IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 September 2022, accepted 4 October 2022, date of publication 6 October 2022, date of current version 13 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212730

RESEARCH ARTICLE

Belief Propagation With Optimized Pool Size for Non-Adaptive Group Testing: An Empirical Study

SHUAI WANG AND QIN HUANG, (Senior Member, IEEE)
School of Electronic and Information Engineering, Beihang University, Beijing 100191, China

Corresponding author: Qin Huang (qhuang.smash@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62071026 and Grant 61941106, and in part by the High Performance Computing (HPC) Resources at Beihang University.

ABSTRACT In this paper, an empirical study shows that positive tests containing multiple defectives are unlikely to provide effective messages in belief propagation (BP) for non-adaptive group testing. Thus, an objective function is proposed to measure the effectiveness of messages over edges, especially in the low-noise region. The maximization of the objective function allows us to optimize the pool size for BP. Simulation results show that the error performance of BP in the low-noise region is significantly improved by our pool size optimization.

INDEX TERMS Group testing, decoding, belief propagation, bipartite graph, LDPC codes.

I. INTRODUCTION

In order to detect few defectives among millions of items, *group testing* was proposed by Dorfman [1] to reduce the huge number of tests required by individual testing. In the following decades, the theory of group testing has been further explored [2], [3], [4] and applied in various fields like information and communication technology [5], [6], [7], [8], data science [9], [10], and pandemic detection [11], [12], [13]. In general, the group testing strategies can be divided into adaptive and non-adaptive ways. The former sequentially designs each test based on the feedback of the previous tests, while the latter designs all tests in advance. This paper focuses on non-adaptive group testing (NAGT), due to its parallelizability and lower computation cost [8]. Further, noisy group testing is considered here because of its practical application [4], where the output of each test can be corrupted by noise.

In the scenario of noisy NAGT, the *belief propagation* (BP) algorithm empirically performs well in classifying the items [4], [14], [15], [16]. In [14], it indicates that different test designs may affect the error performance of BP in NAGT. In general, the Bernoulli random test design with fixed probability $p = \ln 2/K$ is used to extract the maximum amount of information from each test [17], where K denotes the expected number of defectives. As a result, half of the tests are positive and one bit of information is contained in each test [17]. However, it still lacks studies that how much information from tests is effectively used in the BP algorithm.

In this paper, we study the test design for BP to improve the error performance in NAGT with the consideration of the effectiveness of messages over edges. The main contributions of this work are given as follows:

- We distinguish the tests in terms of their transmitted messages in BP. Based on the bipartite subgraphs corresponding to various classes of tests, we show that the positive tests containing multiple defectives are unlikely to provide effective messages to their associated items, especially in the low-noise region.
- Instead of choosing the pool size to make the positive and negative tests approximately equal [15], [17], we only consider the tests with effective messages and propose a novel objective function to measure their effects over edges in the bipartite graph. Then, the optimized pool size in NAGT for BP is determined by the maximization of the objective function.
- We evaluate the error performance and computational complexity of BP with various test designs. Simulations show that the bit error rate (BER) of BP with our optimized pool size is about one order magnitude less than that with conventional test designs [15], [17] under the binary symmetric channel (BSC) with low flipping

The associate editor coordinating the review of this manuscript and approving it for publication was Qinghua Guo.

probability. Moreover, the expected computational complexity of the former is less than that of the latter.

The outline of this paper is as follows. Section II introduces the model of group testing and the BP algorithm. Section III proposes pool size optimization for BP. In Section IV, we test the error performance of BP with optimized pool size. Last, conclusions are drawn in Section V.

II. PRELIMINARIES

In this section, we first describe the basics of group testing. Then, we review the BP algorithm that will be further analyzed in later sections.

A. MODEL OF GROUP TESTING

Consider the detection of a small number of defectives in N items with the defective rate α of each item. In other words, the defective items are independent and identically distributed with the expected number $K = N\alpha$. The random binary vector, $\boldsymbol{X} = (x_1, \ldots, x_j, \ldots, x_N) \in \{0, 1\}^N$, denotes the status of N items, where $x_j = 1$ represents that the j-th item is defective and $\Pr(x_j = 1) = \alpha$ for $1 \leq j \leq N$.

Let T be the number of tests to detect the defectives. For $1 \leq i \leq T$ and $1 \leq j \leq N$, the i-th test is represented by $\boldsymbol{V}_i = (v_{i1}, \ldots, v_{ij}, \ldots, v_{iN}) \in \{0, 1\}^N$, where $v_{ij} = 1$ if the j-th item is in the pool of that test; otherwise, $v_{ij} = 0$. The noiseless output of the i-th test is denoted as z_i. If no defective item participates in that test, $z_i = 0$; otherwise, $z_i = 1$. Formally we have

$$z_i = \bigvee_{j=1}^{N} (v_{ij} \wedge x_j), \tag{1}$$

where the symbols $\vee$ and $\wedge$ stand for Boolean sum and Boolean product, respectively. In practice, the output z_i may be corrupted by noise and its noisy observation is y_i. In this paper, we consider the widely-adopted BSC noise model with flipping probability $\epsilon < 0.5$, denoted as BSC(ϵ). Let n_i be the BSC noise, then we have

