ELSEVIER

Information Processing Letters 58 (1996) 135–140

Information Processing Letters

A note on *PCP* vs. *MIP*

Amnon Ta-Shma*

Institute of Computer Science, The Hebrew University, Givat Ram, 91904 Jerusalem, Israel

Received 10 May 1994; revised 20 July 1995
Communicated by S. Zaks

Abstract

Two variants of interactive proof systems have been used to derive intractability of approximation results. The first is the single-round multi-prover model where one verifier can query many provers who cannot communicate among themselves. The second is the oracle model where the verifier queries a non-adaptive oracle. It is known that the oracle model is at least as strong as the one-round multi-prover model but it is not known whether the opposite is true. We partially close this gap.

Keywords: Algorithms; Complexity; Interactive proof system

1. Introduction

1.1. Definitions

In the one-round multi-prover proof system of Ben-Or, Goldwasser, Killian and Wigderson [4], a probabilistic polynomial time verifier can query several all-mighty but possibly cheating provers, whose aim is to convince him to accept. The provers are limited in that they cannot communicate among themselves.

The oracle model of Fortnow, Rompel and Sipser [6], renamed "probabilistically checkable proofs" by Arora and Safra [1], is similar. In this model the provers are replaced by a non-adaptive oracle, i.e., instead of many provers who cannot communicate among themselves we have one oracle which cannot change its predetermined answers.

Bellare, Goldwasser, Lund and Russell [3] distinguish five important parameters and define the classes $PCP(r,m,b,q,\varepsilon)$ and $MIP_1(r,m,b,q,\varepsilon)$ as follows:

Definition 1.1.

$Lang \in PCP(r(n),m(n),b(n),q(n),\varepsilon(n))$

iff there is a probabilistic, polynomial time verifier V, communicating with an oracle, such that for every $x \in \{0,1\}^n$: V tosses $r(n)$ random bits and computes $m(n)$ questions, $q_1,\ldots,q_{m(n)}$, each of length $\leqslant q(n)$. V asks the oracle the $m(n)$ questions (V asks the questions one by one) and gets $m(n)$ answers $b_1,\ldots,b_{m(n)}$ each of length $\leqslant b(n)$. Then V computes a polynomial time predicate $V(x,r,q_1,\ldots,q_{m(n)},b_1,\ldots,b_{m(n)})$ and accepts iff it is 1. It should hold that:

if $x \in Lang$ then
$\exists O$ $\mathrm{Prob}_r(V^O \text{ accepts}) = 1$,
if $x \notin Lang$ then
$\forall O$ $\mathrm{Prob}_r(V^O \text{ accepts}) \leqslant \varepsilon(n)$,

* This work was supported by BSF grant 92-00043 and by a Wolfeson award administered by the Israeli Academy of Sciences.
Email: am@cs.huji.ac.il.

PII S0020-0190(96)00043-9

where V^O means the verifier V with access to the oracle O.

Analogously they define:

Definition 1.2.

$Lang \in MIP_1(r(n), m(n), b(n), q(n), \varepsilon(n))$

iff there is a probabilistic, polynomial time verifier V, communicating with $m(n)$ provers who cannot communicate among themselves, such that for every $x \in \{0,1\}^n$: V tosses $r(n)$ random bits and computes $m(n)$ questions, $q_1, \ldots, q_{m(n)}$, each of length $\leqslant q(n)$. V asks the ith prover the question q_i (V asks simultaneously all the questions) and gets $m(n)$ answers $b_1, \ldots, b_{m(n)}$ each of length $\leqslant b(n)$. Then V computes a polynomial time predicate $V(x, r, q_1, \ldots, q_{m(n)}, b_1, \ldots, b_{m(n)})$ and accepts iff it is 1. It should hold that:

if $x \in Lang$ then
  $\exists \bar{P} = P_1, \ldots, P_m$
  $\text{Prob}_r(V$ accepts when interacting with $\bar{P}) = 1$,
if $x \notin Lang$ then
  $\forall \bar{P} = P_1, \ldots, P_m$
  $\text{Prob}_r(V$ accepts when interacting with $\bar{P})$
    $\leqslant \varepsilon(n)$.

