Oded (September 1, 2023): On parallel repetition of interactive proof systems

This memo provides an overview of an analysis of the soundness error of parallel repetitions of general (i.e., two-sided error) interactive proof systems.[1] The bottom-line is that the analysis of the general case can be reduced to the analysis of the one-sided error case, by using a general result of [3]. Since a full analysis of the one-sided error case was published in [1, Apdx. C.1], combining these two published sources yields the desired result.

The main text

It is taken for granted that the error probability of interactive proof systems can be reduced by *parallel* repetitions, but this belief requires a precise formulation as well as a proof. Starting with the formulation, recall that a general formulation of interactive proof systems refers both to the completeness error (i.e., the probability that a YES-instance is rejected) and the soundness error (i.e., the probability that a NO-instance is accepted). In one-sided error systems (aka systems with perfect completeness), the completeness error is zero. In general, the error probability of a system is the maximum of the completeness and soundness errors.

The foregoing belief states that, for any interactive proof system with error probability at most $\beta < 1/2$, *executing m copies of the system in parallel and having the verifier decide according to a majority rule* (of the decisions made in the m copies) *yields an interactive proof system with error probability at most* $\exp(-\Omega((0.5-\beta)^2 \cdot m))$. We stress that the verifier acts in each of the m copies according to the original prescribed (single copy) strategy, which means that its actions is each copy are oblivious of the execution of the other copies. In contrast, the cheating prover may behave arbitrarily; in particular, it may base its action in each copy on its current view of all m parallel executions).

Nevertheless, the foregoing belief is correct, but the proof is more complex than one may expect. The case of one-sided error probability (and a verifier deciding according to the conjunction rule) was treated in [1, Apdx. C.1]. In that case, if the original proof systems has soundness error at most s, then the resulting proof system (which uses m copies) has soundness error at most s^m. The proof of this claim is quite straightforward (alas tedious), and is reproduced in the appendix.

Intuitively, the proof (in the one-sided error case) is based on the fact that maximizing the acceptance probability under the *conjunction rule* calls for maximizing the acceptance probability of each copy. Unfortunately, it is unclear whether this intuitive holds under the majority rule, which is employed in the general (i.e., two-sided error) case. We conjecture that the maximum probability that a majority of the m copies accepts is obtained when the prover maximizes the acceptance probability of each copy (independently of the others), but the analysis outlined next does not rely on this conjecture.

Recall that we refer to a parallel execution of m copies of an interactive proof system with two-sided error probability, and, focusing on NO-instances, we wish to upper-bound the probability that the majority of the copies accept (when the prover tries to maximize this probability while possibly acting in each copy in a manner that depends on its view of all copies). We obtain the desired upper bound by reduction to the one-sided error case (and the conjunction rule).

For illustration, suppose that the error bound β is a constant that smaller than $1/4$. In this case, for any NO-instance, the probability that a majority of the copies accept is upper-bounded by

[1] We assume that the reader is familiar with the basic definitions of interactive proof systems.

the probability that there exists an $m/2$-subset I of $[m]$ such that all copies in I accept. This yields an upper bound of $\binom{m}{m/2} \cdot \beta^{m/2}$, which is $\exp(-\Omega(m))$ by the hypothesis that $\beta < 1/4 = (1/2)^2$. (Using an acceptance threshold of 0.66, one can handle $\beta = 1/3$ in an analogous manner.)[2]

To handle arbitrary error bound $\beta < 1/2$, we use a more sophisticated argument. We first observe that the completeness condition follows by using a prover strategy that applies an optimal strategy to each of the copies, independently of the other copies. Focusing on the soundness condition, our starting point is the observation that, for every set I, the probability that all copies in I accept the input (i.e., a NO-instance) is at most $\beta^{|I|}$. Hence, although the m random events that describe the verifier's ruling in the m copies may be related (by the strategy of the prover), we know that the conjunction of each k-subset of these events holds with probability at most β^k. At this point we apply a general result of [3], which states the following.

Theorem 1 (generalized Chernoff Bound): *Let $(\zeta_1, ..., \zeta_m)$ be a joint distribution on $\{0,1\}^m$ and $b_1, ..., b_m \in [0,1]$ be numbers such that, for every subset $I \subseteq [m]$, it holds that*

$$\mathsf{Pr}\left[\bigwedge_{i \in I}(\zeta_i = 1)\right] \leq \prod_{i \in I} b_i.$$

Then, for $b = \sum_{i \in [m]} b_i/m$ and any $c \in (b, 1]$, it holds that

$$\mathsf{Pr}\left[\sum_{i \in [m]} \zeta_i > c \cdot m\right] \leq \exp(-2 \cdot (c-b)^2 \cdot m).$$

