Efficient Rational Proofs with Strong Utility-Gap Guarantees

Jing Chen[1], Samuel McCauley[2], and Shikha Singh[2](✉)

[1] Stony Brook University, Stony Brook, USA
jingchen@cs.stonybrook.edu
[2] Wellesley College, Wellesley, USA
{samuel.mccauley,shikha.singh}@wellesley.edu

Abstract. As modern computing moves towards smaller devices and powerful cloud platforms, more and more computation is being delegated to powerful service providers. Interactive proofs are a widely-used model to design efficient protocols for verifiable computation delegation.

Rational proofs are payment-based interactive proofs. The payments are designed to incentivize the provers to give correct answers. If the provers misreport the answer then they incur a payment loss of at least $1/u$, where u is the *utility gap* of the protocol.

In this work, we tightly characterize the power of rational proofs that are super efficient, that is, require only logarithmic time and communication for verification. We also characterize the power of single-round rational protocols that require only logarithmic space and randomness for verification. Our protocols have strong (that is, polynomial, logarithmic, and even constant) utility gap. Finally, we show when and how rational protocols can be converted to give the completeness and soundness guarantees of classical interactive proofs.

1 Introduction

Most computation today is not done locally by a client, but instead is outsourced to third-party service providers in exchange for money. Trading computation for money brings up two problems—(a) how the client can guarantee correctness of the outsourced computation (without redoing the computation), and (b) how to design the payment scheme. The two problems are closely related: ideally, we want the payment scheme to be such that it incentivizes service providers to perform the computation correctly.

This work has been partially supported by NSF CAREER Award CCF 1553385, CNS 1408695, CCF 1439084, IIS 1247726, IIS 1251137, CCF 1217708, by Sandia National Laboratories, and by the European Union's 7th Framework Programme (FP7/2007-2013) / ERC grant agreement no. 614331. BARC, Basic Algorithms Research Copenhagen, is supported by the VILLUM Foundation grant 16582.

X. Deng (Ed.): SAGT 2018, LNCS 11059, pp. 150–162, 2018.
https://doi.org/10.1007/978-3-319-99660-8_14

Interactive proofs (IP) are the most well-studied and widely-used theoretical framework to verify correctness of outsourced computation [7,10,11,17,18,22,28,33]. In an IP, a weak client (or *verifier*) interacts with powerful service providers (or *provers*) to determine the correctness of their claim. At the end, the verifier probabilistically accepts or rejects the claim.[1] Interactive proofs guarantee that, roughly speaking, the verifier accepts a truthful claim with probability at least 2/3 (*completeness*) and no strategy of the provers can make the verifier accept a false claim with probability more than 1/3 (*soundness*).[2].

Rational proofs are payment-based interactive proofs for computation outsourcing which leverage the incentives of the service providers. In rational proofs, the provers act *rationally* in the game-theoretic sense, that is, they want to maximize their payment. The payment is designed such that when the provers maximize their payment, they also end up giving the correct answer. The model of rational proofs (RIP) was introduced by Azar and Micali in [2]. Since then, many simple and efficient rational protocols have been designed [3,9,13,23,24,26,35].

While rational proofs provide strong theoretical guarantees, there are two main barriers that separate them from what is often desired in practice. First, many rational protocols require a polynomial-time verifier—but a "weak" client is unlikely to be able to spend (say) quadratic time or linear extra space on verification. Second, many of these protocols strongly rely on the rationality of the provers. An honest prover may receive only a fraction of a cent more than a dishonest prover, yet a rational prover is assumed to be incentivized by that small amount. However, service providers may not always be perfectly rational.

The goal of this paper is to give protocols that overcome these barriers.

Utility Gap. The strength of the guarantee provided by rational proofs is captured by the notion of *utility gap*. The high level idea behind utility gap is that provers who are not perfectly rational may not care about small losses in payments and may lazily give the incorrect answer. If a rational protocol has a utility gap of u, then the provers who mislead the verifier to an incorrect answer are guaranteed to lose at least $1/u$. (This is under a normalized budget of 1; if the budget is scaled up to B, such provers can be made to lose at least B/u.) Thus, protocols with small utility gap are sound even against provers with *bounded rationality*; that is, provers who are only sensitive to large losses.

In this paper, we design efficient rational protocols with strong utility gap—that is, polynomial, logarithmic, and even constant utility gap. In Section 5, we show when and how a noticeable utility gap of a rational protocol can be utilized to achieve the strong completeness and soundness guarantees of a classical proof.

