Design & Analysis of Optimal Coin-tossing: New Techniques

Hamidreza Amini Khorasgani
Department of Computer Science, Purdue University, USA
haminikh@purdue.edu

Hemanta K. Maji
Department of Computer Science, Purdue University, USA
hmaji@purdue.edu

Mingyuan Wang
Department of Computer Science, Purdue University, USA
wang1929@purdue.edu

Abstract

Collective coin-tossing allows n processors with private randomness sources to agree on a common public coin. Without loss of generality, one can assume that the output is in the set $\{0,1\}$, and the expected output of a coin-tossing protocol is X. The objective of a coin-tossing protocol is to be robust to adversarial interventions. In this paper, we study Byzantine adversaries who can arbitrarily set the messages of the corrupted processors.

Historically, the study of coin-tossing protocols, with the introduction of even the mildest of variations in its setting, tends to yield surprising and exciting outcomes. We know several optimal or asymptotically optimal protocols like tribes, baton passing, and threshold protocols. Incidentally, there are several variants of coin-tossing where the majority protocol (or, more generally, the threshold protocols) turn out to be asymptotically optimal. In this work, we consider coin-tossing protocols in two security models and study the susceptibility of the optimal coin-tossing protocols in those settings to adversarial attacks.

In the first model, there are n processors and processor i broadcasts her uniformly and independently random message $x_i \in \{0,1\}$. The processors apply a function $f_n \colon \{0,1\}^n \to \{0,1\}$ to the broadcast messages and agree on their common output $f_n(x_1,\ldots,x_n)$. After all the processors broadcast their messages, the adversary may corrupt at most t processors and change their messages arbitrarily. The optimal protocol minimizes the change in the expected output that this adversary causes. We reduce this problem to an isoperimetric inequality over the boolean hypercube and demonstrate that the threshold protocols are the optimal protocols.

In the second model, at time i, processor i broadcasts her message x_i, and her message distribution possibly depends on the previously broadcast messages. We consider an adversary who can take control of one processor and change her message arbitrarily. In this case, we prove that the threshold protocols are asymptotically optimal.

Keywords and phrases Multi-party Coin-tossing, Adaptive Adversaries, Optimal Protocols, Isoperimetric Inequalities

Funding The research effort is supported in part by an NSF CRII Award CNS–1566499, an NSF SMALL Award CNS–1618822, the IARPA HECTOR project, MITRE Innovation Program Academic Cybersecurity Research Award, a Purdue Research Foundation (PRF) Award, and The Center for Science of Information, an NSF Science and Technology Center, Cooperative Agreement CCF–0939370.

1 Introduction

Collective coin-tossing allows n-processors with unbounded computational power to agree on a common random coin. These processors have access to a broadcast channel, and each processor has a private source of randomness. In this paper, we consider *single-turn* protocols, i.e., each processor broadcasts a message only once during the evolution of the coin-tossing protocol. Without loss of generality, we assume that the output of the coin-tossing protocol is a bit, and we denote the expected output of the honest protocol by X_0, referred to as bias-X_0. An adversary may corrupt up to t processors to change the expected output of the protocol (it may either increase or decrease the expected output). In this paper, we consider *Byzantine* adversaries, i.e., the adversary may fix the message of the corrupted processors arbitrarily.

A coin-tossing protocol may proceed in multiple rounds. The distribution of the messages sent by processors prescribed to speak in one round may depend on the messages sent in the previous rounds. For example, in one-round protocols, the distribution over the message space of the coin-tossing protocol is a product space. On the other hand, in n-round protocols, only one processor speaks in a round, and her message distribution possibly depends on all previous messages. As is standard in cryptography, our adversary is always *rushing*, i.e., it can arbitrarily schedule all those processors who are supposed to speak in a round.

Given a setting for coin-tossing and the adversarial model, a typical objective is to identify the optimal or asymptotically optimal protocols realizing bias-X_0 coin-tossing. Historically, the study of coin-tossing protocols, with the introduction of even the mildest of variations in its setting, tends to yield surprising and exciting outcomes. We have encountered several optimal coin-tossing protocols like tribes, baton passing, and threshold protocols, and each of them has had a significant impact on research in discrete mathematics and theoretical computer science. In fact, for most models, we do not know the characterization of the optimal coin-tossing protocol for arbitrary bias-X_0.

In this paper, we study two models for coin-tossing.

1. Single-turn, one-round coin-tossing protocols against a *strongly* adaptive adversary. A strongly adaptive adversary can see a processor's message before deciding to change her message arbitrarily.
2. Single-turn, n-round coin-tossing protocols against an adaptive adversary. Here we consider the typical definition of an adaptive adversary who has to corrupt a processor before learning her honest message.

We shall prove that the *threshold protocols are the optimal coin-tossing protocols* in the first model. Note that the only difference between the model for the original conjecture of Ben-Or and Linial [BL89] from the model we study is that the former considers adaptive adversaries (not strongly adaptive). This result establishes connections to a new form of vertex isoperimetric inequality to characterize the optimal coin-tossing protocol. While in the second model, we prove that *threshold protocols are asymptotically optimal*. However, experimentally, we verify that the optimal protocol corresponds to threshold protocols as well. This conjecture, however, remains to be proven formally. This result uses the potential-based inductive technique introduced recently by [KMM19] to inductively lower bound the performance of the best attack on such coin-tossing protocols.

1.1 Our Contributions

Our first result is in the following setting. There are n processors, and processor i broadcasts her uniformly (and independently) random message $x_i \in \{0,1\}$. The processors apply a function $f_n \colon \{0,1\}^n \to \{0,1\}$ to the broadcast messages and agree on their common output $f_n(x_1,\ldots,x_n) \in \{0,1\}$. Let X_0 represent the expected outcome of the protocol. We consider the following adversarial model. After all processors broadcast their messages, the adversary decides to change at most $t \in \{1,\ldots,n\}$ processor's messages arbitrarily. The objective of the adversary is to deviate the expected outcome away from X_0. The *optimal protocol* ensures that the maximum deviation caused by any adversary in the adversarial model above is minimized.

The theorem below summarizes our result in this setting.

▶ **Theorem 1** (Coin-tossing and Harper's Theorem). *Consider single-turn one-round coin-tossing protocols involving n processors, each processor broadcasts a uniformly and independently random bit, and the expected output of the protocol is X_0. Consider strong adaptive Byzantine adversaries who can corrupt at most $t \in \{1,\ldots,n\}$ processors. The optimal protocol in this setting outputs 1 if and only if $(x_1,\ldots,x_n) \in S_{n,X_0}$, where $S_{n,X_0} \subseteq \{0,1\}^n$ represents the subset of the first $X_0 2^n$ elements in the* simplicial ordering *of $\{0,1\}^n$.*

The simplicial ordering is a total order on $\{0,1\}^n$ defined as follows. For $x, y \in \{0,1\}^n$, we say that $x \leq y$ if and only if (1) The number of ones in x is more than the number of ones in y, or (2) The number of ones x and y are identical, but x has one at the first coordinate where x and y differ.

