Algorithm 2: SCL_D($u_{\mathcal{I}_{X|Y}}, y_0^{N-1}, L$)

input : $u_{\mathcal{I}_{X|Y}}$: Codeword, y_0^{N-1}: Side information, L: List size
output: $\widehat{x}_0^{N-1}$: Reconstructed sequence

// $\mathcal{L}_i$: The set of active paths at phase i.
for $i = 0, 1, \ldots, N-1$ **do**
  if $i \in \mathcal{I}_{X|Y}$ **then**
    Append u_i to each $\widehat{u}_0^{i-1}[l] \in \mathcal{L}_{i-1}$, and obtain $(\widehat{u}_0^{i-1}[l], u_i)$
  else
    Append all $\widehat{u}_i \in \mathcal{X}$ to each $\widehat{u}_0^{i-1}[l] \in \mathcal{L}_{i-1}$;
    Calculate $P_N^{(i)}(\widehat{u}_0^{i-1}[l], \widehat{u}_i | y_0^{N-1})$ for all $(\widehat{u}_0^{i-1}[l], \widehat{u}_i) \in \mathcal{L}_i$;
    Prune all but L paths with highest probabilities.
Return $\widehat{x}_0^{N-1} = (\underset{\widehat{u}_0^{N-1} \in \mathcal{L}_{N-1}}{\operatorname{argmax}} P_N^{(N-1)}(\widehat{u}_0^{N-1} | y_0^{N-1}))\mathbf{G}_N^{-1}$.

2.4 Code Construction

In polar coding, it is assumed that the information set, $\mathcal{I}_{X|Y}$, is known at both encoder and decoder. In order to construct $\mathcal{I}_{X|Y}$ as in (2.5), the set of conditional entropies $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$ must be available. Except a small class of examples, e.g., binary erasure channels in channel coding, analytic solutions do not exist for computing conditional entropy values. As a solution, density evolution is proposed for computing $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$, and constructing $\mathcal{I}_{X|Y}$ for any generic distribution $p_{X,Y}(x, y)$ [13, 14]. In this section, a greedy approximation algorithm for computing $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$ using density evolution is proposed for prime-size alphabets.

2.4.1 Density Evolution

Let $\mathbf{p} = (p_0, p_1, \ldots, p_{q-1}) \in \mathbb{R}^q$ be a q-dimensional probability vector, i.e., $p_i > 0$, $\forall i$ and $\sum_{i=0}^{q-1} p_i = 1$. q-ary entropy function is defined as follows:

$$\mathcal{H}(\mathbf{p}) = \sum_{i=0}^{q-1} p_i \log \frac{1}{p_i} \tag{2.10}$$

For $f : \mathcal{X} \mapsto \mathbb{R}$, a function defined on $\mathcal{X} = \{0, 1, \dots q-1\}$, $[f(x)]_{x=0}^{q-1}$ represents q-dimensional vector formed by the values of f:

$$[f(x)]_{x=0}^{q-1} = [f(0), f(1), \dots, f(q-1)]$$

Let (X, Y) be a pair of random variables over $\mathcal{X} \times \mathcal{Y}$, $\mathcal{X} = \{0, 1, \dots, q-1\}$, with joint distribution $p_{X,Y}(x, y)$. The conditional entropy $H(X|Y)$ is computed as follows:

$$\begin{aligned} H(X|Y) &= \sum_{y\in\mathcal{Y}} p_Y(y) \sum_{x\in\mathcal{X}} p_{X|Y}(x|y) \log \frac{1}{p_{X|Y}(x|y)} \\ &= \sum_{y\in\mathcal{Y}} p_Y(y)\mathcal{H}([p_{X|Y}(x|y)]_{x=0}^{q-1}) \end{aligned} \tag{2.11}$$

Hence, for the computation of $H(X|Y)$, the set of $(q+1)$-dimensional vectors $\{(p_Y(y), [p_{X|Y}(x|y)]_{x=0}^{q-1})\}_{y\in\mathcal{Y}}$ is needed. In density evolution, the objective is to obtain $\{(p_{U_0^{i-1},Y_0^{N-1}}(u_0^{i-1}, y_0^{N-1}), [p_{U_i|U_0^{i-1},Y_0^{N-1}}(u|u_0^{i-1}, y_0^{N-1})]_{u=0}^{q-1})\}_{u_0^{i-1}\in\mathcal{X}^i, y_0^{N-1}\in\mathcal{Y}^N}$ by evolving the distributions of the source through the polar transform so that $H(U_i|U_0^{i-1}, Y_0^{N-1})$ can be computed.

Probability distributions of the pair of random variables (X, Y) are expressed in a list as follows:

$$X|Y \sim \sum_{y\in\mathcal{Y}} p_Y(y)Q([p_{X|Y}(x|y)]_{x=0}^{q-1}) \tag{2.12}$$

Remark: Consider a random variable Z that is independent from X and Y. Then, by notation, $X|Y = X|Y, Z$ since for a fixed y, $[p_{X|Y,Z}(x|y, z)]_{x=0}^{q-1} = [p_{X|Y}(x|y)]_{x=0}^{q-1}$ for all $z \in \mathcal{Z}$, and $p_{Y,Z}(y, z) = p_Y(y)p_Z(z)$ for all y, z, which together imply that Z has no effect in the computation of conditional entropy.