$$y_i = z_i \oplus n_i = \left(\bigvee_{j=1}^{N} (v_{ij} \wedge x_j) \right) \oplus n_i, \tag{2}$$

where the symbol $\oplus$ denotes the XOR operation. Similarly, the noiseless outputs and their observations are denoted as $\boldsymbol{Z} = (z_1, \ldots, z_T)$ and $\boldsymbol{Y} = (y_1, \ldots, y_T)$, respectively. After observing $\boldsymbol{Y}$, the estimation $\hat{\boldsymbol{X}}$ of $\boldsymbol{X}$ is calculated to classify the N items. Three error performance metrics including BER, false negative rate (FNR) and false positive rate (FPR) are considered in the estimation of items, where

$$\text{FNR} = \frac{\text{Num}(\hat{x}_j = 0 | x_j = 1)}{\text{Num}(x_j = 1)}, \tag{3}$$

$$\text{FPR} = \frac{\text{Num}(\hat{x}_j = 1 | x_j = 0)}{\text{Num}(x_j = 0)}. \tag{4}$$

In the above equations, $\text{Num}(\hat{x}_j = 0 | x_j = 1)$ is the number of defectives falsely classified, and the others are similar.

The pool of T tests is denoted as a pooling matrix $\mathbf{V}$, where $\mathbf{V} = [\boldsymbol{V}_1; \ldots; \boldsymbol{V}_i; \ldots; \boldsymbol{V}_T] \in \{0, 1\}^{T \times N}$. For $1 \leq i \leq T$ and $1 \leq j \leq N$, the row weight $\rho_i \in [1, N]$ and the column weight $\gamma_j \in [1, T]$ of $\mathbf{V}$ denote the pool size of the i-th test and the number of tests participated by the j-th item, respectively. If $\mathbf{V}$ has constant row weight ρ and constant column weight γ, it is called (ρ, γ)-regular.

B. THE BP ALGORITHM

From the view of graphical representation, the matrix $\mathbf{V}$ can be described by a bipartite graph [18]. In analogy with the Tanner graphs [19] of *low-density parity-check* (LDPC) codes [20], we use variable nodes (VNs) and check nodes (CNs) to represent items and tests, respectively. As shown in Fig. 1, only the nodes of different types can be connected by the edges. Given this representation, it is convenient to estimate $\boldsymbol{X}$ by the BP algorithm, which has been widely used for decoding LDPC codes [21]. Their main difference in the message update comes from the specific operations to calculate the outputs of group testing and the check sums of LDPC codes. In this part, we briefly review the BP algorithm under the BSC [4], [15].

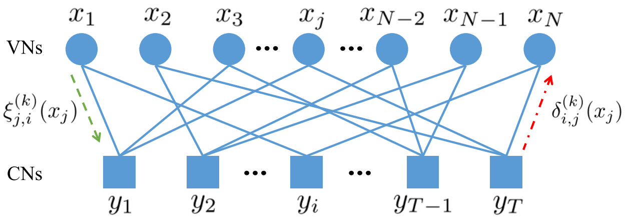


FIGURE 1. Graphical representation of the pooling matrix of NAGT.

The BP algorithm iteratively estimates the posterior marginals for the items in a graph-based model. As shown in Fig. 1, the probabilistic messages passed from the VNs and CNs in the k-th iteration are denoted as $\xi_{j,i}^{(k)}(x_j)$ and $\delta_{i,j}^{(k)}(x_j)$, respectively. These messages are probability distributions on $\{0, 1\}$ [15], i.e., $\xi_{j,i}^{(k)}(x_j) \in [0, 1]$ and $\xi_{j,i}^{(k)}(0)+\xi_{j,i}^{(k)}(1) = 1$, and similarly for $\delta_{i,j}^{(k)}(x_j)$. Define the index sets $M_j = \{i : 1 \leq i \leq T, v_{i,j} \neq 0\}$ for $1 \leq j \leq N$ and $N_i = \{j : 1 \leq j \leq N, v_{i,j} \neq 0\}$ for $1 \leq i \leq T$, respectively. For $k = 0$ with arbitrary i and j, initialize $\xi_{j,i}^{(0)}(1) = \alpha$, and $\xi_{j,i}^{(0)}(0) = 1 - \alpha$. For $k \geq 1$, $\delta_{i,j}^{(k)}(x_j)$ is first calculated in the test-to-item message update as follows. If $y_i = 0$, then

$$\begin{cases} \delta_{i,j}^{(k)}(0) = \lambda\epsilon + \lambda(1 - 2\epsilon) \prod\limits_{j' \in N_i \setminus j} \xi_{j',i}^{(k-1)}(0), \\ \delta_{i,j}^{(k)}(1) = \lambda\epsilon; \end{cases} \tag{5}$$

otherwise,

$$\begin{cases} \delta_{i,j}^{(k)}(0) = \lambda(1 - \epsilon) - \lambda(1 - 2\epsilon) \prod\limits_{j' \in N_i \setminus j} \xi_{j',i}^{(k-1)}(0), \\ \delta_{i,j}^{(k)}(1) = \lambda(1 - \epsilon) \end{cases} \tag{6}$$

where the symbol λ is a positive normalizing factor. Then, $\xi_{j,i}^{(k)}(x_j)$ is calculated in the item-to-test message update by

$$\xi_{j,i}^{(k)}(b) = \lambda\alpha^b(1-\alpha)^{1-b} \prod_{i' \in M_j \backslash i} \delta_{i',j}^{(k)}(b), \quad b \in \{0, 1\}. \tag{7}$$

Finally, the estimation of x_j in the k-th iteration is given by

$$\hat{x}_j^{(k)} \leftarrow \arg \max_{b \in \{0,1\}} \left[\alpha^b(1-\alpha)^{1-b} \prod_{i \in M_j} \delta_{i,j}^{(k)}(b)\right]. \tag{8}$$

In the iterative process, different update schedules can be used. Compared to the flooding BP algorithm which performs message update in parallel, the random scheduling BP (RSBP) algorithm updates messages in a randomized way and improves the error performance [15], [16].