Note that if n is odd and $X_0 = 1/2$ then the majority protocol is the optimal protocol. Here we emphasize that we are claiming that majority is *not* just asymptotically optimal. It is in fact *the* optimal protocol. In general, when $X_0 = \mathrm{Vol}\,(n,k)\,/2^n$,[1] the threshold protocol[2] is the optimal protocol. We prove this result by establishing connections to (a new variant of) vertex isoperimetric inequalities over the boolean hypercube. An interesting problem is to generalize this result to the setting where processors output uniformly random messages from larger sets.

Our next result is in the following setting. There are n processors and at time i processor i broadcasts her message x_i. The distribution of the i-th processor's message possibly depends on the messages broadcast by the first $(i-1)$ processors. At the end of the protocol, parties agree on the output $f_n(x_1,\ldots,x_n) \in \{0,1\}$. The expected output is X_0. We consider an adaptive Byzantine adversary who corrupts at most one processor.

We prove the following result.

▶ **Theorem 2** (Coin-tossing and Potential Argument). *Consider any single-turn n-round coin-tossing protocol involving n processors with an expected output X_0. There exists an adaptive Byzantine adversary who can corrupt at most $t = 1$ processor and change the expected output of the protocol by*

$$\geq \frac{X_0(1-X_0)}{\sqrt{2}} \cdot \frac{1}{\sqrt{n}}.$$

---

[1] $\mathrm{Vol}\,(n,k)$ represents the total number of n-bit bitstrings with $\leq k$ ones. That is, we have $\mathrm{Vol}\,(n,k) = \sum_{i=0}^{k} \binom{n}{i}$.

[2] A threshold protocol outputs 1 if the number of processors who send 1 as their messages exceeds a given threshold. In particular, when $X_0 = \mathrm{Vol}\,(n,k)\,/2^n$, the threshold is $n-k-1$.

Consider odd n and $X_0 = 1/2$. The majority protocol's expected outcome can be changed by at most $\sim \frac{\sqrt{2}}{\sqrt{\pi}} \cdot \frac{1}{\sqrt{n}}$. Therefore, our lower bound is within a constant-factor of the optimal value.

Interestingly, the optimal solutions for the potential argument for small values of odd n turn out to be the majority protocol. This leaves open the possibility that the majority protocol may be the optimal protocol in this setting when $X_0 = 1/2$.

1.2 Prior Works

There is a vast literature that studied coin-tossing protocols against static Byzantine adversaries. The celebrated result of Kahn, Kalai, and Linial [KKL88] implied that any one-round protocol could be fully biased by corrupting $\widetilde{\Omega}(n/\log n)$ processors. On the positive side, Ben-Or and Linial [BL89] constructed a boolean function that is resilient to $\mathcal{O}(n^{0.63})$ corruptions. This result is improved by Ajtai and Linial [AL93], who gave a construction that is resilient to $\mathcal{O}(n/\log^2 n)$ corruptions. Their construction was inexplicit and recently made explicit by Chattopadhyay and Zuckerman [CZ16].

In the multi-round setting, Saks [Sak89] constructed a protocol called the "Baton Passing" game and showed its resilience to $\mathcal{O}(n/\log n)$ corruptions. Alon and Naor [AN90] modified their construction and showed that it is resilient to a constant fraction of corruption. Boppana and Narayanan [BN93] improved the analysis and showed resilience of such protocols to $(1/2 - \delta)$ fraction of corruption, which is optimal. Feige [Fei99] gave an explicit construction that is also resilient to $(1/2 - \delta)$ fraction of corruption. His protocol has round complexity $\mathcal{O}(\log^* n)$,[3] which matches the lower bound proven by Russell, Saks, and Zuckerman [RSZ99].

For adaptive Byzantine adversaries, Ben-Or and Linial [BL89] showed that majority protocol is resilient to $\mathcal{O}(\sqrt{n})$ corruptions, and they conjectured this is asymptotically optimal. That is, no protocol is resilient to more than $\mathcal{O}(\sqrt{n})$ corruption. Lichtenstein, Linial, and Saks[LLS89] proved that this conjecture is true when every processor sends a single bit during the entire protocol. Recently, there have been several works making progress on this conjecture. Goldwasser, Kalai, and Park [GKP15] showed that this conjecture holds if it is a one-round symmetric protocol.[4] En route to this result, they introduced strong adaptive adversaries and showed that any one-round protocol is not resilient to $\widetilde{\mathcal{O}}(\sqrt{n})$ strong adaptive corruptions. Kalai and Komargodski [KK15] showed how to compress the communication complexity of such protocols to $\mathsf{polylog}(n)$ without compromising the security of the protocol. Based on this, Kalai, Komargodski, and Raz [KKR18] proved Ben-Or and Linial's conjecture for all single-turn protocols.

Recently, in the field of fair coin-tossing, Khorasgani, Maji, and Mukherjee [KMM19] introduced the approach of geometric transformation for designing optimal protocols. They showed that this approach yields protocols with less susceptibility than the majority protocols [Blu82, Cle86].

1.3 Technical Overview

Result using Isoperimetric Inequality. Let $S \subseteq \{0,1\}^n$ be the set of all $x \in \{0,1\}^n$ such that $f_n(x) = 1$. For intuition consider the case of $t = 1$. Note that if $x \in S$ is such

[3] $\log^* n$ is defined as the smallest positive integer k such that $\underbrace{\log \cdots \log}_{k \text{ times}} n < 1$.

[4] A one-round protocol is called symmetric if the output of the protocol is oblivious to the order of the messages.

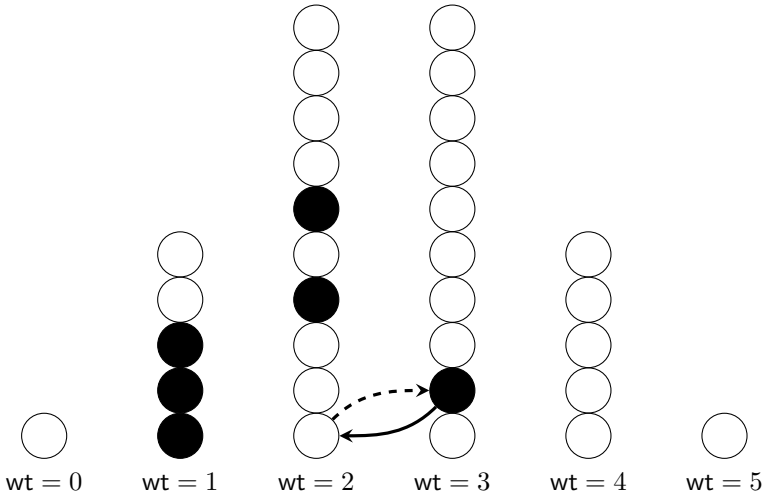

Figure 1 An intuitive example on hypercube $\{0,1\}^5$. Any compression of the "black ball set" is also a compression of the "white ball set." The solid arrow compresses the black ball set, and the dotted arrow compresses the complementary white ball set.

that one of its distance-1 neighbor in the boolean hypercube is in $\overline{S} := \{0,1\}^n \setminus S$ then the adversary can change the outcome of the protocol to 0. On the other hand, if no distance-1 neighbor of x in the boolean hypercube is in $\overline{S}$ then the adversary cannot change the outcome of the protocol to 0.