In order to investigate how input distributions are transformed in polar transformation, let us consider the basic transform given in Figure 2.5 first. Assume that $|\mathcal{X}| = q$ is a prime number and (X_0, Y_0), (X_1, Y_1) are independently drawn from $p_{X,Y}(x, y)$, i.e., they can be expressed identically in the form of 2.12.

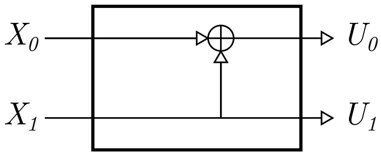

Figure 2.5: Basic polar transform.

The first operation is defined as follows:

$$\begin{aligned} U_0|Y_0,Y_1 &\sim \sum_{y_0,y_1\in\mathcal{Y}} p_{Y_0,Y_1}(y_0,y_1)Q([p_{X_0\oplus X_1|Y_0,Y_1}(x|y_0,y_1)]_{x=0}^{q-1}) \qquad (2.13)\\ &\triangleq \sum_{y_0\in\mathcal{Y}} p_Y(y_0)Q([p_{X|Y}(x|y_0)]_{x=0}^{q-1}) \circledast \sum_{y_1\in\mathcal{Y}} p_Y(y_1)Q([p_{X|Y}(x|y_1)]_{x=0}^{q-1}) \end{aligned}$$

Proposition 2. *The expression in (2.13) can be written in the following form:*

$$U_0|Y_0,Y_1 \sim \sum_{y_0,y_1} p_Y(y_0)p_Y(y_1)Q([\sum_{z=0}^{q-1} p_{X|Y}(z|y_0)p_{X|Y}(x\ominus z|y_1)]_{x=0}^{q-1})$$

Proof. Since (X_0,Y_0) and (X_1,Y_1) are independent and identically distributed, $p_{Y_0,Y_1}(y_0,y_1) = p_Y(y_0)p_Y(y_1)$ for all $y_0,y_1\in\mathcal{Y}$, and the probability distribution $p_{X_0\oplus X_1|Y_0,Y_1}(x|y_0,y_1)$ can be written as the convolution of $p_{X|Y}(x|y_0)$ and $p_{X|Y}(x|y_1)$. □

By using the remark, $U_0|Y_0,Y_1$ can be expressed as $(X_0|Y_0,Y_1)\oplus(X_1|Y_0,Y_1) = X_0\oplus X_1|Y_0,Y_1$. Note that $\circledast$ is a commutative operation.

The second operation is defined as follows:

$$\begin{aligned} U_1|U_0,Y_0,Y_1 &\sim \sum_{z\in\mathcal{X}}\sum_{y_0,y_1\in\mathcal{Y}} p_{X_0\oplus X_1,Y_0,Y_1}(z,y_0,y_1)Q([p_{X_1|X_0\oplus X_1,Y_0,Y_1}(x|z,y_0,y_1)]_{x=0}^{q-1})\\ &\triangleq \sum_{y_0\in\mathcal{Y}} p_Y(y_0)Q([p_{X|Y}(x|y_0)]_{x=0}^{q-1}) \boxast \sum_{y_1\in\mathcal{Y}} p_Y(y_1)Q([p_{X|Y}(x|y_1)]_{x=0}^{q-1}) \end{aligned}$$

Proposition 3. *The operation $\boxast$ can be written in the following form:*

$$U_1|U_0,Y_0,Y_1 \sim \sum_z\sum_{y_0,y_1}\left([\sum_{u=0}^{q-1} p_{X|Y}(u|y_0)p_{X|Y}(z\ominus u|y_1)]p_Y(y_0)p_Y(y_1)\right)\cdot Q(\left[\frac{p_{X|Y}(x|y_1)p_{X|Y}(z\ominus x|y_0)}{[\sum_{u=0}^{q-1} p_{X|Y}(u|y_0)p_{X|Y}(z\ominus u|y_1)]}\right]_{x=0}^{q-1})$$

The proof is similar to Proposition 2.

Note that polar transform is a successive application of the basic transform given in Figure 2.5. The following theorem states that $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$ can be computed by using the density evolution operations successively through the polar transform.

Theorem 2. *Let* $\{(X_i, Y_i)\}_{i=0}^{N-1}$ *be iid pairs of random variables with probability distribution* $p_{X,Y}(x, y)$. *Operations* $\circledast$ *and* $\boxast$ *applied successively through the polar transform suffice to obtain* $\{U_i|U_0^{i-1}, Y_0^{N-1}\}_{i=0}^{N-1}$.

Proof. The proof will be based on induction. Proposition 2 and Proposition 3 together prove the initial step, i.e., for n = 1, $H(U_0|Y_0^1)$ and $H(U_1|U_0, Y_0^1)$ can be computed by using density evolution. For the inductive step, consider 2.6. Assume that the hypothesis is true for $n-1$. In this case, the input variables $\{X_i|Y_i\}_{i=0}^{N/2-1}$ are transformed into $\{R_i|R_0^{i-1}, Y_0^{N-1}\}_{i=0}^{N/2-1}$, and, similarly, $\{X_i|Y_i\}_{i=N/2}^{N-1}$ are transformed into $\{S_i|S_0^{i-1}, Y_0^{N-1}\}_{i=0}^{N/2-1}$. At the last step of polar transform, these vectors are combined as in Figure 2.6.

