PAPER

Connection Choice Codes

Chih-Ming CHEN[†], *Nonmember and* **Ying-ping CHEN**[†a)], *Member*

SUMMARY Luby Transform (LT) codes are the first practical implementation of digital fountain codes. In LT codes, encoding symbols are independently generated so as to realize the *universal property* which means that performance is independent of channel parameters. The universal property makes LT codes able to provide reliable delivery simultaneously via channels of different quality while it may also limit the flexibility of LT codes. In certain application scenarios, such as real-time multimedia transmission, most receivers have tolerable channels whose erasure rates are not fixed, and channels of high erasure rate are outside the design box. In this paper, *Connection Choice* (CC) codes are proposed to trade the universal property for better performance. The key to CC codes is replacement of random selection with *tournament selection*. Tournament selection can equalize the frequency of input symbols to join encoding and change the degree distribution of input symbols. Our study indicates that CC codes with appropriate degree distributions provide better performance than the best known LT code when channels of high erasure rate can be ignored. CC codes enable system designers to customize digital fountain codes by taking into account the distribution of the erasure rate and create a new possibility for setting trade-offs between performance and erasure rate.

key words: *digital fountain code, forward error correction, LT code, erasure rate, connection choice*

1. Introduction

Digital fountain codes are a new category of erasure correcting codes that have been introduced in the last decade. A well-known implementation of digital fountain codes is the Luby Transform codes (LT codes) [1]. LT codes realize the most important characteristic of digital fountain codes called *ratelessness*, which allows unlimited output symbols to be generated for variable code rates. Moreover, it has been proven that the coding scheme of LT codes can offer low overhead and good coding efficiency. First, the encoder decides the degree of each output symbol according to a pre-defined probability distribution. Second, the general but costly decoding approach, Gaussian elimination, is replaced with belief propagation [2] at the receiver side. As a result, reception overhead is slightly increased to secure the benefit of much better decoding efficiency. Based on the design, two conditions are considered in the scenario regarding whether source data can be successfully recovered. The first one is a necessary condition that the receiver must receive all pieces of source data. In other words, each input symbol must be chosen and encoded in received output symbols at least once. The other is that the set of received output symbols must have an appropriate configuration such that belief propagation can successfully unpack all the output symbols. These conditions may be satisfied by adopting applicable degree distributions and receiving sufficient amount of output symbols.

The proposal of LT codes [1] gave rise to a family of degree distributions based on theoretical analysis. The analysis showed that one of the proposed distributions, *ideal soliton distribution* (ISD), has the optimal performance only in the ideal case. The other one, *robust soliton distribution* (RSD), was developed with robustness and flexibility. Two parameters, c and δ, were introduced to adjust the distribution and coding behavior of LT codes. According to the analysis, k input symbols can be recovered with a success probability $1-\delta$ when extra $O(\sqrt{k}\ln^2(k/\delta))$ output symbols are received. Moreover, both soliton distributions are designed to work with an average degree equal to or greater than $O(\ln(k))$ in order to satisfy the necessary condition for successful decoding. The structure of error correction codes can be represented as a Tanner graph with information and check nodes. In LT codes, the degrees of check nodes are decided by the given degree distribution, and information nodes are chosen uniformly at random when the edges are being built. The degree distribution of information nodes is hence binomial. A degree distribution with average degree greater than $O(\ln(k))$ can guarantee that the probability of an isolated information node will be less than $1/k$. Clearly, high average degrees reduce the probability that some input symbols are missing at the receiver side. However, another problem concerning the average degree is the computational cost. To generate an output symbol with degree d requires $d-1$ Xor operations, and therefore, the average degree of the adopted distribution dictates the computational cost of LT codes. For example, the required operations of LT codes with robust soliton distribution are $O(k\ln(k/\delta))$. Such a cost is merely acceptable, and more efficient digital fountain codes are still in need.

A successful improvement, called *Raptor codes* [3], [4], was designed as a two-layer encoding structure. In its second layer, *weakened LT codes* are implemented with degree distributions of which the average degrees are much lower than $O(\ln(k))$. It can be understood that not all input symbols will be chosen to join the encoding process due to low average degrees and selection randomness. Therefore, a set of block codes is integrated as a pre-coder in front of the weakened LT codes to encode source data as intermediate symbols. The pre-coder has a fixed code rate, which

Manuscript received March 22, 2013.
Manuscript revised December 5, 2013.

[†]The authors are with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.
a) E-mail: ypchen@nclab.tw

DOI: 10.1587/transcom.E97.B.1350

allows a fraction of intermediate symbols to be lost while the source data can still be reconstructed. Although the pre-coder requires extra cost of space and computation, the total coding time of Raptor codes is $O(k)$. The success of Raptor codes bases on the cooperation between weakened LT codes and the pre-coder. In brief, the pre-coder shares the workload of LT codes and solves the problem that, when a degree distribution with low average degree is adopted, some input symbols may be missing in encoding or transmission.