Note that every element in the vertex perimeter of $\overline{S}$, represented by $\partial\overline{S}$, is an element where the adversary can change the output from 1 to 0. Furthermore, this adversary cannot change the outcome from 1 to 0 at any other x. So, we conclude that the adversary can reduce the expected outcome by $\left|\partial\overline{S}\right|/2^n$. Analogously, an adversary can increase the expected outcome by $|\partial S|/2^n$.

Consequently, to construct the optimal coin-tossing protocol, our objective is to minimize

$$\max\left\{\left|\partial\overline{S}\right|, |\partial S|\right\}.$$

Harper's theorem minimizes $\left|\partial\overline{S}\right|$ and $|\partial S|$ individually. Our objective is to *simultaneously* minimize them. Is it possible that one can trade off one of these terms for the other?

The proof proceeds by considering a *compression*-based proof of the Harper's theorem. Intuitively, a compression of a set moves the points closer to each other. The crucial observation is the following.

"The complement of any compression of a set
is also a compression of the complement of that set."

Figure 1 provides this intuition for $n = 5$. Therefore, one can translate any compression based proof for Harper's theorem to our context as well. This observation implies that the optimal S is obtained by choosing appropriate number of points in the boolean hypercube according to the simplicial order. Interestingly, the proof implies that it is *impossible* to increase the susceptibility to attacks biasing towards 0 as compared to the optimal protocol while gaining robustness to attacks biasing towards 1.

Result using Potential Arguments. Recently, Khorasgani, Maji, and Mukherjee [KMM19] introduced a potential based inductive technique to account for good adversarial attacks on a coin-tossing protocol. We use that framework for our next result. The potential accounts

for the sum of the positive and negative change in the expected outcome that an adversary can cause.

Suppose we already know the best attack on protocols of depth d. Let $C_d(X)$ represent a lower-bound on the best attack on one-turn d-round coin-tossing protocols involving d processors, and the expected output of the honest protocol is X. Next, we consider any one-turn $(d+1)$-round coin-tossing protocols involving $(d+1)$ processors. Note that in the beginning, the adaptive adversary (who can corrupt at most $t = 1$ processors) can either corrupt the first processor and fix her outcome, or delay the corruption to a later round depending on the evolution of the coin-tossing protocol. We express this decision as a geometric transformation on the curve $C_d(X)$ to obtain the new $C_{d+1}(X)$ curve (see Figure 2).

A closed form expression of this curve seems difficult to obtain. So, we proceed to lower bound it using an easier to express curve. This procedure proceeds inductively in the following manner.

1. We prove that the geometric transformation of Figure 2 preserves the relative ordering of two curves. That is, if one curve is above another curve, then the transformation of the former curve is above the transformation of the latter curve.
2. Next, we obtain a simple lower bound to $C_1(X)$ and inductively proceed to find a similar curve that lower bounds $C_2(X)$, and so on. We show that $D_n(X) := \sqrt{2}X(1-X) \cdot \frac{1}{\sqrt{n}}$ is below $C_n(X)$.

Finally, one observes that there exists an adaptive Byzantine strategy that corrupts at most $t = 1$ processors to change the final outcome by $\frac{1}{2}D_n(X)$ (because, either the positive or negative change in the expected output shall surpass the average).

2 Preliminaries

For a universe Ω and a set $S \subseteq \Omega$, we use $\overline{S}$ to denote the complement of S, i.e., $\Omega \backslash S$. We use $|S|$ to represent the cardinality of S. We use $[n]$ for set $\{1, 2, \ldots, n\}$. For any set S and integer $0 \leq n \leq |S|$, we use $\binom{S}{n}$ to represent the collection of subsets of S of size n, i.e., $\{T \subseteq S \,|\, |T| = n\}$.

For binary strings $x, y \in \{0,1\}^n$, the Hamming weight is defined as $\mathsf{wt}(x) := |\{i \in [n] | x_i = 1\}|$; the Hamming distance is defined as $\mathsf{HD}(x,y) := |\{i \in [n] | x_i \neq y_i\}|$. We use $\mathrm{Vol}(n,k)$ to denote the size of a Hamming ball of radius k, i.e., $\mathrm{Vol}(n,k) := \sum_{i=0}^{k} \binom{n}{i}$.

2.1 Coin-Tossing Protocols

In this work, we consider coin-tossing protocols among n processors in the *full information* model. That is, all processors communicate through one single broadcast channel. This protocol might consist of multiple rounds. Within each round, the processors who are supposed to speak shall broadcast their messages simultaneously. At the end of the protocol, all processors will reconstruct the output $\in \{0,1\}$ by applying the same function on the broadcasted messages. Hence, they will always agree on the output of the protocol. We call it an m-turn protocol if every processor sends at most m messages throughout the protocol. In particular, a single-turn protocol implies that every processor broadcasts a single message during the entire protocol. We do not limit to coin-tossing protocols with unbiased output. That is, the probability of the output being head could be any real number in $[0,1]$.

▶ **Definition 1** ((n, X_0)-Coin-tossing protocols)**.** An (n, X_0)-coin-tossing protocol is a coin-tossing protocol among n processors, where the expectation of the output is X_0.

The *susceptibility* of a coin-tossing protocol is the maximum change (in terms of statistical distance) that the adversary can cause to the distribution of the output of the protocol.

2.2 Adversarial Models

In this work, we focus on *Byzantine* adversaries, i.e., once a processor is corrupted, the adversary takes full control over its behavior. We assume the adversaries are *rushing*, which means it can schedule the order of which processors broadcast their messages within each round. Specifically, we consider the following two types of adversaries.

- **Adaptive Adversary.** An adaptive adversary does not commit to which processors to corrupt before the protocol begins (i.e., *static*). In contrast, it decides on which processors to corrupt in the course of the protocol. However, it cannot alter messages that have already been sent.
- **Strong Adaptive Adversary [GKP15].** A strong adaptive adversary has the additional power to decide on whether to corrupt a processor after seeing its message. Without loss of generality, such adversaries, within each round, first wait for all the processors to broadcast their messages and then decide which processors to corrupt (to alter their messages).

3 Optimal Coin-Tossing Protocols for Strong Adaptive Adversary

In this section, we study strong adaptive adversaries. This adversarial model is proposed and studied by Goldwasser, Kalai, and Park [GKP15]. In particular, we consider a one-round protocol, where every processor sends a uniform bit as its message.

We shall show that, in this setting, majority protocols (more generally, threshold protocols for biased output), are the optimal protocol for such adversaries. Note that we are not claiming that it is asymptotically optimal, it is *the* optimal protocol. We prove our results by drawing connections from isoperimetric inequalities on boolean hypercubes.

In Section 3.1, we provide some basics about isoperimetric inequalities on boolean hypercubes. In Section 3.2, we shall show how one can apply isoperimetric inequalities to coin-tossing problems.

3.1 Isoperimetric inequalities on boolean hypercubes

For a graph G, isoperimetric inequalities consider, of a fixed size, which subgraph of G minimizes the size of its vertex boundary. Let us define vertex boundary first.

