On Minimum Expected Length Prefix Codes Satisfying a (d, k) Runlength-Limited Constraint

Shivkumar K Manickam and Navin Kashyap
Dept. of Electrical Communication Engineering
Indian Institute of Science, Bangalore, India
Email: {shivkumar,nkashyap}@iisc.ac.in

***Abstract*—A prefix code X is said to satisfy the (d, k) runlength-limited (RLL) constraint if all the possible concatenations of the codewords of X satisfy the (d, k) RLL constraint. In this paper, the problem of constructing a minimum expected length prefix code satisfying the (d, k) RLL constraint is studied for certain (d, k) pairs. The question of maximality, with maximality defined relative to the RLL constraint, of these optimal prefix codes is also taken up and answered for some cases.**

I. Introduction to the Problem

A class of problems considered in the literature is that of finding minimum expected length codes among the prefix codes that satisfy the following type of constraint: all of their codewords must come from a given language S. For example, Golin and Na [1] take S to be languages accepted by a certain class of finite automata.

On a related note, in the theory of uniquely decodable codes, Restivo [2] undertook a study of constrained uniquely decodable codes. Here, not only the codewords of these codes had to come from a specified language but all the possible concatenations of the codewords also had to be a part of that language. In this work, he studied the notions of maximality and completeness with respect to this constraint.

Influenced by Restivo's work, we study minimum expected length prefix codes satisfying this tighter constraint. To our knowledge, minimum expected length prefix codes satisfying such constraints have not been studied before. For the constraint, we choose the runlength-limited (RLL) constraint which is of importance in the problem of storage on magnetic or optical disks. This constraint has been well-studied in the channel coding point of view [3].

We will now formally define our problem. Let X be a prefix code over a finite alphabet A for a discrete source with source probability distribution $P = (p_1, p_2, \ldots, p_n)$. In short, throughout this paper, we will just describe this as "X is a prefix code for a probability mass function (PMF) P", without referring to the source. Let l_i be the length of the codeword associated with the probability p_i, for $1 \leq i \leq n$. Its expected length, which we denote by $L_P(X)$, is $\sum_i p_i l_i$. Let each $a \in A$ be called a *letter*. Let A^* be the set of all finite sequences of letters of A. Each $w \in A^*$ is called a *word*. Similarly, for any subset $Y \subset A^*$, let Y^* denote the set of all concatenations of finite number of words belonging to Y. Note that $Y \subset Y^*$. In this paper, we denote the concatenation of two words w_1 and w_2 by $w_1 w_2$. And, whenever we just use the word "code" without any adjective, a prefix code is meant.

Let us now consider the binary alphabet $\{0, 1\}$. For integers d and k with $0 \leq d \leq k \leq \infty$, we say that a word $w \in \{0, 1\}^*$ (*binary word*) satisfies the (d, k) RLL constraint if the number of zeros between any two successive ones is at least d and the length of any consecutive run of zeros is at most k (note that if a word starts with a '0', this constraint allows the length of its run to be less than d). We say that a code X satisfies the (d, k) RLL constraint if all $w \in X^*$ satisfy the (d, k) RLL constraint.

The problem we are concerned with is to determine a prefix code over the binary alphabet — a *binary prefix code* — for a given PMF P, satisfying the (d, k) RLL constraint with the minimum expected length. We call such a code an *optimal* code for P satisfying the (d, k) RLL constraint. This paper contains some preliminary results on this problem. In Section II we show that the problem of obtaining optimal codes for (d, ∞) RLL and $(d, d + 1)$ RLL can be reduced to the problem of finding certain minimum expected cost prefix codes. The latter problem is well studied and has a dynamic programming solution given by Golin and Rote [4]. For the $(0, k)$ RLL constraint, we obtain a code from a specific minimum expected cost code and derive an upper bound on its expected length in terms of the expected length of an optimal code. This is carried out in Section III.

We also explore if the optimal codes are maximal, with maximality defined with respect to the (d, k) RLL constraint. Let X be a binary prefix code that satisfies the (d, k) RLL constraint. We call it a *RLL-maximal* code if for any word w not lying in X, $X \cup \{w\}$ is either not prefix or does not satisfy the (d, k) RLL constraint. This study of maximality adds to the insight into the structure of these codes. Further, the study of an analogously defined notion of maximality constitutes a classical portion of the theory of uniquely decodable codes.

