Two-Prover One-Round Proof Systems: Their Power and Their Problems

(Extended Abstract)

Uriel Feige* László Lovász†

Abstract

We characterize the power of two-prover one-round ($MIP(2,1)$) proof systems, showing that $MIP(2,1) = NEXPTIME$. However, the following intriguing question remains open: Does parallel repetition decrease the error probability of $MIP(2,1)$ proof systems?

We use techniques based on quadratic programming to study this problem, and prove the parallel repetition conjecture in some special cases. Interestingly, our work leads to a general polynomial time heuristic for any NP-problem. We prove the effectiveness of this heuristic for several problems, such as computing the chromatic number of perfect graphs.

1 Introduction

1.1 The Power of $MIP(2,1)$

In a multiple-prover interactive proof system, several computationally unbounded provers, P_1, P_2, ..., try to convince a probabilistic polynomial time verifier V that a common input x belongs to a language L [5]. The verifier follows a prespecified protocol which proceeds in rounds. In each round, the verifier sends to each prover in private a polynomial size query and receives a polynomial size answer. Each prover is restricted to seeing only the queries addressed to him, without the ability to communicate with other provers. When the protocol ends, the verifier evaluates a polynomial time predicate on his coin tosses and on the messages exchanged and decides whether to accept or reject.

Definition 1.1 *A k-prover r-round protocol is an* MIP(k,r) proof system *for language L if:*

1. Completeness:

$\forall x \in L\ \exists(P_1, ..., P_k)$ s.t. $(V, P_1, ..., P_k)$ accepts x.

2. Soundness:

$$\forall x \notin L\ \forall P_1, ..., P_k$$
$$\text{Prob}((V, P_1, ..., P_k) \text{ accepts } x) < 2^{-n}.$$

(n denotes the length of the input x.)

Babai, Fortnow and Lund established that $MIP(2, poly(n)) = NEXPTIME$. ($MIP(poly(n), poly(n)) \subset NEXPTIME$ is simple, see [21].)

Fortnow, Rompel and Sipser [13] initiated the characterization of one round protocols. They claimed that $MIP(1, poly(n)) \subset MIP(2,1)$. However, Fortnow [12] demonstrated that parallel repetition of MIP(2,1) protocols does not decrease the error probability in the same way as sequential repetition (or parallel repetition of MIP(1,1) protocols), invalidating [13]'s proof. A correct proof of $MIP(1, poly(n)) \subset MIP(2,1)$ was eventually given by Cai, Condon and Lipton ([7],[9]). Kilian [19] proved that $MIP(2, poly(n))$ has two-prover two-round proof systems with constant error probability, and Feige [10] proved that $MIP(2, poly(n))$ has two-prover one-round proof systems with constant error probability. However, in view of our failure to analyse parallel repetition of protocols, it was not known how to obtain exponentially small error probability in $MIP(2,1)$ proof systems.

*IBM T.J. Watson Research Center. ufeige@watson.ibm.com

†Eötvös University and Princeton University.

24th ANNUAL ACM STOC - 5/92/VICTORIA, B.C., CANADA
© 1992 ACM 0-89791-512-7/92/0004/0733...$1.50

Recent work of Lapidot and Shamir [16] is the key to solving this question. They introduce the concept of a "quasi-oracle", and use it to prove that $NEXPTIME \subset MIP(4,1)$. Using the same concept (our Lemma 2.7 is based on Lemma 2 in [16]), we establish:

Theorem 1.2 *$NEXPTIME = MIP(2,1)$.*

Unlike [9]'s $MIP(2,1)$ protocol for PSPACE (which uses the particular structure of [22]'s protocol) and [16]'s $MIP(4,1)$ protocol for NEXPTIME (which uses the particular structure of the [3] protocol), our $MIP(2,1)$ protocol treats BFL's protocol for NEXPTIME as a blackbox.

1.2 The Parallel Repetition Conjecture

Our exact characterization of $MIP(2,1)$ is obtained without resolving the deceptive question of parallel repetition of $MIP(2,1)$. In fact, even though our NEXPTIME protocol has exponentially small error probability, we do not know whether parallel repetition decreases its error probability further.

The second part of our paper suggests a new approach in studying this difficult problem. As in [8, 10], we model this problem as a problem on games. (The reader may view this as proofs of membership for the empty language).

Let S and T be finite sets. Let π be a probability distribution over $S \times T$. Let V be a predicate. Then $G = G(V,\pi)$ is the following two person cooperative game of incomplete information: A pair of messages $(s,t) \in S \times T$ is chosen at random according to probability distribution π. The message s is sent to one player (which we call *Left*) and t is sent to the other player (*Right*). A *strategy* of a player is a function from messages to replies. The players' objective is to choose strategies P_1 and P_2 which maximize the probability (over π) that $V(s,t,P_1(s),P_2(t)) = 1$. Let the *value* of a game G, denoted by $\omega(G)$, be the probability of success of the players' optimal strategy in the game G. A game G is *trivial* if $\omega(G) = 1$. Otherwise it is *nontrivial.*

If the probability distribution π of the verifier's question is a product distribution, $\pi = \pi_s \times \pi_t$, where π_s is the distribution of questions to Left player and π_t is a distribution of questions to Right player, then we say that G is a *no-information* game. [1] (By receiving a question, a player does not gain any information about which question the other player received). Otherwise, the game is said to be of *partial information.*

$MIP(2,1)$ proof systems can be modeled as families of games (indexed by the input x), where each player corresponds to a prover. If $x \in L$, then $\omega(G_x) = 1$, whereas if $x \notin L$, then $\omega(G_x) < 1$. By playing the game of incomplete information, the verifier can gather statistical evidence to the claim that $x \in L$. The smaller $\omega(G_x)$ is in cases $x \notin L$, the higher the confidence of the verifier that he is not mistakenly accepting a false input.

Parallel repetition of $MIP(2,1)$ protocols was suggested in [13] as a means of decreasing the error probability. In our games terminology, this corresponds to a *product* game G^n. That is, G is carried out n times in parallel, the predicate V is evaluated independently on each of the n coordinates, and the players succeed whenever all n evaluations are '1'. If the n copies of G are played sequentially, the probability of winning all n games is $(\omega(G))^n$ (this can be proven by induction on n). It was erroneously claimed by Fortnow, Rompel and Sipser [13] that this is also the case for parallel execution. However, Fortnow [12] later constructed an example of a game for which $\omega(G^n) > (\omega(G))^n$ (see Section 3).

Let the *amortized value* of a game G, denoted as $\bar{\omega}(G)$, be $sup_k((\omega(G^k))^{1/k})$. We address the following **parallel repetition conjecture:**

Parallel repetition reduces the value of nontrivial games in an exponential rate. Formally, $\omega(G) < 1$ implies $\bar{\omega}(G) < 1$.

Previous results on the parallel repetition conjecture: Cai, Condon and Lipton [8] proved the parallel repetition conjecture for *no-information* games. The actual bounds that [8] obtained for $\bar{\omega}(G)$ were greatly improved in [15, 20, 10, 2], and there still is room for further improvements. [8] also claimed to have proved the conjecture for arbitrary games (i.e., *partial information* games).

Our results: We construct an explicit example that falsifies [8]'s proof for the partial information case.[2] We then study an alternative approach of proving the conjecture, motivated by an approach used to study the *Shannon Capacity* of graphs [17]. This approach is based on quadratic programming. We derive our *quadratic* bound, denoted by $\bar{\sigma}(G)$, which is an upper bound on the amortized value of games.