For the same purpose, another solution is to reduce the variance of frequency with which an input symbol is chosen and encoded. In this work, a new scheme named *connection choice* (CC) codes is studied by introducing a different selection mechanism for LT code encoding. The adopted selection strategy can reduce the probability of input symbols never being selected and effectively drop the error floor of weakened LT codes. It means that the proportion of recovered input symbols can be enhanced or even a full decoding can be achieved and also that CC codes may be integrated into Raptor codes as a substitution of weakened LT codes. In fact, the new selection mechanism reforms the degree distribution of input symbols. This feature provides the flexibility of CC codes to cooperate with customized degree distributions and obtain better performance in different application scenarios. Moreover, it is noted that, while trading the universal property for better performance, CC codes are still rateless and suitable for the scenarios in which digital fountain codes are suitable but fixed rate block codes may not be applicable or efficient.

For the remainder of this paper, the details of connection choice are firstly introduced in Sect. 2. Although the selection strategy is conceptually simple, its behavior makes the coding behavior highly complicated such that theoretically analyzing the error probability is extremely difficult. Consequently, simulation results are presented to demonstrate the performance of CC codes in Sect. 3. Section 4 then makes a further study of connection choice and illustrates the characteristics of CC codes by using And-Or tree analysis. Finally, Sect. 5 introduces obtained degree distribution instances to confirm the flexibility of CC codes, followed by the conclusion given in Sect. 6.

2. Connection Choice

In LT codes, the relation between input symbols and output symbols can be modeled as a Tanner graph. Each connection edge denotes that an input symbol is a part of some output symbols. As aforementioned, we expect that all input symbols have at least one connection edge for a possible full decoding. Therefore, a selection strategy called *tournament selection* is introduced to stochastically equalize the connection count of each input symbol. Tournament selection is a common technique in the field of evolutionary computation. The form of tournament selection attributed to the unpublished work by Wetzel was studied in Brindle's dissertation [5], and more recent studies using tournament selection can be found in [6]. The operation of tournament selection is fairly simple and hence, a similar concept has also been used in other domains, such as that Mitzenmacher introduced the technique to achieve load balancing in distribution systems [7]. The details and the effect of utilizing tournament selection in LT codes will be presented in this section.

2.1 Tournament Selection

In CC codes, random selection is replaced by tournament selection, used to select an input symbol for encoding. The first step of employing tournament selection is to define a parameter T called *tournament size*. At the time to decide each input symbol for encoding, T input symbols are uniformly randomly selected as candidates, and the final decision is made according to the *connection counters*. Connection counters count the number of output symbols to which an input symbol has been connected. The counter records the degree of input symbols during the encoding process. Out of these candidates, the input symbol with the lowest counter value is chosen to take the part of the current encoding run. Figure 1 illustrates tournament selection with an example. A Tanner graph is used to represent the relation between input (square) and output (circle) symbols. The connection edges indicate the coding structure between these symbols. For example, the output symbol e_1 is exactly the input symbol s_2, and e_2 is generated by s_1, s_2, and s_4. The number in symbol nodes represents its degree or counter value for the moment. The example shows the coding process of e_4 with degree $d = 2$. For the first selection, tournament selection selects T (assuming $T = 2$) input symbols as candidates (dashed lines in sub-figure (b)) and compares their counter values. A solid line in sub-figure (c) connects e_4 with the final decision s_3 that has a lower counter value than s_2. Then, another selection repeats the procedure in sub-figure (d) and (e).

The procedure of CC codes is described as follows:

- Parameters
  - $(s_1, s_2, \ldots, s_k)$: input symbols

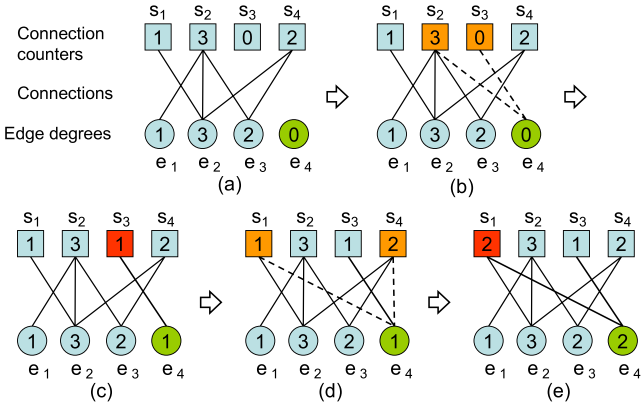

Fig. 1 An example of tournament selection. The scenario here is to generate a new output symbol e_4 of which the degree is 2.