If there exists a word w not lying in X such that $X \cup \{w\}$ is a prefix code satisfying the (d, k) RLL constraint,

The work of N. Kashyap was supported in part by a Swarnajayanti Fellowship awarded by the Government of India.

then we call such a w to be an *add-word* for X. We show in Section III that the optimal codes under (d, ∞) RLL constraint and $(d, d+1)$ RLL constraint are RLL-maximal.

We would like to point out that although it has been mentioned in the literature that such (d, k) RLL constraint prefix codes can be obtained from minimum expected cost codes in the same manner as we have obtained them [4], the optimality of these resulting codes have not been studied before.

II. Some Cases when a Minimum Cost Code Yields an Optimal Code

A. Optimal code satisfying the (d, ∞) RLL constraint

We will start with the (d, ∞) RLL case. For a word $w \in \{0, 1\}^*$ with at least one occurrence of '1', let $z(w)$ be the number of zeros occurring after the last occurrence of '1' in it. If X is a code, then define $z(X)$ to be $z(X) = \min\{z(w) : w \in X \text{ and } w \text{ contains a } 1\}$. Now we have the following result:

Lemma 1. *There exists an optimal code for a PMF P satisfying the (d, ∞) RLL with $z(X) \geq d$.*

Proof. Let X be an optimal code and let $z(X) < d$. Let w_0 be a codeword of X of the form $w_0 = u_0 10^{z(X)}$, for some word u_0. For any $w \in X$, we know that $u_0 10^{z(X)} w$ satisfies the (d, ∞) RLL constraint. Thus all the codewords having a '1' are of the form $0^{d-z(X)}u$. Now we will construct another optimal code $\varphi(X)$ with $z(\varphi(X)) = d$. For a word $w \in X$ containing a '1' of the form $w = 0^{d-z(X)}u$, let $\varphi(w) = u0^{d-z(w)}$. By doing this, we are making $z(\varphi(w)) = d$. If there exists a $w \in X$ that does not contain a '1', then define $\varphi(w) = w$. Let $\varphi(X) = \{\varphi(w) : w \in X\}$. We now show that $\varphi(X)$ is a prefix code. Consider all the codewords of X containing a '1': $\{0^{d-z(X)}u_i\}_{i \in I}$, for some finite set I. Since this set is a prefix code, the set $\{u_i\}_{i \in I}$ is also a prefix code. Further, note that the set obtained by right concatenating each codeword of a prefix code by arbitrary words is also a prefix code. Thus, the codewords of $\varphi(X)$ containing a '1' constitute a prefix code. If $\varphi(X)$ contains an all-zero word, then it cannot be a prefix of any other codeword; for otherwise, it would mean that X is not a prefix code. Also, $\varphi(X)$ satisfies the (d, ∞) RLL constraint. For each $w \in X$ containing a '1', we have (using $|\cdot|$ to denote the length of a word)

$$|\varphi(w)| = |w| + (d - z(w)) - (d - z(X)) \leq |w|.$$

Hence $\varphi(X)$ is an optimal code for P satisfying the (d, ∞) RLL. By construction, we have $z(\varphi(X)) = d$. □

Let $\mathcal{X}$ be the set of all codes X satisfying the (d, ∞) RLL with $z(X) \geq d$. Lemma 1 tells us that to find an optimal code, it is enough if we just restrict ourselves to $\mathcal{X}$.

An operation between uniquely decodable codes called *composition* will serve as a convenient tool to further analyze the problem. Let Z be a prefix code over alphabet $A = \{a_1, a_2, \ldots, a_n\}$. Let Y be another prefix code over alphabet B having n codewords (i.e., having the same size as A): $Y = \{y_1, y_2, \ldots, y_n\}$. Let $\beta_Y : A^* \to Y^*$ be a function with $\beta_Y(a_i) = y_i$, where $a_i \in A$. For $w_1, w_2 \in A^*$, let $\beta_Y(w_1 w_2) = \beta_Y(w_1)\beta_Y(w_2)$. Further, for $U \subset A^*$, let $\beta_Y(U) = \{\beta_Y(u) : u \in U\}$. It can be seen that $\beta_Y(Z)$ is also a prefix code and it is referred to as the code obtained by the *composition* of Z and Y by means of β_Y. Note that throughout this paper whenever β_Y is talked about, it will satisfy this definition.