Theorem 1.3 *For any game G, $\bar{\omega}(G) \leq \bar{\sigma}(G) \leq 1$.*

We use the quadratic bound to prove the parallel repetition conjecture for certain classes of games. Previously, the parallel repetition conjecture was not proved even for a single game of partial information.

[1] This is essentially [8]'s notion of *free* games.

[2] The authors of [8] admitted their error before seeing this explicit example.

1.3 Algorithmic Implications of the Quadratic Bound

Proposition 1.4 *1. NP decision problems have natural representations as games, where triviality of the game corresponds to membership in the NP language.*

2. The quadratic bound is computable in polynomial time, via the ellipsoid method.

The above proposition suggests a hueristic for tackling any NP problem: Represent the input instance x as a game G_x, and compute its quadratic bound. If $\bar{\sigma}(G_x) < 1$, conclude that x is not in the NP-language. Obviously, one would expect that $\bar{\sigma}(G) = 1$ at least for some of the *no* instances, as otherwise $P = NP$. Moreover, by the first part of our paper ($MIP(2,1) = NEXPTIME$), we expect $\bar{\sigma}(G) = 1$ even for games with $\omega(G)$ which is arbitrarily close to 0, unless NEXPTIME = EXPTIME.

We characterize some of the languages for which the above heuristic never errs. It turns out that computationally simple graph properties, such as *connectivity* and *two-colorability*, are always recognized by our procedure. Perhaps more surprisingly, also some complex graph properties, such as the chromatic number of perfect graphs, are always computed correctly by our procedure.

2 $MIP(2,1) = NEXPTIME$

In [10] it was observed that [3]'s proof system for NEXPTIME is *nonadaptive* (i.e., the verifier can prepare all his questions based only on his coin tosses, and not on the provers' previous answers), and thus it can be carried out by polynomially many provers in one round. Thus to prove Theorem 1.2, it is sufficient to prove the following:

Theorem 2.5 *If $L \in MIP(poly(n), 1)$, then $L \in MIP(2,1)$.*

Proof. Consider any MIP(poly(n),1) proof system. By doubling the number of provers, we can decrease the error probability to 2^{-2n}. So our starting point is $(V, P_1, ..., P_m)$, a MIP(m,1) proof system with 2^{-2n} one sided error probability, where $m < n^c$, for some constant c. We transform it into a new MIP(2,1) proof system $(\hat{V}, \hat{P}_1, \hat{P}_2)$ (with 2^{-n} one sided error probability).

In the new protocol, $\hat{V}$ executes all of the original protocol in one step with $\hat{P}_1$. $\hat{V}$ sends the m queries $Q = (q_1, ..., q_m)$, $\hat{P}_1$ replies with $(a_1, ..., a_m)$ of his choice, and $\hat{V}$ checks that V would have accepted the conversation $(q_1, a_1; q_2, a_2; ...; q_m, a_m)$.

The obvious problem in this approach is that for any j, $\hat{P}_1$'s answer to q_j may depend on the whole vector Q, unlike the case we are trying to simulate, in which each prover does not see the messages received by other provers. In order to solve this problem, V performs a consistency check involving $\hat{P}_2$, which forces $\hat{P}_1$ to display *functional behavior*: For any coordinate of the tuple of queries, $\hat{P}_1$'s reply to any query is a function of the query itself, but independent of the queries on other coordinates.

Let l denote an upper bound on the length of any single message that is exchanged in the original protocol $(V, P_1, ..., P_m)$. Let $k = max(l, n, \log(4m))$. $\hat{V}$ chooses a large arbitrary prime N, where $N > 2^{9k}$. (To choose N, $\hat{V}$ may use a probabilistic algorithm, and in the extremely unlikely case that $\hat{V}$ fails to find such a prime, he accepts the protocol outright). All subsequent computations are performed over the finite field Z_N.

We now describe the consistency check on an arbitrary coordinate j. $\hat{V}$ performs this check independently (but in parallel) on each of the m queries sent to $\hat{P}_1$. In square brackets we denote what the good provers should do in order to successfully follow the protocol. We remark that a very similar construction, was already used in [19].

[View each of the m original provers as a function from $\{0,1\}^l$ (the queries q) to Z_N (the answers a). For a function g_j, representing the optimal strategy of P_j, break the l bit arguments into l variables u_1, u_2, ..., u_l, and consider $\hat{g}_j(u_1, u_2, ..., u_l)$, the unique multi-linear representation of g_j over Z_N^l. Formally, let $Q(u_1, u_2, ..., u_l)$ be the l-bit binary string (query q) obtained by concatenating $u_1 \in \{0,1\}$, $u_2 \in \{0,1\}$, ..., $u_l \in \{0,1\}$. Let $s(u)$ be a shorthand representation which means u when $u = 1$, and $1 - u$ when $u = 0$. Then

$$\hat{g}_j = \sum_{u_1, u_2, ..., u_l \in \{0,1\}} g_j(Q(u_1, u_2, ..., u_l)) \prod_{j=1}^{l} s(u_j)$$

$\hat{g}_j$ is a multinomial which is linear in each one of its variables.]

In addition to the query q_j, $\hat{V}$ chooses a point y_j ($\neq q_j$) uniformly at random from $\{[0, N-1]\}^l$. Let $L(q_j, y_j)$ denote the line joining q_j and y_j. To fix a canonical representation of lines, let x_j be the lexicographically first value such that any point on $L(q_j, y_j)$ can be represented as $q_j + tx_j$, for $0 \leq t < N$. (x_j can be computed in polynomial time from q_j and y_j.) $\hat{V}$ sends q_j and x_j to $\hat{P}_1$, and y_j to $\hat{P}_2$. $\hat{P}_1$ is requested to reply with an l degree polynomial $P(t)$.

[$P(t)$ agrees with the values of $\hat{g}_j$ for any value of t.]

$\hat{V}$ extracts $g(q_j) = P(0)$ for use in the simulation of the original protocol.

[$\hat{P}_2$ has to reply with $\hat{g}_j(y_j)$.]

$\hat{V}$ checks that $\hat{P}_2$'s answer agrees with $P(t)$, for t satisfying $q_j + tx_j = y_j$.

The *completeness* property of $(\hat{V}, \hat{P}_1, \hat{P}_2)$ follows from the completeness of the original $MIP(m, 1)$ protocol. If $\hat{P}_1$ and $\hat{P}_2$ follow the strategy outlined in square brackets, $\hat{V}$ accepts whenever V would accept the m-prover protocol.

The proof of *soundness* is more involved, since cheating provers might not follow the strategy outlined in square brackets, and then there is no direct correspondence between executions of the original m-prover protocol and the new two-prover protocol. We highlight a property of the transformation which plays an important role in the proof of soundness.

Low degree polynomials: $\hat{P}_1$'s answer to q_j and $\hat{P}_2$'s answer to y_j have to be related by a low degree polynomial. Since two different l-degree polynomials agree on at most l points, different answers to the same (q_j, x_j) induce different values for almost all arguments y_j. Thus $\hat{P}_2$ cannot adapt himself to inconsistent behavior on the part of $\hat{P}_1$.