▶ **Definition 2** (Vertex Boundary). For a graph $G = (V, E)$ and a subset of vertices $S \subseteq V$, the vertex boundary of S is defined as

$$\partial S := \left\{ \overline{s} \in \overline{S} \;\middle|\; \exists s \in S \text{ s.t. } (s, \overline{s}) \in E \right\}.$$

More generally, the t-vertex boundary is defined as

$$\partial^t S := \left\{ \overline{s} \in \overline{S} \;\middle|\; \exists s \in S \text{ s.t. } \mathsf{dist}(s, \overline{s}) \leq t \right\}.$$

In particular, let graph $G = (V, E)$ represent the boolean hypercube, which is defined as, $V = \{0,1\}^n$ and for any two vertices $u, v \in V$, $(u, v) \in E$ if and only if $\mathsf{HD}(u, v) = 1$. The celebrated Harper's theorem [Har66] states that, for the boolean hypercube G, the subgraph that minimizes its vertex boundary, is exactly the subgraph induced by the prefix of the simplicial ordering, which is defined as follows.

▶ **Definition 3** (Simplicial Ordering). For any two distinct elements $a, b \in \{0,1\}^n$, we say $a < b$ if one of the following two conditions is met.

1. $\mathsf{wt}(a) < \mathsf{wt}(b)$;
2. $\mathsf{wt}(a) = \mathsf{wt}(b)$, but a is smaller than b in lexicographical ordering.

Let $L^n(s)$ be the first s smallest elements of $\{0,1\}^n$ in simplicial ordering. Then Harper's theorem guarantees the following.

▶ **Theorem 3** (Harper's theorem). *For all integers $n > 0$, $0 \le s \le 2^n$, and for all $S \in \binom{\{0,1\}^n}{s}$, we have*

$$|\partial L^n(s)| \le |\partial S|\,.$$

3.2 The connections between isoperimetric inequalities and coin-tossing protocols

Consider an (n, X_0)-coin tossing protocol, where every processor sends a uniform bit as its message. We consider strong adaptive adversaries who are allowed to corrupt at most t processors. In this setting, an (n, X_0)-coin-tossing protocol is solely determined by the function

$$f\colon \{0,1\}^n \longrightarrow \{0,1\},$$

where $\left|f^{-1}(1)\right| = X_0 \cdot 2^n$. Let us use S to represent the set $f^{-1}(1)$. Clearly, the complement of S, i.e., $\overline{S}$, is identical to $f^{-1}(0)$.

Consider a strong adaptive adversary that corrupts at most t processors and aims to deviate the output towards 1. It will first see the messages from all the processors, i.e., $x \xleftarrow{\$} \{0,1\}^n$. If $x \in S$, i.e., $f(x) = 1$, the adversary shall not alter any processor's message. However, if $x \in \overline{S}$, i.e., $f(x) = 0$, the adversary desires to alter x to be x', by changing at most t coordinates of x, such that $f(x') = 0$. Clearly, by the definition of t-vertex boundary, this is possible if and only if $x \in \partial^t S$. Hence, this adversary can alter the probability of the output being 1 to be at most

$$(|S| + \left|\partial^t S\right|)/2^n,$$

which causes a statistical distance change of $\left|\partial^t S\right|/2^n$.

Similarly, a strong adaptive adversary that aims to deviate the output towards 0, can alter the distribution of the output by at most $\left|\partial^t \overline{S}\right|/2^n$. Therefore, the susceptibility of such a protocol in the presence of strong adaptive adversary that corrupts at most t processors is

$$\max\left(\left|\partial^t S\right|/2^n\,,\,\left|\partial^t \overline{S}\right|/2^n\right).$$

Therefore, we reduce the problem of finding the optimal protocol to the objective of, given an integer $0 \le s \le 2^n$, find

$$\operatorname*{argmin}_{S \in \binom{\{0,1\}^n}{s}} \max\left(\left|\partial^t S\right|\,,\,\left|\partial^t \overline{S}\right|\right).$$

We note that Harper's theorem extends naturally to this setting. That is, the optimal choice of S is still the prefix the simplicial ordering on the boolean hypercube. This observation is stated as the following theorem.

▶ **Theorem 4.** *For all integers $n > 0$ and $0 \le s \le 2^n$, for any $S \in \binom{\{0,1\}^n}{s}$ we have,*

$$\left|\partial^t L^n(s)\right| \le \left|\partial^t S\right| \quad \text{and} \quad \left|\partial^t \overline{L^n(s)}\right| \le \left|\partial \overline{S}\right|.$$

That is, the subgraph induced by the prefix of the simplicial ordering minimizes simultaneously the t-vertex boundary of both itself and its complement.

The proof of this theorem uses similar ideas as the original proof of Harper's theorem. For completeness, we provide a proof in Appendix A.

Implications. Our observation implies that the optimal protocol with expected output X_0 will define function f as

$$f(x) = 1 \text{ if and only if } x \in L^n(s),$$

where $s = X_0 \cdot 2^n$. When $X_0 = 1/2$ and n is odd, this is exactly the majority protocol. Hence, majority protocol is exactly the optimal protocol against strong adaptive adversaries among all protocols with unbiased output.

Asymmetry of deviation. We note that, when $X_0 < 1/2$, the optimal protocol is asymmetric in the sense that the amount of deviation towards 1 the adversary can cause are more than that towards 0. For example, let $X_0 = 1 - \mathrm{Vol}(n,k)/2^n$ for some integer $k > n/2$. Then our observation implies that the optimal protocol is the threshold protocol, i.e., the outcome is 1 if $> k$ processors' message are 1. Note that, in the threshold protocol, the amount of deviation towards 1 that a strong adaptive adversary who corrupts at most t processors, can cause is $\sum_{i=0}^{t-1} \binom{n}{k-i}/2^n$, while the amount of deviation towards 0 is $\sum_{i=1}^{t} \binom{n}{k+i}/2^n$. Similarly, when $X_0 > 1/2$, the protocol is more susceptible when the adversary aims to deviate toward 0. Our observation implies that such asymmetry is inherent and no protocols can be more secure by exploiting a trade-off between the deviation towards 0 and 1.

▶ Remark 1. We remark that the proof of Harper's theorem does not trivially extend to larger alphabet because Lemma 1 in Appendix A does not hold in general for larger alphabet. There are some works on extending Harper's theorem into integer lattice $[\ell]^n$ (See, for example, [BE17]). However, we note that the graph structure they studied is different from that of interests in this problem.

4 Optimal Coin-Tossing Protocols for Adaptive Adversary

In this section, we study adaptive adversary. In particular, we focus on single-turn n-round coin-tossing protocols among n-processor. That is, each round consists of one processor sending one message and every processor speaks only once. We consider the susceptibility of such protocols in the presence of adaptive adversary that corrupts at most one processor.

We study this problem through the lens of geometric approach introduced recently by Khorasgani, Maji, and Mukherjee [KMM19]. Let π be an (n, X_0)-coin-tossing protocol. Let $\mathcal{A}$ be an adaptive attack strategy that corrupts at most one processor. We define $\mathsf{Score}(\pi, \mathcal{A})$ as the deviation that $\mathcal{A}$ causes to the output of protocol π. We are interested in

$$C_n(X_0) := \inf_{\pi} \sup_{\mathcal{A}} \mathsf{Score}(\pi, \mathcal{A}).^{5}$$

Intuitively, for all $X_0 \in [0,1]$, $C_n(X_0)$ represents the least susceptibility against the most devastating attack among all protocols with n processors and expected outcome X_0. In particular,

$$C_1(x) = \begin{cases} 0 & x = 0, 1 \\ 1 - x & x \in (0, 1/2] \\ x & x \in (1/2, 1) \end{cases}$$

For the rest of this section, in Section 4.1, we shall introduce a geometric transformation approach such that one can iteratively obtain curve C_{n+1} by applying it on the previous

[5] There are additionally subtleties in this definition that are addressed in Remark 2.

curve C_n. In Section 4.2, we prove a lower bound of C_n. In Section 4.3, we compare our lower bound with the majority protocol.

