Electronic Colloquium on Computational Complexity, Report No. 170 (2014)

On the Space Complexity of Linear Programming with Preprocessing

Yael Tauman Kalai * Ran Raz †

Abstract

Linear Programs are abundant in practice, and tremendous effort has been put into designing efficient algorithms for such problems, resulting with very efficient (polynomial time) algorithms. A fundamental question is: what is the *space complexity* of Linear Programming?

It is widely believed that (even approximating) Linear Programming requires a large space. Specifically, it was shown that (approximating) Linear Programming is P complete with a log-space reduction, thus showing that $n^{o(1)}$-space algorithms for (approximating) Linear Programming are unlikely.

We show that (approximating) Linear Programming is likely to have a large space complexity, even if we allow a preprocessing phase that takes the polyhedron as input and runs in unbounded time and space. Specifically, we prove that (approximating) Linear Programming with such "preprocessing" is P complete with a polylog space and quasi-poly time reduction, thus showing that $2^{(\log n)^{o(1)}}$-space algorithms for Linear Programming with "preprocessing" are unlikely.

We obtain our result using a recent work of Kalai, Raz and Rothblum, showing that every language in P has a no-signalling multi-prover interactive proof with poly-logarithmic communication complexity. To the best of our knowledge, this is the first space hardness of approximation result proved by a PCP based argument.

1 Introduction

Linear Programs often arise in practice, and algorithms for Linear Programming are widely deployed. There has been a major effort to construct fast algorithms for Linear Programming, resulting with very efficient algorithms (e.g., [Dan51, Kha79, Kar84, DV04, BV04, KS06]).

*Microsoft Research. Email: yael@microsoft.com

†Weizmann Institute of Science, Israel, and the Institute for Advanced Study, Princeton, NJ. Research supported by the Israel Science Foundation grant No. 1402/14, by the I-CORE Program of the Planning and Budgeting Committee and the Israel Science Foundation, by the Simons Foundation, by the Fund for Math at IAS, and by the National Science Foundation grant No. CCF-1412958. Email: ran.raz.mail@gmail.com

ISSN 1433-8092

Recall that a linear program is a constrained optimization problem of the form:

$$\begin{aligned} &\text{maximize } c \cdot x \\ &\text{subject to } Ax \leq b; x \in \mathbb{R}^d \end{aligned}$$

where $c \in \mathbb{R}^d$ and $b \in \mathbb{R}^n$ are column vectors, and A is an $n \times d$ matrix. The vector c is the objective function, and the set $H = \{x : A \cdot x \leq b\}$ is the set of feasible points. If it is non-empty, H is a convex polyhedron.

1.1 The Space Complexity of Linear Programming

The space complexity of Linear Programming was first studied in the late 70's. Dobkin, Lipton and Reiss [DLR79], followed by a work of Serna [Ser91], proved that approximating Linear Programming is P-complete with a log-space reduction, thus proving that if Linear Programming has a log-space algorithm, then any language $L \in \mathsf{P}$ would have a log-space algorithm.[1]

For the special case of positive Linear Programming (where all the coefficients are positive), Luby and Nisan [LN93] gave a fast parallel approximation algorithm that runs in poly-logarithmic time using a linear number of processors.

1.2 Our Results

We consider the problem of Linear Programming *with preprocessing*, where the algorithm runs in two phases. In the first phase, the algorithm is given the polyhedron and it may run in unbounded time and space. In the second phase, the algorithm is given the objective function and it gives an output. We only measure the time and space complexity of the second phase of the algorithm. We show that even in this (seemingly easier) setting, it is unlikely that there exist small-space Linear Programming algorithms.

Specifically, we show that (approximating) Linear Programming with such "preprocessing" is P complete with a polylog space and quasi-poly time reduction, thus showing that a $2^{(\log n)^{o(1)}}$-space algorithm for Linear Programming with "preprocessing" is unlikely.

To this end, we show that for every language $L \in \mathsf{P}$, any instance $x \in \{0,1\}^n$ can be reduced to a Linear Program, with a *fixed* polyhedron that depends on n but otherwise is independent of x,[2] and only the objective function depends on x. The reduction runs in quasi-poly time and polylog space. Moreover, the resulting Linear Program has the property that if x is in the language L then the maximal value of the objective function on the polyhedron is 1, whereas if x is not in the language L then the maximal value of the objective function on the polyhedron is less than $2^{-\mathsf{polylog}(n)}$.

[1] Feige and Kilian [FK97] gave an alternative proof of this fact.

[2] The polyhedron also does not depend on the language L, only on its runtime.

