Efficient Compression Algorithm For File Updates Under Random Insertions And Deletions

Qiwen Wang*, Muriel Médard†, and Mikael Skoglund*

*School of Electrical Engineering, Royal Institute of Technology
†Research Laboratory of Electronics, Massachusetts Institute of Technology

Email: {qiwenw, skoglund}@kth.se, {medard}@mit.edu

Abstract—The problem of one-way file synchronization from the client to the data-center, namely, file updates, is studied. The problem is investigated in particular when an old version of a file which can be available at both client and data-center, is edited by the client to a new version. The edits are modeled as random insertions and deletions (InDels). Based on the updated and the previous version of the file, the client transmits a message to the data-center via a noiseless link, such that the data-center can update the file. A dynamic-programming-run-length-coding (DP-RLC) scheme is proposed for the message encoding in this paper. The lower order terms of the achievable rate are computed explicitly. It is worth noting that these terms match the information-theoretic lower bound derived in our previous work [1]. Therefore, when the insertion and deletion probabilities are small, the achievable rate is nearly optimal.

I. INTRODUCTION

In distributed file backup systems, files are often edited lightly by clients, hence needed to be updated at the data-center. The problem of synchronization of correlated files with edits has been studied extensively in the literature, see Table 1 in [1] for a summary. One most related work [2] studies a one-way file synchronization problem with Markov deletion process. In [2], when the deletion probability is small, the lower order terms of the minimum compression rate to enable file updates is derived, which equal the entropy rate to describe the locations of insertions and deletions, and the contents of insertions and deletions in different runs. This case is more intricate. We characterize the entropy contributed to nature’s secret by this type of events by global alignment in [4], and prove (Lemma 5 of [4]) that the entropy rate of global alignment is also at least of second order in insertion/deletion probabilities.

In Example 1, the uncertainty is caused by the interaction of insertions and deletions in the same run, which occurs with probability of at least second order in insertion/deletion probabilities [3]. Hence, cases like this contribute to nature’s secret in second or higher order terms. In Example 2, the uncertainty of edit pattern is caused by interaction of insertions and deletions in different runs. This case is more intricate. We characterize the entropy contributed to nature’s secret by this type of events by global alignment in [4], and prove (Lemma 5 of [4]) that the entropy rate of global alignment is also at least of second order in insertion/deletion probabilities.

In [1], a simple scheme combining dynamic programming and entropy coding, hence called the DP-EC algorithm, is proposed. For the model with random InDels, the rate achieved by the DP-EC algorithm is shown to be within a constant additive gap to the information-theoretic lower bound, where the gap equals the first order term of nature’s secret. Although this gap decreases and converges to zero as the source alphabet size increases, for small source alphabet, this gap cannot be neglect. Hence, the problem lies whether this gap can be closed,
at least in first order terms, in our random InDel model.

In this work, we propose a dynamic-programming-run-length-coding (DP-RLC) algorithm. Briefly speaking, the algorithm firstly runs a dynamic program on the pair of original and new sequences, and outputs an edit pattern. The algorithm then groups the edits output from the dynamic program according to lengths of the runs where these edits occur in the original sequence. Finally, the algorithm uses entropy codes to compress the grouped edits. Our DP-RLC algorithm shares the same essence with the algorithm in [3], that is, to encode edits according to the lengths of the runs where they occur. The major contributions of our work is as follows.

At first, we study a more “natural” model (described in detail in Section II-A) of that in [3], especially for modeling editing behavior. In the model in [3], both the original sequence and the new sequence are generated from an auxiliary sequence through deletion processes. Then, the reverse process from the original sequence to the auxiliary sequence is regarded as an insertion process. Hence, the original sequence in [3] has variable length. In our work, the original sequence is a fixed length sequence of uniform i.i.d. symbols. The new sequence is generated by passing the original sequence through a random InDel process, with a natural stopping rule – reaching the end of file.