We now give a detailed proof of soundness:

Fix the optimal strategies for $\hat{P}_1$ and $\hat{P}_2$. With any tuple of queries $Y = (y_1, ..., y_m)$ to $\hat{P}_2$ we associate m functions $(f_1, ..., f_m)$, each operating on a single coordinate of the tuple $(q_1, ..., q_m)$. Each function f_j is defined in the following way: $f_j(v)$ is the most successful contents of the j^{th} coordinate in $\hat{P}_1$'s reply to $(q_1, ..., q_{j-1}, v, q_{j+1}, ..., q_m)$, where the probability is taken over the distribution of the vectors Q, conditioned on v being the j^{th} coordinate and on the prespecified value of Y. (Note that once Q and Y are specified, the vector X, specifying the choice of lines through Q, is uniquely determined, and thus there is no need to consider X in computing the probability.) In determining which reply is most successful, we take the reply with the highest record of successfully passing the consistency check against y_j, ignoring failures, and breaking ties arbitrarily.

Definition 2.6 *For a tuple $(y_1, ..., y_m)$, prover $\hat{P}_1$ displays* functional behavior *on $q_j = v$, if the event that $\hat{P}_1$ replies differently than $f_j(v)$ and still passes the consistency check has probability at most 2^{-2k} (over the other queries to $\hat{P}_1$).*

W.l.o.g., we consider the first coordinate of Q. Consider any value v for q_1. We shall prove that almost any query sequence $(y_1, ..., y_m)$ to $\hat{P}_2$ induces functional behavior. In fact, we prove an even stronger condition:

Lemma 2.7 *For any v, for any values of $(y_2, ..., y_m)$, for any line L through v, for all but at most a 2^{-3k}-fraction of the points $y_1 \in L$, the prover $\hat{P}_1$ displays functional behavior. (Note that lines through v are disjoint and cover all of Z_N^l.)*

Proof. Assume the contrary and consider the following experiment: Select a random value for y_1 (from $L \setminus \{v\}$) and two random and independent sequences, Q_1 and Q_2, each according to the probability distribution induced by the condition that $q_1 = v$. Consider the event E that $\hat{P}_1$'s reply on v in the two cases differ, but the consistency check against y_1 is passed successfully (in both cases). We analyse the probability of E in two different ways, reaching a contradiction and proving the lemma.

Pick y_1 at random. By our assumption, with probability greater than 2^{-3k} the resulting Y does not induce functional behavior. Not having functional behavior, combined with the fact that l is a bound on the reply size, implies that the probability of $\hat{P}_1$'s most likely answer is at least $2^{-2k}2^{-l} \geq 2^{-3k}$. Now if two query sequences to $\hat{P}_1$ are completed independently at random, the probability that $\hat{P}_1$ succesfully gives the most successful answer for v on Q_1 is at least 2^{-3k}, and the probability that $\hat{P}_1$ succesfully gives a different answer for v on Q_2 is at least 2^{-2k}. This gives a lower bound of 2^{-8k} on the probability of E.

Alternatively, pick Q_1 and Q_2 at random, and assume that $\hat{P}_1$ gives two conflicting answers for v. Since these answers are accompanied by l degree polynomials through v, they can lead to simultaneous successes on at most l values of y_1. Our choice of N implies that $Prob(E) \leq l2^{-9k}$, leading to a contradiction. ∎

Lemma 2.8 *The fraction of Y tuples which induce functional behavior for any possible query on any possible coordinate of Q is at least $1 - m2^{-2k}$.*

Proof. There are m coordinates and l bits in a query, implying at most $m2^l$ different cases to take care of. Now the proof follows from Lemma 2.7. ∎

We are ready to complete the proof of Theorem 2.5. Assume that $x \notin L$ but the protocol $(\hat{V}, \hat{P}_1, \hat{P}_2)(x)$ succeeds with probability greater than 2^{-n}. On at least $2^{-(n+1)}$ of the choices of Y the protocol $(\hat{V}, \hat{P}_1, \hat{P}_2)(x)$ succeeds with probability $2^{-(n+1)}$.

From Lemma 2.8 it follows that at least one of these Ys induces functional behavior $(f_1, ..., f_m)$. Use these functions as the strategies for $(P_1, ..., P_m)$ in the original MIP(m,1) protocol. The probability of success is now at least $2^{-(n+1)} - m2^{-2k} \geq 2^{-(n+2)}$. This contradicts our assumption that the error probability is at most 2^{-2n} (for $n > 2$). ∎

3 Counter-Intuitive Effects

Cai, Condon and Lipton [8] proved the parallel repetition conjecture for games of no-information. Then they argued that any nontrivial game G of partial information can be extended to a game $\hat{G}$ of no information, such that $\omega(G) < \omega(\hat{G}) < 1$ (this argument is correct). Then they claimed the parallel repetition conjecture for games of partial information (Theorem 2.2 in [8]), by implicitly assuming that for any k, $\omega(G^k) < \omega(\hat{G}^k)$. We present a counter example to this assumption. Our example is a variation on Fortnow's example which first demonstrated that $\omega(G^2) = (\omega(G))^2$ does not always hold [12].

Protocol F: We describe the game $F(V, \pi)$. $S = T = \{0, 1\}$. π is uniform over $(0,0)$, $(0,1)$, and $(1,0)$ (excluding the pair of queries $(1,1)$). On any query, each player must answer either 0 or 1. V is satisfied iff $(s \vee P_1(s)) \neq (t \vee P_2(t))$.

Proposition 3.9 $\omega(F) = \omega(F^2) = 2/3$.

Proof. We present only the players' strategy on F^2. Both players follow the same strategy: If a player receives $\langle 0; 0\rangle$, he replies with $\langle 0; 0\rangle$. Otherwise he replies with $\langle 1; 1\rangle$. ∎

Remark: The first example of a game satisfying $\omega(G) = \omega(G^2)$, the *noninteractive agreement protocol*, was presented and analysed in [10].

F is a game of partial information. We use the procedure described in [8] to extend it to a game $\hat{F}$ of no information. This is done by making π uniform over $S \times T$ (allowing the pair of queries $(1,1)$), but modifying the acceptance condition to $V(\hat{F}) = (s \wedge t) \vee ((s \vee P_1(s)) \neq (t \vee P_2(t)))$, giving the players automatic success whenever $(1,1)$ is asked.

Proposition 3.10 $\omega(\hat{F}) = 3/4$. $\omega(\hat{F}^2) = 10/16$.

Proof. The lower bound on $\omega(\hat{F}^2)$ follows from the following strategy: Each player always replies $\langle 0; 0\rangle$, except for the following two cases: $P_1(\langle 0; 0\rangle) = \langle 0; 1\rangle$ and $P_2(\langle 0; 0\rangle) = \langle 1; 0\rangle$. The upper bound on $\omega(\hat{F}^2)$ requires detailed case analysis, and is omitted. ∎

Since $\hat{F}$ is [8]'s notion of a no information extension of the partial information game F, then the combination of Propositions 3.9 and 3.10 falsifies the proof of Theorem 2.2 in [8].

4 The Quadratic Programming Bound

Previously, there was not even a single game of partial information (in contrast to no information) for which a nontrivial upper bound on its amortized value was known. We proceed to give such bounds, showing in particular that $\bar{\omega}(F) < 1$.

Consider a game $G(V, \pi)$. We can formulate the problem of finding the best strategy for the provers as a quadratic program as follows. Recall that S and T are the sets of possible questions to the Left and Right prover and let U and W be the sets of answers they can give. The strategy of the Left prover is a mapping $L : S \to U$, and it can be encoded in the vector $\ell = (\ell_{su}) \in \{0, 1\}^{S \times U}$, defined by

$$\ell_{su} = \begin{cases} 1, & \text{if the prover answers } u \text{ to question } s, \\ 0, & \text{otherwise.} \end{cases}$$

The strategy R of the Right prover can be encoded in a vector $r = (r_{tw})$ similarly.