  - $(c_1, c_2, \dots, c_k)$: connection counters
  - $\pi(d)$: degree distribution
  - T : tournament size
  - e : new output symbol

- Procedure to generate an output symbol
  - Step 1) Sample a degree d from the distribution $\pi(d)$
  - Step 2) For $i = 1, \dots, d$ do
    - Step 2.1) Generate a random number sequence $(r_1, r_2, \dots, r_T)$ to mark T input symbols $(s_{r_1}, s_{r_2}, \dots, s_{r_T})$ as connection candidates
    - Step 2.2) Find the symbol with the minimum connection count, say, s_{m_i}; i.e., $c_{m_i} = \min(c_{r_1}, c_{r_2}, \dots, c_{r_T})$.
    - Step 2.3) If s_{m_i} has already been selected, discard s_{m_i} and go to Step 2.1
    - Step 2.4) Update the connection count $c_{m_i} = c_{m_i} + 1$
  - Step 3) Output $e = s_{m_1} \bigoplus s_{m_2} \dots \bigoplus s_{m_d}$

In step 2.1, whether or not T candidates are distinct create two variants of tournament selection. The little difference between the two implementations could be ignored when $T \ll k$. Both methods can be used for CC codes, and duplicated candidates are allowed in this paper for convenience on analysis. If there is a tie in step 2.2, a solution to break it is to select one symbol uniformly at random among the symbols with the same count value. Connection counters record the encoding history of input symbols. They help to identify the symbols that have fewer connections and should be selected with a higher probability. As a result, the connections distributed on input symbols are equalized.

2.2 Probability of Isolation

Luby has interpreted the process of connection construction as a ball-bin problem [1]. If there are k input symbols and N connections in the Tanner graph, encoding can be imagined as throwing N balls into k bins uniformly at random. The ball-bin model is also widely used to study the situations of job allocation or supermarket queuing. Tournament selection can effectively reduce the variance of number of balls in each bin and solve the problem of load balancing. The literature [8] gives the proof of that the upper bound of maximum loading is greatly reduced by the greedy selection strategy. For a similar effect, we introduce tournament selection into LT codes to reduce the probability of an isolated information node in the Tanner graph or an empty bin in the ball-bin model. The probability of a particular bin is empty can be calculated as

$$(1 - 1/k)^N \approx e^{-N/k} = e^{-K\times\overline{d}/k}, \tag{1}$$

where the total connections N is expressed by the number of received K output symbols and average degree $\overline{d}$ of the adopted degree distribution. Suppose the reception overhead is small and $K \approx k$,

$$e^{-K\times\overline{d}/k} \approx e^{-\overline{d}}. \tag{2}$$

Hence, the probability depends solely on the average degree when the number of received output symbols roughly equals the size of input symbols. For the same situation in the case of tournament selection, T candidates are chosen uniformly at random before the actual input symbol is determined to be encoded. It can be classified into two cases if an input symbol is not chosen in one tournament selection event. First, the input symbol is not included in the T candidates. Second, there is a tie and another symbol is chosen. The second case occurs at most $k-1$ times. If an input symbol is never chosen to be encoded, it must be missed in at least $(K \times \overline{d} - (k-1)) \times T$ random selections. Thus, we can derive the upper bound of the probability that an input symbol is never encoded as

$$(1 - 1/k)^{(K\times\overline{d}-(k-1))\times T} \approx e^{-(\overline{d}-1)\times T}. \tag{3}$$

Such a result indicates that tournament size T exponentially influences the probability, and therefore the probability can be reduced by increasing the average degree $\overline{d}$ or the new parameter, tournament size T.

3. Simulation

For observation, simulations are conducted in this section to demonstrate the performance of CC codes. Connection choice being able to reduce the isolation probability means that degree distributions with a lower average degree might become feasible. Because we have no intention to introduce more new elements into the coding scheme, for simplicity and convenience, ideal soliton distributions with a shorter length are adopted in the simulation.

3.1 Short Ideal Soliton Distribution

Ideal soliton distribution, $\rho_k(d)$, for input symbols size k is

$$\rho_k(d) = \begin{cases} \frac{1}{k} & \text{for } d = 1 \\ \frac{1}{d(d-1)} & \text{for } d = 2, 3, \dots, k \end{cases}.$$

The average degree of $\rho_k(d)$ is the sum of the harmonic series up to k, $H(k) \approx \ln(k)$. The length, k, influences the probability of degree one and average degree of ISD. Since CC codes can adopt distributions with lower average degrees, a shorter ISD form with the parameter of $\ln(k)$ instead of k is considered, denoted as *Short ideal soliton distribution* (SISD) $\eta_k(d)$ and given as

$$\eta_k(d) = \begin{cases} \rho_u(d) & \text{if } d \le u \\ 0 & \text{otherwise} \end{cases}, \text{ where } u = \lceil \ln(k) \rceil.$$

SISD follows the form of ISD, and we bound the maximum encoding degree by $\lceil \ln(k) \rceil$. The average degree of $\eta_k(d)$ is hence less than $\ln(\ln(k))$.