Efficient Protocols. In this paper, we focus on designing rational protocols with very small overheads in terms of verification time, space, communication cost and number of rounds. In particular, we design constant-round rational

[1] In classical interactive proofs there is no payment—simply acceptance or rejection.

[2] More formally, given an input x and a language L, if $x \in L$, the verifier accepts with probability at least 2/3 (*completeness*); if $x \notin L$, then no strategy of the provers can make the verifier accept with probability more than 1/3 (*soundness*).

protocols where the verification time and communication cost are logarithmic in the input size n. We also design single-round rational protocols that have only logarithmic overhead on the verifier's use of space and randomness.

1.1 Results and Contributions

In this section, we summarize our results and contributions.

Time-Efficient Rational Proofs. We study the effect of different communication costs and an additional prover on the power of rational proofs with a highly time-efficient verifier. The utility gap of these protocols is polynomial.

- **Constant Communication.** We show that multiple provers do not add any power when the communication complexity of the protocol is restricted to be extremely small—a constant number of bits. That is, we show that the class of languages that admit a multi-prover rational proof with a $O(\log n)$-time verifier and $O(1)$ communication is exactly $\mathsf{UniformTC_0}$, which is the same as the power of single-prover version under the same costs [3,23]. $\mathsf{UniformTC_0}$ is the class of constant depth, polynomial size uniform threshold circuits, that includes problems such as integer division and iterated multiplication [1,25].
- **Logarithmic Communication.** We show that any rational proof with polynomial communication can be simulated by a rational proof with logarithmic communication that uses an additional prover. Using this property, we improve the communication complexity of Azar and Micali's [3] single-prover rational protocol and show that the class of languages that admit a two-prover rational proof with logarithmic communication is exactly the class of languages decidable by a polynomial time machine that can make polynomially many queries in parallel to an NP oracle, denoted $\mathsf{P}_{||}^{\mathsf{NP}}$.[3] This is an important class (e.g., see [8,30,34]) and includes optimization problems such as maximum clique, longest paths, and variants of the traveling salesman problem.

Space-Efficient Rational Proofs. We achieve even better utility gap guarantees when the verifier's use of space and randomness is extremely small—logarithmic, but its running time may be polynomial. In particular, we exactly characterize the class of single-round rational proofs with $\gamma(n)$ utility gap and logarithmic space and randomness as the class of languages decidable by a polynomial-time machine that makes $O(\gamma(n))$ queries to an NP oracle, denoted $\mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}$. Even when $\gamma(n) = O(1)$ this bounded-query class is still sufficiently powerful and contains many of the optimization problems mentioned above.

Rational Proofs with Completeness and Soundness Guarantees. Finally, we closely compare the two proof systems—rational and classical. We construct a condition on the expected payments of rational proofs which, if satisfied, turns them into a classical interactive proof with completeness and

[3] For parallel oracle queries, both notations $\mathsf{P}_{||}^{\mathsf{NP}}$ [34] and $\mathsf{P}^{||\mathsf{NP}}$ [3] are used in literature.

soundness guarantees. We first show how to convert a payment-based protocol for a language L to an accept-reject protocol (without payments) for L such that the expected payment of the former is exactly the probability with which the verifier accepts in the latter. We use this to prove that if the expected payments of all inputs $x \in L$ are noticeably far away from that of all inputs $x \notin L$, the rational protocol can be converted to a classical interactive protocol.

1.2 Additional Related Work

Azar and Micali [3] also characterize the classes $\mathsf{UniformTC_0}$ and $\mathsf{P}_{||}^{\mathsf{NP}}$. Their characterization of $\mathsf{P}_{||}^{\mathsf{NP}}$ requires polynomial communication, which we improve to logarithmic using a second prover. We also note that all protocols in [3] have a polynomial utility gap (under a constant budget).

Rational arguments, super-efficient rational proofs where the prover is restricted to be polynomial time, were introduced by Guo et al. [23]. Rational arguments for all languages in P were given in [24]. Campanelli and Rosario [9] study sequentially composable rational proofs. Zhang and Blanton [35] design protocols to outsource matrix multiplications to a rational cloud.

The model of multi-prover rational interactive proofs was introduced by Chen et al. [13], where they study the power of the model in its full generality (that is, polynomial-time verifier and polynomial communication). In this paper, we restrict our focus on proofs with log-time verifiers and log-size communication.