Moreover, the achievable rate of the algorithm in [3] is shown to be second order additive factor away from optimal. However, there is no explicit single-letter form expression for either a lower bound on the optimal compression rate, or the achievable rate by the algorithm.1 In this work, we compute explicitly the achievable rate, which matches our lower bound [1] in first order terms, including the first order term of nature’s secret. Therefore, DP-RLC algorithm outperforms our previous DP-EC algorithm. In addition, our previously derived lower bound is tight up to first order terms.

A. Notational Convention

Uppercase nonboldface symbols such as X are used to denote random variables. Sequences of random variables are denoted by boldface symbols, e.g., X is a sequence of the random variable X. Sets are denoted by calligraphic symbols, such as S. The cardinality of a set S is denoted by |S|. We use H(·) for entropy and conditional entropy of random variables. We denote standard binary entropy by H(·), that is, H(p) = -p log p - (1-p) log (1-p). All logarithms are binary.

II. MODEL

A. Source Sequences and Edit Sequences

Pre-edit source sequence (PreESS): The source initially has a pre-edit source sequence X = (X₁, X₂, ..., Xₙ), a length-n sequence of symbols drawn i.i.d. uniformly at random from the source alphabet . We artificially append an end of file symbol Xₙ₊₁ = eof to the end of X, to set a stopping rule for the InDel process.

InDel sequences: The edit process consists a sequence of edit operations (O₁, O₂, ...), modelled by a stationary three-state Markov model for the InDel process: Starting in front of the first symbol of X, at each step, the process inserts a symbol uniformly drawn from a with probability , reads one symbol rightwards and deletes it with probability , reads one symbol rightwards and does nothing with probability 1 - . The process stops when it reaches the end of file Xₙ₊₁ = eof.

Post-edit source sequence (PosESS): The sequence Y is obtained from X by applying random InDel E. The source encodes the source sequences (X, Y) into a transmission Enc(X, Y) and sends it to the decoder through a noiseless link. The decoder receives Enc(X, Y), and regenerates the PosESS Y’ from (Enc(X, Y), X).

Fig. 1: Markov model for the InDel process: Starting in front of the first symbol of X, at each step, the process inserts a symbol uniformly drawn from A with probability , reads one symbol rightwards and deletes it with probability , reads one symbol rightwards and does nothing with probability 1 - . The process stops when it reaches the end of file Xₙ₊₁ = eof.

Fig. 2: Communication Model: The source has access to both PreESS X and PosESS Y. The sequence Y is obtained from X by applying random InDel E. The source encodes the source sequences (X, Y) into a transmission Enc(X, Y) and sends it to the decoder through a noiseless link. The decoder receives Enc(X, Y), and regenerates the PosESS Y’ from (Enc(X, Y), X).

Markov chain as illustrated in Fig. 1, with initial distribution Pr(O₁ = ) = , Pr(O₁ = Δ) = δ, Pr(O₁ = η) = 1 - - . Specifically, the three states are: -the insertion state : insert a symbol uniformly drawn from A; -the deletion state Δ: read one symbol rightwards in X, and delete the symbol; -the no-operation state η: read one symbol rightwards in X, and do nothing.

The InDel process stops when it reaches the end of file Xₙ₊₁ = eof. Note that with the transition probabilities in Fig. 1, the edit operations are actually i.i.d. distributed. Hence, the edit operations (O₁, O₂, ...) form an i.i.d. process with a determined stopping time [6].² Let K_I denote the random variable of the number of insertions. The edit operation sequence Oⁿ⁺K_I is a length-n⁺K_I sequence of i.i.d. edit operations. Let a length-K_I sequence C⁰K_I over the source alphabet A denote the contents of insertions. The edit pattern under this model can be described by the edit operations and insertion contents, denoted by E = (Oⁿ⁺K_I, C⁰K_I).

Post-edit source sequence (PosESS): The post-edit source sequence Y = Y(X, E) is a sequence obtained from X by applying the edit pattern E.