Notice, that in both cases, we consider the oblivious model where the verifier has to prepare all his questions before he gets any answer.

1.2. Previous work

It is simple to show that $MIP_1(r, m, b, q, \varepsilon) \subseteq PCP(r, m, b, q + \mathrm{O}(\log(m)), \varepsilon)$ [6]. However the converse is not immediate. The problem is that the *PCP* model is "safer" because the verifier knows that for the same question he will always get the same answer, which is not true in the MIP_1 model, where the answer may also depend on the identity of the prover.

Two simulation results in the opposite direction are implicit in Fortnow, Rompel and Sipser's work [6]:

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1\left(r, 2, m \cdot b, m \cdot q, 1 - \frac{1-\varepsilon}{m}\right),$$

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1\left(r + m \cdot \text{polylog}\left(\frac{m}{\varepsilon}\right), \mathrm{O}\left(m \cdot \log\left(\frac{m}{\varepsilon}\right)\right), b, q, 2\varepsilon\right).$$

Bellare, Goldreich and Safra [2] showed that the identity transformation (i.e., given a *PCP* system with m questions, build an *MIP* system with m provers by distributing the m questions among the m provers) amplifies the error by at most m^m. Thus:

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1(r, m, b, q, m^m \cdot \varepsilon).$$

For the sake of completeness, the proof appears in the appendix. This shows that for a constant m the two models are almost equivalent.

Bellare, Goldwasser, Lund and Russell [3] raised the question whether the two models can be related more closely, i.e., whether:

$$PCP(r, m, b, q, \varepsilon) \stackrel{?}{\subseteq} MIP_1(\mathrm{O}(r), \mathrm{O}(m), \mathrm{O}(b), \mathrm{O}(q), \mathrm{O}(\varepsilon)).$$

In this paper we get closer to the above target and we show that:

Corollary 1.3. *If* $q = \mathrm{O}(r)$ *and* $1/\varepsilon, m \leqslant \text{poly}(r)$ *then*:

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1(\mathrm{O}(r), m+1, \mathrm{O}(b \cdot r^2), \mathrm{O}(q), 3\varepsilon).$$

2. Forcing the same behavior on many provers

In this section we show how to force many provers to have the same behavior. We achieve this using the techniques of [5, 7]. We stress that the results of [5, 7] only refer to the *MIP* model and do not imply a *PCP* to *MIP* transformation.

We will use the following well-known lemma:

Lemma 2.1 (Low-degree extension). *Let* F *be a field,* $H_1, \ldots, H_m \subseteq F$, *and* $f : H_1 \times \cdots \times H_m \mapsto F$ *be any function. There is a unique multi-variate polynomial* $\tilde{f} : F^m \mapsto F$ *such that*

- $\forall y \in H_1 \times \cdots \times H_m \ \tilde{f}(y) = f(y)$.
- *The degree of* $\tilde{f}$ *as a polynomial in the ith variable* $d_i(\tilde{f})$ *satisfies* $d_i(\tilde{f}) \leqslant |H_i| - 1$.

Theorem 2.2. *Let h, l, f, ε be positive integers such that $h^l \geq 2^q$ and $f \geq 2m \cdot l \cdot h/\varepsilon^2$, then:*

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1(r + l \cdot \log(f), m + 1, h \cdot b \cdot l \cdot \log(f), 2l \cdot \log(f), 3\varepsilon).$$

Proof. Let $Lang \in PCP(r, m, b, q, \varepsilon)$ be accepted by a verifier V and an oracle P.

We build a new proof system with a verifier $\hat{V}$ and $m + 1$ provers $\hat{P}_0, \ldots, \hat{P}_m$.

Structure of proof: First we formally describe the algorithm, i.e., what the verifier asks and what he does with the answers. While describing the algorithm, we develop some notation. Then we study the case $x \in Lang$ and show what "honest" provers should do. Finally, we show that when $x \notin Lang$, no matter what the provers do, the verifier rejects with high enough probability.

