3} , the proposed decoding algorithm provides SNR advantages of 0.9 dB, and when the BER is 10^{-4} , the SNR advantages are 0.6 dB. Hence, we can conclude that with the proposed generalization BP decoding and PSO, performance of the belief propagation decoding algorithm for polar codes could be improved.

Additionally, one important thing to note is that, since the proposed algorithm is implemented depending on the probability messages modifying factors, while values of these factors are optimized by the PSO algorithm. Hence, the time complexity of the existing BP decoding is lower than that of

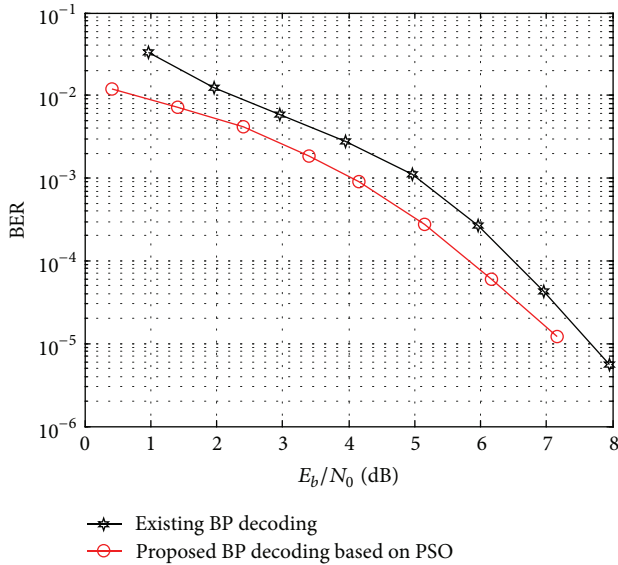


FIGURE 5: Comparison of BER between the proposed and existing BP decoding algorithms for polar codes with (4, 8). The maximum iteration number is set to 60.

the proposed decoding, which is tightly depending on the convergence time of the PSO algorithm. And the work for the time complexity optimization of the proposed algorithm will be conducted in our future research.

6. Conclusion

In this work, we proposed a generalization belief propagation (BP) decoding algorithm based on particle swarm optimization (PSO) to improve the performance of the polar codes. Through the analysis of the existing BP decoding algorithm for polar codes, we first introduced a probability modifying factor to each node of the BP decoder so as to enhance the error-correcting capacity of the BP decoding. Then, we generalized the BP decoding algorithm based on these modifying factors and drive the probability update equations for the proposed generalization BP decoding. Based on these new probability update equations, we further proved the intrinsic relationship of the existing decoding algorithms. Finally, in order to achieve the best performance, we formulated an optimization problem to find the optimal probability modifying factors for the proposed generalization BP decoding. Furthermore, a method based on the modified PSO algorithm was also introduced to solve the optimization problem. Numerical results showed that the proposed generalization BP decoding algorithm achieved better performance than that of the existing decoding, which suggested the effectiveness of the proposed decoding algorithm.

Conflict of Interests

The authors declare that they do not have any commercial or associative interests that represent a conflict of interests in connection with the work submitted.

References

- [1] E. Arıkan, "Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.
- [2] E. Arıkan and E. Telatar, "On the rate of channel polarization," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '09)*, pp. 1493–1495, Seoul, South Korea, July 2009.
- [3] S. B. Korada, E. Şaşıoğlu, and R. Urbanke, "Polar codes: characterization of exponent, bounds, and constructions," *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, vol. 56, no. 12, pp. 6253–6264, 2010.
- [4] I. Tal and A. Vardy, "List decoding of polar codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '11)*, pp. 1–5, St Petersburg, Russia, August 2011.
- [5] I. Tal and A. Vardy, "List decoding of polar codes," <http://arxiv.org/abs/1206.0050>.
- [6] K. Chen, K. Niu, and J. R. Lin, "Improved successive cancellation decoding of polar codes," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3100–3107, 2013.
- [7] K. Niu and K. Chen, "CRC-aided decoding of polar codes," *IEEE Communications Letters*, vol. 16, no. 10, pp. 1668–1671, 2012.
- [8] A. Alamdar-Yazdi and F. R. Kschischang, "A simplified successive-cancellation decoder for polar codes," *IEEE Communications Letters*, vol. 15, no. 12, pp. 1378–1380, 2011.
- [9] G. Sarkis and W. J. Gross, "Increasing the throughput of polar decoders," *IEEE Communications Letters*, vol. 17, no. 4, pp. 725–728, 2013.
- [10] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: algorithm and implementation," <http://arxiv.org/abs/1306.6311>.
- [11] P. Giard, G. Sarkis, C. Thibeault, and W. J. Gross, "A fast software polar decoder," <http://arxiv.org/abs/1306.6311>.
- [12] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. IT-8, pp. 21–28, 1962.
- [13] E. Arıkan, "A performance comparison of polar codes and reed-muller codes," *IEEE Communications Letters*, vol. 12, no. 6, pp. 447–449, 2008.
- [14] N. Hussami, S. B. Korada, and R. Urbanke, "Performance of polar codes for channel and source coding," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '09)*, pp. 1488–1492, Seoul, South Korea, July 2009.
- [15] E. Arıkan, "Polar codes: a pipelined implementation," in *Proceedings of the 4th International Symposium on Broadband Communication (ISBC '10)*, pp. 11–14, July 2010.
- [16] A. Eslami and H. Pishro-Nik, "On bit error rate performance of polar codes in finite regime," in *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing, Allerton (AACC'10)*, pp. 188–194, Allerton, Ill, USA, October 2010.
- [17] A. Eslami and H. Pishro-Nik, "On finite-length performance of polar codes: stopping sets, error floor, and concatenated design," <http://arxiv.org/abs/1211.2187>.
- [18] D. Divsalar and C. Jones, "Protograph LDPC codes with node degrees at least 3," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '06)*, pp. 1–5, San Francisco, Calif, USA, November 2006.
- [19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, December 1995.

- [20] C.-L. Sun, J.-C. Zeng, and J.-S. Pan, "An improved vector particle swarm optimization for constrained optimization problems," *Information Sciences*, vol. 181, no. 6, pp. 1153–1163, 2011.
- [21] S. He, E. Prempan, and Q. H. Wu, "An improved particle swarm optimizer for mechanical design optimization problems," *Engineering Optimization*, vol. 36, no. 5, pp. 585–605, 2004.
- [22] X. Hu and R. C. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," in *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics*, 2002.
- [23] F. Y. H. Ahmed, S. M. Shamsuddin, and S. Z. M. Hashim, "Improved SpikeProp for using particle swarm optimization," *Mathematical Problems in Engineering*, vol. 2013, Article ID 257085, 13 pages, 2013.
- [24] A. Y. Alanis, E. Rangel, J. Rivera, N. Arana-Daniel, and C. Lopez-Franco, "Particle swarm based approach of a real-time discrete neural identifier for linear induction motors," *Mathematical Problems in Engineering*, vol. 2013, Article ID 715094, 9 pages, 2013.
- [25] A. M. Arasomwan and A. O. Adewumi, "An investigation into the performance of particle swarm optimization with various chaotic maps," *Mathematical Problems in Engineering*, vol. 2014, Article ID 178959, 17 pages, 2014.
- [26] N. G. Hedeshi and M. S. Abadeh, "Coronary artery disease detection using a fuzzy-boosting PSO approach," *Mathematical Problems in Engineering*, vol. 2014, Article ID 783734, 12 pages, 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