III. POOL SIZE OPTIMIZATION FOR BELIEF PROPAGATION

In this section, we investigate the effects of tests in BP by considering their transmitted messages in the associated subgraphs. It shows that the positive tests containing multiple defectives are difficult to transmit effective messages, especially in the low-noise region and later iterations of BP. Therefore, we focus on other tests with effective messages and propose an objective function to measure their effects over edges. Pool size optimization maximizes the objective function, and hence facilitates the transmission of effective messages in BP decoding.

A. CLASSES OF TESTS

There are two types of tests in our model: with or without corrupted outputs. First, we show that the tests with corrupted outputs usually have negative effects to their associated items in BP decoding. Specifically speaking, if $x_j = 1$ and its associated positive test is corrupted with $y_i = 0$, we have $\delta_{i,j}^{(k)}(0) \geq \delta_{i,j}^{(k)}(1)$ based on (5) and $\epsilon < 0.5$. As a conclusion, this corrupted test provides either misleading information or no information to the j-th item, if "$>$" or "$=$" is true in the above inequality. Similarly, if $x_j = 0$ and its associated negative test is corrupted with $y_i = 1$, we have $\delta_{i,j}^{(k)}(0) \leq \delta_{i,j}^{(k)}(1)$ based on (6) and draw a consistent conclusion. Or if $x_j = 0$ and its associated positive test is corrupted with $y_i = 0$, although no misleading information is transmitted from the i-th test to the j-th item, but the defective items associated with the i-th test are misled by its messages.

Thus, we focus on the tests with correct outputs, which can be further divided into three classes: (i) containing multiple defective items with $y_i = 1$; (ii) containing only one defective item with $y_i = 1$; (iii) containing no defective item with $y_i = 0$. Corresponding subgraphs of three classes of tests are depicted in Fig. 2. The number "0" or "1" marked on each node is its noiseless true value. Then, the message update of various classes of tests is analyzed as below.

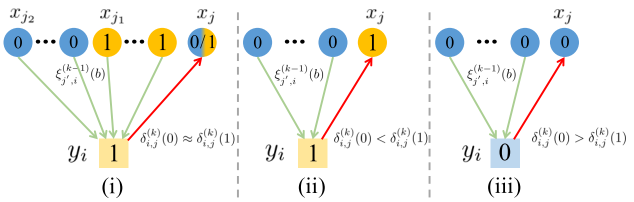

x_{j_2}
x_{j_1}
x_j
0
0
1
1
0/1
$\xi_{j',i}^{(k-1)}(b)$
y_i
1
$\delta_{i,j}^{(k)}(0) \approx \delta_{i,j}^{(k)}(1)$
(i)
x_j
0
0
1
$\xi_{j',i}^{(k-1)}(b)$
y_i
1
$\delta_{i,j}^{(k)}(0) < \delta_{i,j}^{(k)}(1)$
(ii)
x_j
0
0
0
$\xi_{j',i}^{(k-1)}(b)$
y_i
0
$\delta_{i,j}^{(k)}(0) > \delta_{i,j}^{(k)}(1)$
(iii)

FIGURE 2. Subgraphs of three classes of tests.

Without the loss of generality, we consider after a certain number of BP iterations in a ρ-regular graph, there still exist a few decoding errors, and then the BP algorithm goes to the k-th iteration:

(1) Suppose the i-th test belongs to class (i) as shown in Fig. 2-(i). First, it provides no effective messages to the associated non-defective items, since $\delta_{i,j}^{(k)}(0) \leq \delta_{i,j}^{(k)}(1)$ based on (6). Second, as long as one of its associated defective items with index j_1 is correctly decoded as $\hat{x}_{j_1}^{(k-1)} = 1$ in the $(k-1)$-th iteration, it is likely that $\xi_{j_1,i}^{(k-1)}(0) \to 0$, especially in the low-noise region. As a result, $\delta_{i,j}^{(k)}(0) \approx \delta_{i,j}^{(k)}(1)$ for each item $j \in N_i \backslash j_1$ based on (6). In this case, the i-th test transmits almost no effective messages to the rest of the defective items, including those with decoding errors. Third, as long as one of its associated non-defective items with index j_2 is incorrectly decoded as $\hat{x}_{j_2}^{(k-1)} = 1$ with $\xi_{j_2,i}^{(k-1)}(0) \to 0$, we have $\delta_{i,j}^{(k)}(0) \approx \delta_{i,j}^{(k)}(1)$ for each item $j \in N_i \backslash j_2$ based on (6). In this case, the i-th test is not helpful to correct the decoding errors for the j_2-th item and other defective items. Fourth, if no item is decoded as $\hat{x}_j^{(k-1)} = 1$, the restriction of $y_i = 1$ will lead to at least one item to be decoded as 1 in latter iteration. Therefore, this case can be transformed into the above second or third case.