For example, let $Z = \{ab, ba\}$ be a code over $A = \{a, b\}$, $Y = \{0, 10\}$ be a code over $B = \{0, 1\}$, with $\beta_Y(a) = 0$ and $\beta_Y(b) = 10$. Then, $\beta_Y(X) = \{010, 100\}$.

Since Y is uniquely decodable, the function β_Y is one-to-one and so there is the inverse function β_Y^{-1} as well. Continuing the previous example, take the word $100010 \in Y^*$. For this, we have $\beta_Y^{-1}(100010) = baab$.

Lemma 2. *Let $Y = \{0, 10^d\}$ and for the alphabet $A = \{a, b\}$, let β_Y be such that $\beta_Y(a) = 0$ and $\beta_Y(b) = 10^d$. For every $X \in \mathcal{X}$, there exists a prefix code Z over $\{a, b\}$ with $X = \beta_Y(Z)$.*

Proof. Since X satisfies the (d, ∞) RLL constraint and $X \in \mathcal{X}$, each $w \in X$ can be written as a concatenation of codewords of Y. Now define $Z = \{\beta_Y^{-1}(w) : w \in X\}$. Since X is a prefix code, it can be seen that Z is also a prefix code. By way of construction, we have $X = \beta_Y(Z)$. □

Lemmas 1 and 2 allow us to convert our problem into the problem of obtaining a minimum expected cost code. We will now formally define the terms. Let X be a code over an alphabet $A = \{a_1, a_2, \ldots a_m\}$ for a PMF P. We now have costs C associated with the letters of A. Let each symbol a_i be associated with a cost $c(a_i)$ which is a positive real number. Cost of a word w in A^*, which we write as $c(w)$, is the sum of the costs of each letter of w. Expected cost of X is defined as $L_P(X, C) = \sum_{x \in X} p(x)c(x)$.

Fix $A = \{a, b\}$ with costs C: $c(a) = 1$ and $c(b) = d + 1$. Take $Y = \{0, 10^d\}$ and $\beta_Y(a) = 0$, $\beta_Y(b) = 10^d$. Note that for any prefix codes Z over A, $\beta_Y(Z) \in \mathcal{X}$.

Theorem 1. *Let Z_{opt} be a minimum expected cost code for a PMF P. Then $X_{opt} = \beta_Y(Z_{opt})$ is an optimal code for P satisfying the (d, ∞) RLL constraint.*

Proof. The theorem follows from Lemmas 1 and 2, and the observation that if Z is a prefix code over A, then $L_P(\beta_Y(Z)) = L_P(Z, C)$. □

Remark 1. *For each optimal code X for P satisfying the (d, ∞) RLL constraint with $z(X) \geq d$, we have a minimum expected cost code Z over A such that $X = \beta_Y(Z)$.*

Thus, the problem of finding an optimal code satisfying the (d, ∞) RLL constraint can be reduced to the problem of finding a minimum expected cost binary code with costs 1 and $d + 1$.

B. Optimal code satisfying the $(d, d+1)$-RLL constraint

Let us now consider the problem of finding an optimal code for a PMF P satisfying the $(d, d+1)$ RLL constraint. Note that the $(0, 1)$-RLL constraint and $(1, \infty)$-RLL constraint are complements of each other and we have already seen how to handle the latter case. So it will suffice if we take $d \geq 1$. As in the previous case, it turns out that we can restrict our attention to a subset of codes.