4.1 Geometric transformation

Suppose we have curve C_n, we are interested in the next curve C_{n+1}. Let us use Figure 2 as an intuitive example to understand how is $C_{n+1}(x)$ related to curve C_n.

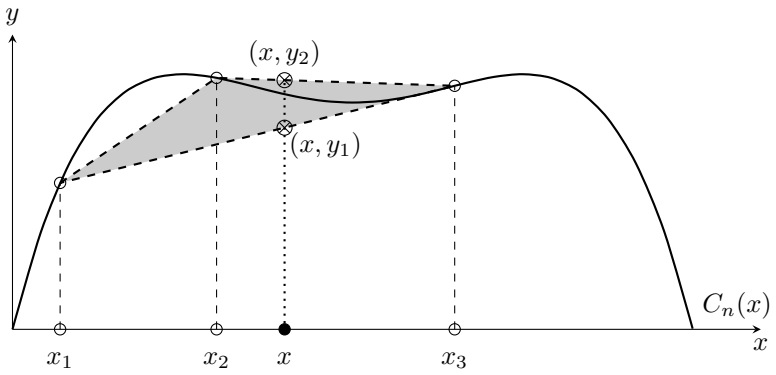

Figure 2 An intuitive example of the geometric transformation

Let π be an $(n+1, x)$-coin-tossing protocol. Suppose there are three possible messages that the first processor might send, namely m_1, m_2, and m_3. Conditioned on the first message being m_1, m_2, and m_3, the expected output is x_1, x_2, and x_3, respectively. The probability of the first message being m_1, m_2, and m_3, are p_1, p_2, and p_3, respectively. Note that after the first processor sends message m_i, the remaining protocol π_i becomes a (n, x_i)-coin-tossing protocol.

An adaptive adversary that corrupts at most one processor have four choices for the first processor. Either it can carry out the attack now by fixing the first processor's message to be m_i, for $i \in \{1, 2, 3\}$, or it can defer the attack to subprotocols π_1, π_2, and π_3. If it fixes the first processor's message to be m_i, this will result in a deviation of $|x_i - x|$. On the other hand, if it defers the attack to each subprotocol, by the definition of curve C_n, it can ensure a deviation of at least $C_n(x_i)$ in subprotocol π_i. Overall, it ensures a deviation of

$$p_1 \cdot C_n(x_1) + p_2 \cdot C_n(x_2) + p_3 \cdot C_n(x_3).^{6}$$

Note that it must hold that $x = p_1x_1 + p_2x_2 + p_3x_3$. Therefore, $p_1 \cdot C_n(x_1) + p_2 \cdot C_n(x_2) + p_3 \cdot C_n(x_3)$ must lie between y_1 and y_2 in Figure 2.

The most devastating attack will do the attack based on which strategy results in the largest deviation, which is

$$\max\left(|x - x_1|, |x - x_2|, |x - x_3|, p_1 \cdot C_n(x_1) + p_2 \cdot C_n(x_2) + p_3 \cdot C_n(x_3)\right).$$

The protocol designer shall, however, pick $x_1, \ldots, x_\ell$ and $p_1, \ldots, p_\ell$ accordingly to minimize

[6] There are some subtleties here that are addressed in Remark 2.

the above quantity. Therefore,

$$C_{n+1}(x) = \inf_{\substack{x_1,\ldots,x_\ell\in[0,1]\\ p_1,\ldots,p_\ell\in[0,1]\\ p_1+\cdots+p_\ell=1\\ p_1x_1+\cdots+p_\ell x_\ell=x}} \max\left(|x-x_1|,\ldots,|x-x_\ell|,\sum_{i=1}^{\ell} p_i\cdot C_n(x_i)\right).$$

For convenience, let us define geometric transformation T, which takes any curve C on $[0,1]$ as input, and outputs a curve $T(C)$ defined as

$$T(C)(x) := \inf_{\substack{x_1,\ldots,x_\ell\in[0,1]\\ p_1,\ldots,p_\ell\in[0,1]\\ p_1+\cdots+p_\ell=1\\ p_1x_1+\cdots+p_\ell x_\ell=x}} \max\left(|x-x_1|,\ldots,|x-x_\ell|,\sum_{i=1}^{\ell} p_i\cdot C(x_i)\right).$$

Hence, we now have C_{n+1} is defined by $T(C_n)$.

▶ Remark 2. We remark on some subtleties in the geometric transformation definition of C_n. When the attacker defers the attacks to sub-protocols. It is not guaranteed that the optimal attack in each sub-protocols will deviate towards the same direction. For instance, if in sub-protocol π_1, the optimal attack can deviate toward 0 by 1%, while in sub-protocol π_2, the optimal attack can deviate toward 1 by 1% and the probability of the first message being m_1 and m_2 are both 1/2. Then the overall deviation should be 0. However, in the geometric transformation definition, the score is in fact 1%. Therefore, $C_n(x)$ does not represent the deviation of the most devastating attack on the optimal protocol. However, this shall not be an issue. The deviation of the most devastating attack on the optimal protocol is still lower bounded by $\frac{1}{2}\cdot C_n(x)$. This is because $C_n(x)$ can be bipartition into two attacks, one deviates toward 0 and the other towards 1; And the summation of the deviations of these two attacks shall equal to $C_n(x)$. Hence, at least one of the attack ensures a deviation of $\geq \frac{1}{2}\cdot C_n(x)$.

4.2 Lower bounding C_n

Given our observation of the geometric transformation, it still remains elusive to obtain a close form representation of curve C_n. In this section, we shall show how one can use the idea of the geometric transformation to obtain a lower bound.

For any two curves A and B on $[0,1]$, we say $A \preccurlyeq B$ if we have that for all $x\in[0,1]$, $A(x)\leq B(x)$. Intuitively, it means that curve A is strictly below curve B. We have the following claim.

▶ **Claim 1.** *If $A \preccurlyeq B$, then $T(A)\preccurlyeq T(B)$.*

Proof of Claim 1 . Trivially, for all x, $x_1,\ldots,x_\ell$, and $p_1,\ldots,p_\ell$, we have

$$\max\left(|x-x_1|,\ldots,|x-x_\ell|,\sum_{i=1}^{\ell} p_i\cdot A(x_i)\right)\leq \max\left(|x-x_1|,\ldots,|x-x_\ell|,\sum_{i=1}^{\ell} p_i\cdot B(x_i)\right).$$

Therefore, by definition, for all x, $T(A)(x)\leq T(B)(x)$, or equivalently $T(A)\preccurlyeq T(B)$. ◀

Next, we have the following claim.

