Linial's Algorithm and Systematic Deletion-Correcting Codes

Yuting Li* and Farzad Farnoud†
*School of Mathematical Science, Peking University, 1901110017@pku.edu.cn
†Electrical and Computer Engineering, University of Virginia, farzad@virginia.edu

***Abstract*—In this paper, we present a universal method to construct systematic codes correcting a constant number of errors that are traditionally hard to handle, such as insertions and deletions. Our method is based on Linial's distributed graph coloring algorithm, and the codes have polynomial-time encoding and decoding complexity, with a redundancy that is about twice the Gilbert-Varshamov bound. As an application, for $q = O(\text{poly}(n))$, where q is the size of the alphabet and n is the code length, we construct systematic codes correcting t deletions and s substitutions with redundancy $(4t + 4s)\log n + (3t + 6s)\log q + O(\log\log n)$ and systematic codes correcting t deletions or s substitutions with redundancy $4\max\{t, s\}\log n + 3\max\{t, 2s\}\log q + O(\log\log n)$. We also show that the celebrated 'syndrome compression' technique, proposed by Sima et al. (ISIT 2020), can be viewed as an application of Linial's algorithm. Thus our method is a generalization of syndrome compression.**

I. Introduction

Codes correcting a constant number of deletions have been studied in several recent works. In [1], Brakensiek et al. constructed a family of t-deletion-correcting codes with redundancy $O(t^2 \log t \log n)$ and near-linear decoding complexity, where t is a constant and n is the length of the code. In [2], Sima et al. constructed a family of t-deletion-correcting codes with redundancy roughly $8t\log n$, which are polynomial-time encodable and decodable. Then in [3], Sima et al. presented a general technique called *syndrome compression*, using which they constructed a family of t-deletion-correcting codes with redundancy roughly $4t\log n$, which are polynomial-time encodable and decodable.

Syndrome compression can be briefly described as follows. For $\boldsymbol{x} \in \Sigma_2^n$ and a positive integer t, define

$$B_t(\boldsymbol{x}) \triangleq \{\boldsymbol{y} \in \Sigma_2^n : \text{LCS}(\boldsymbol{x}, \boldsymbol{y}) \geq n - t\} \setminus \{\boldsymbol{x}\},$$

where $\text{LCS}(\boldsymbol{x}, \boldsymbol{y})$ stands for the length of a longest common subsequence of $\boldsymbol{x}$ and $\boldsymbol{y}$. A function $f : \Sigma_2^n \to [2^{R(n)}]$ is called a labeling if for any $\boldsymbol{x} \in \Sigma_2^n$ and $\boldsymbol{y} \in B_t(\boldsymbol{x})$, $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$, and $R(n) = o(\log n \log\log n)$ [3]. Note that if we have a labeling $f : \Sigma_2^n \to [2^{R(n)}]$, then it is easy to construct a t-deletion-correcting codes with redundancy roughly $R(n)$. Syndrome compression is a technique that can lower the redundancy of labeling-based codes by "compressing" the label. Concretely, syndrome compression takes an old labeling over $[2^{o(\log n \log\log n)}]$ and produces a new labeling

$$f_{new} : \Sigma_2^n \to [2^{2\log|B_t(\boldsymbol{x})| + o(\log n)}].$$

As $|B_t(\boldsymbol{x})| = O(n^{2t})$, the resulting t-deletion-correcting codes have redundancy $4t\log n + o(\log n)$, which is about twice the Gilbert-Varshamov bound.

While syndrome compression is an effective technique for constructing low redundancy codes correcting a constant number of deletions [3]–[6], it relies on the existence of a labeling over $[2^{o(\log n \log\log n)}]$. Finding such a labeling is often a challenging and complicated task. In this paper, we present a method for constructing a labeling of small size directly, resulting in a universal way to construct systematic codes correcting a constant number of deletions. Our method is based on Linial's distributed graph coloring algorithm.