Our reduction uses the recent result of [KRR14], that shows that any language in P has a multi-prover interactive proof with polylog communication complexity, which is secure against *no-signaling* cheating provers.

1.3 Multi-Prover Interactive Proofs with No-Signaling Provers

Multi-prover interactive proofs (MIPs) were introduced by [BGKW88]. In such a proof system a set of provers wish to convince a verifier of the validity of a statement. The verifier sends each prover a query and each prover responds with an answer. An MIP has the guarantee that if the statement is valid then the (honest) provers will convince the verifier to accept with probability 1. On the other hand, if the statement is invalid then any set of cheating provers will convince the verifier to accept only with negligible probability, assuming they do not interact, and each prover sees only its own query (and does not see any of the other queries). Babai, Fortnow and Lund [BFL90] showed that any language in NEXP has an MIP (where the verifier runs in polynomial time).

The study of multi-prover interactive proofs (MIPs) that are secure against *no-signaling* provers was motivated by the study of MIPs with provers that share entangled quantum states. No-signaling provers are more powerful than classical as well as quantum provers. Loosely speaking, no-signaling provers are allowed to use arbitrary strategies, as long as their strategies cannot be used for communication between any two disjoint sets of provers.[3] More specifically, in a no-signaling strategy the answer given by each prover is allowed to depend on the queries to all other provers, as long as for any subset of provers S, and any queries given to the provers in S, the distribution of the answers given by the provers in S is independent of all the other queries. In particular, the answer of each prover can depend on the queries to all other provers as a function, but not as a random variable.

More formally, fix any MIP consisting of k provers, and fix any set of cheating provers $\{P_1^*, \ldots, P_k^*\}$ who may see each other's queries (and thus each answer may depend on the queries sent to all the provers). The provers are said to be *no-signaling* if for every subset of provers $\{P_i^*\}_{i \in S}$, and for every two possible query sets $\{q_i\}_{i \in [k]}$ and $\{q'_i\}_{i \in [k]}$ such that $q_i = q'_i$ for every $i \in S$, it holds that the distributions of answers $\{a_i\}_{i \in S}$ and $\{a'_i\}_{i \in S}$ are *identical*, where $\{a_i\}_{i \in S}$ is the the answers of the provers in S corresponding to the queries $\{q_i\}_{i \in [k]}$, and $\{a'_i\}_{i \in S}$ is the answers of the provers in S corresponding to the queries $\{q'_i\}_{i \in [k]}$.

No-signaling strategies were first studied in physics in the context of Bell inequalities by Khalfin and Tsirelson [KT85] and Rastall [Ras85], and they gained much attention after they were reintroduced by Popescu and Rohrlich [PR94]. MIPs that are secure against no-signaling provers were extensively studied in the literature (see for example [Ton09, BLM+05, AII06, KKM+08, IKM09, Hol09, Ito10]). It was known that they are contained in EXP, and recently [KRR14] showed that they also contain EXP, thus giving a full characterization of their exact power.

[3] By the physical principle that information cannot travel faster than light, a consequence of Einstein's special relativity theory, it follows that all the strategies that can be realized by provers that share entangled quantum states are no-signaling strategies.

Informal Theorem 1. *[KRR14] For any language L computable in time $t = t(n)$, there exists an* MIP *that is secure against no-signaling cheating provers. The number of provers and the communication complexity is* $\mathsf{polylog}(t)$*. The verifier runs in time* $n \cdot \mathsf{polylog}(t)$ *(and the provers run in time* $\mathsf{poly}(t)$*). Moreover, the verifier only runs in time* $\mathsf{polylog}(t)$ *if he is given oracle access to a (specific) encoding of x,*[4] *where each entry of the encoding can be computed from x in time $\tilde{O}(n)$ and space $O(\log n)$.*

In this work, we use this theorem for languages in P. Note that this theorem implies that for languages in P the verifier runs in $\tilde{O}(n)$ time and in $\mathsf{polylog}(n)$ space. Thus, we restate the theorem as follows.