(2) Suppose the i-th test belongs to class (ii) as shown in Fig. 2-(ii). Since $\delta_{i,j}^{(k)}(0) \leq \delta_{i,j}^{(k)}(1)$ for each $j \in N_i$ based on (6), the i-th test can only transmit effective message to the associated defective item. Specifically speaking, if its associated non-defective items with indexes $\{j' : j' \in N_i \backslash j\}$ are correctly decoded as 0's in the $(k-1)$-th iteration,[1] each $\xi_{j',i}^{(k-1)}(0)$ is far from 0, since $\delta_{i,j'}^{(k-1)}(0) \leq \delta_{i,j'}^{(k-1)}(1)$ and $\delta_{i,j'}^{(k-1)}(0) \cdot \xi_{j',i}^{(k-1)}(0) > \delta_{i,j'}^{(k-1)}(1) \cdot \xi_{j',i}^{(k-1)}(1)$ based on (7) and (8). Thus, the i-th test transmits $\delta_{i,j}^{(k)}(0) < \delta_{i,j}^{(k)}(1)$ to the only defective item with index $j \in N_i$. In particular, in the low-noise region where $\xi_{j',i}^{(k-1)}(0) \to 1$, the difference between $\delta_{i,j}^{(k)}(1)$ and $\delta_{i,j}^{(k)}(0)$ tends to $\lambda(1-2\epsilon)$.

(3) Suppose the i-th test belongs to class (iii) as shown in Fig. 2-(iii). Since $\delta_{i,j}^{(k)}(0) \geq \delta_{i,j}^{(k)}(1)$ for each $j \in N_i$ based on (5), the i-th test can provide effective messages to all the ρ associated non-defective items. In particular, in the low-noise region where $\xi_{j,i}^{(k-1)}(0) \to 1$ for each $j \in N_i$, the difference between $\delta_{i,j}^{(k)}(0)$ and $\delta_{i,j}^{(k)}(1)$ tends to $\lambda(1-2\epsilon)$.

In conclusion, the above analysis shows that the tests of class (i) are difficult to transmit effective messages to their

[1] This case with the associated non-defective items correctly decoded is quite common, since the FPR of the BP algorithm is several orders magnitude less than its FNR as shown in the simulation in Section IV.

associated items. In contrast, each test of class (ii) only transmits effective message to the associated defective item, while each test of class (iii) can transmit effective messages to all the ρ associated items. The occurrence probabilities of three classes of tests are denoted as P_1, P_2 and P_3, respectively, where $P_1 = 1 - \epsilon - P_2 - P_3$,

$$P_2 = (1-\epsilon)\rho\alpha(1-\alpha)^{(\rho-1)}, \tag{9}$$

$$P_3 = (1-\epsilon)(1-\alpha)^{\rho}. \tag{10}$$

B. POOL SIZE OPTIMIZATION VIA OBJECTIVE FUNCTION

In conventional test designs, the parameters of pooling matrix are usually chosen to maximize the entropy of each test. For instance, in the Bernoulli design [17], each item is included in each test independently at random with the probability $p = \ln 2/K$. As a result, the occurrence probability of positive tests, denoted as P_{pos}, tends to 0.5 in the asymptotic region of $N \to \infty$. Then, the binary entropy function is used to measure the entropy of each test as follows [17]:

$$H(P_{\text{pos}}) = -P_{\text{pos}} \log_2 P_{\text{pos}} - (1-P_{\text{pos}}) \log_2(1-P_{\text{pos}}). \tag{11}$$

Similarly, a randomly chosen (ρ, γ)-regular test design [15] sets $\rho = \ln 2/\alpha$ and $\gamma = T\rho/N$, so that nearly half of the tests are positive and about one bit of information is contained in each test.

According to the analysis in the previous subsection, we design a ρ-regular pooling matrix $\mathbf{V}$ for BP by only considering the positive tests of class (ii) and the negative tests of class (iii). Similar to (11), a metric denoted as J_{m} is firstly defined to measure the information contained in each test of classes (ii) and (iii) as follows:

$$J_{\text{m}} = -P_2' \log_2 P_2' - (1-P_2') \log_2(1-P_2'),$$

where P_2' is a normalization of the probability distribution:

$$P_2' = \frac{P_2}{P_2+P_3} = \frac{\rho\alpha}{1+(\rho-1)\alpha}.$$

Based on the previous analysis of BP, each positive test of class (ii) can only provide effective message to one defective item, while each negative test of class (iii) can provide effective messages to ρ non-defective items. Thus, we regard the maximum amount of information provided by these tests in the test-to-item message update as

$$J_{\text{sum}} = TP_2 \cdot J_{\text{m}} + TP_3 \cdot \rho J_{\text{m}}.$$

After that, we define the objective function J_{o} as follows:

$$J_{\text{o}} = \frac{J_{\text{sum}}}{T\rho} = (1-\epsilon)(1-\alpha)^{(\rho-1)} J_{\text{m}}, \tag{12}$$

which is related to the average of the maximum amount of information provided by the tests of class (ii) and (iii) on each edge. Given the parameters ϵ and α, J_{o} is a function of ρ. We believe that J_{o} is able to reflect the effectiveness of the transmitted messages in the test-to-item message update, at least in the low-noise region and later iterations of BP. Thus, the optimized pool size is set as