Different variants of the rational-proof models have also been studied. Chen et al. [14] consider rational proofs where the rational provers are *non-cooperative* [14]. Inasawa and Kenji [27] consider rational proofs where the verifier is also rational and wants to minimize the payment to the provers.

Interestingly, the log-space verifier studied in this paper also happens to be a *streaming algorithm*, that is, the verifier does not need to look again at any input or message bits out of order. Thus, our space-efficient rational proofs are closely related to the work on streaming interactive proofs [11,17,18].

Refereed games is another multi-prover interactive-proof model that leads to game-theoretic characterizations of various complexity classes (e.g. [12,20,21,29,32]). The model of refereed games requires at least one honest prover.

2 Preliminaries

We begin by reviewing the model of rational proofs [2,13].

Let L be a language, x be an input string and $n = |x|$. An *interactive protocol* is a pair $(V, \vec{P})$, where V is the *verifier* and $\vec{P} = (P_1, \ldots, P_{p(n)})$ is the vector of *provers*, and $p(n)$ a polynomial in n. The goal of the verifier is to determine if $x \in L$. In general, the verifier runs in time polynomial in n and uses polynomial space as well. In Sect. 3, the verifier's running time is $O(\log n)$. In Sect. 4, the verifiers may use polynomial time but are restricted to use $O(\log n)$ space and randomness. The provers are computationally unbounded.[4]

The verifier can communicate with each prover privately, but no two provers can communicate with each other. In a *round*, either each prover sends a message to the verifier, or the verifier sends a message to each prover, and these two cases alternate. Without loss of generality, provers send the first round of messages. The first bit of the first round is the *answer bit*, denoted by c, and indicates whether $x \in L$; that is, $x \in L$ iff $c = 1$. We define the *communication* of the protocol to be the maximum number of total bits transmitted (summed over all provers and all rounds) during the protocol.

Let r be the random string used by V. Let $\vec{m}$ be the vector of all messages exchanged. At the end, the verifier computes the total payment to the provers, given by a payment function $R(x, r, \vec{m})$. We restrict the verifier's budget to be constant, that is, $R \in [0, 1]$ for convenience. We may use negative payments to emphasize penalties but they can shifted to be non-negative. The protocol (including the payment function R) is public knowledge.

The verifier outputs the answer bit c at the end of the protocol—thus the verifier always agrees with the provers.

Each prover P_i can choose a *strategy* $s_{ij} : \{0,1\}^* \rightarrow \{0,1\}^*$ for each round j, which maps the transcript he has seen up until the beginning of round j to the message he sends in round j. Note that P_i does not send any message when j is even; in this case s_{ij} can be treated as a constant function. Let $s_i = (s_{i1}, \ldots, s_{ik})$ be the strategy vector of P_i and $s = (s_1, \ldots, s_{p(n)})$ be the strategy profile of the provers. Given any input x, randomness r and strategy profile s, we may write the vector $\vec{m}$ of messages exchanged in the protocol more explicitly as $(V, \vec{P})(x, r, s)$.

The provers are *cooperative* and jointly act to maximize the total expected payment. Thus, before the protocol starts, the provers pre-agree on a strategy profile s that maximizes $u_{(V,\vec{P})}(s; x) = \mathbf{E}_r[R(x, r, (V, \vec{P})(x, r, s))]$. When $(V, \vec{P})$ and x are clear from the context, we write $u(s)$ for $u_{(V,\vec{P})}(s; x)$.

Definition 1 ([13]). *For any language L, an interactive protocol $(V, \vec{P})$ is a rational interactive proof protocol for L if, for any $x \in \{0,1\}^*$ and any strategy profile s of the prover(s) such that $u(s) = \max_{s'} u(s')$, $c = 1$ if and only if $x \in L$.*

Similar to classical proofs, single-prover rational interactive protocols, that is, when $\vec{P} = P$, are denoted by RIP. Multi-prover interactive protocols, where $\vec{P} = (P_1, \ldots, P_{p(n)})$ are denoted by MRIP. In this paper we study both single-prover and multi-prover rational proof protocols.

We use poly(n) as a shorthand for a polynomial n^k, for some constant k.

2.1 Utility Gap and Budget in Rational Proofs

In the above definitions, we assume that a prover is fully rational, and will give the correct answer for *any* increase in expected payment, no matter how small.

[4] While the model allows for extremely powerful provers, those considered in this paper essentially only need to be powerful enough to determine if $x \in L$ or $x \notin L$.

However, a prover may be lazy, and unwilling to give the correct answer unless the correct answer increases its payment by some minimum amount.