▶ **Claim 2.** *$Qx(1-x) \preccurlyeq T\Big(Px(1-x)\Big)$ if we have $P - Q - PQ^2/4 \geq 0$.*

Proof of Claim 2 . Let us use Figure 3 for intuition. In Figure 3, $D(x)$ is defined as $P{\cdot}x(1{-}x)$ for some constant $P > 0$, the choice of x_1 and x_2 satisfies that

$$|x_1 - x| = |x_2 - x| = \frac{1}{2} \cdot (D(x_1) + D(x_2)) .$$

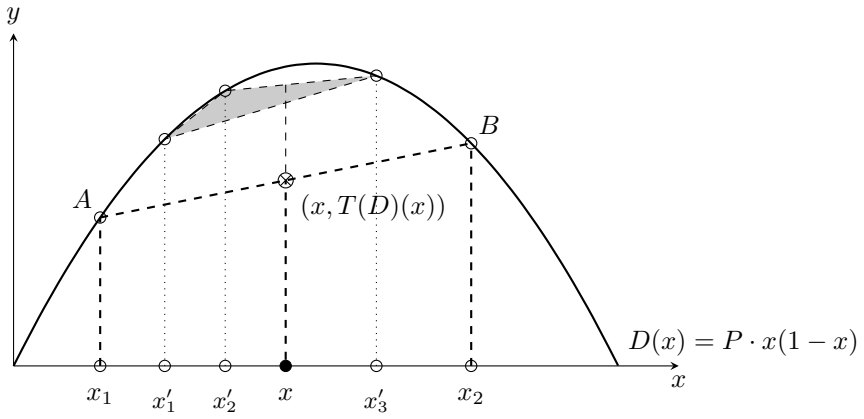

Figure 3 The geometric transformation of curve $D(x) = P \cdot x(1-x)$ for constant $P > 0$. Intuitively, if x'_1, x'_2, and x'_3 are between x_1 and x_2, the shaded region is always above line segment AB by the convexness of D.

Suppose x_1 and x_2 exist, let us show that

$$|x_1 - x| \leq \inf_{\substack{x'_1,\ldots,x'_\ell \in [0,1] \\ p_1,\ldots,p_\ell \in [0,1] \\ p_1+\cdots+p_\ell = 1 \\ p_1 x'_1 + \cdots + p_\ell x'_\ell = x}} \max\left(|x - x'_1|, \ldots, |x - x'_\ell|, \sum_{i=1}^{\ell} p_i \cdot D(x'_i) \right)$$

Firstly, if there exists an x'_i such that $|x - x'_i| \geq |x_1 - x|$, then the statement trivially holds. Next, if for all i, $|x - x'_i| \leq |x_1 - x|$, then by the convexness of curve D,

$$\frac{1}{2} \cdot (D(x_1) + D(x_2)) \leq \sum_{i=1}^{\ell} p_i \cdot D(x'_i).$$

And hence the statement again holds.

Now, let us see why x_1 and x_2 always exist. Let $\delta = |x_1 - x| > 0$. δ shall satisfy that

$$\delta = \frac{1}{2} \cdot (P \cdot (x-\delta)(1-x+\delta) + P \cdot (x+\delta)(1-x-\delta)) .$$

Solving this, we have

$$\delta = \frac{-1 + \sqrt{1 + 4P^2 x(1-x)}}{2P}.$$

Therefore, δ always exists and so are x_1 and x_2. By definition, $\delta = T(D)(x)$.[7] Hence, we require $\delta \geq Q \cdot x(1-x)$ for all x. This implies

$$PQ^2 x(1-x) + Q - P \leq 0,$$

[7] It might be the case that the solution $x_1, x_2 \notin [0,1]$. Then, δ is not exactly the definition of $T(D)(x)$. However, note that δ is still a lower bound of $T(D)(x)$. And that is all we care about.

which holds if we have

$$P - Q - PQ^2/4 \geq 0. \quad \blacktriangleleft$$

Define $\Gamma_n := \sqrt{\frac{2}{n}}$. We note that the constraint of Claim 2 is always satisfied if we set $P = \Gamma_n$ and $Q = \Gamma_{n+1}$. Now, define curve D_n as

$$D_n(x) := \Gamma_n \cdot x(1-x).$$

Initially, one can easily verify that $D_1 \preccurlyeq C_1$. Moreover,

$$D_n \preccurlyeq C_n \xRightarrow{\text{Claim 1}} T(D_n) \preccurlyeq T(C_n) \xRightarrow{\text{Claim 2}} D_{n+1} \preccurlyeq C_{n+1}.$$

Hence, by induction, one can trivially show that curve C_n is always lower bounded by curve D_n.

4.3 Comparison to Majority protocol

We note that $D_n(x) = \sqrt{\frac{2}{n}} \cdot x(1-x)$ is only a lower bound of the curve C_n. We do not obtain an upper bound of C_n. In this section, we provide additional perspectives by comparing this lower bound with majority protocol.

For a n-processor majority protocol, an adaptive adversary that corrupts at most one processor can deviate the output of the protocol towards 1 by fixing the first processor's message to be 1. Asymptotically, this results in a deviation of $\binom{n}{n/2}/2^n$. By Stirling approximation,

$$\binom{n}{n/2} \cdot 2^{-n} \sim \sqrt{\frac{2}{\pi}} \cdot \frac{1}{\sqrt{n}}.$$

On the other hand, our results show that for any n-processor protocol with expected output $1/2$, there at least exists an attack that deviates the protocol by $\frac{1}{2} \cdot C_n(1/2)$, which is lower bounded by

$$\frac{1}{2} \cdot D_n(1/2) = \frac{\sqrt{2}}{8} \cdot \frac{1}{\sqrt{n}}.$$

Therefore, our results show that, asymptotically, majority protocol is the optimal protocol modulo a constant factor.

References

AL93 Miklós Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, 13(2):129–145, 1993. 4

AN90 Noga Alon and Moni Naor. Coin-flipping games immune against linear-sized coalitions (extended abstract). In *31st Annual Symposium on Foundations of Computer Science*, pages 46–54, St. Louis, MO, USA, October 22–24, 1990. IEEE Computer Society Press. doi:10.1109/FSCS.1990.89523. 4

BE17 Ben Barber and Joshua Erde. Isoperimetry in integer lattices. *arXiv preprint arXiv:1707.04411*, 2017. 9

BL89 Michael Ben-Or and Nathan Linial. Collective coin flipping. *Advances in Computing Research*, 5:91–115, 1989. 2, 4

Blu82 Manuel Blum. Coin flipping by telephone. *Proc. of COMPCON, IEEE, 1982*, 1982. 4

BN93 Ravi B. Boppana and Babu O. Narayanan. The biased coin problem. In *25th Annual ACM Symposium on Theory of Computing*, pages 252–257, San Diego, CA, USA, May 16–18, 1993. ACM Press. doi:10.1145/167088.167164. 4

Cle86 Richard Cleve. Limits on the security of coin flips when half the processors are faulty (extended abstract). In *18th Annual ACM Symposium on Theory of Computing*, pages 364–369, Berkeley, CA, USA, May 28–30, 1986. ACM Press. doi:10.1145/12130.12168. 4