Lemma 3. *For any PMF P, there exists an optimal code X satisfying the $(d, d+1)$ RLL constraint having a codeword ending in a '1'.*

Proof. First of all note that a prefix code with at least two codewords satisfying the $(d, d+1)$ RLL constraint cannot have an all-zero word. Let X be an optimal code and let us suppose that all its codewords end in 10^j, for $1 \leq j \leq d+1$. Let us first take the case where X contains a codeword ending in 10^{d+1}. So all the codewords have to begin with a '1'. For each $x = 1u \in X$, for some word u, let $\varphi_1(x) = u1$. Let $\varphi_1(X) = \{\varphi_1(x) : x \in X\}$. Arguing as in the proof of Lemma 1, we can show that $\varphi_1(X)$ is a prefix code. Next, note that all the codewords of $\varphi_1(X)$ by themselves satisfy the $(d, d+1)$ RLL constraint. If $\varphi_1(X)$ does not satisfy the $(d, d+1)$ RLL constraint, then there exist codewords $u_1 1$ and $u_2 1$ in $\varphi_1(X)$ such that $w = u_1 1 u_2 1$ does not satisfy the constraint. More specifically, there exist two successive '1's in w, say v_1 and v_2, having either less than d zeros or more than $d+1$ zeros between them. If the last '1' of w is either v_1 or v_2, then $1u_2 1u_2$ will also not satisfy the $(d, d+1)$ RLL constraint, which is not possible as X satisfies the $(d, d+1)$ RLL constraint. If both v_1 and v_2 are not the last '1' of w, then $1u_1 1u_2$ will also not satisfy the constraint which again is not possible. Thus $\varphi_1(X)$ satisfies the $(d, d+1)$ RLL constraint, and so is an optimal code with all its codewords ending in a '1'.

Let us now take the case where X does not contain a word ending in 10^{d+1}. Recall the definition of $z(\cdot)$ from Section II-A. Let $z = \min\{z(x) : x \in X\}$. For each $x = u10^j \in X$, let $\varphi_2(x) = 0^z u 10^{j-z}$. Let $\varphi_2(X) = \{\varphi_2(x) : x \in X\}$. The fact that X is a prefix code satisfying the $(d, d+1)$ RLL constraint implies that $\varphi_2(X)$ is also a prefix code satisfying the $(d, d+1)$ RLL constraint. To see this, suppose that $\varphi_2(X)$ is not a prefix code. So there exist two words $0^z u_1 10^{j_1 - z}$ and $0^z u_2 10^{j_2 - z}$ of $\varphi_2(X)$ such that the former is a prefix of the latter: $u_2 10^{j_2 - z} = u_1 10^{j_1 - z} w$, for some word w. If w has a '1', then it begins with a run of z_1 zeros such that $j_1 - z + z_1 \geq d$. This is because the word $u_2 10^{j_2}$ is a codeword of X and so all of its prefixes satisfy the $(d, d+1)$ RLL constraint. Since we are considering the case where none of the codewords end in 10^{d+1}, we have that $j_1 \leq d$. This means that $u_1 10^{j_1}$ is a prefix of $u_2 10^{j_2}$, which is not possible as X is a prefix code. If w does not have a '1', then $u_1 = u_2$ which again leads to a contradiction. Thus $\varphi_2(X)$ is a prefix code. If $\varphi_2(X)$ does not satisfy the $(d, d+1)$ RLL constraint, then it contains two codewords $0^z u_1 10^{j_1 - z}$ and $0^z u_2 10^{j_2 - z}$ such that $0^z u_1 10^{j_1 - z} 0^z u_2 10^{j_2 - z}$ does not satisfy the $(d, d+1)$ RLL constraint. If this is the case, then the word $w = u_1 10^{j_1} u_2 10^{j_2}$ also does not satisfy the constraint. But this is not possible as $w \in X^*$. Thus, $\varphi_2(X)$ is a prefix code satisfying the $(d, d+1)$ RLL constraint. By construction, we have that $\varphi_2(X)$ is optimal with at least one of its codewords ending in '1'. □

Let us define $\mathcal{X}$ to be the set of all codes satisfying the $(d, d+1)$ RLL constraint and having a codeword ending in a '1'. It is enough if we restrict our attention to this set.