Runs: We define a run as a maximal block of consecutively identical symbols. To avoid confusion, we occasionally use X-run to emphasize the run is from sequence X.

B. Communication Model

The communication system is as shown in Fig. 2. It is a problem of source compression with decoder side information, where the side-information is PreESS X. The encoder has both PreESS X and PosESS Y. The sequence Y is a function of X and edit pattern E. The sequences X and Y are encoded to a transmission Enc(X, Y) to the decoder through a noiseless link. Taking as inputs the transmission Enc(X, Y)
and PreESS $\mathbf{X}$, the decoder reconstructs the PosESS $\mathbf{Y}$ as $\mathbf{Y'}$. The code $C_{n,\delta}$ comprises the encoder-decoder pair. The average rate $R$ of the code $C_{n,\delta}$ is the average number of bits per source symbol transmitted by the encoder, defined as $\frac{1}{n} \sum p(\mathbf{X}, \mathbf{Y}) \log |\text{Enc}(\mathbf{X}, \mathbf{Y})|$. A code $C_{n,\delta}$ is $(1 - P_e)$-good if the average probability of error, i.e. $\Pr(\mathbf{Y}' \neq \mathbf{Y})$, is less than $P_e$. A rate $R_{e,\delta}$ is said to be achievable on average if for any $P_e \geq 0$, for all sufficiently large $n$, there is a code with average rate $R_{e,\delta}$ that is $(1 - P_e)$-good. In this step, we firstly classify the insertions in the $\mathbf{X}$-run, for all $l = 1, 2, \ldots, l_{\max}$.

**Theorem 1.** The optimal average compression rate for enabling updates is bounded from below by $R^*_e,\delta = \lim_{n \to \infty} \frac{1}{n} H(\mathbf{Y}|\mathbf{X}) \geq \mathcal{H}(\delta) + \mathcal{H}(\epsilon) + \epsilon \log \frac{\epsilon}{\delta} C_{|A|} - 56 \max(\epsilon, \delta)^{2-2} + \mathcal{O}(\max(\epsilon, \delta)^{2})$ for some $\tau \in (0, 1)$, where $C_{|A|} = \sum_{l=1}^{\infty} \left(\frac{1}{|A|}\right)^{l-1} \left(1 - \frac{1}{|A|}\right)^2 l \log l$ is a constant that depends only on the alphabet size $|A|$.

### III. DYNAMIC-PROGRAMMING-RUN-LENGTH-CODING (DP-RLC) ALGORITHM

We present the details of the proposed DP-RLC algorithm. **Encoder:** The output of the encoder $\text{Enc}(\mathbf{X}, \mathbf{Y})$ defined in Section II-B is generated as follows:

**Step 1 DP-enc:** The first subroutine of the encoder runs a dynamic program (DP) on the input $(\mathbf{X}, \mathbf{Y})$ to output an edit pattern. Using DP to find the edit distance between two sequences, i.e. the minimum number of edits needed to process on one sequence to get the other, is well-known in the literature. The computation in this step dominates the encoding time-complexity, which takes time $\mathcal{O}(\epsilon \log n^2)$ ([7]).

**Step 2 RL-Grouping-enc:** In this step, we firstly classify insertions into two types as following.

- **Type-E insertions:** This includes insertions which insert the same symbol composing the runs where they occur, hence they extend the lengths of X-runs.
- **Type-O insertions:** This includes all the other insertions, hence they create new runs and/or break X-runs. Specifically, when they occur within runs, they break these runs and form new runs themselves; when they occur at the positions between two runs, if the source alphabet has size at least three, it is possible that they create new runs themselves.

After obtaining the edit pattern output by DP-enc, the encoder groups deletions, and type-E insertions according to lengths of X-runs, as shown in Fig. 3. Specifically, denote the maximum length of X-runs by $l_{\max}$, and the number of X-runs with length $l$ by $n(l)$ for all $l = 1, 2, \ldots, l_{\max}$.