The algorithm: $\hat{V}$ tosses $\bar{r} \in_R \{0,1\}^r$ and simulates V to find the m questions V would have asked. Denote them by $q_1, \ldots, q_m$.

Next, we represent the questions as taken from some domain H^l and we extend them to a larger field F. Denote $H = [h]$ and represent each q_i as being in H^l (this is possible because $h^l \geq 2^q$). Choose a prime $m \cdot l \cdot h/\varepsilon^2 \leq p \leq f$ (this is possible because $f \geq 2m \cdot l \cdot h/\varepsilon^2$), denote $F = GF(p)$, and view each q_i as residing in F^l.

Choose an element $y \in_R F^l$. If for some i, $y = q_i$ accept (this is a very rare case and the reader can ignore it).

Notation 2.3 (*Representing lines*). A line L is a set of p points in F^l such that there is a function $L : F \mapsto F^l$ linear in each of its coordinates, passing through those points. Any two different points s_1, s_2 in F^l determine a unique line $L(s_1, s_2)$, however this line has many different linear representations. We want a representation which does not reveal s_2. One such representation can be the function $L(t) = t \cdot s_1 + (1-t) \cdot x_1$ where x_1 is the lexicographically first point on the line $L(s_1, s_2)$ different from s_1. Given the points s_1, s_2 it is easy to find the point x_1 in polynomial time, and for any point $z \neq s_1$ on the line $L(s_1, s_2)$, the lines $L(s_1, s_2)$ and $L(s_1, z)$ have the same representation. From now on we will denote by $L(a, b)$ both the line passing through a and b, and its representation. A similar representation is given in [5].

Now $\hat{V}$ sends:

- y to $\hat{P}_0$,
- $L(y, q_i)$ to $\hat{P}_i$ $(i = 1, \ldots, m)$.

In response he should get:

- $\hat{P}_0$ answers with an element in F^b,
- each prover $\hat{P}_i$ $(i = 1, \ldots, m)$ answers with b polynomials $\hat{P_{i,j}} : F \mapsto F$ $(j = 1, \ldots, b)$ of degree $d \leq l \cdot (h-1)$.

Notation 2.4. Let $L = L(a, b)$ be a line represented by the function $L : F \mapsto F^l$, and let z be a point on that line, i.e., for some $t \in F$ we have that $L(t) = z$. Denote by $\hat{P}_i(L(a, b))$ the sequence of b polynomials provided by $\hat{P}_i$ on question $L(a, b)$. Denote by $\hat{P}_i(L(a, b))(z)$ the value of these b polynomials at the point z, i.e., $\hat{P}_i(L(a, b))(z) = (\hat{P_{i,1}}(L(a, b))(t), \ldots, \hat{P_{i,b}}(L(a, b))(t))$.

$\hat{V}$ checks that:

- $\forall i \in \{1, \ldots, n\}$ $\hat{P}_i(L(y, q_i))(y) = \hat{P}_0(y)$ where equality is in F^b.
- Let $b_i = \hat{P}_i(L(y, q_i))(q_i) \in F^b$. $\hat{V}$ checks that $b_i \in \{0,1\}^b$ and that $V(\bar{r}, b_1, \ldots, b_m)$ accepts.

If $x \in Lang$:

Since $x \in Lang$ there is an oracle $O : H^l \mapsto \{0,1\}^b$ such that $\text{Prob}_r(V^O \text{ accepts}) = 1$. We can view O as b different functions $O_i : H^l \mapsto \{0,1\}$. Each O_i has a (unique) extension $\tilde{O}_i : F^l \mapsto F$ with degree $\leq h - 1$ in each of its variables (Lemma 2.1).

Hence if L is a line in F^l, i.e., $L : F \mapsto F^l$ then $\tilde{O}_i(L(t))$ is a polynomial (in one variable) of degree $\leq l \cdot (h-1)$.

Now let us define the provers' strategies by:

- $\hat{P}_0$ answers with $\tilde{O}_1(y), \ldots, \tilde{O}_b(y)$,
- $\hat{P_{i,j}}(L(t)) = \tilde{O}_j(L(t))$.