Represent the acceptance condition (V's predicate) as a matrix $V = \{v_{su,tw}\}$, whose rows are indexed by pairs su ($s \in S$, $u \in U$) and whose columns are indexed by pairs tw ($t \in T$, $w \in W$). Let $v_{su,tw} = 1$ if $V(s, u, t, w)$ accepts, and 0 otherwise. Define the *cost* matrix $C = (c_{su,tw})$, as $c_{su,tw} = v_{su,tw}\pi_{s,t}$. Partition C into submatrices of size $|U| \times |V|$, indexed by pairs st of questions, and every entry in the submatrix corresponding to st is either 0 or π_{st}. For Protocol F, the corresponding matrix C_F is:

$$\begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & 0 & 0 & 0 \end{pmatrix}$$

Lemma 4.11 *$\omega(G)$ is the maximum of the following quadratic program:*
Maximize

$$x^{\mathsf{T}}Cy, \tag{1}$$

Subject to

$$x \geq 0, \qquad y \geq 0, \tag{2}$$

$$\sum_{u \in U} x_{su} = 1 \qquad (\textit{for all } s \in S). \tag{3}$$

$$\sum_{w \in W} y_{tw} = 1 \qquad (\textit{for all } t \in T). \tag{4}$$

Proof. Let (L, R) be an optimum strategy for the provers, then $x = \ell$ and $y = r$ satisfy the constraints in the lemma and

$$x^{\mathsf{T}} C y = \sum_{s,t,u,w} V_{su,tw} \pi_{st} \ell_{su} r_{tw}$$

$$= \sum_{s,t} \pi_{st} V_{sL(s),tR(t)} = \omega(G).$$

So the maximum is the quadratic program is at least $\omega(G)$.

Conversely, consider an optimum solution of the quadratic program (1) -(4). Fixing x, the program is linear in y. Hence the optimum is attained at a vertex of the feasible domain for y, which is trivially a 0-1 vector. Similarly, we may assume that x is a 0-1 vector. But then the constraints imply that for every s there is exactly one u with $x_{su} = 1$, i.e., x encodes a strategy for the Left prover. Similarly, y encodes a strategy for the Right prover. This pair of strategies shows that $\omega(G)$ is at least as large as the optimum of the (1)-(4). ∎

In passing, we obtain the following corollary:

Corollary 4.12 *If there is a polynomial time algorithm that approximates the maximum of quadratic programs to within some constant multiplicative factor, then $EXPTIME = NEXPTIME$.*

Proof. Consider an arbitrary NEXPTIME language L and an input x. By Theorem 1.2, there is a two-prover one-round proof system such that the acceptance probability reflects membership of x in L. By Lemma 4.11, this acceptance probability is the maximum of a quadratic program with exponentially many constraints. Applying the polynomial time approximation algorithm to this quadratic program would give an exponential time decision procedure for L. ∎

For more details on the relation between interactive proof systems and approximation problems see [11]. Lemma 4.11 and Corollary 4.12 were independently discovered in [4].

Next we apply the method of "linearizing" this quadratic program (see [6, 23, 18]).

Let us introduce a new variable $p_{su,tw}$ for all $s \in S$, $u \in U$, $t \in T$ and $w \in W$. These variables will "correspond" to the products $x_{su} y_{tw}$. We can arrange these new variables as a matrix $P = (p_{su,tw})$, with rows indexed by pairs su ($s \in S$, $u \in U$) and columns indexed by pairs tw ($t \in T$, $w \in W$). So P is of the same shape as C and has the same natural way of partitioning into submatrices.

Unfortunately, generally we also have to introduce new variables for products $x_{su} x_{s'u'}$ and $y_{tw} y_{t'w'}$. So we consider the set $J = (S \times U) \cup (T \times W)$ (this indexes the "old" variables; we assume that $(S \times U)$ and $(T \times W)$ are disjoint, unless emphasized otherwise), and introduce new variables p_{jk} for all $j, k \in J$. Each new variable will correspond to the product of two old variables. We can arrange the new variables in a $|J| \times |J|$ matrix $\hat{P}$. There is a natural way to partition $\hat{P}$ into submatrices, indexed by pairs st, where $s, t \in S \cup T$. We call these submatrices "blocks". We also "extend" C to the matrix

$$\hat{C} = \frac{1}{2} \begin{pmatrix} 0 & C \\ C^{\mathsf{T}} & 0 \end{pmatrix},$$

which is of the same shape as $\hat{P}$. We may denote the entries of $\hat{C}$ also by c_{jk}.

Consider the following optimization problem:
Maximize

$$\sum_{j,k \in J} c_{jk} p_{jk}, \tag{5}$$

Subject to

$$\hat{P} \geq 0, \tag{6}$$

$$\sum_{jk \in B_{st}} p_{jk} = 1, \qquad (\text{ for each block } B_{st}), \tag{7}$$

$$\hat{P} \text{ is symmetric,} \tag{8}$$

$$\hat{P} \text{ is positive semidefinite.} \tag{9}$$

We can write the objective function as $\hat{C} \cdot \hat{P}$, where "$\cdot$" denotes the entry-wise inner product of two matrices. Also we can write (7) as $Q_{st} \cdot P = 1$, where Q_{st} is the matrix that is 1 on block B_{st} and 0 elsewhere. Also note that (9) can be viewed as an (infinite) family of linear constraints, since it is equivalent to saying that $x^{\mathsf{T}} \hat{P} x \geq 0$ for all $x \in \mathbb{R}^J$.

Every feasible 0-1 solution of (1)-(4) (representing a strategy for the provers), defines a feasible solution of (5)-(9) by taking the products of the variables, and the objective value for this problem is the same. This does not hold conversely. A feasible solution of (6)-(9) is called a *pseudostrategy* for the provers. We denote the optimum value of (5)-(9) by $\sigma(G)$, and call it the *quadratic bound* on the game.

Since (5)-(9) defines a convex program with a linear objective function, it is natural to write up its dual program. This turns out the following.
Minimize

$$\sum_{s,t \in S \cup T} \max_{jk \in B_{st}} (c_{jk} + m_{jk}), \tag{10}$$

Subject to

$$\hat{M} = (m_{jk}) \text{ is positive semidefinite.} \tag{11}$$

Lemma 4.13 *The optimum of (10)-(11) is $\sigma(G)$.*

Proof. Let $\sigma = \sigma(G)$ and let (for this proof) σ' denote the optimum value of (10)-(11). Consider any optimum solution $\hat{P}$ of (5)–(9), and any optimum solution $\hat{M}$ of (10)-(11). Both of these matrices are positive semidefinite, hence

$$\hat{P}\cdot\hat{M} \geq 0.$$

Hence

$$\begin{aligned} \sigma &\leq \hat{P}\cdot\hat{C} \leq \hat{P}\cdot(\hat{C}+\hat{M}) \\ &= \sum_{s,t\in S\cup T}\ \sum_{j,k\in B_{st}} p_{jk}(c_{jk}+m_{jk}) \\ &\leq \sum_{s,t}\left(\sum_{jk\in B_{st}} p_{jk}\right)\max_{jk\in B_{st}}(c_{jk}+m_{jk}) \\ &= \sum_{s,t}\max_{jk\in B_{st}}(c_{jk}+m_{jk}) = \sigma'. \end{aligned}$$