- **Deletions** – Let $\tilde{D}_{l,i}$ denote the number of deletions in the $i$th length-$l$ X-run. The sequence $\tilde{D}_{l,i}$ represents the numbers of deletions in all X-runs with length $l$. We use $\{\tilde{D}_{l,i}\}_{l=1}^{l_{\max}}$ to denote the set of sequences $\{\tilde{D}_{1,i}, \tilde{D}_{2,i}, \ldots, \tilde{D}_{l_{\max},i}\}$.

**Proof:** We firstly show in Lemma 3 below that the entropy of $\{\tilde{D}_{l,i}\}_{l=1}^{l_{\max}}$, $\{\mathbf{I}_{l,i}\}_{l=1}^{l_{\max}}$ and $\mathbf{I}^{O}$ is close to the entropy of similar sets of sequences $\{\tilde{D}_{l,i}\}_{l=1}^{l_{\max}}$, $\{\mathbf{I}_{l,i}\}_{l=1}^{l_{\max}}$ and $\mathbf{I}^{O}$ obtained from the original edit pattern $\mathbf{E}$. We then compute the entropy rates of these three sets of sequences in Lemma 4-6 sequentially. □

**Remark:** From Theorems 1 and 2, we observe that the lower bound and achievable rate match up to first order terms. Therefore, for small $\epsilon$ and $\delta$, DP-RLC is asymptotically
optimal. Moreover, compared with Theorem 3 in [1], which characterizes the rate achieved by DP-EC algorithm, it can be observed that DP-RLC outperforms DP-EC.

Suppose we know the actual edit pattern \( E \), we can group the edits in \( E \) according to lengths of \( X \)-runs with the same procedure in Step 2 of the DP-RLC algorithm, resulting in three similar groups of sequences, denoted by \( \{ D_l \}_{l=1}^{\max} \), \( \{ I_l \}_{l=1}^{\max} \), and \( \{ O_l \}_{l=1}^{\max} \). In Lemma 3 below, we prove that the description length required by the DP-RLC algorithm is close to the description length of these three groups of sequences obtained from the actual edit pattern.

**Lemma 3.** The description length required by the DP-RLC algorithm \( H(\{ D_l \}_{l=1}^{\max}) + H(\{ I_l \}_{l=1}^{\max}) + H(\{ O_l \}_{l=1}^{\max}) \) is close to the description length of sequences \( H(\{ D_l \}_{l=1}^{\max}) + H(\{ I_l \}_{l=1}^{\max}) + H(\{ O_l \}_{l=1}^{\max}) \) corresponding to the actual edit pattern. Specifically,

\[
|H(\{ D_l \}_{l=1}^{\max}) - H(\{ D_l \}_{l=1}^{\max})| \leq n \cdot O(\max(\epsilon, \delta)^2),
\]

\[
|H(\{ I_l \}_{l=1}^{\max}) - H(\{ I_l \}_{l=1}^{\max})| \leq n \cdot O(\max(\epsilon, \delta)^2),
\]

\[
|H(\{ O_l \}_{l=1}^{\max}) - H(\{ O_l \}_{l=1}^{\max})| \leq n \cdot O(\max(\epsilon, \delta)^2).
\]

**Proof:** Note that for the same \( X \) and for any \( l \), \( D_l \) has the same length as \( \hat{D}_l \), because the algorithm groups edits according to the length of \( X \)-runs. We argue below that at most \( O(n(\max(\epsilon, \delta)^2)) \) edits can be performed in \( \{ D_l \}_{l=1}^{\max} \) and \( \{ I_l \}_{l=1}^{\max} \).

We first define the extended run of an \( X \)-run to be the subsequence of \( X \) including the run and its two neighbouring symbols. E.g., in ‘00110’ the extended run of the run ‘11’ is ‘00110’. For an \( X \)-run, the number of deletions in its output by DP may differ from the number of deletions in the original edit-pattern only if one of following two cases occur:

**Case 1:** there is more than one edit in the extended run of this \( X \)-run (recall Example 1 in Section I);

**Case 2:** although there is only one edit in its extended run, the interaction of insertions and deletions in different runs leads to uncertainty in the edit pattern (recall Example 2 in Section I).