Lemma 4. *Let $Y = \{0^d 1, 0^{d+1} 1\}$ and for the alphabet $A = \{a, b\}$, let β_Y be such that $\beta_Y(a) = 0^d 1$ and $\beta_Y(b) = 0^{d+1} 1$. For every $X \in \mathcal{X}$, there exists a prefix code Z over $\{a, b\}$ for which $X = \beta_Y(Z)$ holds.*

Proof. Take an $X \in \mathcal{X}$. Since it contains a codeword ending in '1', all its codewords must begin with at least d zeros. We claim that this implies all the codewords of X end in '1'. Let us suppose that X contains codewords ending in zeros. If that is the case, then they should end in '10'. To see this, suppose that there exists a word w ending in 10^j, for $j \geq 2$. Take a codeword w' of X. We have seen that the first occurrence of '1' in the word w' is preceded by at least d zeros. Thus the word ww' does not satisfy the $(d, d+1)$ RLL constraint. Now, for an $x \in X$ of the form $u10$, let $\varphi(x) = u1$. For those $x \in X$ ending in a '1', let $\varphi(x) = x$. Define $X' = \{\varphi(x) : x \in X\}$. Clearly X' satisfies the $(d, d+1)$ RLL constraint. If X' is not a prefix code, then there exists x_1, $x_2 \in X$ such that $\varphi(x_1)$ is a prefix of $\varphi(x_2)$. First let us take the case $x_1 \neq \varphi(x_1)$. In this case, $\varphi(x_1)$ ends in a '1' and since the constraint does not allow the occurrence of two adjacent '1's, we have that x_1 is a prefix of $\varphi(x_2)$ which in turn is a prefix of x_2 leading to a contradiction. The case $x_1 = \varphi(x_1)$ also implies that x_1 is a prefix of x_2. Thus X' is a prefix code. But this contradicts the optimality of X. Thus, all the codewords of X end in a '1'.

Since X satisfies the $(d, d+1)$ RLL constraint and all its codewords begin with at least d zeros and end in a '1', each of its codeword can be expressed as concatenation of codewords of Y. Let $Z = \{\beta_Y^{-1}(x) : x \in X\}$. We have $X = \beta_Y(Z)$, and since X is a prefix code so is Z. □

Let us fix $A = \{a, b\}$ with costs C: $c(a) = d+1$ and $c(b) = d+2$. Take $Y = \{0^d 1, 0^{d+1} 1\}$ and $\beta_Y(a) = 0^d 1$, $\beta_Y(b) = 0^{d+1} 1$. Note that for any prefix code Z over A, $\beta_Y(Z) \in \mathcal{X}$.

Theorem 2. *Let Z_{opt} be a minimum expected cost code for a PMF P. Then $X_{opt} = \beta_Y(Z_{opt})$ is an optimal code for P satisfying the $(d, d+1)$ RLL constraint.*

This theorem can be proved similar to Theorem 1.

Remark 2. *For each optimal code X for P satisfying the $(d, d+1)$ RLL constraint and the conditions of Lemma 3,* *we have a minimum expected cost code Z over A such that $X = \beta_Y(Z)$.*

Thus, an optimal code satisfying the $(d, d+1)$ RLL constraint can be obtained from a minimum expected cost binary code with costs $d+1$ and $d+2$.

III. Other Observations

Let us now look into the optimal codes constructed in Section II. The optimal codes in both the cases can be represented as a composition of a minimum expected cost code and another prefix code which we will call a *base code.* It can be seen that this base code decides all the other parameters that go into getting an optimal code. For the (d, ∞) case, the base code is $\{0, 10^d\}$ and for the $(d, d+1)$ case, $\{0^d1, 0^{d+1}1\}$ is the base code. Both of them are prefix codes and they can be roughly seen as a sort of basic building blocks of the words satisfying the corresponding RLL constraints. Now for the general (d, k) RLL case, is it possible to reduce the problem of obtaining an optimal code to a problem of finding a minimum expected cost code by suitable defining a base code which is prefix? One can think of $\{0^d1, 0^{d+1}1, \ldots, 0^k1\}$ as a candidate for this. Let us see if it works for the case $d = 0$ or not.

A. A result for the $(0, k)$-RLL case

Let us now consider the case of $d = 0$ and $k > 1$. Take $Y = \{1, 01, 001, \ldots, 0^k1\}$, alphabet $A = \{a_1, a_2, \ldots, a_{k+1}\}$. Recall the function $\beta_Y : A^* \to Y^*$ defined in Section II-A. Let us take $\beta_Y(a_i) = 0^{i-1}1$. For any prefix code Z over A, $\beta_Y(Z)$ satisfies the $(0, k)$ RLL constraint.