To show the converse, let z be a new variable and let K denote the convex cone in $\mathbb{R}^{J\times J}\times\mathbb{R}$, defined by $\hat{P}\geq 0$, $\hat{P}$ positive semidefinite, and $Q_{st}\cdot\hat{P}-z=0$ (for all $s,t\in S\cup T$). By the defnition of σ, every pair (P,z) satisfying these constraints also satisfies $\sigma z-\hat{C}\cdot\hat{P}\geq 0$. This means that $(-C,\sigma)$ is in the polar cone of K. Now K was defined as the intersection of cones (the non-negative cone, the positive semidefinite cone, and the hyperplanes $Q_{st}\cdot\hat{P}-z=0$), and hence its polar is generated by the polars of these. Now the polar of the non-negative cone is itself, and the same holds for the positive semidefinite cone; the polar of $Q_{st}\cdot\hat{P}-z=0$ is the line spanned by $(Q_{st},-1)$. So we have a non-negative matrix N, a positive semidefinite matrix M, and real numbers w_{st} such that

$$-\hat{C} = N+M+\sum_{st} w_{st}Q_{st},$$

and

$$-\sigma = \sum_{st} w_{st}.$$

Now (4) is equivalent to saying that

$$-w_{st} \geq \max_{jk\in B_{st}}(c_{jk}+m_{jk}),$$

and hence

$$\sum_{st}\max_{jk\in B_{st}}(c_{jk}+m_{jk}) \leq -\sum_{st} w_{st} \leq \sigma.$$

Since the left-hand side is a feasible solution value for (10)-(11), this shows that $\sigma'\leq\sigma$. ∎

Our main motivation for introducing the quadratic bound is to obtain an upper bound $\sigma'(G)$ on $\omega(G)$ which would satisfy

$$\sigma'(G_1\times G_2)\leq\sigma'(G_1)\sigma'(G_2);$$

such an upper bound would then automatically be also an upper bound on $\bar{\omega}(G)$. The quadratic bound $\sigma(G)$ introduced in the previous section does not have this property: it is in fact supermultiplicative, i.e. $\sigma(G_1\times G_2)\geq\sigma(G_1)\sigma(G_2)$. To prove submultiplicativity for some version of σ, we want to be able to "multiply" dual solutions.

So we consider the following relaxation of (5)–(9): *Maximize*

$$\sum_{j,k\in J} c_{jk}p_{jk}, \tag{12}$$

Subject to

$$p_{su,tw}\geq 0,\quad \forall\, s\in S, t\in T, u\in U, w\in W \tag{13}$$

$$\sum_{u,w\in U}|p_{su,tw}|\leq 1 \qquad (\text{ for each } s,t\in S), \tag{14}$$

$$\sum_{u,w\in W}|p_{su,tw}|\leq 1 \qquad (\text{ for each } s,t\in T), \tag{15}$$

$$\hat{P} \text{ is symmetric,} \tag{16}$$

$$\hat{P} \text{ is positive semidefinite.} \tag{17}$$

The relaxation consists of omitting about half of (6) and (7), and modifying the rest of (7). We denote by $\bar{\sigma}(G)$ the optimum value of this program. It is clear that $\bar{\sigma}(G)\geq\sigma(G)$.

The dual of this optimization problem is *Minimize*

$$\sum_{s,t\in S}\max_{u,w\in U}|m_{su,tw}| + \sum_{s,t\in T}\max_{u,w\in W}|m_{su,tw}| \tag{18}$$

Subject to

$$\hat{M}=(m_{jk}) \text{ is positive semidefinite,} \tag{19}$$

$$m_{su,tw}\geq c_{su,tw} \tag{20}$$

for all $s\in S, t\in T, u\in U, w\in W$.

Equivalently, we can replace (20) by:

$$m_{su,tw}\leq -c_{su,tw} \tag{21}$$

since scaling the rows and columns corresponding to the Left player by -1 preserves positive semidefiniteness. We shall use these two forms interchangeably.

If we scale the rows and columns of M corresponding to the Left player by any positive λ, and scale those corresponding to the Right player by $1/\lambda$, we get another feasible solution of (19)–(20)). For the optimum solution, this cannot improve the objective function, and hence we see that every optimum solution must satisfy

$$\sum_{s,t\in S} \max_{u,w\in U} |m_{su,tw}| = \sum_{s,t\in T} \max_{u,w\in W} |m_{su,tw}|.$$

Thus in (18), we could replace the sum by either twice the maximum or twice the geometric mean of these two terms.

The optimum value of this program is also $\bar{\sigma}(G)$ (proof given in the full paper). It satisfies:

Lemma 4.14 *For every game, $\bar{\sigma}(G) \leq 1$.*

Proof. Let c_{st} denote the maximum entry of $\hat{C}$ in the block B_{st}. Consider the following matrix M_0:

$$m_{su,tw} = \begin{cases} c_{st}, & \text{if } s \in S, t \in T \\ & \text{or } \ s \in T, t \in S. \\ \sum_{r\in T} c_{sr}, & \text{if } t = s \in S, \\ \sum_{r\in S} c_{rt}, & \text{if } t = s \in T, \\ 0, & \text{otherwise.} \end{cases}$$

If we eliminate identical rows and columns from M_0, we get a matrix in which the diagonal entry is the sum of all other entries in each row. This implies that M_0 is positive semidefinite. Moreover, M_0 satisfies (20) trivially and the value of the objective function is 1. ∎

All these complications have been introduced to allow the proof of the following lemma.

Lemma 4.15 *For any two games,*

$$\bar{\sigma}(G_1 \times G_2) = \bar{\sigma}(G_1)\bar{\sigma}(G_2).$$

Proof. Let C_i be the cost matrix for game i. Then the cost matrix of the product game is $C = C_1 \circ C_2$, and

$$\hat{C} = \frac{1}{2}\begin{pmatrix} 0 & C \\ C^{\mathsf{T}} & 0 \end{pmatrix},$$

Let M_i be an optimum solution of (18)–(20) for game i. Write $M_i = \begin{pmatrix} A_i & D_i \\ D_i^{\mathsf{T}} & B_i \end{pmatrix}$, where the rows and columns of A_i correspond to the question-answer pairs of the Left player, and consider the matrix

$$M = \begin{pmatrix} A_1 \circ A_2 & D_1 \circ D_2 \\ D_1^{\mathsf{T}} \circ D_2^{\mathsf{T}} & B_1 \circ B_2 \end{pmatrix}.$$