The main idea of our method is as follows. Let G_t be a graph whose vertices are sequences in Σ_2^n, and $(\boldsymbol{x}, \boldsymbol{y}) \in \Sigma_2^n \times \Sigma_2^n$ is an edge in G_t if $\boldsymbol{x}, \boldsymbol{y}$ are distinct and have a common subsequence of length $n - t$. From a coloring $h : G_t \to [A], A \in \mathbb{N}$, which is called an A-coloring, one can construct a t-deletion-correcting code with redundancy roughly $\log A$. The greedy coloring algorithm gives an $O(\Delta(G_t))$-coloring, but computing it takes exponential time. Linial's algorithm, proposed in [**?**], provides a (distributed) way to color a graph with N vertices and maximum degree Δ, using $O(\Delta^2)$ colors in $\log^* N + O(1)$ rounds. In this paper, we use the idea of Linial's algorithm to compute a coloring $h : G_t \to [O(\Delta(G_t)^2)]$ in polynomial time. Thereby, we get a family of t-deletion-correcting codes with redundancy roughly $2\log\Delta(G_t)$ and polynomial-time encoding and decoding. Since $\Delta(G_t) = O(n^{2t})$, the redundancy is roughly $4t\log n$, which is about twice the Gilbert-Varshamov bound.

Our method has several desirable properties. First, it is applicable to any class of errors for which the graph has a bounded maximum degree. Specifically, if we replace the G_t in the last paragraph with the $G_{n,q,\varepsilon}$ (defined precisely in Section II) for an error type ε, the idea in the last paragraph still works. Second, codes constructed by our method are systematic. Third, the codes are simpler than those constructed by syndrome compression since we do not need a complex labeling over $[2^{o(\log n \log\log n)}]$. Forth, using our method, we can construct codes for a wider range of parameters than syndrome compression. In Section V, we demonstrate that

This work was supported in part by NSF grants under grant nos. CCF-1816409 and CAREER-2144974.

syndrome compression can be viewed as an application of Linial's algorithm.

As an application of our method, in Section IV, for alphabet size $q = O(\mathrm{poly}(n))$, we construct systematic codes correcting t deletions and s substitutions with redundancy $(4t+4s)\log n + (3t+6s)\log q + O(\log\log n)$ and systematic codes correcting t deletions or s substitutions with redundancy $4\max\{t,s\}\log n + 3\max\{t,2s\}\log q + O(\log\log n)$. Both codes are polynomial-time-encodable and decodable. The state of the art and the results presented in this paper can be compared using Tables I and II. Note that, for $q > \log n$, systematic codes correcting t deletions and s substitutions were not constructed before, and for $n < q = O(\mathrm{poly}(n))$, the redundancy of our q-ary t-deletion-correcting codes is substantially less than that of the codes in [5].

II. Notations and Preliminaries

Let Σ_q be the alphabet $\{0,1,\ldots,q-1\}$ and Σ_q^n denote all the strings of length n over Σ_q. Strings in Σ_q^n are denoted by bold symbols, like $\boldsymbol{u}$. $\boldsymbol{u}_{[i,j]}$ denotes the substring of $\boldsymbol{u}$ that begins at position i and ends at position j. $|\boldsymbol{u}|$ denotes the length of $\boldsymbol{u}$. For a set S, $|S|$ denotes the cardinality of S. For a set family $\mathcal{J}$, $|\mathcal{J}|$ denotes the number of sets in $\mathcal{J}$. Logarithms in the paper are to the base 2. The redundancy of a code $\mathcal{C} \subset \Sigma_q^n$ is defined as $\log(q^n) - \log|\mathcal{C}|$ bits.