Let the cost C associated with each letter $a_i \in A$ be $c(a_i) = i$. Let Z_{opt} be a minimum expected cost code for a PMF P and let X_0 be $\beta_Y(Z_{opt})$. The code X_0 satisfies the $(0, k)$ RLL constraint. The following example shows that X_0 is not always an optimal code.

Take $k = 4$. Let us evaluate X_0 for the uniform distribution U_7. From the algorithm outlined in [5], Z_{opt} comes out to be $\{a_1a_1, a_1a_2, a_1a_3, a_2, a_3, a_4, a_5\}$. So $X_0 = \{11, 01, 101, 001, 1001, 0001, 00001\}$ with its expected length being 23/7. But consider the code $X = \{001, 010, 011, 100, 101, 110, 111\}$. Clearly X satisfies the $(0, 4)$-RLL constraint and its expected length is 3. Thus X_0 is not optimal. But we can say the following:

Lemma 5. *Let $P = (p_1, p_2, \ldots, p_n)$ be a PMF with $p_1 \geq p_2 \geq \ldots \geq p_n$. Let X_0 be the code as defined above and X_{opt} be the optimal code for P satisfying the $(0, k)$ RLL constraint. Then, we have $L_P(X_0) \leq L_P(X_{opt}) + (1 - p_n)$.*

Proof. Without loss of generality, we can take the codeword in X_{opt} associated with the least probability p_n to end with a '1'. To see it, let $w'0$ be one of the longest codewords of X_{opt}. If $w'1$ is a codeword with probability p_k associated to it, then associate p_n to $w'1$ and p_k to $w'0$. If not, we can replace the codeword $w'0$ with $w'1$ and then associate p_n to it. The resulting code is still an optimal code satisfying $(0, k)$ RLL.

Now for each $w \in X_{opt}$, define

$$\varphi(w) = \begin{cases} w1, & \text{if } w \text{ ends with '0'} \\ w, & \text{otherwise} \end{cases}$$

Let $X = \{\varphi(w) : w \in X_{opt}\}$. The code X is prefix and satisfies the $(0, k)$ RLL constraint. Since the codeword associated to p_n ends with a '1', we have

$$L_P(X) \leq L_P(X_{opt}) + (1 - p_n). \tag{1}$$

Since X satisfies the $(0, k)$ RLL constraint and all its codewords end with a '1', each $u \in X$ can be written as a concatenation of codewords of Y uniquely. Thus there exists a code Z over A such that $X = \beta_Y(Z)$. Now we have the following chain of inequalities:

$$L_P(X_0) = L_P(Z_{opt}, C) \leq L_P(Z, C) = L_P(X). \tag{2}$$

Thus from (1) and (2), we have $L_P(X_0) \leq L_P(X_{opt}) + (1 - p_n)$. □

B. Maximality of optimal codes

A prefix code X is said to be a maximal prefix code if it is not a proper subset of any prefix code. It is known that the minimum expected length binary prefix codes in the unconstrained scenario are maximal prefix codes. Are the optimal codes satisfying the (d, k) RLL constraint RLL-maximal (refer Section I)? We will now show that any optimal code satisfying either the (d, ∞) RLL or the $(d, d+1)$ RLL constraint is RLL-maximal. We will first start by showing that the optimal codes that we derived are RLL-maximal.

Let Z be a prefix code over alphabet $A = \{a_1, a_2, \ldots, a_n\}$. Let Y be another prefix code over alphabet B having n codewords (i.e., having the same size as A): $Y = \{y_1, y_2, \ldots, y_n\}$. Let $\beta_Y : A^* \to Y^*$ be such that $\beta_Y(a_i) = y_i$, where $a_i \in A$. Let $X = \beta_Y(Z)$. Recall the definition of add-word from Section I.