The notion of utility gap captures the payment loss incurred by provers who misreport the answer bit. We recall the formal definition below.

Definition 2 ([13]). *Let L be a language with a rational proof protocol $(V, \vec{P})$ and let $\gamma(n) \geq 0$. We say that $(V, \vec{P})$ has an $\gamma(n)$*-utility gap *if for any input x with $|x| = n$, any strategy profile s of $\vec{P}$ that maximizes the expected payment, and any other strategy profile s', where the answer bit c' under s' does not match the answer bit c under s, i.e., $c' \neq c$, then $u(s) - u(s') > 1/\gamma(n)$.*

Relationship Between Utility Gap and Budget. The *budget* is the total expected payment that a verifier can give in a protocol.

Utility gap and budget are closely related. To study utility gaps consistently, we maintain a fixed $O(1)$ budget.[5] This is because utility gap scales naturally with the payment—a polynomial utility gap under a constant budget is the same as a constant utility gap under a sufficiently-large polynomial budget.

2.2 Analyzing Computational Costs of Rational Proofs

Our primary focus in this paper is analyzing the various computational costs of rational interactive proofs. The different parameters fall into two categories.

Verification Costs. A verifier has three main resources: running time, space usage and its randomness.

In Sect. 3, we focus on time-efficient $O(\log n)$ time verifiers. Thus, their space and randomness is also $O(\log n)$. We denote the class of languages that have time-efficient RIP protocols, that is, protocols with a $O(\log n)$ time verifier as RIP^t. Multi-prover notation MRIP^t is analogous. Similar to the literature on "probabilistically checkable proofs of proximity" (PCPPs) [5,6,33], we assume that the verifier has random access to the input string and the proof tape. Thus, if the messages sent by the provers are of size $C(n)$ bits, the verifier needs at least $O(\log C(n))$ time to index a random location of the transcript.

To achieve better utility gap, in Sect. 4, we restrict the verifier's space usage and randomness, instead of its running time and consider verifiers that use $O(\log n)$ space and $O(\log n)$ randomness. We denote the class of languages that have an RIP protocol with space- and randomness-efficient verifiers, that is, verifiers with $O(\log n)$ space and $O(\log n)$ randomness as $\mathsf{RIP}^{s,r}$.

Protocol Costs. A rational interactive proof protocol has three main ingredients: communication cost, number of rounds of interaction and utility gap.[6]

[5] In contrast, Azar and Micali [3] maintain a polynomial-size budget.

[6] The number of provers is an additional parameter in MRIP protocols, but we ignore this so as not to overload notation. All the MRIP protocols in this paper have two provers and all the upper bounds work even with polynomially many provers.

In Section 3, we study the effect of varying the communication complexity of a protocol on its power when we have a logarithmic time verifier. The number of rounds in all the protocols in the paper is $O(1)$.

We denote the class of languages that have an RIP protocol with communication cost $C(n)$, number of rounds $k(n)$ and utility gap $\gamma(n)$ as $\mathsf{RIP}[C(n), k(n), \gamma(n)]$. The multi-prover version is defined similarly.

3 Verification in Logarithmic Time

In this section we consider *time-efficient* verifiers that run in time logarithmic in the input size. We show that for time-efficient verifiers, access to multiple provers is fundamentally linked to the communication cost of the protocol: any single-prover protocol with high communication costs can be reduced to a communication-efficient multi-prover protocol. On the other hand, multiple provers give no extra power for communication-efficient protocols.

Since the utility gap of all the protocols in this section is polynomial in n, we drop it from the notation for simplicity. Thus, an RIP protocol with a $O(\log n)$-time verifier that has communication complexity $C(n)$ and round complexity $k(n)$ is denoted as $\mathsf{RIP}^t[C(n), k(n)]$.

We omit the proofs, which can be found in the full version [15].

Constant Communication. We first show that multiple provers do not increase the power of a rational proof system when the communication complexity of the protocol is very small, that is, only $O(1)$ bits. Recall that with a single prover, $\mathsf{RIP}^t[O(\log n), O(\log n)] = \mathsf{RIP}^t[O(\log n), O(1)] = \mathsf{UniformTC}_0$ [3,23].

Theorem 1. $\mathsf{MRIP}^t[O(1), O(1)] = \mathsf{UniformTC}_0$.

Logarithmic and Polynomial Communication. We characterize the power of MRIP protocols with $O(\log n)$-time verification, when the communication complexity of the protocol is logarithmic and polynomial in n.