We use ε to denote the possible errors that can occur, which is called an error type in this paper. For $\boldsymbol{u}, \boldsymbol{v} \in \Sigma_q^*$, we write $\boldsymbol{u} \xrightarrow{\varepsilon} \boldsymbol{v}$ if $\boldsymbol{u}$ may become $\boldsymbol{v}$ through the errors in ε. For $\boldsymbol{w} \in \Sigma_q^*$, we define

$$I_{q,n,\varepsilon}(\boldsymbol{w}) \triangleq \left\{\boldsymbol{u} \in \Sigma_q^n : \boldsymbol{u} \xrightarrow{\varepsilon} \boldsymbol{w}\right\}$$

and define $G_{q,n,\varepsilon}$ to be the graph whose vertices are Σ_q^n and $(\boldsymbol{u}, \boldsymbol{v})$ is an edge if $\boldsymbol{u}, \boldsymbol{v}$ are distinct elements in some $I_{q,n,\varepsilon}(\boldsymbol{w})$. We denote $\Delta(G_{q,n,\varepsilon})$ (the maximum degree of $G_{q,n,\varepsilon}$) as $\Delta_\varepsilon(q,n)$, and denote

$$L_\varepsilon(q,n) \triangleq \max\{|I_{q,n,\varepsilon}(\boldsymbol{w})| : \boldsymbol{w} \in \Sigma_q^*\}.$$

For a vertex u in a graph, we use $N(u)$ to denote the neighbors of u. For a graph G with vertex set V and a set S, the function $h : V \to S$ is called a coloring if, for each vertex u, $h(u)$ is distinct from all $h(v), v \in N(u)$. The *size* of the coloring h is $|S|$. We say that a coloring h of G can be computed in time T if, for each vertex u of G, $h(u)$ can be computed in time T.

We use $f(n) = O(g(n)) (f(n) = \Omega(g(n)))$ to denote that $f(n) \le cg(n) (f(n) \ge cg(n))$, where c is an absolute constant.

III. Constructing Codes using Linial's Graph Coloring Algorithm

Linial's algorithm, proposed in [**?**], provides a (distributed) way to color the vertices of a graph G, with N vertices and maximum degree Δ, using $O(\Delta^2)$ coloring in $\log^* N + O(1)$ rounds. We briefly recall Linial's algorithm below through Definition 1, Fact 1, Fact 2, and Fact 3, which are known results. A detailed description of Linial's algorithm can be found in [7, Section 3.10].

Linial's algorithm utilizes r-cover-free set families, which are defined in Definition 1. For a graph G, Linial's algorithm uses a $\Delta(G)$-cover-free set family $\mathcal{J}$ over a ground set M to construct a new coloring of size $|M|$ from an old coloring of size $|\mathcal{J}|$. Fact 1 and Fact 2 show an explicit construction of *r-cover-free* set families, and Fact 3 is Linial's algorithm. We will give the proof of Fact 2 and Fact 3 because they may not be exactly the same as the textbook Linial algorithm.

Definition 1. *A family $\mathcal{J}$ of subsets of a "ground" set M is called r-*cover-free *over M if for each set $F \in \mathcal{J}$ the following holds: F is not contained in the union of any other r sets in $\mathcal{J}$.*

Fact 1 ([8, Example 3.2])**.** *Let Q be a prime power, b, r be non-negative integers, and*

$$\mathcal{J} = \big\{\{(x, g(x)) : x \in \mathbb{F}_Q\} : \\ g(x) = a_0 + a_1 x + \cdots + a_b x^b, a_i \in \mathbb{F}_Q\big\}. \quad (1)$$

If $br < Q$, then $\mathcal{J}$ is an r-cover-free set family over $\mathbb{F}_Q \times \mathbb{F}_Q$. There are Q^{b+1} sets in the family $\mathcal{J}$, and the size of the ground set is Q^2.

Remark 1. *We use the notations in Fact 1 and let $F_g \triangleq \{(x, g(x)) : x \in \mathbb{F}_Q\}$ for each polynomial g with degree at most b. To find an element in $F_{g_0} \setminus \bigcup_{i=1}^r F_{g_i}$, one can search for an $x \in \mathbb{F}_Q$ such that $g_i(x) - g_0(x) \neq 0$ for all $i = 1, \cdots, r$. Then $(x, g_0(x)) \in F_{g_0} \setminus \bigcup_{i=1}^r F_{g_i}$. For each candidate $x \in \mathbb{F}_Q$, it takes $O(rb) = O(Q)$ time to compute $g_i(x) - g_0(x)$ for all $i = 1, \cdots, r$, and there are Q candidates. So it takes $O(Q^2)$ time to find such x.*