Alternative proofs of Theorem 1 are given in [3, Sec. 3] and in [2]. The first proof generalizes the standard proof of the Chernoff Bound, whereas the second proof uses a different strategy (and actually derives the standard Chernoff Bound as a special case). Now, combining Theorem 1 with Lemma 4 (of the appendix), we get

Theorem 2 (parallel repetition of interactive proof systems): *Let A and B be arbitrary interactive machines, and $(A,B)(x)$ denote the binary output of B after interacting with A on common input x. Let V be an interactive verifier of soundness error at most $0.5 - \delta$; that is, for any NO-instance x, it holds that $\max_{P^*}\{\mathsf{Pr}[(P^*, V)(x) = 1]\} \leq 0.5 - \delta(|x|)$. For any $m : \mathbb{N} \to \mathbb{N}$, let $\overline{V}_m$ denote a verifier that, on input an $m(n)$-long sequence of n-bit long inputs, executes the corresponding $m(n)$ parallel copies of V, and outputs the majority verdict of these copies. Then, for every $m(n)$-long sequence of NO-instances $(x_1, ..., x_m) \in (\{0,1\}^n)^{m(n)}$ it holds that*

$$\max_{\widehat{P}}\left\{\mathsf{Pr}\left[(\widehat{P}, \overline{V}_m)(x_1, ..., x_{m(n)}) = 1\right]\right\} \leq \exp(-2 \cdot \delta(n)^2 \cdot m(n)).$$

Indeed, in the special case of $x_1 = x_2 = \cdots = x_{m(n)}$, one may use Lemma 3 instead of Lemma 4.

Proof: Let us spell out the straightforward proof. Fixing an arbitrary prover-strategy $\widehat{P}$ for interacting with $\overline{V}_m$, we let $(\zeta_1, ..., \zeta_m)$ represent the verifier's decisions in the $m = m(n)$ copies (i.e., ζ_i represents V's decision regarding the i^{th} copy). Applying Lemma 4, we infer that the hypothesis of Theorem 1 holds, when setting $b_i = 0.5 - \delta(n)$ (for every $i \in [m]$). Using $c = 0.5$ and invoking Theorem 1, the current theorem follows. ■

[2] The completeness error of the resulting proof system vanishes exponentially with $((1-\beta) - 0.66)^2 \cdot m = \Omega(m)$, whereas its soundness error is upper-bounded by $\binom{m}{0.66 \cdot m} \cdot \beta^{0.66 \cdot m} < (2 \cdot \beta^{0.66})^m < 0.99^m$.

Digest. The intriguing aspect of the proof of Theorem 2 is that it reduces the analysis of the two-sided error case to the analysis of the one-sided error case. The work itself is, of course, done by Theorem 1. It is unfortunate that Theorem 1 is not better known.

References

[1] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness.* Algorithms and Combinatorics 17, Springer 1998, ISBN

[2] Russell Impagliazzo and Valentine Kabanets. Constructive Proofs of Concentration Bounds. *ECCC*, TR10-072, 2010.

[3] Alessandro Panconesi and Aravind Srinivasan. Randomized Distributed Edge Coloring via an Extension of the Chernoff–Hoeffding Bounds *SIAM Journal on Computing*, Vol. 26 (2), pages 350–368, 1997.

Appendix: The one-sided error case

Here we are interested in interactive proof systems with perfect completeness and in reducing their soundness error by parallel repetitions. Hence, we consider the case that the resulting verifier decides by a conjunction rule (rather than by a majority rule as in Theorem 2). The rest of the text is reproduced from [1, Apdx. C.1], with minor revisions.

By k parallel repetitions of an interactive proof system, (P,V), we mean a proof system (P_k,V_k) in which the parties play in parallel k copies of (P,V). That is, V_k (resp., P_k) generates k independently distributed random-pads, $r_1,...,r_k$, for V (resp., $\omega_1,...,\omega_k$ for P), and sets its i^{th} message to $\beta_{1,i},...,\beta_{k,i}$, where $\beta_{j,i}=V(r_j,\alpha_{1,j},...,\alpha_{i-1,j})$ (resp., to $\alpha_{1,i},...,\alpha_{k,i}$, where $\alpha_{j,i}=P(\omega_j,\beta_{1,j},...,\beta_{i,j})$). We stress that V_k accepts if and only if V would have accepted in all k copies. We are interested in the soundness error of V_k, which only depends on V and k (and so P_k and P are omitted from the rest of the discussion). For any pair of interactive machines, A and B, let use denote by (A,B) the output of A after interacting with B, on common input x. The Parallel Repetition Theorem for interactive proofs is captured by the following lemma.