For $i = 0, 1, \ldots, N/2-1$,

$$\begin{aligned}(R_i\big|R_0^{i-1}Y_0^{N/2-1}) \oplus (S_i\big|S_0^{i-1}Y_{N/2}^{N-1}) &= (R_i|R_0^{i-1}S_0^{i-1}Y_0^{N-1}) \oplus (S_i|R_0^{i-1}S_0^{i-1}Y_0^{N-1}) \\ &= R_i \oplus S_i|R_0^{i-1}S_0^{i-1}Y_0^{N-1} \\ &= U_{2i}|R_0^{i-1} \oplus S_0^{i-1}, S_0^{i-1}Y_0^{N-1} \\ &= U_{2i}|U_{0,e}^{2i-1}U_{0,o}^{2i-1}Y_0^{N-1} \\ &= U_{2i}|U_0^{2i-1}Y_0^{N-1}\end{aligned}$$

In the first line above, the fact that R_0^{i-1} is independent from S_0^{i-1} and $Y_{N/2}^{N-1}$ is used. The operations on the variables are matched to the density evolution operations similar to the $n = 1$ case. For odd indices,

$$\begin{aligned}S_i|S_0^{i-1}\Big|R_i \oplus S_i|R_0^{i-1}, S_0^{i-1}, Y_0^{N-1} &= S_i|R_i \oplus S_i, R_0^{i-1}, S_0^{i-1}, Y_0^{N-1} \\ &= S_i|U_{2i}, U_0^{2i-1}, Y_0^{N-1} \\ &= U_{2i+1}|U_0^{2i}, Y_0^{N-1}\end{aligned}$$

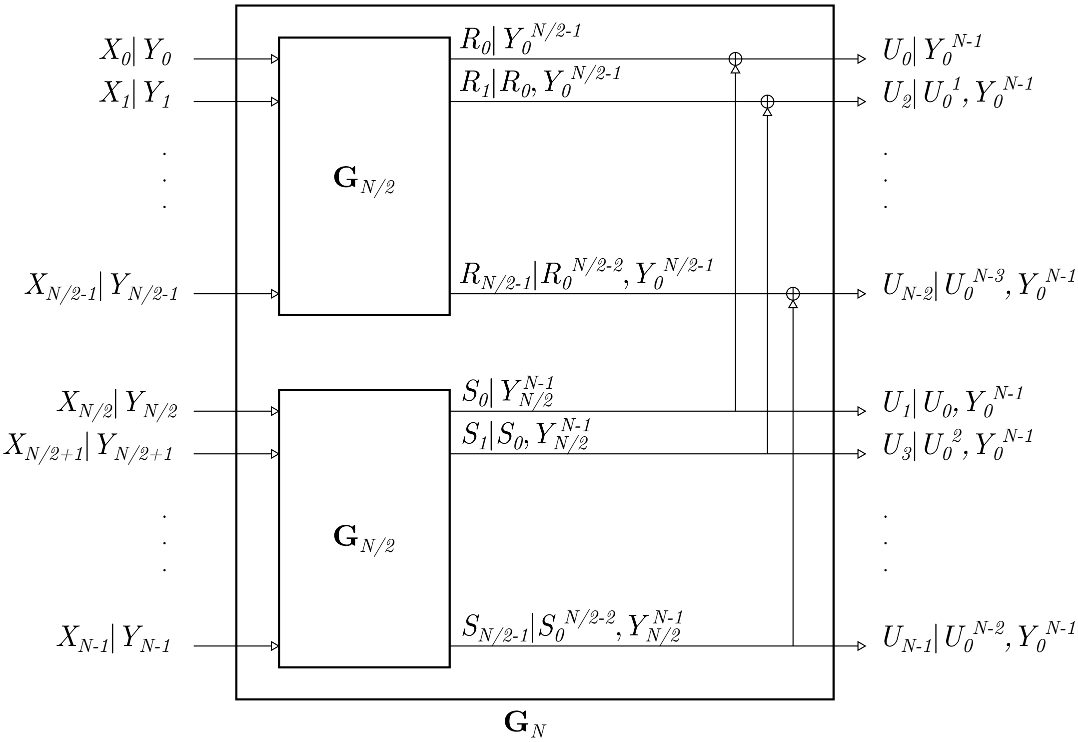

Figure 2.6: Density evolution at block-length N.