Lemma 6. *Let $w \in Y^*$ be an add-word for X. Then, $Z \cup \{\beta_Y^{-1}(w)\}$ is a prefix code.*

Proof. For a $u \in Z$, if either u is a prefix of $\beta_Y^{-1}(w)$ or vice-versa, then it follows that either $\beta_Y(u)$ is a prefix of w or vice-versa. But this is not possible as $\beta_Y(u)$ and w are codewords of $X \cup \{w\}$. □

Lemma 7. *Let $X^{(d,\infty)}$ and $X^{(d,d+1)}$ be optimal codes for a PMF P satisfying the (d, ∞) RLL constraint and $(d, d+1)$ RLL constraint respectively as obtained in Theorem 1 and Theorem 2. Both of them are RLL-maximal.*

Proof. Let $Z^{(d,\infty)}$ and $Z^{(d,d+1)}$ be minimum expected cost codes occurring in Theorem 1 and Theorem 2 respectively. As both these codes are binary minimum expected cost codes, they are maximal prefix. Let $Y_1 = \{0, 10^d\}$ and $Y_2 = \{0^d1, 0^{d+1}1\}$. Let us now suppose that $X^{(d,\infty)}$ is not RLL-maximal and let w be an add-word for it. From Lemma 6, we have that w does not belong to Y_1^*. Let w end in 10^j with $0 \leq j < d$ and let $w' = w0^{d-j}$. It can be seen that w' is also an add-word for $X^{(d,\infty)}$. But $w' \in Y_1^*$ as w' satisfies the (d, ∞) RLL constraint and w' ends in 10^d. This again leads to a contradiction by the application of Lemma 6. Thus, $X^{(d,\infty)}$ is RLL-maximal.

Let us now suppose that $X^{(d,d+1)}$ is not RLL-maximal and let w be an add-word for it. Since all the codewords of $X^{(d,d+1)}$ end in a '1', w begins with 0^d1 or $0^{d+1}1$. Also since $Z^{(d,d+1)}$ is a maximal prefix code, $X^{(d,d+1)}$ has codewords starting from $0^{d+1}1$. So w has to end in a '1' and so $w \in Y_2^*$. But by virtue of Lemma 6, this leads to a contradiction. Thus, $X^{(d,d+1)}$ is also RLL-maximal. □

Now we will show that any optimal code for the (d, ∞) RLL and $(d, d+1)$ RLL cases are RLL-maximal.

Proposition 1. *Any optimal code for a PMF P satisfying the (d, ∞) RLL constraint is RLL-maximal.*

Proof. Let X be an optimal code. It is sufficient if we consider the case $z(X) < d$ (refer section II-A) as the case $z(X) \geq d$ is already covered by Lemma 7 (Remark 1). Suppose that X is not RLL-maximal and let w be its add-word. Let $X' = X \cup \{w\}$. We will first show that X contains an all-zero word and so w cannot be an all-zero word. For this, we are going to use the conclusions drawn in the proof of Lemma 1. Recall, from the proof of Lemma 1, that $\varphi(X)$ is an optimal code with $z(\varphi(X)) = d$. Let $Y = \{0, 10^d\}$ and for the alphabet $A = \{a, b\}$, let β_Y be such that $\beta_Y(a) = 0$ and $\beta_Y(b) = 10^d$. There exists a prefix code Z over A such that $\varphi(X) = \beta_Y(Z)$ (Lemma 2). Now since Z is a binary minimum expected cost code (Remark 1), it is a maximal prefix code and hence one of its codewords is of the form a^k, for some positive integer $k > 0$. Thus, $\varphi(X)$ contains an all-zero word. But this is possible only if X contains an all-zero word. Also, since Z is a maximal prefix code, it has a word beginning in 'b'. Thus $\varphi(X)$ has a word beginning in '1' which implies that X has a word $w_0 = 0^{d-z(X)}1u_0$.

Now we show that $z(w) \geq z(X)$. If not, then ww_0 will not satisfy the (d, ∞) RLL constraint. So $z(X') = z(X) < d$. Now, we have that $\varphi(X') = \varphi(X) \cup \{\varphi(w)\}$. But this contradicts Lemma 7 as $\varphi(X')$ is a prefix code satisfying the (d, ∞) RLL constraint. So, X is RLL-maximal. □

Proposition 2. *Any optimal code for a PMF P satisfying the $(d, d+1)$ RLL constraint is RLL-maximal.*

Proof. Let X be one such optimal code. Let $Y = \{0^d1, 0^{d+1}1\}$ and for the alphabet $A = \{a, b\}$, let β_Y be such that $\beta_Y(a) = 0^d1$ and $\beta_Y(b) = 0^{d+1}1$. If X is of the form $\beta_Y(Z)$, for some prefix code Z over A, then it is RLL-maximal (Remark 2 and Lemma 7). Let us now take that X is not of the form $\beta_Y(Z)$. Suppose that X is not RLL-maximal, and let w be its add-word. We will now show that this leads to a contradiction. Define $X' = X \cup \{w\}$.