Lemma 3 (folklore): *Let V_1 be an interactive machine, and V_k be an interactive machine obtained from V_1 by playing k versions of V_1 in parallel. Let*

$$\begin{aligned} p_1(x) &\stackrel{\text{def}}{=} \max_{P^*}\{\mathsf{Pr}[(P^*,V_1)(x)=1]\}\,, \text{ and} \\ p_k(x) &\stackrel{\text{def}}{=} \max_{P^*}\{\mathsf{Pr}[(P^*,V_k)(x)=1]\}\,. \end{aligned}$$

Then

$$p_k(x) \;=\; p_1(x)^k\,.$$

Proof: Clearly, $p_k(x)\geq p_1(x)^k$. The point is to prove $p_k(x)\leq p_1(x)^k$. We stress that one may not just assume that the optimal prover strategy against V_k consists of playing optimally but independently in each of the k parallel copies. As we shall see below, this conjecture turns out to be correct in the current setting (but is wrong in related settings such as multi-party interactive proofs and computationally-sound proofs). Thus, a proof is due.

We start with a general description of the execution of an interactive proof system, where our point of view is not of the parties themselves but rather of an external (all knowing) analyzer. Fixing a verifier V we consider its interaction with a generic prover on any fixed common input, denoted x. The verifier's random choices can be thought of as corresponding to the contents of its random-tape, called the random-pad. We assume without loss of generality that V sends the first message and that the prover sends the last one. In each round, V's message is chosen depending on the history of the interaction so far and according to some probability distribution induced by V's local random-tape. The history so far corresponds to a fixed subset of possible random-pads, and the possible messages to be sent correspond to a partition of this subset. Thus, each possible message is sent with probability proportional to its part in this subset. The above description corresponds to general interactive proofs. (In case of Arthur-Merlin games the situation is simpler: V merely tosses a predetermined number of coins and sends the outcome to the prover.) As to the prover's messages, they are chosen arbitrarily (but are of length at most $\text{poly}(|x|)$). The interaction goes on, for at most $\text{poly}(|x|)$ rounds at which point the verifier stops outputting either *accept* or *reject*. The messages exchanged till that point are called a *transcript* of the interaction between

the prover and V. To simplify the exposition, we augment the transcript of the interaction by V's random-pad. This way, V's accept/reject decision is determined by the *augmented transcript* (and the input x).

The interaction between the prover and V on common input x may be viewed as a game in which the prover's objective is to maximize the probability that V accepts, and V's strategy is fixed but mixed (i.e., probabilistic). The possible executions of this game are captured by the following notion of a game tree.

Definition 3.1 (the game tree and its value): *Let V and x be fixed.*

- *The tree T_x: The nodes in T_x correspond to prefixes of transcripts of possible interactions of V with an arbitrary prover.*

  1. *The root represents the empty interaction and is defined to be at level 0.*
  2. *For every $i \geq 0$, the edges going out from each $2i^{\text{th}}$ level node correspond to the messages V may send given the history so far, and the edges going out from each $(2i+1)^{\text{st}}$ level node correspond to the messages a prover may send given the history so far.*
  3. *Leaves correspond to augmented transcripts as defined above, and so their direct ancestors correspond to full transcripts.*

- *The value of T_x: The value of the tree is defined bottom-up as follows.*

  1. *The* value of a leaf *is either 0 or 1 depending on whether V accepts in the augmented transcript represented by it or not.*
  2. *The* value of an internal node at level $2i$ (aka verifier-node) *is defined as the weighted average of the values of its children, where the weights correspond to the probabilities of the various verifier messages.* (This definition holds also for the direct ancestors of leaves, when viewing V's random-pad as an auxiliary, fictitious message sent by V.)
  3. *The* value of an internal node at level $2i-1$ (aka prover-node) *is defined as the maximum of the values of its children. This corresponds to the prover's strategy of trying to maximize V's accepting probability.*

  The value of the tree *is defined as the value of its root.*

We may assume, without loss of generality, that the averages taken in even-leveled nodes are plain averages (rather than weighted ones). This is justified by duplicating odd-level nodes. We stress that this modification is applied to the game-tree (not to the verifier), and results in a tree the correspondence of which to the proof system is less obvious. Notice that we are dealing with a general interactive proof, yet our analysis of the game-tree is a mental experiment (which need not be efficiently implementable).

We consider the game-trees of both the basic proof system and the k-repeated proof system. Fixing an input, we denote the first tree by T_1 and the second by T_k. There is a natural 1-1 mapping of nodes in T_k to sequences of k nodes in T_1. Going from the leaves of T_k to its root, we prove by induction that the value of each node is T_k equals the product of the values of the k nodes to which it is mapped (by the above mapping). Specifically, denoting the values of node v in T_1 by $\text{val}(v)$, and the value of node $\overline{v}$ in T_k by $\overline{\text{val}}(\overline{v})$, we prove the following claim by induction (from the leaves).