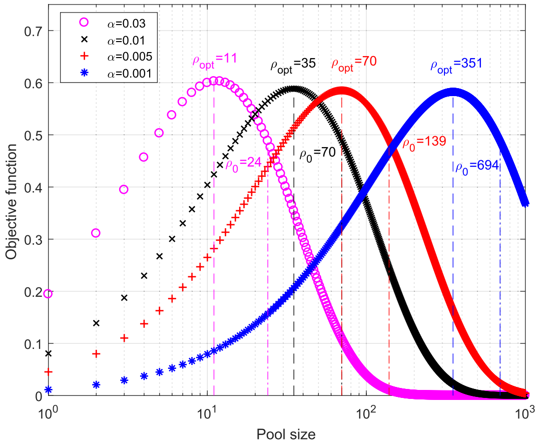


FIGURE 3. Relation between the objective function and the pool size with various defective rates and $\epsilon = 0$.

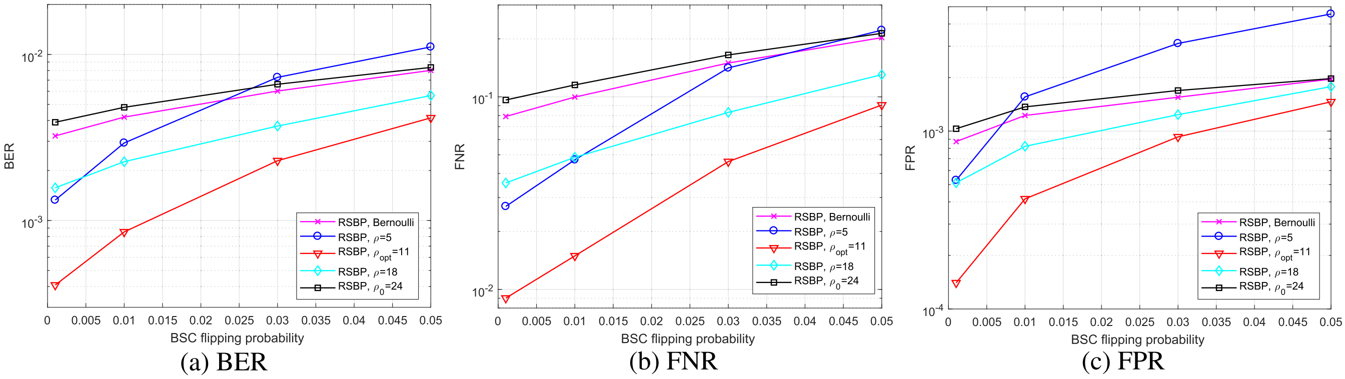
$$\rho_{\text{opt}} = \arg\max_{\rho \in \mathbb{N}_+} (J_{\text{o}}), \tag{13}$$

where $\mathbb{N}_+$ denotes the positive integer set. Accordingly, the column weights of $\mathbf{V}$ are constant or near-constant with the expected number $\overline{\gamma} = T\rho_{\text{opt}}/N$. Simulation results in next section show that the maximization of the objective function is benefit for BP decoding, especially in the low-noise region.

IV. SIMULATION AND COMPARISON

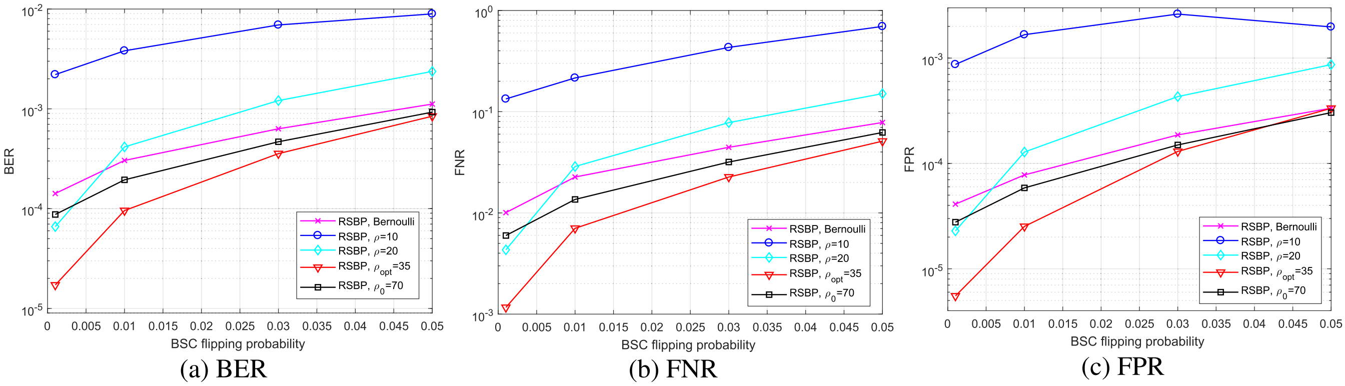
In this section, we compare the error performance and computational complexity of the RSBP algorithm [16] with different test designs. Given the parameters N, T, α and ρ, the pooling matrix $\mathbf{V}$ is generated by the Bernoulli design [17] with fixed probability $p = \ln 2/(N\alpha)$ or the ρ-regular test design. For the Bernoulli design, a specific randomly constructed $\mathbf{V}$ is generated in each of the settings below. For the ρ-regular test design, the progressive-edge-growth algorithm proposed for LDPC codes [22] is used to generate $\mathbf{V}$. The maximum iteration number of the RSBP algorithm is set as 10 times of T, so that each message is likely to be updated several times. In addition, each simulation result is averaged over at least 3000 experiments, among which at least 100 experiments have decoding errors.