According to Sect. 2.2, the probability of an input symbol never being encoded approximates $e^{-\overline{d}}$ for LT codes and can be adjusted to $e^{-(\overline{d}-1)\times T}$ for CC codes. We can compute a value of T such that CC codes would have roughly the same probability as LT codes. Let $\overline{d}_{\rho_k}$ be the average degree of ISD, $\overline{d}_{\eta_k}$ be the average degree of SISD. By letting $e^{-(\overline{d}_{\eta_k}-1)\times T} = e^{-\overline{d}_{\rho_k}}$, we have

$$\begin{aligned} e^{-(\ln(\ln(k))-1)\times T} &= e^{-\ln(k)} \\ T &= \ln(k)/(\ln(\ln(k))-1), \end{aligned}$$

which gives a guideline to decide T when SISD is adopted to work in CC codes. It should be noted that there are lots of approximation in the calculation and it gives a sign that the tournament size grows very slowly to provide the effect for large sizes of k, meaning that little extra cost is required to use tournament selection.

3.2 Simulation Results

To examine the performance of CC codes, the simulation results for input symbols size $k = 1000$ are presented in this section. The performance here is defined as the average reception overhead required for a successful recovery. We repeat the complete encoding/decoding process to obtain the performance indicator. Our simulation includes LT codes adopting robust soliton distributions and CC codes adopting short ideal soliton distribution with different tournament sizes. Moreover, the analysis of RSD gives the upper bound of the error probability of LT codes, and we also know that RSD is not the optimum for a finite size (i.e., $k < \infty$). Many studies [9]–[12] made attempts to optimize the degree distribution for LT codes with finite length. [10] even developed an approach to obtain the optimal degree distribution, while the results are unfortunately limited within $k < 30$ due to the very high order of computational complexity. In order to emphasize the better performance of CC codes, the best degree distribution listed in the literature [12] for $k = 1000$ is included in the comparison. The best degree distribution is named as "BDD" in Table 1.

A pure channel without erasure was implemented and all data points are averaged over 10^6 independent simulation runs. Figure 2 shows the successful full recovery rate under different receiving overheads. The successful full recovery rate is the percentage of runs that source data have fully uncovered over the 10^6 runs. In the figure, LT codes with RSD present the expected performance consistent with the analysis in the literature, BDD obviously outperform RSD, and the results of CC codes depend on the tournament size. According to the derivation in Sect. 2.2, an applicable tournament size T should be around $\ln(k)/(\ln(\ln(k))-1) \approx 7$ for $k = 1000$. In the experiment, the average degree of ISD($\rho_k(d)$) and SISD($\eta_k(d)$) is respectively 7.49 and 2.6. Tournament size more than $7.49/(2.6-1) = 4.68$ ensures a sufficiently small probability of isolated nodes. Hence, CC codes with tournament size $T = 5$ delivers better performance than both LT codes with RSD and BDD.

Table 1 The known best degree distribution from the literature [12] is denoted as "BDD" and the three customized degree distributions with different characteristics for CC codes are "CCD's." All the degree distributions are designed for LT codes with message size $k = 1000$.

Degrees	BDD	CDD1	CDD2	CDD3
1	0.1297	0.0636	0.0798	0.1281
2	0.2661	0.4261	0.3947	0.2471
3	0.3215	0.3586	0.2975	0.4235
5	0.0770	0.0247	0.1391	0.0770
8	0.1245	0.0472	0.0027	0.0074
13	0.0003	0.0328	0.0077	0.0656
21	0.0196	0.0291	0.0556	0.0030
34	0.0336	0.0021	0.0000	0.0075
55	0.0154	0.0000	0.0018	0.0221
89	0.0010	0.0091	0.0000	0.0053
144	0.0001	0.0043	0.0211	0.0011
233	0.0008	0.0023	0.0000	0.0122
377	0.0104	0.0000	0.0000	0.0000
$\overline{d}$	9.6111	5.5764	6.8811	8.1931

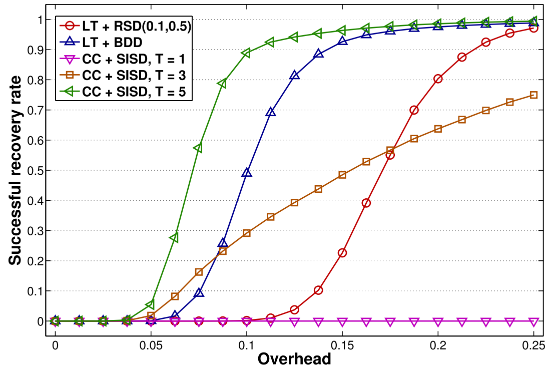

Fig. 2 Simulation results on the successful recovery rate.