Claim 3.2 (the main claim): *For every node, $\overline{v} = (v_1, ..., v_k)$, it holds that $\overline{\mathrm{val}}(\overline{v}) = \prod_{j=1}^{k} \mathrm{val}(v_j)$.*

Proof: The claim is proved by induction from the leaves to the root. The base case (i.e., the values at leaves) follows by the definition of the decision rule of V_k. As for the value of internal nodes, the analysis splits according to the parity of their levels (resp., whether they are prover or verifer nodes).

1. For a prover-node, $\overline{v} = (v_1, ..., v_k)$, denote its children in T_k by $\overline{w}^{\overline{i}} = (w_1^{i_1}, ..., w_k^{i_k})$, where $\overline{i} = (i_1, ..., i_k)$ and w_j^i is the i-th child in T_1 of v_j. Then, by definition of the game trees

$$\begin{aligned} \overline{\mathrm{val}}(\overline{v}) &= \max_{\overline{i}}(\overline{\mathrm{val}}(\overline{w}^{\overline{i}}))\,, \text{ and} & (1)\\ \mathrm{val}(v_j) &= \max_{i}(\mathrm{val}(w_j^i))\,, \text{ for } j = 1, ..., k. & (2) \end{aligned}$$

   By induction, for every $\overline{i} = (i_1, ..., i_k)$,

$$\overline{\mathrm{val}}(\overline{w}^{\overline{i}}) = \prod_{j=1}^{k} \mathrm{val}(w_j^{i_j})) \quad (3)$$

   Combining Equations (1)–(3), and using the "distributivity of maximization and products", we get

$$\begin{aligned} \overline{\mathrm{val}}(\overline{v}) &= \max_{\overline{i}}(\overline{\mathrm{val}}(\overline{w}^{\overline{i}}))\\ &= \max_{\overline{i}} \left(\prod_{j=1}^{k} \mathrm{val}(w_j^{i_j}) \right)\\ &= \prod_{j=1}^{k} \max_{i_j}(\mathrm{val}(w_j^{i_j}))\\ &= \prod_{j=1}^{k} \mathrm{val}(v_j) \end{aligned}$$

   as required.

2. For a verifier-node, $\overline{v} = (v_1, ..., v_k)$, denote its children in T_k by $\overline{w}^{\overline{i}} = (w_1^{i_1}, ..., w_k^{i_k})$, where $\overline{i}$ and the w_j^i's are as above. Then, by definition of the game trees

$$\begin{aligned} \overline{\mathrm{val}}(\overline{v}) &= \mathrm{aver}_{\overline{i}}(\overline{\mathrm{val}}(\overline{w}^{\overline{i}}))\,, \text{ and} & (4)\\ \mathrm{val}(v_j) &= \mathrm{aver}_{i}(\mathrm{val}(w_j^i))\,, \text{ for } j = 1, ..., k. & (5) \end{aligned}$$

   where $\mathrm{aver}_i(x_i)$ denotes the average value of the x_i's which are to be understood from the context. Again, Eq. (3) holds by induction, and so using the "distributivity of summation and products", we get

$$\overline{\mathrm{val}}(\overline{v}) = \mathrm{aver}_{\overline{i}}(\overline{\mathrm{val}}(\overline{w}^{\overline{i}}))$$

$$
\begin{aligned}
&= \mathrm{aver}_{\bar{i}}\left(\prod_{j=1}^{k}\mathrm{val}(w_j^{i_j})\right)\\
&= \prod_{j=1}^{k}\mathrm{aver}_{i_j}(\mathrm{val}(w_j^{i_j}))\\
&= \prod_{j=1}^{k}\mathrm{val}(v_j)
\end{aligned}
$$

as required.

The claim follows. □

Applying Claim 3.2 to the root, the lemma follows. ■

Generalization We comment that the above argument generalizes to the case in which the k copies of V_1 are invoked on possibly different inputs. That is,

Lemma 4 (parallel executions on different inputs): *Let V_1 be an interactive machine, and V_k be an interactive machine obtained from V_1 by playing k versions of V_1 in parallel so that on input $\overline{x} = (x_1, ..., x_k)$ to V_k the i^{th} version of V_1 is invoked on x_i. Let $p_1(x) \stackrel{\text{def}}{=} \max_{P^*}\{\mathsf{Pr}[(P^*, V_1)(x) = 1]\}$, and $p_k(\overline{x}) \stackrel{\text{def}}{=} \max_{P^*}\{\mathsf{Pr}[(P^*, V_k)(\overline{x}) = 1]\}$. Then*

$$p_k(x_1, ..., x_k) = \prod_{i=1}^{k} p_1(x_i)$$