Setting 1: Given $\alpha = 0.03$, $N = 100$ and $T = 50$ for the RSBP algorithm. Based on (12) and (13), the flipping probability ϵ is just a scaling factor which has no influence on the value of ρ_{opt}. For simplicity, the relation between J_o and ρ given $\epsilon = 0$ is shown in the magenta line in Fig. 3. As a result, the optimized pool size via the objective function is set as $\rho_{opt} = 11$. Pool size $\rho_0 = \lceil \frac{ln2}{\alpha} \rceil = 24$, where about half of the tests are positive and the entropy of each test is maximized [15], and the two other pool sizes 5 and 18 are investigated for comparison purpose. The RSBP algorithm with these various pool sizes are tested under the BSC as shown in Fig. 4.



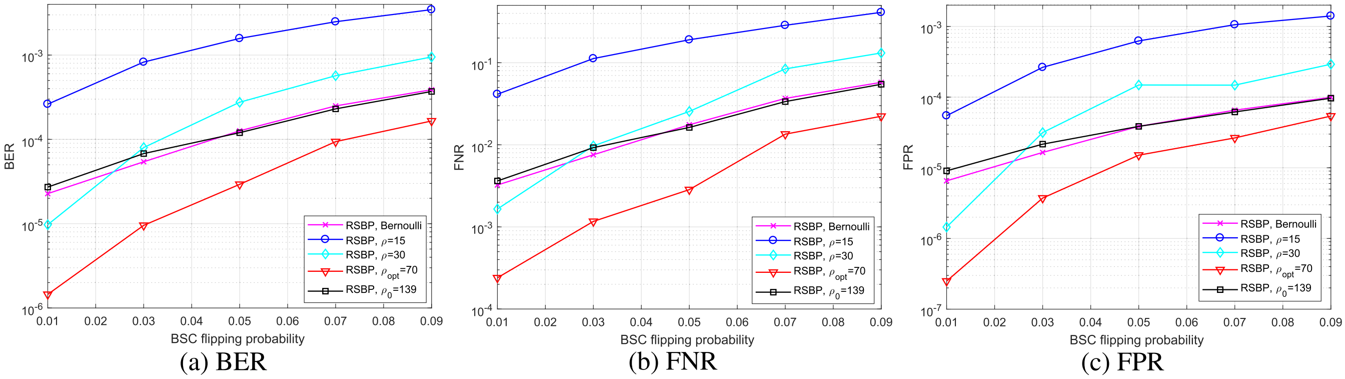
(a) BER (b) FNR (c) FPR

FIGURE 4. Error performance of different strategies with various pool sizes under the BSC when $\alpha = 0.03$, $N = 100$ and $T = 50$.



(a) BER (b) FNR (c) FPR

FIGURE 5. Error performance of different strategies with various pool sizes under the BSC when $\alpha = 0.01$, $N = 1000$ and $T = 200$.



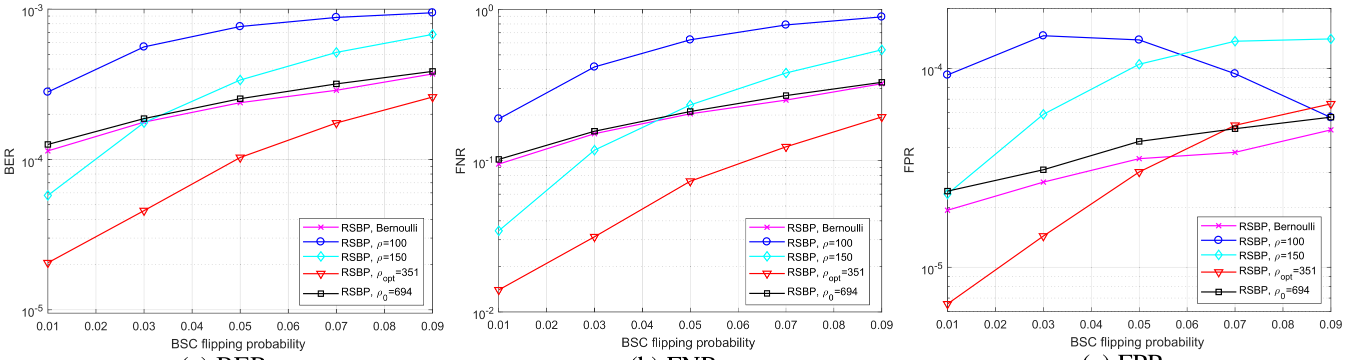
(a) BER (b) FNR (c) FPR

FIGURE 6. Error performance of different strategies with various pool sizes under the BSC when $\alpha = 0.005$, $N = 1000$ and $T = 200$.