Then, it is immediate that:

$$\hat{P_{i,j}}(L(y, q_i))(y) = \tilde{O}_j(y) = \hat{P_{0,j}}(y)$$

and therefore

$$\hat{P}_i(L(y, q_i))(y) = \tilde{O}(y) = \hat{P}_0(y)$$

and the verifier's first check is satisfied.

Since $q_i \in H^l$ and $\tilde{O}$ extends O,

$$b_i = \hat{P}_i(L(y, q_i))(q_i) = \tilde{O}(q_i) = O(q_i),$$

and therefore $b_i \in \{0,1\}^b$.

Finally, as $\hat{V}$ gets the same answers to his questions as V does, and V always accepts, $\hat{V}$ will also approve the second check, and accept.

In short, we saw that if the provers imitate the (extended) oracle, then since the oracle makes the verifier always accept, the provers too make the verifier always accept.

If $x \notin Lang$:

The main idea is the following: we want to force the different provers to have the same behavior. We can do that by asking the same question to different provers, but then we need to choose randomly which prover to ask what question, and we pay in randomness. We overcome this using the nice idea of [7] of handling *strategies*. We compare $\hat{P}_i$'s strategy to $\hat{P}_0$'s strategy. Provers who are not consistent among themselves cannot be (simultaneously) consistent with $\hat{P}_0$. More precisely, suppose $\hat{V}$ wants to ask q and chooses y as his checking point. Two provers succeed in fooling $\hat{V}$ if they differ on q but agree with $\hat{P}_0$ on y. We use the fact that two low-degree polynomials either agree on all points or disagree on most of them, to show the verifier is rarely fooled. Thus, either the verifier rejects with a very high probability, or else the provers are consistent among themselves and there is a convincing oracle.

Formally, let $\hat{P}_0, \ldots, \hat{P}_m$ be optimal provers for x. Suppose, to the contradiction, that:

$$\mathrm{Prob}_{y,\bar{r}}(\hat{V}, \hat{P}_0, \ldots, \hat{P}_m) > 3\varepsilon. \tag{1}$$

Definition 2.5. We say that y is bad for q ($y \in F^l$, $q \in H^l$) if there are $i \neq j$ such that
(1) $\hat{P}_i(L(q,y))(q) \neq \hat{P}_j(L(q,y))(q)$ and
(2) $\hat{P}_i(L(q,y))(y) = \hat{P}_j(L(q,y))(y) = \hat{P}_0(y)$.
We say y is good, if it is not bad.

Definition 2.6. We say that y is bad for $\bar{r} \in \{0,1\}^r$ if V with the random string $\bar{r}$ would have asked the questions $q_1, \ldots, q_m$ and y is bad for some q_i.

Claim 2.7. $\forall q\ \mathrm{Prob}_{y \in F^l}(y \textit{ is bad for } q) \leqslant l \cdot h/|F|$.

Proof. Let $L = L(q, y_0)$ be a line passing through q. Suppose there is a point z on the line L (we denote it by $z \in L$) which is bad for q. Therefore, there is $1 \leqslant k \leqslant b$ and $i \neq j$ such that $\hat{P}_{i,k}, \hat{P}_{j,k}$ differ on q. Since $\hat{P}_{i,k}, \hat{P}_{j,k}$ are different polynomials of degree $\leqslant l \cdot (h-1)$ we have:

$$\mathrm{Prob}_{y \in L}(\hat{P}_{i,k}(L)(y) = \hat{P}_{j,k}(L)(y)) \leqslant \frac{l \cdot (h-1)}{|F|}.$$

By the second requirement of Definition 2.5:

$$\mathrm{Prob}_{y \in L}(y \text{ is bad for } q) \leqslant \mathrm{Prob}_{y \in L}(\hat{P}_i(L)(y) = \hat{P}_j(L)(y))$$

and thus, for every line L passing through q:

$$\mathrm{Prob}_{y \in L}(y \text{ is bad for } q) \leqslant \frac{l \cdot (h-1)}{|F|}.$$

Since the lines passing through q partition F^l, we get the desired result. □

Claim 2.8.