Fact 2 (cf. [7, Section 3.10])**.** *If k is sufficiently large (i.e., larger than some absolute constant), then there exists an explicit r-cover-free family $\mathcal{J}$ of size at least k over a ground set of size at most*

$$\frac{17r^2\log^2 k}{(\log r + \log\log k)^2}.$$

Proof. For positive integers r, k, let

$$b = \left\lceil \frac{2\log k}{\log r + \log\log k} \right\rceil$$

and let Q be a prime power in the interval $(rb, 2rb]$. Construct the family $\mathcal{J}$ based on Fact 1. □

Fact 3 (cf. [**?**, Theorem 4.1])**.** *Let $\Delta = \Delta(G)$ and suppose $\mathcal{J} = \{F_1, F_2, \ldots, F_k\}$ is a Δ-cover-free subset family over a ground set M. Given an old coloring $h : G \to [k]$, we can get a new coloring $h_1 : G \to M$. Moreover, if the Δ-cover-free subset family is that of Fact 1 and h can be computed in time T, then h_1 can be computed in $O(|M| + \Delta T)$ time.*

Proof. For a vertex v in G, since $\mathcal{J}$ is Δ-cover-free,

$$F_{h(v)} \setminus \bigcup_{i \in h(N(v))} F_i \neq \emptyset.$$

We define $h_1(v)$ to be any element in $F_{h(v)} \setminus \bigcup_{i \in h(N(v))} F_i$. It is easy to see that h_1 is a legal coloring of G.

TABLE I
PRIOR WORK

	Alphabet size q	Error type	Redundancy	Systematic
[6]	$q \leq \log n$	t deletions and s substitutions	$\left(4t+4s-1-\left\lfloor \frac{2s-1}{q} \right\rfloor\right)\log n + o(\log n)$	yes
[5]	$\log n < q \leq n$	t deletions	$2t(1+\varepsilon)(2\log n + \log q) + o(\log n)$	no
[5]	$n < q$	t deletions	$(30t+1)\log q$	no

TABLE II
THIS WORK

Alphabet size q	Error type	Redundancy	Systematic
$q = O(\mathrm{poly}(n))$	t deletions and s substitutions	$(4t+4s)\log n + (3t+6s)\log q + O(\log\log n)$	yes
$q = O(\mathrm{poly}(n))$	t deletions or s substitutions	$4\max\{t,s\}\log n + 3\max\{t,2s\}\log q + O(\log\log n)$	yes

To compute $h_1(v)$, one has to first compute $h(v)$ and $h(N(v))$, which takes $O(\Delta T)$ time. If one uses the Δ-cover-free family in Fact 1, then by Remark 1, one can perform an exhaustive search over $F_{h(v)}$ for an element in $F_{h(v)} \setminus \bigcup_{i\in h(N(v))} F_i$, which takes $O(|M|)$ time. Thus, the total time is $O(|M| + \Delta T)$. □

Now we use Linial's algorithm to get a coloring of $G_{q,n,\varepsilon}$. For our purpose, we only need the first two rounds of Linial's algorithm. We have three steps. First, we apply Fact 3 with respect to the cover-free set families in Fact 2 and get Lemma 1. Second, we apply Lemma 1 to graphs with $2^{O(\mathrm{poly}(n)\log n)}$ vertices and get Lemma 2, which gives an $O(\Delta^2)$-coloring for graphs with $2^{O(\mathrm{poly}(n)\log n)}$ vertices and maximum degree Δ. Third, if $q = O(\mathrm{poly}(n))$, then $G_{q,n,\varepsilon}$ has $2^{O(n\log n)}$ vertices. So we apply Lemma 2 to $G_{q,n,\varepsilon}$ and get Theorem 1, which is the main theorem of this paper.