Setting 2: Given $\alpha = 0.01$, $N = 1000$ and $T = 200$ for the RSBP algorithm. When $\epsilon = 0$, the relation between J_o and ρ given $\epsilon = 0$ is shown in the black line in Fig. 3. Based on (13), we have $\rho_{opt} = 35$. Pool size $\rho_0 = \lceil \frac{ln2}{\alpha} \rceil = 70$ and the two other smaller pool sizes 10 and 20 are investigated for comparison purpose. The RSBP algorithm with these various pool sizes are tested under the BSC as shown in Fig. 5.

Setting 3: Given $\alpha = 0.005$, $N = 1000$ and $T = 200$ for the RSBP algorithm. When $\epsilon = 0$, the relation between J_o and ρ given $\epsilon = 0$ is shown in the red line in Fig. 3. Based on (13), we have $\rho_{opt} = 70$. Pool size $\rho_0 = \lceil \frac{ln2}{\alpha} \rceil = 139$ and the two other smaller pool sizes 15 and 30 are investigated for comparison purpose. The RSBP algorithm with these various pool sizes are tested under the BSC as shown in Fig. 6.

Setting 4: Given $\alpha = 0.001$, $N = 3000$ and $T = 100$ for the RSBP algorithm. When $\epsilon = 0$, the relation between J_o and ρ given $\epsilon = 0$ is shown in the blue line in Fig. 3. Based on (13), we have $\rho_{opt} = 351$. Pool size $\rho_0 = \lceil \frac{ln2}{\alpha} \rceil = 694$ and the two other smaller pool sizes 100 and 150 are investigated for comparison purpose. The RSBP algorithm with these various pool sizes are tested under the BSC as shown in Fig. 7.



(a) BER (b) FNR (c) FPR

FIGURE 7. Error performance of different strategies with various pool sizes under the BSC when $\alpha = 0.001$, $N = 3000$ and $T = 100$.

As shown in Fig. 4∼7, given different values of ϵ and α, the RSBP algorithm with our optimized pool size performs best in terms of BER, FNR and FPR in the region of low ϵ. For instance, its BER is about one order magnitude less than that of the RSBP algorithm with ρ_0 or the Bernoulli design when $(\epsilon, \alpha) \in \{(0.001, 0.03), (0.001, 0.01), (0.01, 0.005), (0.01, 0.001)\}$, and about one to two orders of magnitude less than those of the RSBP algorithm with smaller pool sizes.

Now we discuss the computational complexity of the message update in the RSBP algorithm. Suppose $\mathbf{V}$ has constant row weight ρ and constant column weight γ. In each iteration ($k \geq 1$) of the RSBP algorithm, one CN with index i is randomly chosen to compute the messages $\delta_{i,j}^{(k)}(x_j)$'s for $j \in N_i$, and then all the messages $\xi_{j,i_1}^{(k)}(x_j)$'s for $i_1 \in M_j$ are required to compute. In addition, the messages on the rest of the edges are unchanged and need no update. First, we consider $\delta_{i,j}^{(k)}(x_j)$'s based on (5) and (6). The calculation of each $\delta_{i,j}^{(k)}(1)$ is omitted here for its simplicity. To compute $\delta_{i,j}^{(k)}(0)$'s, $\rho + 1$ real multiplications are required for $(1 - 2\epsilon) \prod_{j' \in N_i} \xi_{j',i}^{(k-1)}(0)$, followed by one real multiplication and one real addition to obtain each $\delta_{i,j}^{(k)}(0)$. Thus, $2\rho + 1$ real multiplications and ρ real additions are required for all the $\delta_{i,j}^{(k)}(x_j)$'s. Then, we consider $\xi_{j,i_1}^{(k)}(x_j)$'s based on (7). To compute $\xi_{j,i_1}^{(k)}(0)$'s or $\xi_{j,i_1}^{(k)}(1)$'s, $\gamma + 1$ real multiplications are required for $(1 - \alpha) \prod_{i' \in M_j} \delta_{i',j}^{(k)}(0)$ or $\alpha \prod_{i' \in M_j} \delta_{i',j}^{(k)}(1)$, followed by one more real multiplication to obtain each $\xi_{j,i_1}^{(k)}(0)$ or $\xi_{j,i_1}^{(k)}(1)$. Thus, for all the ρ VNs associated with the chosen CN, it requires $2\rho(2\gamma + 1)$ real multiplications to compute $\xi_{j,i_1}^{(k)}(x_j)$'s.

To summarize, $4\rho(\gamma+1)+1$ real multiplications and ρ real additions are required for the message update in each iteration of the RSBP algorithm. Based on this result we provide a complexity comparison between various test designs. For the Bernoulli design, the expectations of the row weights and the column weights are $\overline{\rho} = \ln 2/\alpha$ and $\overline{\gamma} = T \ln 2/(N\alpha)$, respectively. For the ρ-regular test design, $\overline{\rho} = \rho$ and $\overline{\gamma} = T\rho/N$. Using Setting 2 as an example, the expected computational complexity per iteration of the RSBP algorithm with various test designs is listed in Table 1. It can be seen that the expected computational complexity increases quadratically as $\overline{\rho}$ increases. For the RSBP algorithm with ρ_{opt}, its computational complexity per iteration is about 3.6 times less than the RSBP algorithm with ρ_0 or the Bernoulli design.