$$\mathrm{Prob}_{y \in F^l}(\mathrm{Prob}_{\bar{r} \in \{0,1\}^r}(y \textit{ is bad for } \bar{r}) \geqslant \varepsilon) \leqslant \varepsilon.$$

Proof. For any $\bar{r} \in \{0,1\}^r$:

$$\mathrm{Prob}_{y \in F^l}(y \text{ is bad for } \bar{r}) = \mathrm{Prob}_{y \in F^l}(\exists i\ y \text{ is bad for the } i\text{th query of } \bar{r}) < \frac{m \cdot l \cdot h}{|F|} \leqslant \varepsilon^2.$$

The first inequality by Claim 2.7, and the second by choice of F.

Therefore:

$$\mathrm{Prob}_{y \in F^l, \bar{r} \in \{0,1\}^r}(y \text{ is bad for } \bar{r}) \leqslant \varepsilon^2$$

and the claim follows, by an averaging argument. □

On the other hand Eq. (1) implies by an averaging argument that:

$$\mathrm{Prob}_{y \in F^l}(\mathrm{Prob}_{\bar{r} \in \{0,1\}^r}((\hat{V}, \hat{P}_0, \ldots, \hat{P}_m) \text{ accept}) > 2\varepsilon) > \varepsilon. \tag{2}$$

Therefore combining Claim 2.8 and Eq. (2) together we see that there is some $y \in F^l$ such that
(1) $\mathrm{Prob}_{\bar{r} \in \{0,1\}^r}(y \text{ is bad for } \bar{r}) < \varepsilon$,
(2) if $\hat{V}$ chose y then

$$\mathrm{Prob}_{\bar{r} \in \{0,1\}^r}((\hat{V}, \hat{P}_0, \ldots, \hat{P}_m) \text{ accept}) > 2\varepsilon.$$

From now on we fix this y and call it y_0.

Definition 2.9. Let R_V be the set of $\bar{r} \in \{0,1\}^r$ such that y_0 is good for $\bar{r}$ and $(\hat{V}, \hat{P}_0, \ldots, \hat{P}_m)$ accept with $(\bar{r}, y_0)$.

Claim 2.10. $\text{Prob}_{\bar{r}\in\{0,1\}^r}(\bar{r} \in R_V) > \varepsilon$.

Proof.

$$\begin{aligned}&\text{Prob}_{\bar{r}\in\{0,1\}^r}(\bar{r} \in R_V)\\ &\quad \geqslant \text{Prob}_{\bar{r}\in\{0,1\}^r}((\hat{V}, \hat{P}_0, \ldots, \hat{P}_m) \text{ accept})\\ &\qquad - \text{Prob}_{\bar{r}\in\{0,1\}^r}(y_0 \text{ is bad for } \bar{r})\\ &\quad > 2\varepsilon - \varepsilon = \varepsilon. \qquad \square\end{aligned}$$

Now we define an oracle $O' : H^l \mapsto \{0,1\}^b$ as follows:

$$O'(q) = \begin{cases} \hat{P}_i(L(q, y_0))(q), \\ \qquad \text{if there is an } \bar{r} \in R_V \text{ such that} \\ \qquad \text{for that } \bar{r}\ V \text{ asks the questions} \\ \qquad q_1, \ldots, q_m \text{ and } q_i = q, \\ 0^b, \quad \text{otherwise.} \end{cases}$$

Claim 2.11. *O' is well defined.*

Proof. Let $\bar{r}_1 \in R_V$ with questions $q_{1,1}, \ldots, q_{1,m}$ and $q_{1,i} = q$, and $\bar{r}_2 \in R_V$ with questions $q_{2,1}, \ldots, q_{2,m}$ and $q_{2,j} = q$. Since $\bar{r}_1, \bar{r}_2 \in R_V$, they make $\hat{V}$ accept, hence: $\hat{P}_i(L(q, y_0))(y_0) = \hat{P}_0(y_0) = \hat{P}_j(L(q, y_0))(y_0)$.