Hence, all conditional entropies can be computed using $\circledast$ and $\boxast$ operations. $\square$

Theorem 2 indicates that $\{U_i|U_0^{i-1}, Y_0^{N-1}\}_{i=0}^{N-1}$ can be computed by successive application of the density evolution operations. Next, the procedure to obtain $U_i|U_0^{i-1}, Y_0^{N-1}$ will be described. Let the right-hand side of (2.12) be denoted as $\chi_i^{(0)}$. In order to compute $U_i|U_0^{i-1}, Y_0^{N-1}$ where the binary expansion of i is $(b_0 b_1 \dots b_{n-1})_2$, the following procedure is applied [7]:

$$\chi_i^{(k+1)} = \begin{cases} \chi_i^{(k)} \circledast \chi_i^{(k)} & \text{, if} \quad b_k = 0 \\ \chi_i^{(k)} \boxast \chi_i^{(k)} & \text{, if} \quad b_k = 1 \end{cases} \tag{2.14}$$

for $k = 0, 1, \dots, n-1$. The output of this procedure is $\chi_i^{(n)}$ such that $U_i|U_0^{i-1}, Y_0^{N-1} \sim \chi_i^{(n)}$.

In order to estimate the complexity of density evolution, the growth in the alphabet sizes of the conditioned terms is used. In Figure 2.5, the alphabet sizes of the conditioned terms of $U_0|Y_0, Y_1$ and $U_1|U_0, Y_0, Y_1$ are $|\mathcal{Y}|^2$

and $|\mathcal{Y}|^2 q$, respectively. Note that if two different y, say y_1 and y_2, in the RHS of (2.12) have the same $[p_{X|Y}(x|y)]_{x=0}^{q-1}$, then they can be unified as $[p_Y(y_1) + p_Y(y_2)]Q([p_{X|Y}(x|y_1)]_{x=0}^{q-1})$, which implies that the alphabet sizes are upper bounds. At block-length N, the alphabet size becomes $O(q^{N-1})$. Thus, the direct application of density evolution in polar code construction infeasible. Approximation methods are proposed to solve this problem [14]. In the following subsection, approximation methods for q-ary code construction based on [14] will be proposed.

2.4.2 Greedy Approximation Algorithm for Code Construction

The basic idea in the approximation algorithm is to unify symbols $y, \bar{y} \in \mathcal{Y}$, whose unification changes the entropy of $X|Y$ minimally, successively until the alphabet size becomes lower than a given parameter μ. By such an approximation, the growth in the number of $(p_Y(y), [p_{X|Y}(x|y)]_{x=0}^{q-1})$ through the polar transform can be controlled, hence efficient code construction can be performed.

Assume that we have $X|Y \sim \sum_{y\in\mathcal{Y}} p_Y(y)Q([p_{X|Y}(x|y)]_{x=0}^{q-1})$ where $|\mathcal{Y}| > \mu$ for a given positive integer μ. Denote the conditional probability of X given a realization y by $\mathbf{p}_{X|y}$, i.e., $\mathbf{p}_{X|y} = [p_{X|Y}(x|y)]_{x=0}^{q-1}$. For $y, \bar{y} \in \mathcal{Y}$, the following operation is performed to reduce the alphabet size of the side information:

$$p_Y(y)Q(\mathbf{p}_{X|y}) + p_Y(\bar{y})Q(\mathbf{p}_{X|\bar{y}}) \mapsto (p_Y(y) + p_Y(\bar{y}))Q(\mathbf{p}) \tag{2.15}$$

for a valid probability vector $\mathbf{p}$. The new alphabet, $\widehat{\mathcal{Y}} = \mathcal{Y}\backslash\{y, \bar{y}\} \cup \{\widehat{y}\}$, has cardinality $|\mathcal{Y}| - 1$. In greedy approximation algorithm, symbol pairs $y, \bar{y}$ and unified distribution $\mathbf{p}$ are chosen in an intelligent way at each step, and the alphabet size is reduced by one. By successively applying the operation in (2.15), the alphabet size is reduced below the given parameter, i.e., $|\widehat{\mathcal{Y}}| < \mu$.

A different approximation algorithm is proposed for non-binary polar code construction in [18]. The main difference is that [18] involves parametrized quantization levels, whereas the method presented here is based on a greedy algorithm as in [14] and [19]. The algorithm that will be proposed here can be considered as a modification of the mass merging algorithm in [19]. For detailed description and analysis of mass merging and mass transportation algorithms, we refer to [19].

Let $X|Y \sim \chi$ be a source as defined in (2.12) with $\chi = \sum_{y\in\mathcal{Y}} p_Y(y)Q(\mathbf{p}_{X|Y}(x|y))$, $y, \bar{y} \in \mathcal{Y}$ be given symbols for merging, and $\gamma \in [0, 1]$. Merging of these masses with weight γ is the following transformation:

$$p_Y(y)Q(\mathbf{p}_{X|y}) + p_Y(\bar{y})Q(\mathbf{p}_{X|\bar{y}}) \mapsto (p_Y(y) + p_Y(\bar{y}))Q(\gamma\mathbf{p}_{X|y} + (1-\gamma)\mathbf{p}_{X|\bar{y}}) \quad (2.16)$$

This transformation corresponds to mass transportation if $\gamma = 0$ and $\gamma = 1$, and degrading approximation (and mass merging as defined in [19]) if $\gamma = \frac{p_Y(y)}{p_Y(y)+p_Y(\bar{y})}$. In order to define the approximation error due to (2.16), consider the following function:

$$\begin{aligned} f_{y,\bar{y}}(\gamma) = (p_Y(y) + p_Y(\bar{y}))\mathcal{H}(\gamma\mathbf{p}_{X|y} + (1-\gamma)\mathbf{p}_{X|\bar{y}}) \\ - \; p_Y(y)\mathcal{H}(\mathbf{p}_{X|y}) - p_Y(\bar{y})\mathcal{H}(\mathbf{p}_{X|\bar{y}}). \end{aligned} \quad (2.17)$$