CZ16 Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. In Daniel Wichs and Yishay Mansour, editors, *48th Annual ACM Symposium on Theory of Computing*, pages 670–683, Cambridge, MA, USA, June 18–21, 2016. ACM Press. doi:10.1145/2897518.2897528. 4

Fei99 Uriel Feige. Noncryptographic selection protocols. In *40th Annual Symposium on Foundations of Computer Science*, pages 142–153, New York, NY, USA, October 17–19, 1999. IEEE Computer Society Press. doi:10.1109/SFFCS.1999.814586. 4

GKP15 Shafi Goldwasser, Yael Tauman Kalai, and Sunoo Park. Adaptively secure coin-flipping, revisited. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *ICALP 2015: 42nd International Colloquium on Automata, Languages and Programming, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 663–674, Kyoto, Japan, July 6–10, 2015. Springer, Heidelberg, Germany. doi:10.1007/978-3-662-47666-6_53. 4, 7

Har66 Lawrence H Harper. Optimal numberings and isoperimetric problems on graphs. *Journal of Combinatorial Theory*, 1(3):385–393, 1966. 7

KK15 Yael Tauman Kalai and Ilan Komargodski. Compressing communication in distributed protocols. In Yoram Moses, editor, *Distributed Computing - 29th International Symposium, DISC 2015, Tokyo, Japan, October 7-9, 2015, Proceedings*, volume 9363 of *Lecture Notes in Computer Science*, pages 467–479. Springer, 2015. 4

KKL88 Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on Boolean functions (extended abstract). In *29th Annual Symposium on Foundations of Computer Science*, pages 68–80, White Plains, NY, USA, October 24–26, 1988. IEEE Computer Society Press. doi:10.1109/SFCS.1988.21923. 4

KKR18 Yael Tauman Kalai, Ilan Komargodski, and Ran Raz. A lower bound for adaptively-secure collective coin-flipping protocols. In Ulrich Schmid and Josef Widder, editors, *32nd International Symposium on Distributed Computing, DISC 2018, New Orleans, LA, USA, October 15-19, 2018*, volume 121 of *LIPIcs*, pages 34:1–34:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. 4

KMM19 Hamidreza Amini Khorasgani, Hemanta K. Maji, and Tamalika Mukherjee. Estimating gaps in martingales and applications to coin-tossing: Constructions and hardness.

In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019: 17th Theory of Cryptography Conference, Part II*, volume 11892 of *Lecture Notes in Computer Science*, pages 333–355, Nuremberg, Germany, December 1–5, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-36033-7_13. 2, 4, 5, 9

LLS89 David Lichtenstein, Nathan Linial, and Michael E. Saks. Some extremal problems arising form discrete control processes. *Combinatorica*, 9(3):269–287, 1989. 4

RSZ99 Alexander Russell, Michael E. Saks, and David Zuckerman. Lower bounds for leader election and collective coin-flipping in the perfect information model. In *31st Annual ACM Symposium on Theory of Computing*, pages 339–347, Atlanta, GA, USA, May 1–4, 1999. ACM Press. doi:10.1145/301250.301337. 4

Sak89 Michael E. Saks. A robust noncryptographic protocol for collective coin flipping. *SIAM J. Discrete Math.*, 2(2):240–244, 1989. 4

A Proof of Theorem 4

Similar to the original proof of Harper's theorem, we prove Theorem 4 by induction on the dimension n and distance t. The proof of the inductive step relies on the technique called *compression*. On a high level, starting from an arbitrary set $S \subseteq \{0,1\}^n$, the compression argument shall use a sequence of steps to transform S into $L^n(|S|)$, with the additional guarantee that each step is monotone decreasing in terms of the t-vertex boundary of itself and its complement.

If a set $S \subseteq \{0,1\}^n$ consists of the first $|S|$ elements of the simplicial ordering, we call S a *prefix*. Similarly, if S consists of the last $|S|$ elements, then we call it a *suffix*. The inductive proof relies on the following lemma.

▶ **Lemma 1.** *If S is a prefix, then $S \cup \partial S$ is also a prefix. Similarly, if S is a suffix, then $S \cup \partial S$ is also a suffix.*

Next, we shall first prove Lemma 1 and then prove Theorem 4 using Lemma 1.

Proof of Lemma 1 . Let $S \subseteq \{0,1\}^n$ be any prefix. Let $a, b \in \{0,1\}^n$ be any two elements such that $a < b$ in simplicial ordering. We will show that if $b \in S \cup \partial^t S$, then $a \in S \cup \partial^t S$ must also hold. This is sufficient to imply the claim regarding prefix in Lemma 1.

If $a = 0^n$, the statement is trivial. Hence, for the rest of the proof, we consider $a \neq 0^n$. Obviously, we also have $b \neq 0^n$.

Let a' (resp., b') be the *smallest element* whose distance to a (resp., b) is at most t.[8] The crucial observation is that, if $a < b$, we must have $a' \leq b'$. Therefore, the facts that S is a prefix and $b \in S \cup \partial^t S$ together imply that $b' \in S$, which further imply $a' \in S$ and finally imply that $a \in S \cup \partial^t S$.

The proof for the suffix part is essentially the same. One can define a'' and b'' to be the largest elements whose distance is at most t from a and b respectively. Then observe that $a > b$ implies $a'' \geq b''$. The rest of the proof shall follow naturally. ◀

We note that the proof of Harper's theorem shall only rely on the prefix part of Lemma 1. However, for our purpose, we also require the suffix part. As mentioned above, we prove our observation by induction on dimension n and distance t. The base case, (1) $n = 1$ and arbitrary t, or (2) arbitrary n and $t = 0$, can be verified trivially. Hence, we only need to prove the inductive step. In the following proof, we assume that the statement is correct for $(n-1, t-1)$ and $(n-1, t)$. This inductive step shall show that the statement is also correct for (n, t). This is sufficient to imply that the statement is correct for all dimension n and distance t.

Proof of the inductive step. Now, suppose the statement is correct for $(n-1, t-1)$ and $(n-1, t)$, we shall prove its correctness for (n, t). For each $i \in [n]$, let

$$H_0^i := \{x \in \{0,1\}^n \mid x_i = 0\} \quad \text{and} \quad H_1^i := \{x \in \{0,1\}^n \mid x_i = 1\}$$

Note that H_0^i and H_1^i are two hypercubes in $n-1$ dimension. Define a compression operator $\mathcal{C}_i$ as follows.

1. Given a set $S \subseteq \{0,1\}^n$, bipartition S as $S_0^i = S \cap H_0^i$ and $S_1^i = S \cap H_1^i$.

[8] If $\mathsf{wt}(a) \leq t$, then $a' = 0^n$. Otherwise, a' is exactly a with the first t 1's in a replaced by 0.

2. Let $T_0^i \subseteq H_0^i$ be of the same size as S_0^i such that elements in T_0^i without the i^{th} coordinates form a prefix in the $n-1$ dimension. Similarly, let $T_1^i \subseteq H_1^i$ be of the same size as S_1^i such that elements in T_1^i without the i^{th} coordinates form a prefix in the $n-1$ dimension.
3. Define
$$\mathcal{C}_i(S) := T_0^i \cup T_1^i.$$

Intuitively, the compression operator first divides the n dimension hypercube into two $n-1$ dimension hypercubes based on the i^{th} coordinate, i.e., H_0^i and H_1^i. Then, it compresses the partitions of S to be the prefix of the simplicial ordering on each sub-hypercube individually.