This is a submatrix of the Kronecker product $M_1 \circ M_2$ in symmetric position, and hence it is symmetric and positive semidefinite. Moreover, $2D_1 \circ D_2 \geq C_1 \circ C_2$, so $2M$ satisfies (20) (the factor of 2 is needed to compensate for the factor of 1/2 in the definition of $\hat{C}$). Let us determine the first term of the objective function:

$$2 \sum_{\substack{s_1,t_1\in S_1 \\ s_2,t_2\in S_2}} \max_{\substack{u_1,w_1\in U_1 \\ u_2,w_2\in U_2}} |m^{(1)}_{s_1u_1,t_1w_1} \cdot m^{(2)}_{s_2u_2,t_2w_2}|$$

$$= 2 \sum_{\substack{s_1,t_1\in S_1 \\ s_2,t_2\in S_2}} \max_{u_1,w_1\in U_1} |m^{(1)}_{s_1u_1,t_1w_1}| \cdot \max_{u_2,w_2\in U_2} |m^{(2)}_{s_2u_2,t_2w_2}|$$

$$= 2 \left(\sum_{s_1,t_1\in S_1} \max_{u_1,w_1\in U_1} |m^{(1)}_{s_1u_1,t_1w_1}| \right)$$

$$\cdot \left(\sum_{s_2\in S_2,t_2\in S_2} \max_{u_2,w_2\in U_2} |m^{(2)}_{s_2u_2,t_2w_2}| \right)$$

$$= \frac{1}{2}\bar{\sigma}(G_1)\bar{\sigma}(G_2).$$

The other term in the objective function is the same, hence

$$\bar{\sigma}(G_1 \times G_2) \leq \bar{\sigma}(G_1)\bar{\sigma}(G_2).$$

The opposite inequality follows by a similar argument applied to the optimum primal solutions. ∎

This Lemma implies:

Theorem 4.16 *For every game G, we have*

$$\bar{\omega}(G) \leq \bar{\sigma}(G).$$

For investigating the parallel repetition conjecture, the crucial test is whether $\bar{\sigma}(G) = 1$, or in our terminology, whether the provers have a perfect pseudostrategy. Luckily, the following lemma simplifies matters:

Lemma 4.17 *For any game, $\bar{\sigma}(G) = 1$ if and only if $\sigma(G) = 1$.*

Proof. The "if" part is obvious. Assume that $\bar{\sigma}(G) = 1$, and let $\hat{P}$ be an optimal solution of (12)–(17). We may assume that every block B_{ss} in $\hat{P}$ is diagonal; for if it has an entry $p_{su,sw}$ which is nonzero, then we can consider the matrix $\hat{P}'$ defined by

$$(\hat{P}')_{jk} = \begin{cases} -p_{su,sw}, & \text{if } j = su,\ k = sw \\ & \text{or } \ j = sw,\ k = su, \\ |p_{su,sw}|, & \text{if } j = k = su \\ & \text{or } \ j = k = sw, \\ 0, & \text{otherwise.} \end{cases}$$

This matrix is clearly positive semidefinite and hence $\hat{P} + \hat{P}'$ is another optimal solution of (12)-(17). We can replace $\hat{P}$ by $\hat{P} + \hat{P}'$ and repeat until all diagonal blocks are themselves diagonal matrices. Since $\hat{P}$ is positive semidefinite, its diagonal entries are nonnegative. We may also assume that the sum of entries in each diagonal block is exactly 1, since raising diagonal entries to achieve this only improves the solution.

Next, note that the matrix $\hat{M}_0$ constructed in the proof of lemma 4.14 is an optimal solution of the dual. By the appropriate complementary slackness condition, this implies that

$$\hat{P} \cdot \hat{M}_0 = 0. \qquad (22)$$

Now notice that the matrix $\hat{M}_0$ can be written as follows. Let, for $s \in S$ and $t \in T$, the vector $v_{st} \in \mathbb{R}^J$ be defined by

$$(v_{st})_j = \begin{cases} 1, & \text{if } j = su \text{ for some } u \in U, \\ -1, & \text{if } j = tw \text{ for some } w \in W, \\ 0, & \text{otherwise.} \end{cases}$$

(We choose the dual satisfying (21), and hence the -1 term in the case $j = tw$). Then we have

$$\hat{M}_0 = \sum_{s \in S, t \in T} c_{st} v_{st} v_{st}^{\mathsf{T}},$$

and hence

$$\hat{P} \cdot \hat{M}_0 = \sum_{s \in S, t \in T} c_{st} v_{st}^{\mathsf{T}} \hat{P} v_{st}.$$

Thus it follows from (22) that for every s, t with $c_{st} > 0$, we must have

$$v_{st}^{\mathsf{T}} \hat{P} \cdot v_{st} = 0.$$

Since $\hat{P}$ is positive semidefinite, this implies that

$$\hat{P} \cdot v_{st} = 0. \qquad (23)$$

This means in particular that for every $s \in S$, $t \in T$ and $u \in U$, we have

$$\sum_{w \in W} p_{su,tw} = p_{su,su}.$$

So for all $s \in S$ and $t \in T$ we have

$$\sum_{u \in U, w \in W} p_{su,tw} = \sum_{u \in U} p_{su,su} = 1.$$

So $\hat{P}$ satisfies (7). Equation (23) also implies that $\hat{P}$ sums to 1 on each block. But by (14, 15), the sum of absolute values of entries is at most one in each block B_{st} with $s, t \in S$ or $s, t \in T$, it follows that these blocks are also non-negative. Thus P is non-negative and it is a feasible solution of (6)-(9). This proves that $\sigma(G) = 1$. ∎

Having developed some machinery, we can use the quadratic bound to prove the parallel repetition conjecture for several protocols, including Protocol F. One such class of protocols is that of games with the *uniqueness* property, that is, games such that for all s, t, u there is at most one w such that $V(s, t, u, w)$ holds, and for all s, t, w there is at most one u such that $V(s, t, u, w)$ holds. These games were first defined by [8].

Theorem 4.18 *For any game G with the uniqueness property, $\omega(G) = 1$ if and only if $\bar{\sigma}(G) = 1$.*

Proof. $\omega(G) = 1 \Longrightarrow \bar{\sigma}(G) = 1$ is trivial. We prove $\bar{\sigma}(G) = 1 \Longrightarrow \omega(G) = 1$. By Lemma 4.17, it is sufficient to prove that $\sigma(G) = 1 \Longrightarrow \omega(G) = 1$.

Consider any pseudo-strategy $\hat{P}$ which satisfies (6) - (9), for which $\hat{C} \cdot \hat{P} = 1$, where $\hat{C}$ corresponds to a game G which has the uniqueness property. We construct a perfect strategy for G.

We eliminate from $\hat{P}$ any row (and column) which is all 0. As in the proof of Lemma 4.17, we may assume that all diagonal blocks of $\hat{P}$ are themselves diagonal matrices, and the sum of the entries in each diagonal block is exactly 1. Consider any block B_{st}, where $s \in S$, $t \in T$, and (s, t) is in the support of G. Since $\hat{P}$ represents a perfect pseudo-strategy, then $\hat{P}_{sutw} > 0$ implies $\hat{C}_{sutw} > 0$. Since G has the uniqueness property, then for any $u \in U$, there exist at most one $w \in W$ for which $\hat{P}_{sutw} > 0$. By (7) and (9), it follows that the nonzero entries of block B_{st} are equal to the corresponding diagonal entries of $\hat{P}$, and they form a permutation matrix.

Consider any nonzero entry $\hat{P}_{ij}$ in a support block of G. Let v be a vector such that $v_i = 1$, $v_j = -1$, and $v_k = 0$ for $k \neq i, j$. Then by the above discussion it follows that $v^T \hat{P} v = 0$. Since $\hat{P}$ is positive semidefinite, $\hat{P} v = 0$. Thus columns i and j are equal in $\hat{P}$.

Now we describe a greedy algorithm for extracting a perfect true strategy (P_1, P_2) from the perfect pseudo-strategy $\hat{P}$. If the game G can be partitioned into k components (disjoint sets $S_i \subset S$ and $T_i \subset T$, $1 \leq i \leq k$, such that if (s, t) is in the support of G, then there exists j such that $s \in S_j$ and $t \in T_j$), the greedy algorithm should be repeated for each of the components.