It is straightforward that Case 1 occurs with probability at least in second order term, i.e., \( O(\max(\epsilon, \delta)^2) \). Case 2 is more intricate. It corresponds to an ambiguous local alignment event defined in [4]. We proved in Lemma 5 in [4] that on average (over \( X \) and \( E \)), Case 2 also occurs with probability at most \( O(n \max(\epsilon, \delta)^2) \). The intuition is that, not too many \( X \)'s are possible to have ambiguous local alignment (uncertainty in the edit pattern). Specifically, the fraction of such \( X \)'s decreases geometrically with the distance between the interacting edits. If the edits that interact occur close to each other in \( X \), it happens with probability \( O(n \max(\epsilon, \delta)^2) \). If they occur at locations far away in \( X \), the fraction of such \( X \)'s which is possible to have uncertainty in the edit pattern goes to zero. From Case 1 and Case 2, on average at most \( O(n \max(\epsilon, \delta)^2) \) terms are different in \( \{ D_l \}_{l=1}^{\max} \) and \( \{ D_l \}_{l=1}^{\max} \). Hence, for some \( n \), we have \( E[n(l)] = n \left( 1 - \frac{1}{|X|} \right)^2 \left( \frac{1}{|X|} \right)^{l-1} \).

For type-E insertions, recall that \( \Gamma_0 = (\Gamma_0, \ldots, \Gamma'_{\infty}) \) and \( \Gamma = (\Gamma_0, \ldots, \Gamma'_{\infty}) \), it is obvious that \( \hat{H}(\Gamma_{\infty}) \) and \( H(\Gamma_{\infty}) \) have the same length. For similar reasons, they differ in at most \( O(\max(\epsilon, \delta)^2) \) edit locations. Hence, we also have \( |H(\{ D_l \}_{l=1}^{\max}) - H(\{ D_l \}_{l=1}^{\max})| \leq n \cdot O(\max(\epsilon, \delta)^2) \).

**Lemma 4.** The asymptotic entropy rate of \( \{ D_l \}_{l=1}^{\max} \) obtained from the original edit pattern is \( \lim_{n \to \infty} \frac{1}{n} H(\{ D_l \}_{l=1}^{\max}) = \mathcal{H}(\hat{\delta}) - \delta \sum_{l=1}^{\infty} \left( \frac{1}{|A|} \right)^2 \left( \frac{1}{|X|} \right)^{l-1} \log l + O(\max(\epsilon, \delta)^2), \) for some \( \tau \in (0, 1) \).

**Proof:** The number of deletions in a length-\( l \) \( X \)-run is Binomial\( l, \frac{\epsilon}{|A|} \) distributed. Denote its probabilistic distribution by \( \Pr_{\text{DP}} \), i.e., \( \Pr_{\text{DP}}(d) = C(l, d) \frac{(1 - \epsilon - \delta)}{(1 - \epsilon)}^{l-1} \). Hence,

\[
H(\Pr_{\text{DP}}) = \sum_{d=0}^{l} \left( \frac{l}{d} \right) \frac{\epsilon}{|A|} \frac{d}{(1 - \epsilon)^l} \left( \frac{1}{1 - \epsilon} \right)^{l-1} \log l + \sum_{d=2}^{l} \left( \frac{l}{d} \right) \frac{\epsilon}{|A|} \frac{d}{(1 - \epsilon)^l} \log l + \sum_{d=2}^{l} \left( \frac{l}{d} \right) \frac{\delta}{(1 - \epsilon)^l} \log l
\]

\[
\leq \mathcal{H}(\hat{\delta}) - \delta \log |A| + O(\max(\epsilon, \delta)^2) + \sum_{d=2}^{l} \left( \frac{l}{d} \right) \frac{\delta}{(1 - \epsilon)^l} \log l
\]

\[
\leq \mathcal{H}(\hat{\delta}) - \delta \log |A| + O(\max(\epsilon, \delta)^2) - \delta^2 \log l \left( \frac{1 - \epsilon}{1 - \epsilon^3} \right).
\]

For some \( \tau \in (0, 1) \). Recall that \( n(l) \) denotes the number of \( X \)-runs with length \( l \), we have \( E[n(l)] = \left( 1 - \frac{1}{|A|} \right)^2 \left( \frac{1}{|X|} \right)^{l-1} \).