The approximation error is defined as the change in the conditional entropy due to the mass merging transformation:

$$\epsilon_{y,\bar{y}}(\gamma) = |f_{y,\bar{y}}(\gamma)|. \quad (2.18)$$

For any $y, \bar{y} \in \mathcal{Y}$, the following proposition holds:

Proposition 4. *$f_{y,\bar{y}}(\gamma) = 0$ has exactly one root in the interval $(0, 1]$.*

Proof. The case $\mathcal{H}(\mathbf{p}_{X|y}) = \mathcal{H}(\mathbf{p}_{X|\bar{y}})$ is trivial: $\gamma = 1$ satisfy the claim. In order to prove that the proposition holds for the non-trivial case, first, it is proved that $f_{y,\bar{y}}$ is a concave function of $\gamma \in [0, 1]$. For $0 \leq \lambda, \gamma_1, \gamma_2 \leq 1$ and $\overline{\lambda} = 1 - \lambda$,

$$\begin{aligned} f_{y,\bar{y}}(\lambda\gamma_1 + \overline{\lambda}\gamma_2) &= [p_Y(y) + p_Y(\bar{y})]\mathcal{H}(\tilde{\mathbf{p}}) - \sum_{y'\in\{y,\bar{y}\}} p_Y(y')\mathcal{H}(\mathbf{p}_{X|y'}) \\ &\geq \lambda f_{y,\bar{y}}(\gamma_1) + \overline{\lambda} f_{y,\bar{y}}(\gamma_2) \end{aligned}$$

where $\tilde{\mathbf{p}} = \lambda(\gamma_1\mathbf{p}_{X|y} + \overline{\gamma}_1\mathbf{p}_{X|\bar{y}}) + \overline{\lambda}(\gamma_2\mathbf{p}_{X|y} + \overline{\gamma}_2\mathbf{p}_{X|\bar{y}})$. The inequality follows from the concavity of $\mathcal{H}$.

In the non-trivial case, $f_{y,\bar{y}}(\gamma)$ has different signs at the boundary points, $\gamma = 0$ and $\gamma = 1$. Since $f_{y,\bar{y}}$ is a concave and continuous function of $\gamma \in [0,1]$, this property implies that $f_{y,\bar{y}}$ has only one zero crossing on $[0,1]$. □

Proposition 4 implies that for any $y, \bar{y} \in \mathcal{Y}$, the approximation error $\epsilon_{y,\bar{y}}(\gamma)$ can attain its minimum value 0 by solving the logarithmic equation $f_{y,\bar{y}}(\gamma) = 0$ that has a unique solution on $[0,1]$. Fast-converging bisection method can be used to solve this problem [20]. Since this method is based on a greedy algorithm, the error in $H(U_i|U_0^{i-1}, Y_0^{N-1})$ due to the approximation cannot be analyzed easily. However, the approximation error $\epsilon_{y,\bar{y}}(\gamma)$ has a close relation to the error in the computation of $H(U_i|U_0^{i-1}, Y_0^{N-1})$ as numerical examples will indicate.

For a given source $X|Y \sim \chi$ and pair of symbols $y, \bar{y}$, mass merging operation is summarized in Algorithm 3.

Algorithm 3: merge$(y, \bar{y})$

input : $y, \bar{y}$: Symbols to be merged

1 Find γ^* such that $f_{y,\bar{y}}(\gamma^*) = 0$;
2 Set $\mathbf{p}_{X|y'} = \gamma^* \mathbf{p}_{X|y} + (1-\gamma^*)\mathbf{p}_{X|\bar{y}}$;
3 Set $\chi = \sum_{y'' \in \mathcal{Y} \setminus \{y,\bar{y}\}} p_Y(y'') Q(\mathbf{p}_{X|y''}) + [p_Y(y) + p_Y(\bar{y})] Q(\mathbf{p}_{X|y'})$.

Algorithm 3 provides the basic tool to reduce the alphabet size of the side information Y by one without changing the entropy of the source for any $y, \bar{y}$. Thereafter, the question of how to choose the symbols y_1, y_2 to be merged comes up. At this point, our consideration is to deviate a source as little as possible so that the effect of mass merging on $H(U_i|U_0^{i-1}, Y_0^{N-1})$ is kept small. For any $y, \bar{y} \in \mathcal{Y}$, the following error function is defined:

$$\widehat{\epsilon}_{y,\bar{y}} = \max(|p_Y(\bar{y})[\mathcal{H}(\mathbf{p}_{X|y}) - \mathcal{H}(\mathbf{p}_{X|\bar{y}})]|, |p_Y(y)[\mathcal{H}(\mathbf{p}_{X|\bar{y}}) - \mathcal{H}(\mathbf{p}_{X|y})]|) \quad (2.19)$$