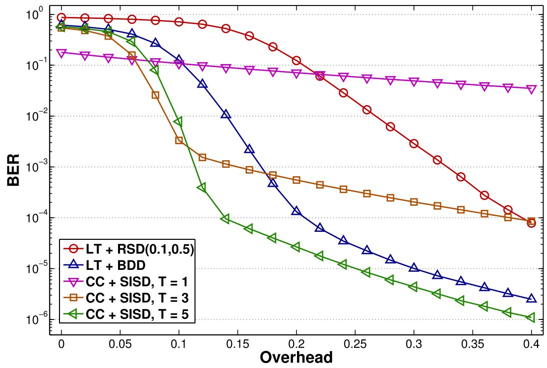

Fig. 3 Simulation results on the bit error rate.

Figure 3 examines the simulation results from a different aspect, the bit error rate (BER), which represents the ratio of unsolved input symbols to all input symbols. A common behavior is shown where the curves go down rapidly to their respective error floors when a sufficient reception overhead is received. The behavior reflects the two conditions described in Sect. 1. High error rate before the drop is due to the failure of belief propagation. Many input symbols are still unsolved even though the number of received output symbols has been greater than k. Once the chain reaction of belief propagation occurs, most symbols can be recovered except for the symbols in a trapping set, which causes error floors of error correcting codes. A trapping set may be attributed by the input symbols never encoded or a non-vanishing cycle. CC codes integrate the tournament selection and lower node degrees. Most output symbols with lower degrees is advantageous for belief propagation. The error rate of CC codes with appropriate T size can fall early in the figure. In addition, tournament selection contributes to eliminate possible isolation nodes and effectively drop the error floor. We can obtain the same conclusion from the observation in Fig. 3 that the error floor of CC codes reduces as tournament size increases.

On both performance indicators, the successful decoding rate and the bit error rate, CC codes show a better performance than LT codes with RSD and even the known best degree distribution to the our limited knowledge. Although the modification in CC codes is confined within the mechanism of symbol selection, the encoding behavior becomes too complicated to be analyzed by using a method similar to that used in [1]. Consequently, we consider an indirect approach to analyze CC codes in next section.

4. Investigation

Many theoretical studies have been proposed to demonstrate the performance of LT codes with particular degree distributions. Most of them are based on estimating the size of ripples. However, the method does not seem to work in the case of CC codes because encoding symbols are not independent anymore. To analyze CC codes, a series generation of encoding symbols should be considered in the same time. For the situation, we look for help from an convenient analysis tool called And-Or tree analysis [13], also introduced by Luby. The tool provides an intuitive framework to carry out the analysis for random processes. In error correction, it was utilized to analyze the failure probability while belief propagation works on a Tanner graph in which degree distributions of information and check nodes are both known. This section will describe how to obtain the degree distribution of input symbols in CC codes, and then different degree distributions of output symbols are examined by And-Or tree analysis.

4.1 Degree Distribution of Input Symbols

Before the use of And-Or tree analysis, it is necessary to know the node degree distribution of the Tanner graph. The degree distribution of check nodes is user-defined in both LT codes and CC codes. In contrast, the degree distribution of information nodes is a binomial distribution since LT codes build the connections uniformly at random, while it is not so intuitive with tournament selection. The history of the ball-bin process influences the location of the next ball. It is extremely difficult to consider all the possibilities and compute accurate results. As a secondary solution, we develop a dynamic programming algorithm to approximate the distribution of the ball-bin model with tournament selection. Let X be the random variable of the number of balls in a bin, and $f_n(x)$ denotes the probability mass function of X after totally n balls were thrown. Consider $f_n(x)$ as the probability function at n-th moment. At beginning of the moment, all bins are empty and we set the initial probability function as $f_0(0) = 1$. When balls are thrown in bins one by one, we estimate the change based on current probability function and update it for the next moment. For large bins size k, the probability, $f_n(x)$, means a proportion of bins which have x balls. The proportion may increase or decrease depends on the location of the new ball at this moment. Let $Pm_n^T(x)$ presents the probability that the minimum number of balls in T candidates bins is x at the n-th moment; in other words, the probability means that the $n+1$-th ball will be allocated in a bin which has x balls. Therefore, the proportion of bins with x balls will decrease with probability $Pm_n^T(x)$ and increase with $Pm_n^T(x-1)$ as

$$f_{n+1}(x) = f_n(x) - Pm_n^T(x)/k + Pm_n^T(x-1)/k. \qquad (4)$$

To complete the dynamic programming, the remaining job is to calculate the probability, $Pm_n^T(x)$, which can be considered as that the number of balls in all candidate bins are greater than x and at least one candidate has x balls. Given the cumulative distribution function of X, $F_n(x) = \sum_{i=1}^{x} f_n(i)$, the probability can be calculated as