Consider an arbitrary query $s \in S$ and an arbitrary answer $u \in U$ such that $\hat{P}_{susu} > 0$. Set $P_1(s) = u$. Consider all queries $t \in T$ such that (s, t) is in the support of G. For each such t there exists a unique $w \in W$ such that $\hat{P}_{sutw} = \hat{P}_{susu}$. Set $P_2(t) = w$. Now the crucial point to notice is that each such column tw

is equal to column su. Now we can continue the process of defining answers to queries s' for which (s', t) is in the support of G without fear of conflict between the different columns (i.e., there is one u' such that $V(s', u', t, w)$ accepts for any of the previous (t, w)). Continuing this process, each $s \in S$ and each $t \in T$ are visited exactly once, and so we extract a perfect strategy for P_1 and P_2. ∎

If a game with the uniqueness property is also a game of no-information, then a stronger statement is provable:

Theorem 4.19 *For any no-information game G with the uniqueness property, $\bar{\omega}(G) \leq (\omega(G))^{1/4}$.*

The proof will be given in the full version of the paper. It is not based on the quadratic bound.

5 The Quadratic Bound as an Algorithm

Theorem 5.20 *The value $\sigma(G)$ is polynomial time computable.*

This follows from the ellipsoid method, since (6)–(9) define a convex region for which the separation problem is polynomial time solvable. See [14]. Let us add that to determine the optimum of the quadratic program (1)–(4) is NP-hard.

As noted earlier, games represent language recognition problems in a natural way, where triviality of a game corresponds to membership in the language. It is natural to investigate how $\sigma(G)$ performs as a language recognition procedure, independently of the issue of parallel repetition. Theorem 4.18 can be interpreted as evidence that $\sigma(G)$ is sensitive enough to serve as a polynomial time (though inefficient in practice) language recognition procedure for languages whose containment problem can be represented as games with the uniqueness property. These languages seem to be related to the class *randomized LOGSPACE.* On the one hand, there exists a randomized LOGSPACE procedure which determines whether the players in such a game have a perfect strategy. On the other hand, the classical random-LOGSPACE language of undirected connectivity [1] has a two-prover one-round proof systems with the uniqueness property:

s-t Cut: Can the nodes of a graph x be colored with two colors such that s is colored white, t is colored black, and adjacent nodes have the same color?

The following game has a perfect strategy if and only if x has an s-t cut: Construct self loops on each of the nodes of x. V selects an edge $(s', t') \in x$ at random, and requests the color of s' from Left player and the color of t' from Right player. V accepts if the player's answers match, and do not contradict the colors of s and t. It is easy to see that this game has the uniqueness property.

We note that also two-colorability (which is in randomized LOGSPACE) has a $MIP(2,1)$ proof system with the uniqueness property.

We now develop additional machinery to show the effectiveness of our algorithm to more complex problems.

Let G_1 and G_2 be two graphs. We write that $G_1 \longleftarrow G_2$ if there is a homomorphism of G_1 into G_2, i.e. a mapping of $V(G_1)$ into $V(G_2)$ which maps adjacent nodes to adjacent nodes. Special cases of this notion are: G_1 is k-colorable (choose G_2 the complete k-graph), or G_2 has a k-clique (choose G_1 the complete k-graph).

Suppose that the true Provers want to convince the Verifier that such a mapping exists. There is a simple protocol for this: the two Provers (are supposed to) agree on a mapping. The Verifier asks each of them to name the image of a node of G_1, uniformly distributed over all pairs. His criteria of acceptance are: if he asks the same node, he should get the same node of G_2; if he asks adjacent nodes of G_1, he should get adjacent nodes of G_2. It is clear that if the there is no homomorphism from G_1 to G_2, then the provers fail with probability at least $1/|V(G_1)|^2$.

The above game is *symmetric*: $S = T$, $U = W$, $V_{us,tw} = V_{tw,us}$ and $\pi_{st} = \pi_{ts}$. In addition, if both players get the same question, the verifier only accepts the same answer. We call such a game *strongly symmetric*. In this case the existence of a perfect pseudostrategy can be characterized in a simpler way.

Note that to define the pseudostrategy, we have to artifially make T and S as well as U and W disjoint; but we have natural bijections $\phi : S \to T$ and $\psi : U \to W$.

We define a *reduced pseudostrategy* as matrix $P \in \mathbb{R}^{(S\times U)\times(S\times U)}$ with the following properties:

$$P \geq 0, \tag{24}$$

$$\sum_{jk\in B_{st}} p_{jk} = 1 \quad \text{(for each block } B_{st}\text{)}, \tag{25}$$

$$P \text{ is symmetric}, \tag{26}$$

$$P \text{ is positive semidefinite}. \tag{27}$$

For each reduced pseudostrategy, we consider the objective function

$$\sum_{j,k\in(S\times U)} c_{jk} p_{jk},$$

We say that the reduced pseudostrategy is *perfect*, if this value is 1. Just as above, we obtain that a reduced pseudostrategy is perfect if and only if

$$P_{su,tv} = 0 \quad \text{whenever } V_{su,tv} = 0 \text{ and } \pi_{st} \neq 0.$$

We will call a perfect reduced pseudostrategy a *hoax*.

Lemma 5.21 *A strongly symmetric game has a hoax if and only if it has a perfect pseudostrategy.*

Proof. If P is a hoax then $\hat{P} = \begin{pmatrix} P & P \\ P & P \end{pmatrix}$ is a perfect pseudostrategy.

Conversely, let $\hat{P}$ be a perfect pseudostrategy. We can write $P = \begin{pmatrix} P_1 & P_2 \\ P_2^{\mathsf{T}} & P_3 \end{pmatrix}$. We claim that $P_1 = P_2 = P_3$.

By our assumption on the verifier, the block $B_{s\phi(s)}$ of P_2 is diagonal and hence by property (23) of perfect pseudostrategies, it must be the same as the block B_{ss} in P_1 as well as the block $B_{\phi(s)\phi(s)}$ is P_3. So the diagonals of the P_i are the same. Since $\hat{P}$ is positive semidefinite, we can write it as a Gram matrix, i.e., there exist vectors v_j $(j \in J)$ such that $p_{jk} = v_j^{\mathsf{T}} v_k$. Now Comparing the diagonal entries of P_1, P_2 and P_3, we see that

$$v_{su}^{\mathsf{T}} v_{\phi(s)u} = v_{su}^{\mathsf{T}} v_{su} = v_{\phi(s)u}^{\mathsf{T}} v_{\phi(s)u},$$

which is only possible if $v_{su} = v_{\phi(s)u}$. But then $P_1 = P_2 = P_3$ follows. Let P denote this matrix. Since $P = P_1$, P is positive semidefinite. Since $P = P_2$, it satisfies conditions (5) of perfect pseudostrategies. So P is a hoax. ∎

One more transformation of the condition is the following. Assume that P is a hoax for a symmetric game. We can write P as a Gram matrix:

$$P_{su,tw} = v_{su}^{\mathsf{T}} v_{tw}$$

for appropriate vectors $v_{su} \in \mathbb{R}^N$ for some N. Now equation (23) implies that for every $s \in S$, $t_1, t_2 \in T$, and $u \in U$, we have

$$\sum_{w \in W} v_{su}^{\mathsf{T}} v_{t_1 w} = \sum_{w \in W} v_{su}^{\mathsf{T}} v_{t_2 w},$$

or

$$v_{su}^{\mathsf{T}} \left(\sum_{w \in U} u_{t_1 w} - \sum_{w \in U} u_{t_2 w} \right) = 0.$$

Appropriate linear combination of these equations yields

$$\left(\sum_{w \in U} v_{t_1 w} - \sum_{w \in U} v_{t_2 w} \right)^2 = 0,$$

and hence

$$\left(\sum_{w \in U} v_{t_1 w} = \sum_{w \in U} v_{t_2 w} \right).$$

So we get that

$$\tilde{v} = \sum_{w \in U} v_{tw} \tag{28}$$

is independent of t.