TABLE 1. Expected computational complexity per iteration of the RSBP algorithm with various test designs when $\alpha = 0.01$, $N = 1000$ and $T = 200$.

Test design	$\overline{\rho}$	Average number of operations	
		RM	RA
Bernoulli	69.3	4120.2	69.3
ρ-regular	$\rho_0 = 70$	4201	70
	$\rho_{\text{opt}} = 35$	1121	35
	20	401	20
	10	121	10

RM: real multiplication RA: real addition

V. CONCLUSION

This paper presents an empirical study to optimize the pool size in NAGT for BP. Our analysis shows the effectiveness of the messages from tests in BP and leads to the proposed objective function for pool size optimization. Simulation results show that the RSBP algorithm with our optimized pool size achieves lower error rates in the low-noise region than those required to maximize the entropy of each test.

REFERENCES

[1] R. Dorfman, "The detection of defective members of large populations," *Ann. Math. Stat.*, vol. 14, no. 4, pp. 436–440, 1943.

[2] D.-Z. Du and F. K. Hwang, *Combinatorial Group Testing and its Applications*, 2nd ed. Singapore: World Scientific, 2000.

[3] M. Aldridge, L. Baldassini, and O. Johnson, "Group testing algorithms: Bounds and simulations," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3671–3687, Jun. 2014.

[4] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: An information theory perspective," *Found. Trend. Comms. Inf. Theory*, vol. 15, nos. 3–4, pp. 196–392, Dec. 2019.

[5] J. Wolf, "Born again group testing: Multiaccess communications," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 185–191, Mar. 1985.

[6] E. S. Hong and R. E. Ladner, "Group testing for image compression," *IEEE Trans. Image Process.*, vol. 11, no. 8, pp. 901–911, Aug. 2002.

[7] S. Wang, Q. Huang, S. Zheng, and Z. Wang, "Design of non-adaptive querying policies based on error control coding," *IEEE Access*, vol. 8, pp. 14513–14522, Jan. 2020.

[8] L. Zhou and A. O. Hero, "Resolution limits for the noisy non-adaptive 20 questions problem," *IEEE Trans. Inf. Theory*, vol. 67, no. 4, pp. 2055–2073, Apr. 2021.

[9] J. Wang, E. Lo, and M. L. Yiu, "Identifying the most connected vertices in hidden bipartite graphs using group testing," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 10, pp. 2245–2256, Oct. 2013.

[10] D. M. Malioutov, K. R. Varshney, A. Emad, and S. Dash, "Learning interpretable classification rules with Boolean compressed sensing," in *Transparent Data Mining for Big Small Data*. Cham, Switzerland: Springer, 2017, pp. 95–121.

[11] C. Gollier and O. Gossner, "Group testing against COVID-19," *Covid Econ.*, vol. 1, no. 2, pp. 32–42, 2020.

[12] B. Abdalhamid, C. R. Bilder, E. L. McCutchen, S. H. Hinrichs, S. A. Koepsell, and P. C. Iwen, "Assessment of specimen pooling to conserve SARS CoV-2 testing resources," *Amer. J. Clin. Pathol.*, vol. 153, no. 6, pp. 715–718, Jun. 2020.

[13] M. Aldridge, "Conservative two-stage group testing in the linear regime," 2020, *arXiv:2005.06617*.

[14] D. Sejdinovic and O. Johnson, "Note on noisy group testing: Asymptotic bounds and belief propagation reconstruction," in *Proc. 48th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Allerton, IL, USA, Sep. 2010, pp. 998–1003.

[15] A. Coja-Oghlan, M. Hahn-Klimroth, P. Loick, and M. Penschuck, "Efficient and accurate group testing via belief propagation: An empirical study," 2021, *arXiv:2105.07882*.

[16] E. Karimi, A. Heidarzadeh, K. R. Narayanan, and A. Sprintson, "Scheduling improves the performance of belief propagation for noisy group testing," 2021, *arXiv:2110.10110*.

[17] M. Aldridge, "The capacity of Bernoulli nonadaptive group testing," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7142–7148, Nov. 2017.

[18] T. Wadayama, "An analysis on non-adaptive group testing based on sparse pooling graphs," in *Proc. IEEE Int. Symp. Inf. Theory*, Istanbul, Turkey, Jul. 2013, pp. 2681–2685.

[19] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533–547, Sep. 1981.

[20] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, 1963.

[21] S. Lin and D. J. Costello, *Error Control Coding*. 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2004.

[22] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Trans. Inf. Theory*, vol. 51, no. 1, pp. 386–398, Jan. 2005.

SHUAI WANG received the B.E. degree in electronic and information engineering from Beihang University, Beijing, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering. His research interests include error control codes of storage and communication systems.

QIN HUANG (Senior Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from Southeast University, Nanjing, China, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California at Davis, Davis, CA, USA, in 2011.

He joined at Link-A-Media Devices Corporation, Santa Clara, CA, USA, in 2011. He is currently a Professor with the School of Electronic and Information Engineering, Beihang University (BUAA), Beijing, China. His research interests include classical and modern coding theory, signal processing, and their applications on communications and storage systems. He served as an Associate Editor for the IEEE Transactions on Communications.