Since y_0 is good for $\bar{r}_1$ ($\bar{r}_1 \in R_V$), y_0 is good for q and therefore by the above and Definition 2.5, $\hat{P}_i(L(q, y_0))(q) = \hat{P}_j(L(q, y_0))(q)$. Thus, O' is well defined. $\square$

Claim 2.12. *For any $\bar{r} \in R_V$, $\hat{V}$ with access to the oracle O' and the random string $\bar{r}$ accepts.*

Proof. Let $\bar{r} \in R_V$, and let $q_1, \ldots, q_m$ be the m questions $\hat{V}$ generates with $\bar{r}$. $\hat{V}$ gets answers according to the oracle O', and since $\bar{r} \in R_v$ then by definition of O', for any $1 \leqslant i \leqslant m$, $O'(q_i) = \hat{P}_i(L(q_i, y_0))(q_i)$. However, by the definition of R_V, $(\hat{V}, \hat{P}_0, \ldots, \hat{P}_m)$ accept with $(\bar{r}, y_0)$. Therefore, $\hat{V}$ accepts with $\bar{r}$. $\square$

Thus:

$$\begin{aligned}&\text{Prob}_{\bar{r}\in\{0,1\}^r}((V, O') \text{ accept})\\ &\quad \geqslant \text{Prob}_{\bar{r}\in\{0,1\}^r}(r \in R_V) > \varepsilon\end{aligned}$$

and this is a contradiction, as $x \notin \mathit{Lang}$. Thus assumption (1) is false, and our new proof system $(\hat{V}, \hat{P}_0, \ldots, \hat{P}_m)$ has at most 3ε error as we wanted to show. $\square$

Proof of Corollary 1.3. Taking $h = \Theta(r)$ and $l = \Theta(r/\log(r))$ guarantees that $h^l \geqslant 2^q$. Take $f = 2m \cdot l \cdot h/\varepsilon^2$. This implies that $\log(f) = O(\log(r))$. Plugging these into Theorem 2.2 we get the corollary. $\square$

The extra prover we are using is not essential, as:

Theorem 2.13. *Let h, l, f, k, ε be positive integers such that $h^l \geqslant 2^q$, $f \geqslant 2l \cdot h \cdot m^3 \cdot b^3/\varepsilon^4$ and $2 \leqslant k \leqslant m$, then:*

$$\begin{aligned}&MIP_1(r, m, b, q, \varepsilon)\\ &\quad \subseteq MIP_1(r + k \cdot l \cdot \log(f), \lceil m/k + 1 \rceil,\\ &\qquad O(k \cdot h \cdot b \cdot l \cdot \log(f)),\\ &\qquad O(k \cdot l \cdot \log(f)), 4\varepsilon).\end{aligned}$$

Proof (*Sketch*). A small variation of the proof in [5,7]. The main point is that we can choose k independent points $y_1, \ldots, y_k \in F^l$ and ask each prover k questions. Suppose, in the $MIP_1(r, m, b, q, \varepsilon)$ system we ask the questions $q_1, \ldots, q_m$. In the new MIP system we send the lines $L(q_{k(i-1)+1}, y_1)$, $L(q_{k(i-1)+2}, y_2), \ldots, L(q_{ki}, y_k)$ to the prover $\boldsymbol{P}_i$, and we send $y_1, \ldots, y_k$ to $\boldsymbol{P}_0$. Then we verify that each $\boldsymbol{P}_i$ is consistent with $\boldsymbol{P}_0$ on $y_1, \ldots, y_k$ and we compute the answer according to the answers to $q_1, \ldots, q_m$. As long as no prover receives two lines passing through the same point, no prover can gain any information on what these points are, and the analysis of [5,7] is still valid. Thus, when choosing k points we need $m/k + 1$ provers. Notice that in [5,7] m points are used, and therefore two provers suffice. $\square$

Acknowledgements

I would like to thank Ilan Kremer, Muli Safra, Dror Lapidot and Shafi Goldwasser for many interesting discussions. I want to thank the anonymous referees

for many helpful comments which significantly clarified the presentation, and for drawing my attention to the result by [2]. Most of all, it is a great pleasure to thank my advisor Noam Nisan.