$$Pm_n^T(x) = (1 - F_n(x-1))^T - (1 - F_n(x))^T. \qquad (5)$$

However, another important factor not yet considered is the erasure rate of communication channel. The function of tournament selection is based on the history of encoding processes. If erasure occurs, the sender and receiver will have asynchronous state of node degree distributions. Assume that the event of erasure occurs with erasure rate p, the $1/(1-p)$ times of encoding symbols should be sent out for receiver that there is sufficient information to complete the recovery. Following such a situation, we reform the ball distribution in the ball-bin model by throwing $N/(1-p)$ balls and considering that each ball may be erased with probability p. First, the proportion, p, of $N/(1-p)$ balls will be erased so there is totally N balls on average. Second, a bin with x balls will decline according to the binomial distribution $B(x, p)$. More precisely, the probability of a bin with x balls reducing to i is $\binom{x}{i} \cdot p^{x-i} \cdot (1-p)^i$. To integrate the steps in this section, the procedure can help us to estimate the ball distribution in the ball-bin model for any given four parameters, N balls, k bins, tournament size T, and erasure rate p. Figure 4 shows the examples for ball-bin model with different parameters. Involving tournament selection effectively reduces the variance of the distribution, but the randomness of the erasure presents a counter force to reform the result back to the binomial distribution. The erasure rate may influence the performance of CC codes.

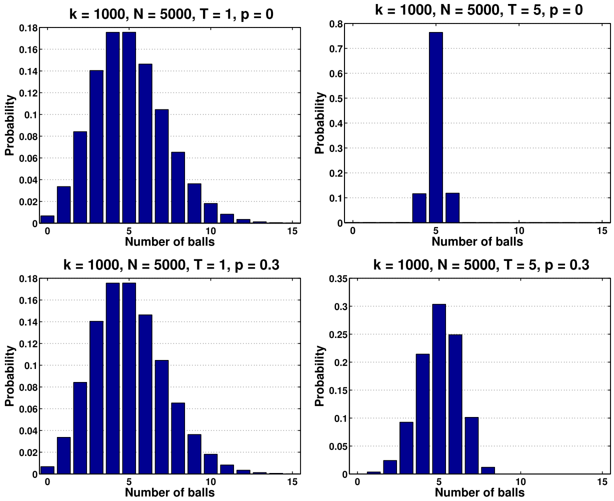

Fig. 4 The ball distributions in the ball-bin model with different parameters.

4.2 And-Or Tree Analysis

To make a further study on CC codes, And-Or tree analysis is utilized to explain the reason why some degree distributions perform better. And-Or tree analysis measures the failure rate of belief propagation by iteratively analyzing the greedy edge punning in a Tanner graph. Edges in a Tanner graph always connect an information node and a check node. We define the left/right degree of an edge is the information node degree or check node degree of the edge. Given the degree distribution of information/check nodes, $\Omega(x)/\Lambda(x)$, degree distributions of edges can be denoted as $\omega(x) = \frac{\Omega'(x)}{\Omega'(1)}$ and $\lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)}$, which respectively represent the distribution of information node degree and check node degree of edges. According to the edge degree distributions, And-Or tree analysis yields the iterative equation as

$$y_{l+1} = \lambda(1 - \omega(1 - y_l)), \tag{6}$$

where y_l denotes the ratio of unpacked edges at the l-th iteration, and the boundary condition is $y_0 = 1$. Let $y = \lim_{l\to\infty} y_l$, we expect that the result of y will converge to 0 after iterative decoding. To achieve it, the iterative equation must satisfy the critical condition, $y_{l+1} < y_l$ for $y_l \in (0, 1]$.

The condition indicates a criterion to examine the ability of error correction codes which employ an iterative decoding algorithm, including CC codes. When the degree distribution of output symbols $\Lambda(x)$ and a constant of reception overhead ϵ are decided, the total number of edges is $N = k \cdot \epsilon \cdot \Lambda'(1)$. By the procedure given in Sect. 4.1, the degree distribution of input symbols $\Omega(x)$ can be estimated with the given tournament size T and erasure rate p. Finally, we can build the iterative function for any particular degree distribution of output symbols and observe the influence of parameter T and p. Figure 5 gives the And-Or tree analysis result of the normalized function that $f(y_l) = y_{l+1}/y_l$. The dotted line, $y_{l+1}/y_l = 1$, presents the fundamental criterion for a successful decoding. It is obvious in the figure that RSD exceeds the based line and cannot finish the decoding when the reception overhead is only 10%. Both BDD and SISD show reasonable curves for successful recovery. We can find that the effect of tournament selection may be offset as the increase of erasure rate. It seems that CC codes with SISD only works with little erasure and more reliable degree distributions for CC codes are in need.

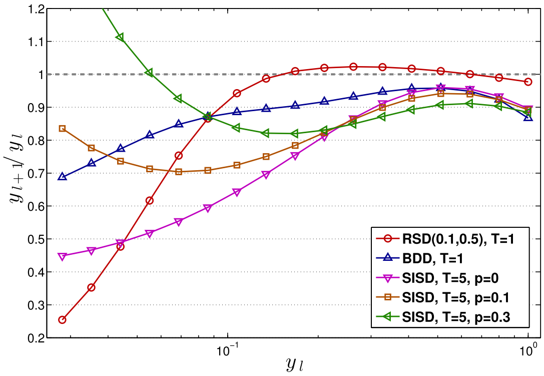

Fig. 5 The influence of different erasure rates is investigated for CC codes.