Theorem 2. *[KRR14] If $L \in \mathsf{P}$, then there exists an* MIP *for L with* $\mathsf{polylog}(n)$ *provers, and with soundness error* $2^{-\mathsf{polylog}(n)}$ *against no-signaling strategies. The verifier runs in time $\tilde{O}(n)$ and space* $\mathsf{polylog}(n)$ *(and the provers run in polynomial time). Each query and answer is of length* $\mathsf{polylog}(n)$*.*

1.4 Our Results in More Detail

We use Theorem 2 to show a reduction from any language $L \in \mathsf{P}$ to a Linear Program. Our reduction runs in quasi-poly time and polylog space. In particular, our reduction takes an instance of size n and converts it into a linear program of size quasi-polynomial in n, where the polyhedron is on quasi-polynomial number of variables (i.e., quasi-polynomial dimensions). This polyhedron is fixed, independent of the instance x (and depends only on its size $n = |x|$).[5]

Our Main Theorem. *There exists a fixed family of polyhedrons $H = \{H_t\}_{t \in \mathbb{N}}$ such that the following holds: For every language $L \in \mathsf{P}$ computable by a Turing Machine with runtime $t = t(n)$, there exists a* polylog *space and* quasi-poly *time reduction, that converts any instance $x \in \{0,1\}^n$ into a Linear Program with the polyhedron $H_{t(n)}$ (and an objective function that depends on x), such that if $x \in L$ then the maximum value of the objective function on the polyhedron is 1, and if $x \notin L$ then the maximum value of the objective function on the polyhedron is smaller than* $2^{-\mathsf{polylog}(n)}$*.*

2 Preliminaries

2.1 Notation

For a vector $a = (a_1, \ldots, a_k)$ and a subset $S \subseteq [k]$, we denote by a_S the sequence of elements of a that are indexed by indices in S, that is, $a_S = (a_i)_{i \in S}$. In general, we denote by a_S a sequence of elements indexed by S, and we denote by a_i the i^{th} coordinate of a vector a.

[4] This encoding is the low-degree extension encoding. We refer the reader to [KRR14] for details.

[5] The polyhedron is also independent of the language L, and depends only on its time complexity.

For a distribution $\mathcal{A}$, we denote by $a \in_R \mathcal{A}$ a random variable distributed according to $\mathcal{A}$ (independently of all other random variables).

2.2 Multi-Prover Interactive Proofs

Let L be a language and let x be an input of length n. In a one-round k-prover interactive proof, k computationally unbounded provers, $P_1, \ldots, P_k$, try to convince a (probabilistic) $\mathsf{poly}(n)$-time verifier, V, that $x \in L$. The input x is known to all parties.

The proof consists of only one round. Given x and her random string, the verifier generates k queries, $q_1, \ldots, q_k$, one for each prover, and sends them to the k provers. Each prover responds with an answer that depends only on her own individual query. That is, the provers respond with answers $a_1, \ldots, a_k$, where for every i we have $a_i = P_i(q_i)$. Finally, the verifier decides wether to accept or reject based on the answers that she receives (as well as the input x and her random string).

We say that $(V, P_1, \ldots, P_k)$ is a one-round multi-prover interactive proof system (MIP) for L if the following two properties are satisfied:

1. **Completeness:** For every $x \in L$, the verifier V accepts with probability 1, after interacting with $P_1, \ldots, P_k$.
2. **Soundness:** For every $x \notin L$, and any (computationally unbounded, possibly cheating) provers $P_1^*, \ldots, P_k^*$, the verifier V rejects with probability $\geq 1-\epsilon$, after interacting with $P_1^*, \ldots, P_k^*$, where ϵ is a parameter referred to as the *error* or *soundness* of the proof system.

Important parameters of an MIP are the number of provers, the length of queries, the length of answers, and the error.

2.3 No-Signaling MIPs

We consider a variant of the MIP model, where the cheating provers are more powerful. In the MIP model, each prover answers her own query locally, without knowing the queries that were sent to the other provers. The no-signaling model allows each answer to depend on all the queries, as long as for any subset $S \subset [k]$, and any queries q_S for the provers in S, the distribution of the answers a_S, conditioned on the queries q_S, is independent of all the other queries.

Intuitively, this means that the answers a_S do not give the provers in S information about the queries of the provers outside S, except for information that they already have by seeing the queries q_S.

Formally, denote by D the alphabet of the queries and denote by Σ the alphabet of the answers. For every $q = (q_1, \ldots, q_k) \in D^k$, let $\mathcal{A}_q$ be a distribution over Σ^k. We think of $\mathcal{A}_q$ as the distribution of the answers for queries q.

We say that the family of distributions $\{\mathcal{A}_q\}_{q\in D^k}$ is *no-signaling* if for every subset $S \subset [k]$ and every two sequences of queries $q, q' \in D^k$, such that $q_S = q'_S$, the following two random variables are identically distributed:

- a_S, where $a \in_R \mathcal{A}_q$
- a'_S where $a' \in_R \mathcal{A}_{q'}$

An MIP, $(V, P_1, \ldots, P_k)$ for a language L is said to have soundness ϵ against no-signaling strategies (or provers) if the following (more general) soundness property is satisfied:

2. **Soundness:** For every $x \notin L$, and any no-signaling family of distributions $\{\mathcal{A}_q\}_{q\in D^k}$, the verifier V rejects with probability $\geq 1 - \epsilon$, where on queries $q = (q_1, \ldots, q_k)$ the answers are given by $(a_1, \ldots, a_k) \in_R \mathcal{A}_q$, and ϵ is the error parameter.