Hence, \( H(\{ D_l \}_{l=1}^{\max}) = E[\sum_{l=1}^{\max(l)} E[n(l)] H(\Pr_{\text{DP}})] = n \mathcal{H}(\hat{\delta}) - \delta \sum_{l=1}^{\infty} \left( \frac{1 - \epsilon}{1 - \epsilon^3} \right) \frac{1}{|X|^{l-1}} \log l + n \cdot O(\max(\epsilon, \delta)^2) \).
Lemma 5. The asymptotic entropy rate of \( \{I^E_{l=1}^{l_{max}}\} \) obtained from the original edit pattern is
\[
\lim_{n \to \infty} n H(I_{l=1}^{l_{max}}^{E}) = (2 - \frac{1}{|A|}) H(\epsilon) + n \log |A| + \epsilon \sum_{i=1}^{\infty} \left(1 - \frac{1}{|A|}\right)^{2} l^{-1} l \log l + O(\epsilon^{2-\tau}), \text{ for some } \tau \in (0, 1).
\]

Proof: The number of insertions in a length-\( l \) \( X \)-run is negative binomial \( NB(l + 1, \epsilon) \) distributed. Each insertion extends the run with probability \( \frac{1}{|A|} \). Denote the probabilistic distribution of number of type-E insertions in a length-\( l \) \( X \)-run by \( Pr_{T_E} \), we have
\[
Pr_{T_E}(i) = \sum_{k=i}^{\infty} \binom{k + l}{l} (1 - \epsilon)^{l+1} \binom{k}{i} \left(1 - \frac{1}{|A|}\right)^{k-i} \left(\frac{1}{|A|}\right)^{i}.
\]

With similar calculations as in Lemma 4, we have
\[
H(Pr_{T_E}) \leq \frac{\epsilon}{|A|} H(\epsilon) + (l + 1) \frac{1}{|A|} \log |A| - \frac{1}{|A|} (l + 1) \log (l + 1) + O(\epsilon^{2-\tau}) \text{ for some } \tau \in (0, 1).
\]

Hence,
\[
H(I_{l=1}^{l_{max}}^{E}) = E\left[\sum_{l=1}^{l_{max}} n(l) H(Pr_{T_E})\right] = n H(\epsilon) \left(\frac{2}{|A|} - \frac{1}{|A|^2}\right) + n \epsilon \log |A| \left(\frac{2}{|A|} - \frac{1}{|A|^2}\right) - cn \sum_{l=1}^{\infty} \left(1 - \frac{1}{|A|}\right)^{2} l^{-1} \log l + O(\epsilon^{2-\tau}) n.
\]

The intuition of the coefficient \( \frac{2}{|A|} - \frac{1}{|A|^2} \) in Lemma 5 is as follows. From the perspective of the whole sequence \( X \), insertions within \( X \)-runs extend runs with probability \( \frac{1}{|A|} \), and insertions between two \( X \)-runs extend runs with probability \( \frac{1}{|A|^2} \) (they may extend the run on either left side or right side). On average, there are \( n(1 - \frac{1}{|A|^2}) \) \( X \)-runs. Hence, out of \( n \) possible positions for insertions, on average \( n(1 - \frac{1}{|A|^2}) \) of them are between two \( X \)-runs.\(^4\) Hence, on average, there are \( n(1 - \frac{1}{|A|^2}) \) \( \epsilon \cdot \frac{2}{|A|} + n \frac{1}{|A|^2} \) type-E insertions.