5. Customization

Since theoretical analysis on CC codes is difficult, a general form of degree distributions cannot be derived for the time being. In this study, we employ techniques from heuristic optimization and present the obtained distribution instances to confirm the applicability of CC codes. In the literature[12], the source of BDD, an evolutionary algorithm was introduced to optimize sparse degree distributions for LT codes. The optimization framework can also be used for CC codes to search for good degree distributions. By the estimation given in the ball-bin model, tournament selection brings the advantage to make different degree distributions of input symbol possible, but degree distributions will be affected by channel erasure. CC codes trade the universal property for better performance and can be customized/optimized in certain range of erasure rates. There is a trade-off between the minimum reception overhead and practical range of the erasure rate. To demonstrate the trade-off, three customized degree distributions named CDD1, CDD2, and CDD3 are given in Table 1 and different characteristics of them are shown in the Fig. 6. The figure plots the performance variance for different erasure rates. The results of instances without tournament selection ($T = 1$) are two horizontal lines which indicate the universal property that performance is independent of the erasure rate. In contrast, the other instances seek chances to further reduce the reception overhead within a range of erasure rates. To compare with SISD, the performance of CCD's is less sensitive and the same result can be obtained by And-Or tree analysis. Figure 7 presents the And-Or tree analysis with reception overhead $\epsilon = 0.05$ for BDD and CDD1. The performance

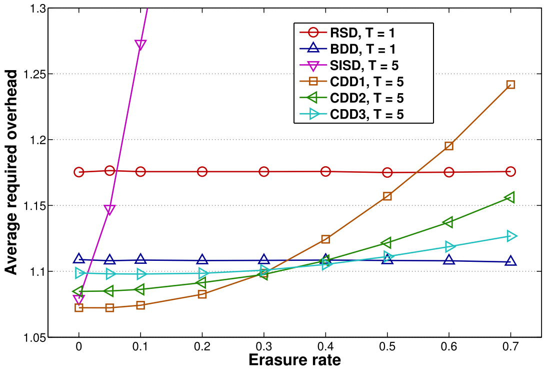

Fig. 6 The performance variance for different erasure rates.

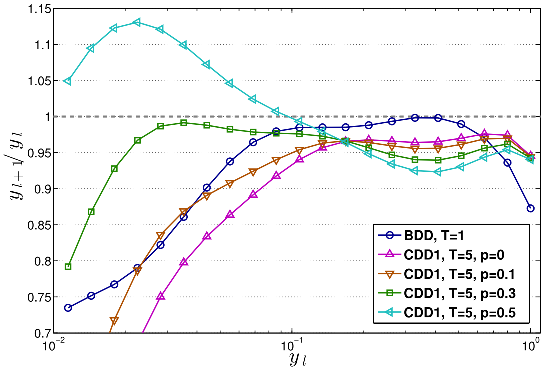

Fig. 7 The influence of different erasure rates for CDD1.

of CCD1 is still comparable when the erasure rate increases up to 0.3.

6. Conclusions

The main contribution of this study is the proposal of CC codes, which are more general than LT codes within the realm of rateless codes. The parameter T creates a new dimension for controlling the coding behavior. Figure 8 shows some well-known coding methods as instances of CC codes on a two-dimensional plane formed by the average degree of the adopted degree distribution and the randomness of connections. Average degree equal to one means that no encoding operation is executed and tournament size affects the randomness of output symbol generation from fully random to sequential. LT codes represent the class with the maximum randomness. Linear random fountain codes [14] are a simple implementation of rateless codes in which each input symbol has a probability of 0.5 to be chosen for encoding. The figure shows that all these codes may be considered as instances of CC codes with different combinations of average degree and tournament size.

The simulation results are presented to illustrate the performance of CC codes with a preliminary design of degree distribution. Even though tournament selection makes CC codes lose the universal property when tournament size is greater than one, heuristic algorithms proposed in the literature were employed to search for good distribution instances to work with CC codes. Three customized degree distributions were presented to demonstrate the potentiality of CC codes that the better performance than LT codes could be achieved within certain range of erasure rates. As a result of our observation, it is a trade-off, seeking better performance or fault tolerance, presented to the system designer by CC codes. If the knowledge of the communication channel is available or the service provider can estimate the distribution of end users' erasure rates, CC codes can be used to aim at a particular range of erasure rates and the adopted degree distribution may be optimized for most users. Although some block codes can be optimized for a given erasure rate, CC codes can provide good performance over a range of erasure rates, instead of one given value, as well as all the advantages of rateless codes and digital fountain codes. Moreover, the flexibility of CC codes allow the possibility of a better inner codes. CC codes could also be customized to serve as a new component and to cooperate with the pre-coder in Raptor codes or with even more recent, advanced pre-coders [15]. The improvement of the inner codes would benefit these state-of-the-art coding schemes.