Theorem 2. $\mathsf{MRIP}^t[\text{poly}(n), \text{poly}(n)] = \mathsf{MRIP}^t[O(\log(n)), O(1)] = \mathsf{P_{||}}^{\mathsf{NP}}$.

Azar and Micali [3] characterized the class $\mathsf{P}_{||}^{\mathsf{NP}}$ in terms of single-prover rational proofs with $O(\log n)$ verification and $O(\text{poly}(n))$ communication. In particular, they proved that $\mathsf{RIP}[O(\text{poly}(n)), O(1)] = \mathsf{P}_{||}^{\mathsf{NP}}$.

To prove Theorem 2, we first show that using two provers reduces the communication complexity of the RIP protocol for $\mathsf{P}_{||}^{\mathsf{NP}}$ exponentially. In fact, we show prove a more general statement—any MRIP protocol (thus any RIP protocol as well) with a logarithmic time verifier and polynomial communication can be simulated using two provers, five rounds and logarithmic communication.

Lemma 1. *A MRIP protocol with $p(n)$ procers, $k(n)$ rounds, verification complexity $T(n)$, and communication complexity of $C(n)$ can be simulated by an MRIP protocol with 2 provers, 5 rounds, verification complexity $O(T(n) + \log C(n))$ and communication complexity $O(T(n) + \log C(n))$.*

The main idea behind the proof of Lemma 1 is to use the first prover to obtain the entire "effective transcript" of the original protocol. An effective transcript is all the bits that, for a given randomness r, a log-time time verifier ever accesses in the original protocol. The size of the effective transcript is at most $T(n)$. Then, the second prover is used to verify the correctness of this transcript.

Lemma 1 demonstrates the importance of two provers over one to save on communication cost in rational proofs.

Corollary 1. $\mathsf{RIP}^t[O(\text{poly}(n)), O(1)] = \mathsf{P}_{||}^{\mathsf{NP}} \subseteq \mathsf{MRIP}^t[O(\text{poly}(n)), O(\text{poly}(n)] \subseteq \mathsf{MRIP}^t[O(\log n), O(1)]$.

To complete the proof Theorem 2, we show the following upper bound.

Lemma 2. $\mathsf{MRIP}^t[O(\log(n)), O(1)] \subseteq \mathsf{P}_{||}^{\mathsf{NP}}$.

4 Verification in Logarithmic Space

The protocols in Section 3 have a polynomial utility gap. For a constant budget this means that the provers who mislead the verifier to an incorrect answer lose at least $1/\text{poly}(n)$ of their expected payment.

As utility gap is analogous to the soundness gap in classical proofs, which is constant (independent of n), it is desirable to have rational protocols with constant utility gap as well.

Constant utility gap is difficult to achieve when the verifier is $O(\log n)$ time and cannot even read the entire input. This is true even for classical proofs with a $O(\log n)$-time verifier where the soundness conditioned is weakened to design PCPPs [5,6,33]. In particular, the soundness guarantees of such proofs depend on how far (usually in terms of hamming distance) the input string x is from the language L. We note that all existing $O(\log n)$-time rational proofs [3,23,24] have polynomial utility gap (under a constant budget).

To design protocols with a strong utility gap such as logarithmic or constant, in this section we consider verifier's that use only $O(\log n)$ space and randomness.

Let $\gamma(n)$ be a polynomial-time computable and polynomially bounded function, e.g., $O(1)$, $\log n$, or $\sqrt{n}$. We prove the characterization for utility gap $\gamma(n)$.

Theorem 3. *Let $\mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}$ be a polynomial-time Turing machine that can make $O(\gamma(n))$ non-adaptive queries to an* NP *oracle. This class is equivalent to the class of languages that have a one-round RIP protocol with a logspace verifier, polynomial communication and $\gamma(n)$-utility gap. That is,*

$$\mathsf{RIP}^{r,s}[\text{poly}(n), 1, \gamma(n)] = \mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}.$$

First, we give a space-efficient RIP for the class NP using the log-space interactive proof for the language given by Condon and Ladner [16] as a blackbox.

Lemma 3. $\mathsf{NP} \in \mathsf{RIP}^{r,s}[\text{poly}(n), 1, \gamma(n)]$.

For the lower bound, we use a different but equivalent complexity class. Let $\mathbf{L}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}$ be a logarithmic space machine that can make $O(\gamma(n))$ non-adaptive queries to an NP oracle. Wagner [34] showed that $\mathbf{L}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]} = \mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}$.