Based on the above observations, on average there are \( n \epsilon \left(1 - \frac{2}{|A|} + \frac{1}{|A|^2}\right) \) type-O insertions, which provides an intuition of the coefficient in Lemma 6 below. We omit the rigorous proof here due to page limit.

Lemma 6. The asymptotic entropy rate of \( I^O \) from the original edit pattern is
\[
\lim_{n \to \infty} \frac{1}{n} H(I^O) = \left(1 - \frac{2}{|A|} + \frac{1}{|A|^2}\right) H(\epsilon) + \left(\frac{2}{|A^2|} - \frac{1}{|A|^3}\right) \epsilon \log |A| + O(\max(\epsilon, \delta)^{2-\tau}), \text{ for some } \tau \in (0, 1).
\]

Combining Lemma 3-6, we have that the asymptotic compression rate of DP-RLC algorithm is bounded by
\[
\lim_{n \to \infty} \frac{1}{n} \left(H(D_{l=1}^{l_{max}}) + H(I_{l=1}^{l_{max}}^{E}) + H(I^O)\right) \leq \lim_{n \to \infty} \frac{1}{n} H(D_{l=1}^{l_{max}}) + \lim_{n \to \infty} \frac{1}{n} H(I_{l=1}^{l_{max}}^{E}) + \frac{1}{n} H(I^O) + O(\max(\epsilon, \delta)^{2})
\]
\[
\leq H(\delta) + H(\epsilon) + \epsilon \log |A| + \left(\frac{1}{|A^2|} - \frac{1}{|A|^3}\right) \epsilon \log |A|
\]
\[
(\delta + \epsilon) \sum_{l_{min}}^{\infty} \left(1 - \frac{1}{|A|}\right)^{2} \left(\frac{1}{|A|}\right)^{l-1} l \log l + O(\max(\epsilon, \delta)^{2-\tau}),
\]
for some \( \tau \in (0, 1) \), hence proved Theorem 2.

\(^4\)In fact, there are \( n \) possible positions for insertion, and on average \( n(1 - \frac{1}{|A|^2}) \) \(-1\) positions are between two \( X \)-runs. However, asymptotically as \( n \) grows, these boundary effects are negligible.

V. Conclusion and Discussion

We have studied the optimal compression rate for file updates via random insertions and deletions, with the old file present at both the encoder and the decoder. We have proposed an algorithm, and computed explicitly its achievable rate. The achievable rate matches the lower bound up to first order terms, including the first order term of nature's secret.

In [9], the authors proved that the capacity of the i.i.d. deletion channel is \( \lim_{n \to \infty} \frac{1}{n} \max_{p(X)} I(X; Y) \). One interesting observation is that, if this can be extended to the InDel channel with random InDels as in our model, our tight bound leads to the maximum achievable rate with uniform i.i.d. input, hence can provide a lower bound on the channel capacity of such InDel channels. This is still to be proved.

One extension of our work is to consider also substitutions. Briefly speaking, considering that an edit operation in our model has probability \( \theta \) to be a substitution, our algorithm only needs to add one more group, i.e. substitutions, in Step 2. Substitutions behave similarly as type-O insertions, that is, they break and create runs. Hence, it is not difficult to see that it only adds \( H(\theta) + \theta \log |A| - 1 \) to the achievable rate. The lower bound is more intricate to analyze. We might conjecture (based on the analysis for the case with only insertions and deletions in [4]) that the interaction of substitutions and InDels also contribute to second order terms of nature’s secret. This is currently under investigation.

Another extension is to consider more general Markov edit models. The model studied in this work has independent small insertion and deletion probabilities, hence generates isolated InDels. An interesting extension is to allow the edit process to stay in insertion (deletion) state with fixed probabilities, which generates isolated blocks of insertions (deletions) with geometrically distributed lengths. In [2] the authors studied general Markov deletion model. The case with generak Markov InDel model is one direction of our future work.

References