3 Our Main Result

In this section we prove our main result.

Theorem 3. *There exists a fixed family of polyhedrons $H = \{H_t\}_{t\in\mathbb{N}}$ such that the following holds: For every language $L \in \mathsf{P}$ computable by a Turing Machine with runtime $t = t(n)$, there exists a* polylog *space and* quasi-poly *time reduction, that converts any instance $x \in \{0,1\}^n$ into a Linear Program with the polyhedron $H_{t(n)}$ (and an objective function that depends on x), such that if $x \in L$ then the maximum value of the objective function on the polyhedron is 1, and if $x \notin L$ then the maximum value of the objective function on the polyhedron is smaller than $2^{-\mathsf{polylog}(n)}$.*

Proof. Let L be any language in P. By Theorem 2, the language L has an MIP, $(V, P_1, \ldots, P_k)$, where $k = \mathsf{polylog}(n)$, with communication complexity $\mathsf{polylog}(n)$ and soundness $2^{-\mathsf{polylog}(n)}$ against no-signaling provers (where n is the instance size).

We define a reduction $\mathcal{R}$ that takes as input an instance $x \in \{0,1\}^n$ and converts it into a Linear Program, as follows: Consider all possible no-signaling families of distributions of cheating provers in the MIP. For each such possible no-signaling family of distributions $\{\mathcal{A}_q\}_{q\in D^k}$, denote by

$$p_{q,a} = \Pr_{A\in_R\mathcal{A}_q}[A = a].$$

Note that $\{\mathcal{A}_q\}_{q\in D^k}$ is a no-signaling family of distributions if and only if the following conditions are satisfied (the first two conditions hold if and only if each $\mathcal{A}_q$ is a distribution, and the last condition holds if and only if these distributions are no-signaling):

1. For every $q = (q_1, \ldots, q_k) \in D^k$ and for every $a \in \Sigma^k$,

$$p_{q,a} \geq 0.$$

2. For every $q = (q_1, \ldots, q_k) \in D^k$,
$$\sum_{a \in \Sigma^k} p_{q,a} = 1.$$

3. For every $S \subseteq [k]$, for every $q = (q_1, \ldots, q_k) \in D^k$ and $q' = (q'_1, \ldots, q'_k) \in D^k$ for which $q_S = q'_S$, and for every $a_S \in \Sigma^S$, it holds that
$$\sum_{a' : a'_S = a_S} p_{q,a'} = \sum_{a' : a'_S = a_S} p_{q',a'}.$$

Denote by p_q the probability that V sends the provers queries $q = (q_1, \ldots, q_k) \in \mathcal{D}^k$. The fact that $(V, P_1, \ldots, P_k)$ is an MIP that is secure against no-signaling strategies (with soundness $2^{-\mathsf{polylog}(n)}$ and perfect completeness), implies that if $x \notin L$ then
$$\sum_q p_q \sum_{a : V(x,q,a)=1} p_{q,a} \leq 2^{-\mathsf{polylog}(n)},$$
and if $x \in L$ then there exists a (classical) strategy for which
$$\sum_q p_q \sum_{a : V(x,q,a)=1} p_{q,a} = 1.$$

Thus, the reduction $\mathcal{R}$ converts $x \in \{0,1\}^n$ into the Linear Program with the polyhedron defined by:
$$p_{q,a} \geq 0, \ \forall q \in D^k \text{ and } \forall a \in \Sigma^k. \tag{1}$$
$$\sum_{a \in \Sigma^k} p_{q,a} = 1, \ \forall q \in D^k. \tag{2}$$
$$\sum_{a' : a'_S = a_S} p_{q,a'} = \sum_{a' : a'_S = a_S} p_{q',a'}, \ \forall S \subseteq [k], \ \forall q, q' \in D^k \text{ s.t. } q_S = q'_S, \ \forall a_S \in \Sigma^S. \tag{3}$$

Note that this polyhedron is fixed and does not depend on the instance x. The objective function is
$$\max_{\{p_{q,a}\}} \sum_q p_q \sum_{a : V(x,q,a)=1} p_{q,a}, \tag{4}$$
where for every q, p_q is a fixed value defined by the verifier in the underlying MIP, and $\{p_{q,a}\}$ are the variables. Note that if $x \in L$ then the maximum of this objective function on the polyhedron is 1, whereas if $x \notin L$ then the maximum of this objective function on the polyhedron is at most $2^{-\mathsf{polylog}(n)}$. Thus, determining whether x is in the language or not reduces to approximating the objective function.