Suppose that a codeword of X ends in 10^{d+1}. So all the words of X' are of the form $1u$. We now use some conclusions from the proof of Lemma 3. Recall φ_1 from the proof of Lemma 3. We have $\varphi_1(X') = \varphi_1(X) \cup \{\varphi_1(w)\}$, where $\varphi_1(X)$ is an optimal code, and the set $\varphi_1(X')$ is a prefix code satisfying the constraint. But this is not possible as $\varphi_1(X)$ is RLL-maximal (Remark 2 and Lemma 7). Thus no codeword of X can end in 10^{d+1}.

Let us consider the following cases for w: (i) let w end in a '1'. From Lemma 4, there exists a prefix code Z over A such that $X' = \beta_Y(Z)$; but this is not possible as X is not of this form. (ii) Let w be of the form $u10^{j_0}$ for some word u and $j_0 < d+1$. Let $z(X) = \min\{z(x) : x \in X\}$. We now show that $j_0 \geq z(X)$. Recall $\varphi_2(\cdot)$ from the proof of Lemma 3 which also shows that $\varphi_2(X)$ is an optimal prefix code satisfying the $(d, d+1)$ RLL constraint. From Lemma 4 we have the existence of a prefix code Z over A such that $\varphi_2(X) = \beta_Y(Z)$. From Remark 2, we have that Z is a maximal prefix code and so $\varphi_2(X)$ contains words starting from 0^d1. But this is possible only if X contains a word w_0 starting from $0^{d-z(X)}1$. Since ww_0 satisfies the RLL constraint, $j_0 \geq z(X)$. So we have $\varphi_2(X') = \varphi_2(X) \cup \{\varphi_2(w)\}$, with $\varphi_2(X')$ being a prefix code satisfying the constraint. But this is not possible as $\varphi_2(X)$ is RLL-maximal (from Remark 2 and Lemma 7). (iii) As the final case, let w end in 10^{d+1}. Define $w' = w10^{z(X)}$. The fact that X' is a prefix code satisfying the constraint implies that $X'' = X \cup \{w'\}$ is also a prefix code satisfying the constraint. Now this is the same as the second case which we have already shown is not possible. Thus, our assumption that X is not RLL-maximal leads to a contradiction. Hence any optimal code is RLL-maximal. □

IV. Concluding Remarks

The results in this paper on finding optimal codes satisfying the (d, k) RLL constraint rely on constructing some appropriate minimum expected cost codes. Also, we have shown that any optimal code for either the (d, ∞) RLL or $(d, d+1)$ RLL case is RLL-maximal and not just those that can be obtained from a minimum expected cost code by composition. This result gives us a strong hope to expect that any optimal code of sufficiently large size for any general (d, k) RLL constraint is RLL-maximal.

References

[1] M. J. Golin and H. Na, " Optimal prefix-free codes that end in a specified pattern and similar problems: the uniform probability case," *Proc. of DCC* 2001, pp. 143–152.

[2] A. Restivo, "Codes and Local Constraints," *Theoret. Comp. Sci.*, vol. 72, no. 1, pp. 55–64, 1990.

[3] K.A.S. Immink, *Codes for Mass Data Storage Systems*, 2nd ed., Shannon Foundation Publishers, 2004.

[4] M. J. Golin and G. Rote, "A Dynamic Programming Algorithm for Constructing Optimal Prefix-Free Codes with Unequal Letter Costs," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1770–1781, 1998.

[5] M. J. Golin and N. Young, "Prefix Codes: Equiprobable Words, Unequal Letter Costs," *Proc. of ICALP* 1994, pp. 605–617.