$\widehat{\epsilon}_{y,\bar{y}}$ corresponds to the maximum of the approximation errors if y and $\bar{y}$ are unified using mass transportation algorithm. The pair $y, \bar{y}$ that has the minimum error (2.19) is chosen as the input to the mass merging algorithm since their unification is likely to affect $H(U_i|U_0^{i-1}, Y_0^{N-1})$ minimally. Therefore, for each $y \in \mathcal{Y}$, the pair is defined as follows:

$$\pi(y) = \underset{\widehat{y} \in \mathcal{Y} \setminus \{y\}}{\operatorname{argmin}} \; \widehat{\epsilon}_{y,\widehat{y}} \tag{2.20}$$

Given $X|Y \sim \chi$, the following symbol y is chosen for merging together with its pair $\pi(y)$:

$$y = \underset{y \in \mathcal{Y}}{\operatorname{argmin}} \; \widehat{\epsilon}_{y,\pi(y)} \tag{2.21}$$

Since the greedy approximation method calls the mass merging function successively, the above procedure for finding the most appropriate pair of symbols for merging is inefficient. In order to improve efficiency at the expense of a slight performance degradation, the following procedure, an extension of the approach in [14], can be applied:

- Sort $(p_Y(y_i), \mathbf{p}_{X|y_i})$ such that $\mathcal{H}(\mathbf{p}_{X|y_i}) \leq \mathcal{H}(\mathbf{p}_{X|y_{i+1}})$ for all $i \in \{0, 1, \ldots, |\mathcal{Y}| - 1\}$,
- Compute $\widehat{\epsilon}_{y_i,y_{i+1}}$ for all i,
- Choose y_i, y_{i+1} that has minimum $\widehat{\epsilon}_{y_i,y_{i+1}}$ for mass merging.

In this approach, $\widehat{\epsilon}_{y_i,y_{i+1}}$ values can be computed once, and after each application of the merge(y, y_{i+1}) function, $\widehat{\epsilon}_{y_{i-1},y_i}$ and $\widehat{\epsilon}_{y_i,y_{i+1}}$ are updated. In Algorithm 4, the greedy mass merging function that follows the second approach is summarized.

Combining the mass merging algorithm with the code construction scheme (2.14), the efficient polar code construction algorithm is implemented as in Algorithm 5.

Using mass merging algorithm with parameter μ on $\chi_i^{(k)}$, the size of $\chi_i^{(k+1)}$ is bounded above by $\mu^2 q$, which follows from (2.14). Therefore, complexity is controlled through the polar transform. Since the size of $\chi_i^{(k)}$, $k = 1, \ldots, n$ changes through polar transform and mass merging, a modified doubly linked list data structure is proposed in [14], and computational complexity of the algorithm is found as $O(\mu^2 \log \mu)$. Similar data structures can be utilized to implement the greedy code construction algorithm proposed in this subsection.

Algorithm 4: mass_merging(χ, μ)

```
input : χ: Source distribution, μ: Maximum allowed cardinality for 𝒴
```

input : χ: Source distribution, μ: Maximum allowed cardinality for $\mathcal{Y}$

if $|\mathcal{Y}| < \mu$ **then**
  Exit.
else
  Sort $(p_Y(y_i), \mathbf{p}_{X|y_i})$ such that $\mathcal{H}(\mathbf{p}_{X|y_i}) \leq \mathcal{H}(\mathbf{p}_{X|y_{i+1}})$;
  Compute $\widehat{\epsilon}_{y_i, y_{i+1}}$ for all $i = 0, 1, \ldots, |\mathcal{Y}| - 2$;
  while $|\mathcal{Y}| > \mu$ **do**
    Find $y_i = \underset{y_j : j=0,1,\ldots,|\mathcal{Y}|-2}{\text{argmin}} \ \widehat{\epsilon}_{y_j, y_{j+1}}$;
    merge(y_i, y_{i+1});
    Update $\widehat{\epsilon}_{y_{i-1}, y_i}$, $\widehat{\epsilon}_{y_i, y_{i+1}}$;

Algorithm 5: code_construction(χ, N, μ)

input : $X|Y \sim \chi$: Input source, N: Block-length, μ: Maximum allowed alphabet size for side information
output: $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$

for $i = 0, 1, \ldots, N-1$ **do**
  $(b_0 b_1 \ldots b_{n-1})_2 = i$;
  $\chi_i^{(0)} = \chi$;
  for k=0,1,…,n-1 **do**
    if $b_k = 0$ **then**
      $\chi_i^{(k+1)} = \chi_i^{(k)} \circledast \chi_i^{(k)}$;
    else
      $\chi_i^{(k+1)} = \chi_i^{(k)} \boxast \chi_i^{(k)}$;
    mass_merging$(\chi_i^{(k+1)}, \mu)$;
  $H(U_i|U_0^{i-1}, Y_0^{N-1}) = H(\chi_i^{(n)})$;
Return $\{H(U_i|U_0^{i-1}, Y_0^{N-1})\}_{i=0}^{N-1}$.

2.5 Numerical Results

In this section, the performance of the proposed data compression scheme is investigated. The figures of merit in this investigation are the block error rate P_b and symbol error rate P_s for a fixed code rate R. Code constructions in all examples are performed with mass merging algorithm with parameter $\mu = 16$.