It remains to prove that the space complexity of $\mathcal{R}$ is $\mathsf{polylog}(n)$ (and hence the runtime is at most $\mathsf{quasi\text{-}poly}(n)$). Since the polyhedron is fixed, it suffices for the reduction $\mathcal{R}$ to generate the objective function, as defined in Equation (4). Namely, $\mathcal{R}$ needs to compute p_q

for every q, and $V(x, q, a)$ for every q and a. $\mathcal{R}$ computes p_q by enumerating over all possible random coin tosses of the MIP verifier. Note that the MIP verifier tosses at most $\mathsf{polylog}(n)$ coins (this follows from the fact that the MIP verifier runs in $\mathsf{polylog}(n)$ time given oracle access to an encoding of x, and this oracle can be implemented by a deterministic algorithm; see Informal Theorem 1). This, together with the fact that the space complexity of V is $\mathsf{polylog}(n)$, implies that the space complexity of $\mathcal{R}$ is $\mathsf{polylog}(n)$, as desired.

□

Acknowledgments. We thank Boaz Barak for illuminating discussions.

References

[AII06] David Avis, Hiroshi Imai, and Tsuyoshi Ito. On the relationship between convex bodies related to correlation experiments with dichotomic observables. *Journal of Physics A: Mathematical and General, 39(36)*, 39(36):11283, 2006.

[BFL90] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 16–25. IEEE Computer Society, 1990.

[BGKW88] Michael Ben-Or, Shafi Goldwasser, Joe Kilian, and Avi Wigderson. Multi-prover interactive proofs: How to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 113–131, 1988.

[BLM+05] Jonathan Barrett, Noah Linden, Serge Massar, Stefano Pironio, Sandu Popescu, and David Roberts. Nonlocal correlations as an information-theoretic resource. *Physical Review A, 71(022101)*, 71(2):022101, 2005.

[BV04] Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *J. ACM*, 51(4):540–556, 2004.

[Dan51] G. B. Dantzig. Maximization of linear function of variables subject to linear inequalities. pages 339–347, 1951.

[DLR79] David P. Dobkin, Richard J. Lipton, and Steven P. Reiss. Linear programming is log-space hard for P. *Inf. Process. Lett.*, 8(2):96–97, 1979.

[DV04] John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 315–320, 2004.

[FK97] Uriel Feige and Joe Kilian. Making games short (extended abstract). In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 506–516. ACM, 1997.

[Hol09] Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009.

[IKM09] Tsuyoshi Ito, Hirotada Kobayashi, and Keiji Matsumoto. Oracularization and two-prover one-round interactive proofs against nonlocal strategies. In *IEEE Conference on Computational Complexity*, pages 217–228, 2009.

[Ito10] Tsuyoshi Ito. Polynomial-space approximation of no-signaling provers. In *ICALP (1)*, pages 140–151, 2010.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–396, 1984.

[Kha79] L. G. Khachiyan. A polynomial algorithm in linear programming. *In Doklady Akademia Nauk SSSR*, pages 1093–1096, 1979.

[KKM+08] Julia Kempe, Hirotada Kobayashi, Keiji Matsumoto, Ben Toner, and Thomas Vidick. Entangled games are hard to approximate. In *FOCS*, pages 447–456, 2008.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. How to delegate computations: the power of no-signaling proofs. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 485–494, 2014.

[KS06] Jonathan A. Kelner and Daniel A. Spielman. A randomized polynomial-time simplex algorithm for linear programming. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 51–60, 2006.

[KT85] Leonid A. Khalfin and Boris S. Tsirelson. Quantum and quasi-classical analogs of Bell inequalities. In *In Symposium on the Foundations of Modern Physics*, pages 441–460, 1985.

[LN93] Michael Luby and Noam Nisan. A parallel approximation algorithm for positive linear programming. In S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal, editors, *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 448–457. ACM, 1993.

[PR94] Sandu Popescu and Daniel Rohrlich. Quantum nonlocality as an axiom. *Foundations of Physics*, 24(3):379–385, 1994.

[Ras85] Peter Rastall. Locality, Bell's theorem, and quantum mechanics. *Foundations of Physics*, 15(9):963–972, 1985.

[Ser91] Maria J. Serna. Approximating linear programming is log-space complete for P. *Inf. Process. Lett.*, 37(4):233–236, 1991.

[Ton09] Ben Toner. Monogamy of non-local quantum correlations. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 465(2101):59–69, 2009.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de