Lemma 1. *Suppose $\Delta(G) = \Delta$, and there exists an old coloring of size k (larger than some absolute constant) that can be computed in time T. Then we can get a new coloring of size*

$$O\left(\Delta^2 \frac{\log^2 k}{(\log\Delta + \log\log k)^2}\right)$$

that can be computed in time

$$O\left(\Delta T + \Delta^2 \frac{\log^2 k}{(\log\Delta + \log\log k)^2}\right).$$

Proof. By Fact 2 and Fact 3, the lemma is proved. □

Lemma 2. *Suppose $p(n)$ is a polynomial in n, G has $2^{O(p(n)\log n)}$ vertices, and $\Delta(G) = \Delta = \Omega(n)$. Then there exists a coloring h_1 of size $O\big(\Delta^2(p(n))^2\big)$ that can be computed in time $O\big(\Delta^2 p(n)^2\big)$. In addition, there exists a coloring h_2 of size $O\big(\Delta^2\big)$ that can be computed in time $O\big(\Delta^3 p(n)^2\big)$.*

Proof. Let h be the trivial coloring of G which colors all the vertices with different colors, then h can be computed in $O(1)$ time. By Lemma 1, one gets a new coloring h_1 of size

$$O\left(\Delta^2\left(\frac{p(n)\log n}{\log\Delta + \log(p(n)\log n)}\right)^2\right) = O\big(\Delta^2 p(n)^2\big)$$

which can be computed in time $T_1 = O\big(\Delta^2 p(n)^2\big)$. Now we apply Lemma 1 again, viewing h_1 as the old coloring, we can get a new coloring h_2 of size

$$O\left(\Delta^2\left(\frac{\log\big(\Delta^2 p(n)^2\big)}{\log\Delta + \log\log(\Delta^2 p(n)^2)}\right)^2\right) = O\big(\Delta^2\big) \qquad (2)$$

that can be computed in time $O\big(\Delta T_1 + \Delta^2\big) = O\big(\Delta^3 p(n)^2\big)$. Note that (2) holds because $\Delta = \Omega(n)$. □

Theorem 1. *For an error type ε, suppose $\Delta_\varepsilon(q,n) = O\big(q^a n^b\big)$ and $q = O(\mathrm{poly}(n))$, then there exists a coloring*

$$h : G_{q,n,\varepsilon} \to \big[O\big(q^{2a}n^{2b}\big)\big],$$

which can be computed in $O\big(q^{3a}n^{3b+2}\big)$ time.

Proof. Because $q = O(\mathrm{poly}(n))$, $G_{q,n,\varepsilon}$ has $2^{O(n\log n)}$ vertices. By Lemma 2, there exists a coloring $h : G_{q,n,\varepsilon} \to \big[O\big(q^{2a}n^{2b}\big)\big]$ that can be computed in $O\big(q^{3a}n^{3b+2}\big)$ time. □

IV. SYSTEMATIC CODES CORRECTING 'DELETIONS AND SUBSTITUTIONS' AND SYSTEMATIC CODES CORRECTING 'DELETIONS OR SUBSTITUTIONS'

As an application of Theorem 1, we consider deletions and substitutions and their combinations. In this section, we assume ε can only be 't deletions and s substitutions' or 't deletions or s substitutions'. The goal of this section is to construct systematic ε-correcting codes. Lemma 3 provides codes of short length that correct errors in ε. Lemma 4 uses the short codes in Lemma 3 to produce systematic ε-correcting codes. Note that the notation defined in Section II is used in this section.

Lemma 3. *For an error type ε and non-negative integer constants a, b, c, d, suppose $\Delta_\varepsilon(q,n) = O\big(q^a n^b\big)$ and*

$$r(n) = a + c + (d\log n + (b+1)\log\log n)/\log q.$$

Then there exists a code $\mathcal{C} \subset \Sigma_q^{r(n)}$ of size $\Omega\Big(q^c n^d \log^{1/2} n\Big)$ that can correct errors in ε. Moreover, if $q = O(\mathrm{poly}(n))$, then $\mathcal{C}$ has polynomial time encoding and decoding algorithms.