Lemma 4. $\mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]} = \mathbf{L}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]} \subseteq \mathsf{RIP}^{r,s}[\mathrm{poly}(n), 1, \gamma(n)]$.

The main idea of the proof is that the prover sends all messages (the overall answer bit, the answer bit of all NP queries and their proofs) in one round. The verifier checks all oracle queries simultaneously using the blackbox protocol [16] and scales the payment appropriately; see full version [15] for the proof.

To complete the proof of Theorem 3 we prove the following upper bound.

Lemma 5. $\mathsf{RIP}^{r,s}[\mathrm{poly}(n), 1, \gamma(n)] \subseteq \mathsf{P}_{||}^{\mathsf{NP}[\gamma(\mathsf{n})]}$.

5 Relationship Between Classical and Rational Proofs

In this section, we show under what conditions does a rational interactive proof reduces to a classical interactive proof. The results in this section are stated in terms of the multi-prover model (that is, MRIP and MIP) which is more general, and thus they also hold for the single prover model (that is, RIP and IP).

To compare the two proof models, we explore their differences. In rational interactive proofs, the provers are allowed to claim $c = 1$ (that is, $x \in L$) or $c = 0$ (that is, $x \notin L$) based on their incentives.[7] Furthermore, for a particular input x of size n, if the provers' claim c about x is incorrect, they lose at least a $1/\gamma(n)$, where $\gamma(n)$ is the utility gap.

On the other hand, in classical proofs, the provers are only allowed to prove $x \in L$. Furthermore, given completeness and soundness parameters c and s respectively, where $0 \leq s < c \leq 1$, for any $x \in L$, there exists a strategy such that V accepts with probability $\geq c$ and for any $x \notin L$, for any strategy V rejects with probability $\leq s$. Thus, given L, the guarantees are independent of x.

In this section, we show when a rational proof reduces to a classical proof. Intuitively, this happens when the utility gap guarantee of a rational protocol is made to hold for all x and in particular, it is enforced to be the gap between the expected payments for all $x \in L$ and all $x \notin L$.

We first show that without loss of generality we can restrict the payments of the provers in a rational proof protocol to be either 1 or 0, where 1 corresponds to "accept" and 0 to "reject" respectively.

Lemma 6. *Any MRIP protocol $(V, \vec{P})$ with payment $R \in [0, 1]$ and utility gap $\gamma(n)$ can be simulated by a MRIP protocol $(V', \vec{P})$ with payment $R' \in \{0, 1\}$ and utility gap $\gamma(n)/2$. In particular, for any strategy s and any input x,*

$$u_{(V,\vec{P})}(x; s) \leq u_{(V',\vec{P})}(x; s) \leq u_{(V,\vec{P})}(x; s) + \gamma(n)/2.$$

V' uses $1 + \lceil \log_2 \gamma(n) \rceil$ more random bits than V.

[7] Thus it is not surprising that rational proofs are closed under complement.

In the proof of Lemma 6, V' simulates V, but instead of giving a payment $R \in [0, 1]$, it gives a payment of 1 with probability R, and 0 otherwise. This preserves the expected reward for each transcript (and thus for each strategy).

Given any rational protocol with zero-one payments, we note that it immediately gives us an accept-reject protocol such that for a given x, the probability that the verifier accepts is exactly the expected payment of the original protocol. More formally let $(V, \vec{P})$ be a rational protocol with $R \in \{0, 1\}$ and utility gap $\gamma(n)$. Let $(V', \vec{P'})$ be defined as follows: V' simulates V, ignores the answer bit c, and if the payment in $(V, \vec{P})$ is $R = 1$ then accept, else reject.

Thus, for a given input string x, the expected payment in $(V, \vec{P})$ is equal to the probability that V' accepts in $(V', \vec{P'})$. That is,

$$u_{(V,\vec{P})}(x; s) = \mathbf{E}_r[R(x, r, (V, \vec{P})(x, r, s))] = \sum_r \Pr(r \mid R(x, r, (V, \vec{P})(x, r, s)) = 1)$$

$$= \sum_r \Pr(r \mid V'\text{accepts } (V', \vec{P'})) = \Pr(V' \text{ accepts } (V', \vec{P'})). \tag{1}$$

Furthermore, $(V', \vec{P'})$ satisfies the following: for any $x \in L$, let s^* denote the optimal strategy of the provers $\vec{P}$, that is, s^* maximizes their expected payment. Then for $\vec{P'}$ following s^*, V' accepts with probability exactly $c(x, n) = u_{(V,\vec{P})}(x; s^*)$. Furthermore, we know from the utility gap condition that for any $x \notin L$, for any strategy s', the probability that V' accepts is at most $u_{(V,\vec{P})}(x; s') < u_{(V,\vec{P})}(x; s^*) - 1/\gamma(n)$, that is, the probability that V' accepts is at most $s(x, n) < c(x, n) - 1/\gamma(n)$. Similar guarantees hold for any $x \notin L$.