Appendix A

In this section we describe the simulation result by Bellare, Goldreich and Safra, showing that:

Proposition A.1 (Bellare, Goldreich, Safra [2]).

$$PCP(r, m, b, q, \varepsilon) \subseteq MIP_1(r, m, b, q, m^m \cdot \varepsilon).$$

Proof (*Bellare, Goldreich, Safra* [2]). Let $Lang \in PCP(r, m, b, q, \varepsilon)$, accepted by a verifier V and the oracle O, and let $x \in \{0,1\}^n$. We build a new proof system with a verifier $\hat{V}$ and m provers $\hat{P}_1, \ldots, \hat{P}_m$.

The algorithm:

$\hat{V}$ tosses $\bar{r} \in_R \{0,1\}^r$ and simulates V to find the m questions V would have asked. Denote them by $q_{\bar{r},1}, \ldots, q_{\bar{r},m}$. Then $\hat{V}$ asks $\hat{P}_i$ the question $q_{\bar{r},i}$, gets the answers, and accepts iff V with these answers accepts.

$x \in L$: Let each $\hat{P}_i$ answer the same way O does. It is clear that the probability $\hat{V}$ accepts is 1.

$x \notin L$: Suppose there are provers $\hat{P}_1, \ldots, \hat{P}_m$ such that $\text{Prob}_{\bar{r}}(\hat{V} \text{ accepts}) = \varepsilon'$.

Define a random oracle O' in the following way: for every question q, randomly pick $i \in \{1, \ldots, m\}$, and we let $O'(q) = \hat{P}_i(q)$.

Then:

$$\begin{aligned}
&\text{Prob}_{O',\bar{r}}(V \text{ accepts with } \bar{r} \text{ and the oracle } O') \\
&\quad \geqslant \text{Prob}_{\bar{r}}(\hat{V} \text{ accepts with } \bar{r} \text{ and } \hat{P}_1, \ldots, \hat{P}_m) \cdot \\
&\qquad \text{Prob}_{\bar{r},O'}(\forall 1 \leqslant i \leqslant m \; O'(q_{\bar{r},i}) = \hat{P}_i(q_{\bar{r},i})) \\
&\quad \geqslant \varepsilon' \cdot \left(\frac{1}{m}\right)^m.
\end{aligned}$$

The last inequality holds, since w.l.o.g. all the questions $q_{\bar{r},1}, \ldots, q_{\bar{r},m}$ are different (we started with a *PCP* system) and therefore the events "$O'(q_{\bar{r},i}) = \hat{P}_i(q_{\bar{r},i})$" are independent.

Therefore, there is some oracle achieving this error rate $\varepsilon' \cdot (1/m)^m$. However, since we started with a *PCP* system with ε error, it means that $\varepsilon' \leqslant \varepsilon \cdot m^m$, and the proposition follows. □

References

[1] S. Arora and S. Safra, Probabilistic checking of proofs; a new characterization of NP, in: *Proc. 33rd Ann. IEEE Symp. on the Foundations of Computer Science* (1992) 2-13.

[2] M. Bellare, O. Goldreich and S. Safra, Private communication.

[3] M. Bellare, S. Goldwasser, C. Lund and A. Russell, Efficient probabilistically checkable proofs and applications to approximation, in: *Proc. 25th Ann. ACM Symp. on the Theory of Computing* (1993) 294-304.

[4] M. Ben-Or, S. Goldwasser, J. Killian and A. Wigderson, Multi-prover interactive proofs: how to remove intractability assumptions, in: *Proc. 20th Ann. ACM Symp. on the Theory of Computing* (1988) 113-131.

[5] U. Feige and L. Lovasz, Two-prover one round proof systems: their power and their problems, in: *Proc. 24th Ann. ACM Symp. on the Theory of Computing* (1992).

[6] L. Fortnow, J. Rompel and M. Sipser, On the power of multi-prover interactive protocols, in: *Proc. 3rd Structures Theory of Computing* (1988).

[7] D. Lapidot and A. Shamir, Fully parallelized multi-prover protocols for NEXPtime, in: *Proc. 32rd Ann. IEEE Symp. on the Foundations of Computer Science* (1991) 13-18.