Proof. Let $\mathcal{C}$ be the maximum independent set of $G_{q,r(n),\varepsilon}$. Then

$$\begin{aligned}|\mathcal{C}| &\geq \frac{q^{r(n)}}{\Delta_\varepsilon(q,r(n))+1}\\ &= \Omega\bigg(\frac{q^{r(n)}}{q^a r(n)^b}\bigg)\\ &= \Omega\bigg(q^c n^d \frac{(\log n)^{b+1}}{r(n)^b}\bigg).\end{aligned}$$

Note that $r(n) \leq (d+1)\log n$ when n is sufficiently large. So

$$|\mathcal{C}| = \Omega\bigg(q^c n^d \frac{(\log n)^{b+1}}{((d+1)\log n)^b}\bigg) = \Omega\Big(q^c n^d \log^{1/2} n\Big).$$

If $q = O(\mathrm{poly}(n))$, then the size of $\Sigma_q^{r(n)}$ is a polynomial in n. So $\mathcal{C}$ has polynomial-time encoding and decoding algorithms. □

Lemma 4. *For a positive integer N, suppose $h : \Sigma_q^n \to [N]$ is a coloring of $G_{q,n,\varepsilon}$, and $E' : [N] \to \Sigma_q^{r(n)}$ is an encoder that can correct the errors in ε. Then $E : \Sigma_q^n \to \Sigma_q^{n+r(n)}$, $E(\boldsymbol{u}) \triangleq (\boldsymbol{u}, E'(h(\boldsymbol{u})))$ is a systematic encoder that can correct errors in ε, which has redundancy $r(n)\log q$.*

Let D', D be the decoders corresponding to E' and E, respectively. Suppose E' and D' can be computed in time t' and T', and h can be computed in time t. Then E can be computed in time $t' + t$, and D can be computed in time $T' + L_\varepsilon(q,n)t$. In particular, if t ,t', T', and $L_\varepsilon(q,n)$ are polynomials in n, then E and D have polynomial time complexity.

Proof. We first describe D. For a received sequence $\boldsymbol{c}'$, we search for the unique $\boldsymbol{v} \in I_{q,n,\varepsilon}\Big(\boldsymbol{c}'_{[1,|\boldsymbol{c}'|-r(n)]}\Big)$ such that $h(\boldsymbol{v}) = D'\Big(\boldsymbol{c}'_{[n+1,|\boldsymbol{c}'|]}\Big)$ and let $D(\boldsymbol{c}') = \boldsymbol{v}$.

We prove that D is a decoder of E. Suppose $\boldsymbol{c} = E(\boldsymbol{u})$ and $\boldsymbol{c} \xrightarrow{\varepsilon} \boldsymbol{c}'$, since ε is 't deletions and s substitutions' or 't deletions or s substitutions', we have

$$E'(h(\boldsymbol{u})) \xrightarrow{\varepsilon} \boldsymbol{c}'_{[n+1,|\boldsymbol{c}'|]}$$

and

$$\boldsymbol{u} \xrightarrow{\varepsilon} \boldsymbol{c}'_{[1,|\boldsymbol{c}'|-r(n)]}.$$

Thus,

$$D'\Big(\boldsymbol{c}'_{[n+1,|\boldsymbol{c}'|]}\Big) = h(\boldsymbol{u}),$$

and

$$\boldsymbol{u} \in I_{q,n,\varepsilon}\Big(\boldsymbol{c}'_{[1,|\boldsymbol{c}'|-r(n)]}\Big).$$

Hence, $D(\boldsymbol{c}') = \boldsymbol{u}$. So, D is a decoder of E.

It is clear that E is systematic and takes $t' + t$ time. To compute $D(\boldsymbol{c}')$, we have to compute $D'\Big(\boldsymbol{c}'_{[n+1,|\boldsymbol{c}'|]}\Big)$ and the colors of elements in $I_{q,n,\varepsilon}\Big(\boldsymbol{c}'_{[1,|\boldsymbol{c}'|-r(n)]}\Big)$, so D takes $T' + L_\varepsilon(q,n)t$ time. Thus, if t, t', T' and $L_\varepsilon(q,n)$ are polynomials in n, then E and D has polynomial time complexity. □

Putting Theorem 1, Lemma 3, and Lemma 4 together, one can construct systematic ε-correcting codes.