However, if we want $(V', \vec{P'})$ to be an interactive proof protocol in the classical sense, that is, with completeness and soundness guarantees that hold for all $x \in L$ and for all $x \notin L$ respectively, we need to impose restrictions on the expected payment function of the rational protocol.

Theorem 4. *Let $(V, \vec{P})$ be an MRIP protocol for a language L such that*

$$\min_{x \in L} u_{(V,\vec{P})}(x; s^*) > \max_{x \notin L} u_{(V,\vec{P})}(x; s^*) + \frac{1}{\gamma(n)} \tag{2}$$

where x is any input of length n, s^* is the strategy of the provers that maximizes their expected payment in $(V, \vec{P})$ and $\gamma(n)$ is any function such that $\gamma(n) > 1$ and $\gamma = O(\text{poly}(n))$. Then, $(V, \vec{P})$ can be simulated by a MIP protocol for L.

We prove this theorem in two parts. First, we show prove the following lemma which proves Theorem 4 with weak completeness and soundness guarantees.

Lemma 7. *Let $(V, \vec{P})$ be an MRIP protocol for a language L that satisfies the condition 2 in Theorem 4. Then, $(V, \vec{P})$ can be simulated by MIP protocol with completeness and soundness parameters $c(n)$ and $s(n)$ respectively such that $c(n) > s(n) + 1/2\gamma(n)$ and $c(n), s(n) \geq 0$.*

We amplify the "gap" of an MIP by repeating the protocol sufficiently many times and then using Chernoff bounds. The techniques are mostly standard, although the parameters must be set carefully to deal with the case $s(n) = 0$.

Lemma 8. *Given an MIP protocol for a language L, with completeness $c(n) > 0$ and soundness $s(n) \geq 0$ such that $c(n) > s(n) + 1/\gamma'(n)$ for some $\gamma'(n) > 1$ and $\gamma' = O(\text{poly}(n))$, can be converted to an MIP protocol for L with completeness at least $1 - 1/\text{poly}(n)$ and soundness at most $1/\text{poly}(n)$.*

Remark 1. The repetition of the MIP protocol to amplify its completeness and soundness guarantee used in Lemma 8 is not efficient as it blows up the number of rounds. There exist more efficient techniques to amplify IP guarantees by parallel repetition that can be used instead; for example, see [4,19,31].

References

1. Allender, E., Hertrampf, U.: On the power of uniform families of constant depth threshold circuits. In: Symposium on Mathematical Foundations of Computer Science, pp. 158–164 (1990)
2. Azar, P.D., Micali, S.: Rational proofs. In: Proceedings of 44th Symposium on Theory of Computing, pp. 1017–1028 (2012)
3. Azar, P.D., Micali, S.: Super-efficient rational proofs. In: Proceedings of 14th Conference on Electronic Commerce, pp. 29–30 (2013)
4. Bellare, M., Goldreich, O., Goldwasser, S.: Randomness in interactive proofs. Comput. Complex. **3**(4), 319–354 (1993)
5. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.: Short PCPs verifiable in polylogarithmic time. In: Proceedings of Conference on Computational Complexity, pp. 120–134 (2005)
6. Ben-Sasson, E., Goldreich, O., Harsha, P., Sudan, M., Vadhan, S.: Robust PCPs of proximity, shorter PCPs, and applications to coding. SIAM J. Comput. **36**(4), 889–974 (2006)
7. Bitansky, N., Chiesa, A.: Succinct arguments from multi-prover interactive proofs and their efficiency benefits. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 255–272. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_16
8. Buhrman, H., Kadin, J., Thierauf, T.: On functions computable with nonadaptive queries to NP. In: Proceedings of 9th Structure in Complexity Theory Conference, pp. 43–52 (1994)
9. Campanelli, M., Gennaro, R.: Sequentially composable rational proofs. In: Proceedings of Decision and Game Theory for Security, pp. 270–288 (2015)
10. Canetti, R., Riva, B., Rothblum, G.N.: Refereed delegation of computation. Inf. Comput. **226**, 16–36 (2013)
11. Chakrabarti, A., Cormode, G., McGregor, A., Thaler, J., Venkatasubramanian, S.: Verifiable stream computation and Arthur-Merlin communication. In: Proceedings of Conference on Computational Complexity, pp. 217–243 (2015)
12. Chandra, A.K., Stockmeyer, L.J.: Alternation. In: Proceedings of 17th Symposium on Foundations of Computer Science, pp. 98–108 (1976)