In Figure 2.7, the performance of polar codes for compressing a ternary source with probability distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{10}$ under SC-D and SCL-D with list sizes $L = 2, 4, 8, 32$ is illustrated. It must be noted that SCL-D outperforms SC-D, and increasing L beyond $L = 8$ at large code rates does not make significant improvement in P_b.

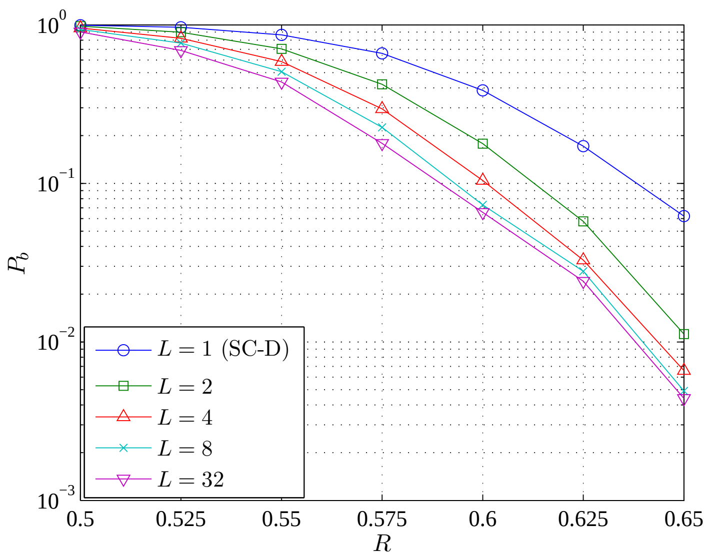

Figure 2.7: Block error rates in the compression of a source with distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{10}$ under SCL-D with $L = 1, 2, 4, 8, 32$.

For the same source, symbol error rates, P_s, are given in Figure 2.8.

In order to investigate the effect of increasing N on P_b and P_s for a fixed source distribution, performance of lossless polar compression scheme for source distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{12}$ is given in Figure 2.9. This example indicates that block error rates decrease at rates above the minimum source coding rate as N increases as expected. At a fixed rate $R > H(X) = 0.5$,

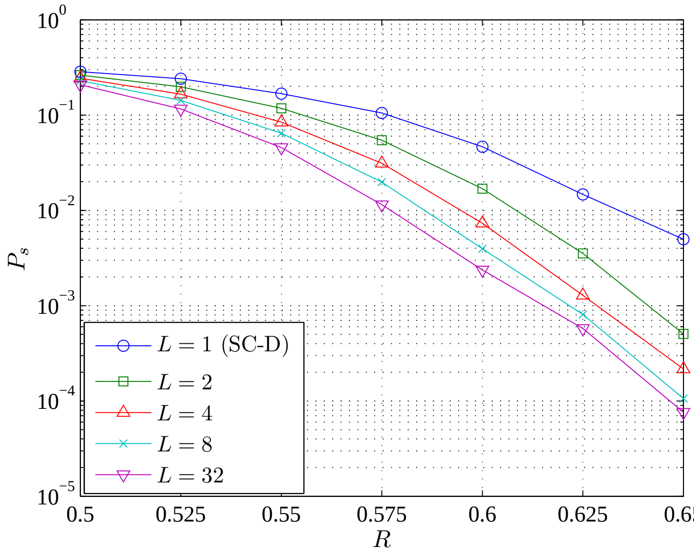

Figure 2.8: Symbol error rates in the compression of a source with distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{10}$ under SCL-D with $L = 1, 2, 4, 8, 32$.

an arbitrary P_b can be obtained at a sufficiently large N.

For the same source, symbol error rates, P_s, are given in Figure 2.10.

For analyzing the performance of the coding scheme at a larger alphabet size, block error rate performance of a quinary source with distribution $p_X = (0.05,\ 0.05,\ 0.055,\ 0.055,\ 0.79)$ at block-length $N = 2^{10}$ is given in Figure 2.12. For the same quinary source, symbol error rate performance is shown in Figure 2.12. The last example indicates that at a fixed block-length and base-q source entropy, similar performance can be obtained at an increased alphabet size.

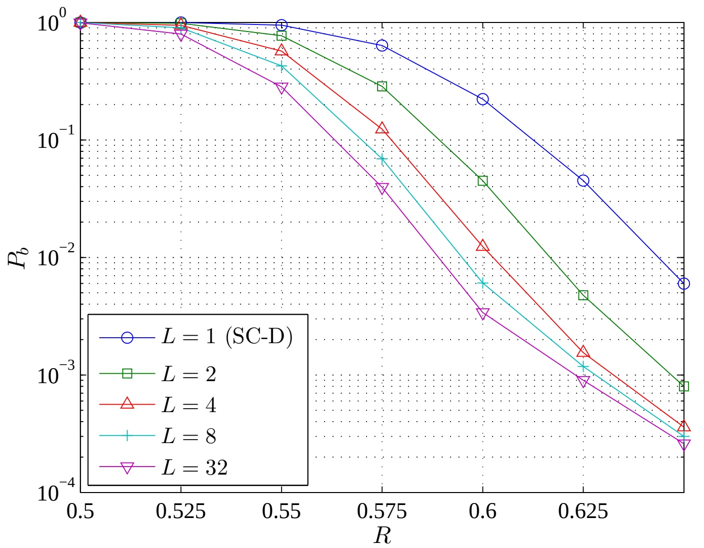

Figure 2.9: Block error rates in the compression of a source with distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{12}$ under SCL-D with $L = 1, 2, 4, 8, 32$.