It follows from the strong symmetry and condition (5) that $p_{su,sw} = 0$ for $u \neq w$, and hence the vectors v_{su}, $u \in U$ are mutually orthogonal for each s. This implies that

$$\tilde{v}^{\mathsf{T}} v_{su} = |v_{su}|^2, \tag{29}$$

and

$$|\tilde{v}^{\mathsf{T}}| = 1. \tag{30}$$

Moreover, we must have that

$$v_{su}^{\mathsf{T}} v_{tw} = 0 \text{ if } \pi_{st} > 0 \text{ but } V_{su,tw} = 0 \tag{31}$$

Conversely, any system of vectors satisfying conditions (28)-(31) yields a hoax.

Let us write $G_1 \Longleftarrow G_2$ if there is a hoax for the homomorphism game. We need the following lemma:

Lemma 5.22 *If $G_1 \Longleftarrow G_2$ and $G_2 \Longleftarrow G_3$, then $G_1 \Longleftarrow G_3$.*

Proof. If P is a hoax for $G_1 \Longleftarrow G_2$ and P' is a hoax for $G_2 \Longleftarrow G_3$, then define P'' by

$$p''_{su,tw} = \sum_{a,b \in V(G_2)} p_{sa,tb} p_{au,bw}$$

$$(s, t \in V(G_1), u, w \in V(G_3)).$$

It is easy to verify that this is a hoax for $G_1 \Longleftarrow G_3$. ∎

Every pair of graphs G_1 and G_2 with $G_1 \not\longleftarrow G_2$ defines two classes of finite graphs:

$$\mathcal{K} = \{H : \ G_1 \longleftarrow H\}, \ \ \mathcal{K}' = \{H : \ H \longleftarrow G_2\},$$

which are trivially disjoint, and both are in NP. An outstanding open problem in complexity theory asks if any two such classes can be separated in P, i.e., if there is a polynomial time decidable class $\mathcal{K}''$ of graphs such that

$$\mathcal{K} \subseteq \mathcal{K}'', \qquad \mathcal{K}' \cap \mathcal{K}'' = \emptyset.$$

While we do not know the answer to this general question, we can show that it is in the affirmative if we make the stronger assumption that $G_1 \not\Longleftarrow G_2$.

Theorem 5.23 *Let G_1 and G_2 be two graphs such that $G_1 \not\Longleftarrow G_2$. Then the class*

$$\mathcal{K}'' = \{H : G_1 \Longleftarrow H\}$$

is in P and separates $\mathcal{K}$ and $\mathcal{K}'$.

The proof is obvious. To apply this theorem, we have to find interesting pairs of graphs with $H_1 \not\Longleftarrow H_2$. One example is given by the following:

Lemma 5.24 $K_k \not\Longleftarrow K_{k-1}$

The proof is based on characterization (28)-(31).

This lemma implies that the class of graphs containing a k-cliques can be separated from the class of $(k-1)$-colorable graphs by a class in P (in particular, giving the chromatic number of perfect graphs). This has been known [17], and in fact the methods obtaining it served as a motivation for our quadratic bound.

It is a challenging research problem to find other applications of Theorem 5.23.

References

[1] R. Aleliunas, R. Karp, R. Lipton, L. Lovász, C. Rackoff, "Random Walks, Universal Traversal Sequences, and the Complexity of Maze Problems", 20th *FOCS, 1979, pp. 218-223.*

[2] N. Alon, "Probabilistic Methods in Extremal Finite Set Theory", *to appear in the Proc. of the Conference on Extremal Problems for Finite Sets, Hungary, 1991.*

[3] L. Babai, L. Fortnow, C. Lund, "Non-Deterministic Exponential Time has Two-Prover Interactive Protocols", 31st *FOCS, 1990, pp. 16-25.*

[4] M. Bellare, P. Rogaway, "The Complexity of Continuous Optimization", *manuscript.*

[5] M. Ben-or, S. Goldwasser, J. Kilian, A. Wigderson, "Multi Prover Interactive Proofs: How to Remove Intractability", 20th *STOC, 1988, pp. 113-131.*

[6] Boros and Hammer.

[7] J. Cai, A. Condon, R. Lipton, "On Bounded Round Multi-Prover Interactive Proof Systems", *Structures 1990, pp. 45-54.*

[8] J. Cai, A. Condon, R. Lipton, "Playing Games of Incomplete Information", *STACS 1990.*

[9] J. Cai, A. Condon, R. Lipton, "PSPACE is Provable by Two Provers in One Round", *Structures 1991, pp. 110-115.*

[10] U. Feige, "On the Success Probability of the Two Provers in One Round Proof Systems", *Structures 1991, pp. 116-123.*

[11] U. Feige, S. Goldwasser, L. Lovasz, M. Safra, M. Szegedy, "Approximating Clique is Almost NP-Complete", 32nd *FOCS, 1991, pp. 2-12.*

[12] L. Fortnow, "Complexity-Theoretic Aspects of Interactive Proof Systems", *Ph.D. Thesis, MIT/LCS/TR-447, 1989.*

[13] L. Fortnow, J. Rompel, M.Sipser, "On the Power of Multi-Prover Interactive Protocols", *Structures 1988, pp. 156-161.* Erratum in *Structures 1990, pp. 318-319.*

[14] M. Grötschel, L. Lovász, A. Schrijver, "Geometric Algorithms and Combinatorial Optimization", *Springer-Verlag, 1988.*

[15] D. Lapidot, A. Shamir, "A One-Round, Two-Prover, Zero-Knowledge Protocol for NP", *Crypto, 1991.*

[16] D. Lapidot, A. Shamir, "Fully Parallelized Multi Prover Protocols for NEXP-time" 32nd *FOCS, 1991, pp. 13-18.*

[17] L. Lovász, "On the Shanon Capacity of a Graph", *IEEE Trans. on Information Theory, Vol. 25, pp. 1-7, 1979.*

[18] L. Lovász, A. Schrijver, "Cones of Matrices and Setfunctions, and 0-1 Optimization", *1990.*

[19] J. Kilian, "Strong Separation Models of Multi Prover Interactive Proofs" *DIMACS Workshop on Cryptography, October 1990.*

[20] D. Peleg, "On the Maximal Number of Ones in Zero-One Matrices with No Forbidden Rectangles", *manuscript, 1990.*

[21] G. L. Peterson, J. H. Reif, "Multiple-Person Alternation", 20th *FOCS, 1979, pp. 348-363.*

[22] A. Shamir, "IP=PSPACE" 31st *FOCS, 1990, pp. 11-15.*

[23] D. Sherali, W. Adams, "A Hierarchy of Relaxations Between the Continuous and Convex Hull Representations for Zero-One Programming Problems", *preprint, 1988.*