13. Chen, J., McCauley, S., Singh, S.: Rational proofs with multiple provers. In: Proceedings of 7th Innovations in Theoretical Computer Science Conference, pp. 237–248 (2016)
14. Chen, J., McCauley, S., Singh, S.: Rational proofs with non-cooperative provers. arXiv preprint arXiv:1708.00521 (2017)
15. Chen, J., McCauley, S., Singh, S.: Efficient Rational Proofs with Strong Utility-Gap Guarantees http://arxiv.org/abs/1807.01389 (2018)
16. Condon, A., Ladner, R.: Interactive proof systems with polynomially bounded strategies. J. Comput. Syst. Sci. **50**(3), 506–518 (1995)
17. Cormode, G., Thaler, J., Yi, K.: Verifying computations with streaming interactive proofs. Proc. VLDB Endow. **5**(1), 25–36 (2011)
18. Daruki, S., Thaler, J., Venkatasubramanian, S.: Streaming verification in data analysis. In: Elbassioni, K., Makino, K. (eds.) ISAAC 2015. LNCS, vol. 9472, pp. 715–726. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48971-0_60
19. Feige, U., Kilian, J.: Two prover protocols: low error at affordable rates. In: Proceedings of 26th Symposium on Theory of Computing, pp. 172–183 (1994)
20. Feige, U., Kilian, J.: Making games short. In: Proceedings of 29th Symposium On Theory of Computing, pp. 506–516 (1997)
21. Feigenbaum, J., Koller, D., Shor, P.: A game-theoretic classification of interactive complexity classes. In: Proceedings of 10th Structure in Complexity Theory Conference, pp. 227–237 (1995)
22. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: interactive proofs for muggles. In: Proceedings of 40th Symposium on Theory of Computing, pp. 113–122 (2008)
23. Guo, S., Hubáček, P., Rosen, A., Vald, M.: Rational arguments: single round delegation with sublinear verification. In: Proceedings of 5th Innovations in Theoretical Computer Science, pp. 523–540 (2014)
24. Guo, S., Hubáček, P., Rosen, A., Vald, M.: Rational sumchecks. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 319–351. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_12
25. Hesse, W., Allender, E., Barrington, D.A.M.: Uniform constant-depth threshold circuits for division and iterated multiplication. J. Comput. Syst. Sci. **65**(4), 695–716 (2002)
26. Hubáček, P.: Rationality in the Cryptographic Model. Ph.D thesis, Department Office Computer Science, Aarhus University (2014)
27. Inasawa, K., Yasunaga, K.: Rational proofs against rational verifiers. Fundam. Electron. Commun. Comput. Sci. **100**(11), 2392–2397 (2017)
28. Kalai, Y.T., Rothblum, R.D.: Arguments of proximity. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 422–442. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_21
29. Koller, D., Megiddo, N.: The complexity of two-person zero-sum games in extensive form. Games Econ. Behav. **4**(4), 528–552 (1992)
30. Krentel, M.W.: The complexity of optimization problems. J. Comput. Syst. Sci. **36**(3), 490–509 (1988)
31. Raz, R.: A parallel repetition theorem. SIAM J. Comput. **27**(3), 763–803 (1998)
32. Reif, J.H.: The complexity of two-player games of incomplete information. J. Comput. Syst. Sci. **29**(2), 274–301 (1984)
33. Rothblum, G.N., Vadhan, S., Wigderson, A.: Interactive proofs of proximity: delegating computation in sublinear time. In: Proceedings of 45th Symposium on Theory of Computing, pp. 793–802 (2013)

34. Wagner, K.W.: Bounded query classes. SIAM J. Comput. **19**(5), 833–846 (1990)
35. Zhang, Y., Blanton, M.: Efficient secure and verifiable outsourcing of matrix multiplications. In: Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S.M. (eds.) ISC 2014. LNCS, vol. 8783, pp. 158–178. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13257-0_10