Theorem 2. *For an error type ε and non-negative integer constants a, b, suppose $\Delta_\varepsilon(q,n) = O\big(q^a n^b\big)$ and $q = O(\mathrm{poly}(n))$. Then there exists a systematic encoder $E : \Sigma_q^n \to \Sigma_q^{n+r(n)}$ correcting errors in ε, where*

$$r(n) = 3a + (2b\log n + (b+1)\log\log n)/\log q.$$

Hence, the redundancy is $2b\log n + 3a\log q + O(\log\log n)$. Moreover, E and its decoder have polynomial time complexity.

Proof. By Theorem 1, there exists a coloring $h : G_{q,n,\varepsilon} \to [N]$, where $N = O\big(q^{2a}n^{2b}\big)$. By letting $c = 2a, d = 2b$ in Lemma 3, there exists an encoder $E' : [N] \to \Sigma_q^{r(n)}$ that corrects the errors in ε. By Lemma 4, $E : \Sigma_q^n \to \Sigma_q^{n+r(n)}$, $E(\boldsymbol{u}) \triangleq (\boldsymbol{u}, E'(h(\boldsymbol{u})))$ is a systematic encoder that can correct the errors in ε. Note that E', D' (decoder for E'), and h can be computed in polynomial time. Since $L_\varepsilon(q,n)$ is a polynomial in n, by Lemma 4, E and its decoder can also be computed in polynomial time. □

Corollary 1. *If $q = O(\mathrm{poly}(n))$, then there exists a systematic code $\mathcal{C} \subset \Sigma_q^n$ correcting t deletions and s substitutions with redundancy*

$$(4t + 4s)\log n + (3t + 6s)\log q + O(\log\log n).$$

Moreover, $\mathcal{C}$ has polynomial-time encoding and decoding algorithms.

Proof. If ε is 't deletions and s substitutions', then

$$\Delta_\varepsilon(q,n) = O\big(q^{t+2s}n^{2t+2s}\big).$$

By Theorem 2, the corollary is proved. □

Corollary 2. *If $q = O(\mathrm{poly}(n))$, then there exists a systematic code $\mathcal{C} \subset \Sigma_q^n$ correcting t deletions or s substitutions with redundancy*

$$4\max\{t,s\}\log n + 3\max\{t,2s\}\log q + O(\log\log n).$$

Moreover, $\mathcal{C}$ has polynomial-time encoding and decoding algorithms.

Proof. If ε is 't deletions or s substitutions', then

$$\Delta_\varepsilon(q,n) = O\Big(q^{\max\{t,2s\}}n^{2\max\{t,s\}}\Big).$$

By Theorem 2, the corollary is proved. □

Remark 2. *In [6] a precoding technique is introduced to reduce redundancy. Precoding is compatible with Linial's algorithm. For example, if one uses single deletion-correcting codes in [9] to precode, then the coefficient of $\log n$ in Corollary 1 can be reduced to $4t + 4s - 1$.*

V. Relationship Between Linial's Algorithm and Syndrome Compression

We now show that syndrome compression can be viewed as an application of Linial's algorithm, where an existing coloring of size $k = 2^{o(\log n \log\log n)}$ is used to obtain a new coloring of size $O\big(\Delta^2 2^{o(\log n)}\big)$. In contrast to the typical use of polynomials to construct cover-free set families for Linial's graph coloring, syndrome compression uses an upper bound on the number of divisors, as described next.

Fact 4 ([3, Lemma 3], cf. [10])**.** *For a positive integer $N \geq 3$, the number of divisors of N is upper bounded by*

$$2^{1.6\frac{\log N}{\log(\log N/\log e)}}.$$

Construction 1. *Let k and r be positive integers and*

$$A = r2^{1.6\frac{\log k}{\log(\log k/\log e)}} + 1.$$

Define

$$F_i = \{(a, i \bmod a) : a \in [A]\}, i \in [k].$$

Construct $\mathcal{J} = \{F_1, \ldots, F_k\}$.