We shall prove that, for all $i \in [n]$, compression operator $\mathcal{C}_i$ reduces both the t-vertex boundary of itself and its complement. That is the following claim.

▶ **Claim 3.** *For all $i \in [n]$,*

$$\left|\partial^t \mathcal{C}_i(S)\right| \leq \left|\partial^t S\right| \quad \text{and} \quad \left|\partial^t \overline{\mathcal{C}_i(S)}\right| \leq \left|\partial^t \overline{S}\right|.$$

Proof. Let us first show $|\partial^t \mathcal{C}_i(S)| \leq |\partial^t S|$. In fact, we show the stronger statement that $\mathcal{C}_i(S)$ has smaller t-vertex boundary set than S in both sub-hypercubes H_0^i and H_1^i, i.e.,

$$\left|\partial^t \mathcal{C}_i(S) \cap H_0^i\right| \leq \left|\partial^t S \cap H_0^i\right| \quad \text{and} \quad \left|\partial^t \mathcal{C}_i(S) \cap H_1^i\right| \leq \left|\partial^t S \cap H_1^i\right|.$$

Let us zoom into H_0^i. One can trivially verify that

$$\left|\partial^t \mathcal{C}_i(S) \cap H_0^i\right| \leq \left|\partial^t S \cap H_0^i\right| \iff \left|\left(\mathcal{C}_i(S) \cup \partial^t \mathcal{C}_i(S)\right) \cap H_0^i\right| \leq \left|\left(S \cup \partial^t S\right) \cap H_0^i\right|.$$

For any set A, let us call $\left(A \cup \partial^t A\right) \cap H_0^i$ the *t-set* of A in H_0^i. Therefore, it suffices to show that the t-set of $\mathcal{C}_i(S)$ in H_0^i is smaller than t-set of S in H_0^i.

Note that the t-set of $\mathcal{C}_i(S) = T_0^i \cup T_1^i$ in H_0^i is the union of the t-set of T_0^i and T_1^i. Similarly, the t-set of $S = S_0^i \cup S_1^i$ in H_0^i is the union of the t-set of S_0^i and S_1^i.

Firstly, by inductive hypothesis, the t-set of T_0^i is smaller than the t-set of S_0^i.

Secondly, the t-set of T_1^i (in H_0^i) is exactly the $(t-1)$-set of $\widetilde{T_1^i}$, where $\widetilde{T_1^i}$ is the set of vertices obtained by flipping the i^{th} coordinate the vertices in T_1^i. Similarly, the t-set of S_1^i is exactly the $(t-1)$-set of $\widetilde{S_1^i}$, where $\widetilde{S_1^i}$ is the set of vertices obtained by flipping the i^{th} coordinate the vertices in S_1^i. By inductive hypothesis, the $(t-1)$-set of $\widetilde{T_1^i}$ is smaller than the the $(t-1)$-set of $\widetilde{S_1^i}$. Consequently, the t-set of T_1^i is smaller than the t-set of S_1^i.

Finally, the t-set of $\mathcal{C}_i(S)$ in H_0^i is the union of the t-set of T_0^i and T_1^i. Here, we use Lemma 1 to see that both these two sets are the prefix of the simplicial ordering and hence one must be included in the other. Therefore, the t-set of $\mathcal{C}_i(S)$ in H_0^i is either the t-set of T_0^i or the t-set of T_1^i, and their sizes are bounded by the size of the t-set of S_0^i and S_1^i, respectively. Therefore, $\mathcal{C}(S)$ must have smaller t-set in H_0^i than S.

The proof of H_1^i is essentially identical.

The proof that $\mathcal{C}_i$ is monotone in terms of the t-vertex boundary of the complement is essentially the same. On a high level, the proof consists of the following steps.

1. The complement of $\mathcal{C}_i(S)$ is the union of (1) the complement of T_0^i in H_0^i and (2) the complement of T_1^i in H_1^i, i.e.,
$$\overline{\mathcal{C}_i(S)} = \left(H_0^i \backslash T_0^i\right) \cup \left(H_1^i \backslash T_1^i\right).$$

2. $H_0^i \backslash T_0^i$ and $H_1^i \backslash T_1^i$ are *suffix* of simplicial ordering in $n-1$ dimension.
3. Similar to the proof above, we argue that the $\overline{S}$ has larger t-boundary than $\overline{\mathcal{C}_i(S)}$ in both H_0^i and H_1^i. This part of the proof relies on the suffix part of the claim in Lemma 1. ◀

Now, start from the original set S, we repetitively apply the compression operators

$$S \xrightarrow{\mathcal{C}_1} S_1 \xrightarrow{\mathcal{C}_2} S_2 \xrightarrow{\mathcal{C}_3} \cdots \xrightarrow{\mathcal{C}_n} S_n \xrightarrow{\mathcal{C}_1} S_{n+1} \xrightarrow{\mathcal{C}_2} \cdots$$

Clearly, after a finite number of steps,[9] this process will stabilize at a set T such that

1. The t-vertex boundary of both itself and its completement decreases compare to S. That is, $|\partial^t T| \leq |\partial^t S|$ and $\left|\partial^t \overline{T}\right| \leq \left|\partial^t \overline{S}\right|$.
2. T is a fix point for all compression operators. That is, for all $i \in [n]$, $\mathcal{C}_i(T) = T$.

Fix points for all compression operators. We shall show that subsets that are the fix points for all compression operators are almost always the simplicial prefix. Suppose T is a fix point for all compression operators, and T is *not* a simplicial prefix. Then, there exists a $x \in T$ and $y \notin T$ such that $y < x$. If there exists an i, such that $x_i = y_i$, then $y < x$ implies $y_{-i} < x_{-i}$ in the $n-1$ dimension, which implies that T is not a fix point for operator $\mathcal{C}_i$. Therefore, for all i, $x_i \neq y_i$. This implies that such a pair of x, y must be unique. Also, there does not exist a z such that $y < z < x$. Otherwise, either $z \in T$ or $z \notin T$ will result in contradictions.

Clearly, the *only* possible scenario that there exists two *consecutive* elements who disagree on every coordinate is when $n = 2m+1$.[10] And

$$y = \underbrace{1\cdots1}_{m}\underbrace{0\cdots0}_{m+1} \quad \text{and} \quad x = \underbrace{0\cdots0}_{m}\underbrace{1\cdots1}_{m+1},$$

where $T = \{x\} \cup \{z \,|\, \mathsf{wt}(z) \leq m \text{ and } z \neq y\}$. Trivially, we can verify that, for this particular case,

$$\left|\partial^t T\right| > \left|\partial^t L^n(|T|)\right|$$

and

$$\left|\partial^t \overline{T}\right| > \left|\partial^t \overline{L^n(|T|)}\right|.$$

This observation completes the proof of the inductive step. ◀

[9] The compression operator always replaces elements with other elements that are smaller in the simplicial ordering. Therefore, this process can only go on for a finite number of steps.

[10] Another possibility is when $n = 2$, $x = 01$ and $y = 10$. However, in general, when n is even, no such x and y exist.