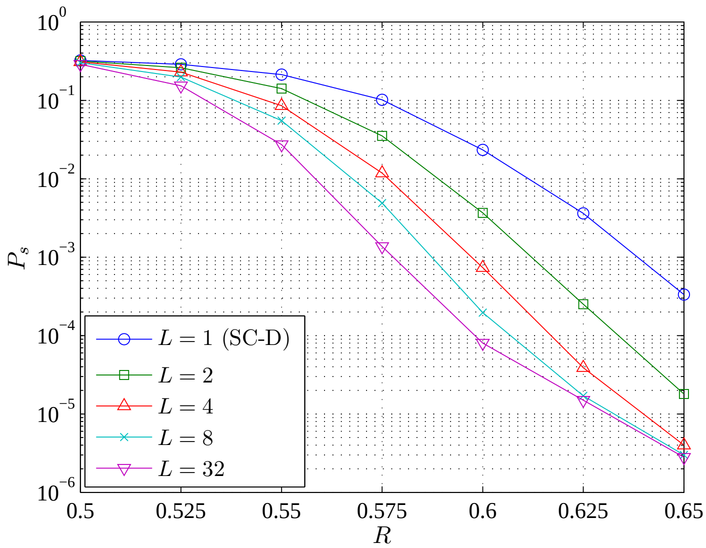

Figure 2.10: Symbol error rates in the compression of a source with distribution $p_X = (0.84,\ 0.09,\ 0.07)$ at block-length $N = 2^{12}$ under SCL-D with $L = 1, 2, 4, 8, 32$.

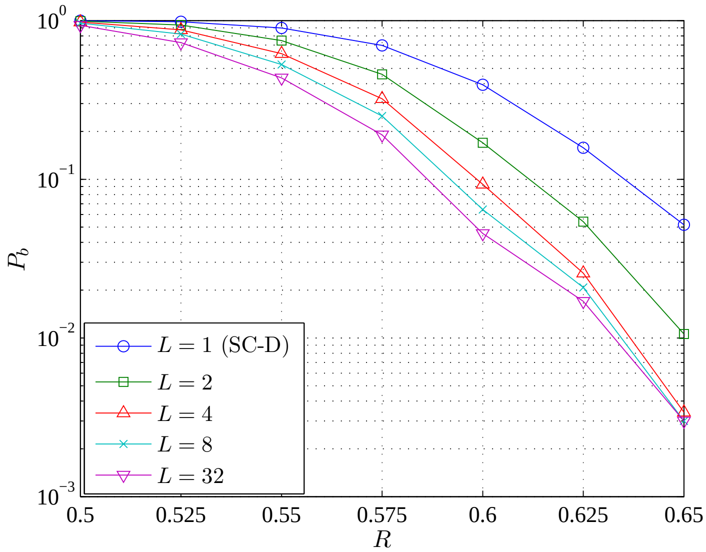

Figure 2.11: Block error rates in the compression of a source with distribution $p_X = (0.05,\ 0.05,\ 0.055,\ 0.055,\ 0.79)$ at block-length $N = 2^{10}$ under SCL-D with $L = 1, 2, 4, 8, 32$.

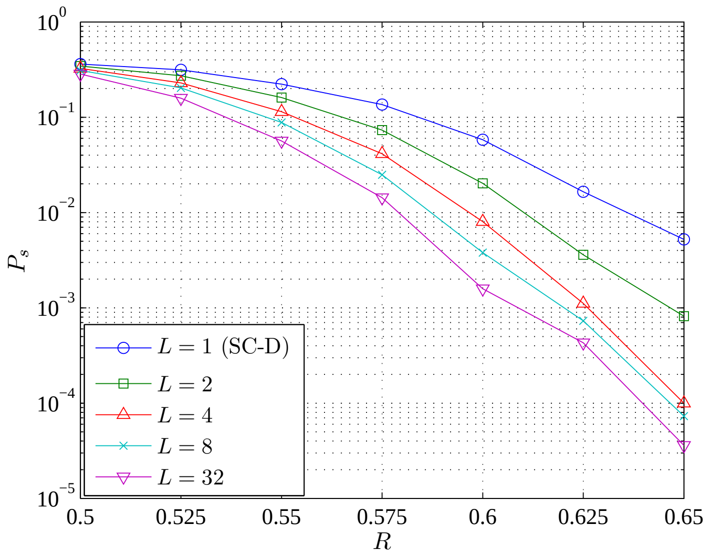

Figure 2.12: Symbol error rates in the compression of a source with distribution $p_X = (0.05,\ 0.05,\ 0.055,\ 0.055,\ 0.79)$ at block-length $N = 2^{12}$ under SCL-D with $L = 1, 2, 4, 8, 32$.