Lemma 5. *The family $\mathcal{J}$ created in Construction 1 is an r-cover-free set family over $[A] \times [A]$. The size of $\mathcal{J}$ is k, and the size of the ground set is*

$$O\Big(r^2 2^{3.2\frac{\log k}{\log(\log k/\log e)}}\Big).$$

Proof. For each F_i, F_j $(i \neq j)$, $|F_i \cap F_j|$ is the number of divisors of $|i-j|$, which is at most

$$2^{1.6\frac{\log k}{\log(\log k/\log e)}},$$

by Fact 4. So, $r|F_i \cap F_j| < A$ for $i \neq j$. Hence,

$$F_{i_0} \setminus \bigcup_{j=1}^{r} F_{i_j} \neq \emptyset$$

if $i_0 \notin \{i_1, \ldots, i_r\}$. Therefore, $\mathcal{J}$ is an r-cover-free set family. □

Theorem 3 is a restatement of syndrome compression obtained by applying Linial's algorithm (Fact 3) with respect to the cover-free set families given in Construction 1.

Theorem 3. *Let $\Delta = \Delta(G)$ and suppose there exists an old coloring of size $2^{o(\log n \log\log n)}$. Then one can get a new coloring of size $O\big(\Delta^2 2^{o(\log n)}\big)$.*

Proof. By assumption, G has an old coloring of size k, where $k = 2^{w(n)\log n}$ and $1 < w(n) = o(\log\log n)$. By Construction 1, there exists a Δ-cover-free set family of size k over a ground set of size

$$O\Big(\Delta^2 2^{3.2\frac{w(n)\log n}{\log(w(n)\log n/\log e)}}\Big) = O\Big(\Delta^2 2^{o(\log n)}\Big).$$

By Fact 3, we can get a new coloring of size $O\big(\Delta^2 2^{o(\log n)}\big)$. □

The main difference between our method and syndrome compression is the construction of r-cover-free set families. Note that to get an r-cover-free set family of size k, Construction 1 requires a larger ground set compared to Fact 2. In other words, for a specific ground set, Construction 1 allows a smaller r-cover-free set family than Fact 2. Therefore, if one wants to apply Construction 1 and Fact 3 to get a new coloring, one needs an old coloring of a small size in advance, given by the labeling. This restriction may complicate code construction. By contrast, in our method, we do not need a labeling of size $2^{o(\log n \log\log n)}$, and so the code construction is simpler.

References

[1] J. Brakensiek, V. Guruswami, and S. Zbarsky, "Efficient low-redundancy codes for correcting multiple deletions," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3403–3410, 2017.

[2] J. Sima and J. Bruck, "On optimal k-deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 67, no. 6, pp. 3360–3375, 2020.

[3] J. Sima, R. Gabrys, and J. Bruck, "Syndrome compression for optimal redundancy codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 751–756.

[4] ——, "Optimal systematic t-deletion correcting codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 769–774.

[5] ——, "Optimal codes for the q-ary deletion channel," in *2020 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2020, pp. 740–745.

[6] W. Song, N. Polyanskii, K. Cai, and X. He, "Systematic codes correcting multiple-deletion and multiple-substitution errors," *IEEE Transactions on Information Theory*, vol. 68, no. 10, pp. 6402–6416, 2022.

[7] L. Barenboim and M. Elkin, "Distributed graph coloring: Fundamentals and recent developments," *Synthesis Lectures on Distributed Computing Theory*, vol. 4, 07 2013.

[8] P. Erdös, P. Frankl, and Z. Füredi, "Families of finite sets in which no set is covered by the union ofr others," *Israel Journal of Mathematics*, vol. 51, pp. 79–89, 1985.

[9] G. Tenengolts, "Nonbinary codes, correcting single deletion or insertion (corresp.)," *IEEE Transactions on Information Theory*, vol. 30, no. 5, pp. 766–769, 1984.

[10] J.-L. Nicolas, "On highly composite numbers," in *Ramanujan revisited (Urbana-Champaign, Ill., 1987)*. Academic Press, Boston, MA, 1988, pp. 215–244.