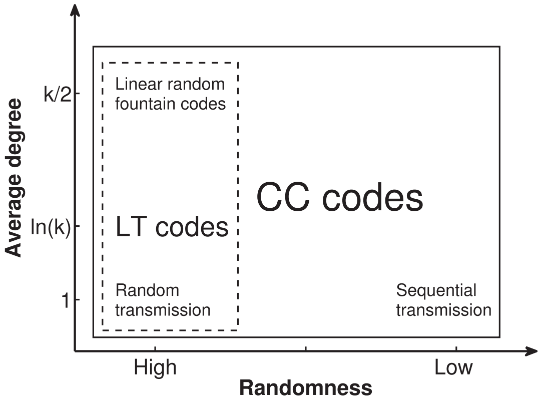

Fig. 8 The relation between CC codes and well-known coding methods.

Acknowledgments

The work was supported in part by the National Science Council of Taiwan under Grant NSC 99-2221-E-009-123-MY2. The authors are grateful to the National Center for High-performance Computing for computer time and facilities.

References

[1] M. Luby, "LT codes," Proc. 43rd Symposium on Foundations of Computer Science, pp.271–280, 2002.

[2] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach," Proc. American Association of Artificial Intelligence National Conference on AI, pp.133–136, 1982.

[3] A. Shokrollahi, "Raptor codes," Proc. International Symposium on Information Theory, p.36, 2004.

[4] A. Shokrollahi, "Raptor codes," IEEE Trans. Inf. Theory, vol.52, no.6, pp.2551–2567, 2006.

[5] A. Brindle, Genetic algorithms for function optimization (Doctoral dissertation and Technical Report TR81-2), Ph.D. thesis, 1981.

[6] D. Goldberg, B. Korb, and K. Deb, "Messy genetic algorithms: Motivation, analysis, and first results," Complex Systems, vol.3, no.5, pp.493–530, 1989.

[7] M. Mitzenmacher, "The power of two choices in randomized load balancing," IEEE Trans. Parallel Distrib. Syst., vol.12, pp.1094–1104, 2001.

[8] Y. Azar, A.Z. Broder, A.R. Karlin, and E. Upfal, "Balanced allocations," SIAM J. Comput., vol.29, no.1, pp.180–200, 1999.

[9] E. Hyytiä, T. Tirronen, and J. Virtamo, "Optimizing the degree distribution of LT codes with an importance sampling approach," Proc. 6th InternationalWorkshop on Rare Event Simulation (RESIM 2006), pp.64–73, 2006.

[10] E. Hyytiä, T. Tirronen, and J. Virtamo, "Optimal degree distribution for LT codes with small message length," Proc. 26th IEEE International Conference on Computer Communications (INFOCOM 2007), pp.2576–2580, 2007.

[11] E.A. Bodine and M.K. Cheng, "Characterization of Luby Transform codes with small message size for low-latency decoding," Proc. IEEE International Conference on Communications, pp.1195–1199, 2008.

[12] C.M. Chen, Y.p. Chen, T.C. Shen, and J. Zao, "On the optimization of degree distributions in LT codes with covariance matrix adaptation evolution strategy," Proc. 2010 IEEE Congress on Evolutionary Computation (CEC 2010), pp.3531–3538, 2010.

[13] M.G. Luby, M. Mitzenmacher, and M.A. Shokrollahi, "Analysis of random processes via and-or tree evaluation," Proc. ninth annual ACM-SIAM symposium on Discrete algorithms, SODA'98, pp.364–373, Society for Industrial and Applied Mathematics, 1998.

[14] D.J.C. MacKay, "Fountain codes," IEE Proc. Commun., vol.152, no.6, pp.1062–1068, 2005.

[15] K. Kasai, D. Declercq, and K. Sakaniwa, "Fountain coding via multiplicatively repeated non-binary LDPC codes," IEEE Trans. Commun., vol.60, no.8, pp.2077–2083, 2012.

Chih-Ming Chen received the B.S. degree in 2006 and currently studies for the Ph.D. degree in Computer Science at National Chiao Tung University, Taiwan.

Ying-ping Chen is currently an Associate Professor in the Department of Computer Science, National Chiao Tung University, Taiwan. His research interests include data grid and MapReduce technologies in distributed computation as well as theories, working principles, and dimensional/facet-wise models in genetic and evolutionary computation. He received the B.S. degree and the M.S. degree in Computer Science and Information Engineering from National Taiwan University, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree in 2004 from the Department of Computer Science, University of Illinois at Urbana-Champaign, Illinois, USA.