Packing LPs are Hard to Solve Accurately, Assuming Linear Equations are Hard

Rasmus Kyng *
Department of Computer Science
ETH Zurich

Di Wang †
Google

Peng Zhang ‡
Department of Computer Science
Yale University

Abstract

We study the complexity of approximately solving packing linear programs. In the Real RAM model, it is known how to solve packing LPs with N non-zeros in time $\widetilde{O}(N/\epsilon)$. We investigate whether the ϵ dependence in the running time can be improved.

Our first main result relates the difficulty of this problem to hardness assumptions for solving dense linear equations. We show that, in the Real RAM model, unless linear equations in matrices $n \times n$ with condition number $O(n^{10})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.01}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves a $O(n)\times O(n)$ packing LPs (where $N = O(n^2)$) in time $\widetilde{O}(n^2\epsilon^{-0.0003})$. It would be surprising to solve linear equations in the Real RAM model this fast, as we currently cannot solve them faster than $\widetilde{O}(n^\omega)$, where ω denotes the exponent in the running time for matrix multiplication in the Real RAM model (and equivalently matrix inversion). The current best bound on this exponent is roughly $\omega \leq 2.372$. Note, however, that a fast solver for linear equations does not directly imply faster matrix multiplication. But, our reduction shows that if fast and accurate packing LP solvers exist, then either linear equations can be solved much faster than matrix multiplication or the matrix multiplication constant is very close to 2.

Instantiating the same reduction with different parameters, we show that unless linear equations in matrices with condition number $O(n^{1.5})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.372}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(n^2\epsilon^{-0.067})$. Thus smaller improvements in the exponent for ϵ in the running time of Packing LP solvers also imply improvements in the current state-of-the-art for solving linear equations.

Our second main result relates the difficulty of approximately solving packing linear programs to hardness assumptions for solving sparse linear equations: In the Real RAM model, unless well-conditioned sparse systems of linear equations can be solved faster than $\widetilde{O}((\text{no. non-zeros of matrix})\sqrt{\text{condition number of matrix}})$, no algorithm $(1-\epsilon)$-approximately solves packing LPs with N non-zeros in time $\widetilde{O}(N\epsilon^{-0.165})$. This running time of $\widetilde{O}((\text{no. non-zeros of matrix})\sqrt{\text{condition number of matrix}})$ is obtained by the classical Conjugate Gradient algorithm by a standard analysis. Our reduction implies that if sufficiently good packing LP solvers exist, then this long-standing best-known bound on the running time for solving well-conditioned systems of linear equations is sub-optimal[1]. While we prove results in the Real RAM model, our condition number assumptions ensure that our results can be translated to fixed point arithmetic with $(\log n)^{O(1)}$ bits per number.

1 Introduction

Packing and covering linear programs (LPs) are LPs formulated with non-negative coefficients, constants and variables. Formally, a packing LP in its generic form can be written as

$$\max_{\boldsymbol{x}\geq \boldsymbol{0}}\{\boldsymbol{c}^\top \boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}, \tag{1.1}$$

where $\boldsymbol{c} \in \mathbb{R}^n_{\geq 0}, \boldsymbol{b} \in \mathbb{R}^m_{\geq 0}$ and $\boldsymbol{A} \in \mathbb{R}^{m\times n}_{\geq 0}$. With the same matrix and vectors $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$, a covering LP can be written as

$$\min_{\boldsymbol{y}\geq \boldsymbol{0}}\{\boldsymbol{b}^\top \boldsymbol{y} : \boldsymbol{A}^\top \boldsymbol{y} \geq \boldsymbol{c}\}.$$

Packing LPs and covering LPs are dual to each other, and thus they have the same optimal value. In general, taken non-negative $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$ as input, a packing / covering LP solver returns both a packing LP solution and a covering LP solution as a primal-dual pair [LN93, KY14, AZO15b]. Thus packing LPs and covering LPs are usually considered equivalent from algorithm design or hardness perspectives. In this work we will focus on packing LPs. We remark that by a slightly modification of our reduction, the same lower bounds hold for covering LPs as well.[2]

A packing LP is always feasible and has optimum at least 0 because of the all 0 solution. Let OPT denote

*email: kyng@inf.ethz.ch.

†email: di.wang@cc.gatech.edu.

‡email: peng.zhang@yale.edu. Supported in part by NSF Grant CCF-1562041, Office of Naval Research Award N00014-16-2374, and a Simons Investigator award from the Simons Foundation to Daniel Spielman.

[1] This bound is the best known for solving sparse linear equations when only a bound on the condition number is known. In other regimes, better results are known.

[2] We note that packing and covering LPs are also called positive LPs in the literature, although the usage of positive LPs is also used to refer to the more general class of LPs where both packing (i.e. $\leq$) constraints and covering (i.e. $\geq$) constraints appear simultaneously.

the optimal value of a packing LP of the form (1.1). Given an error parameter $\epsilon > 0$, we say $\boldsymbol{x}$ is a $(1-\epsilon)$-approximate solution to this packing LP iff $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$ and $\boldsymbol{c}^\top \boldsymbol{x} \geq (1-\epsilon)\mathrm{OPT}$.

Throughout this paper, we are interested in nearly-linear time solvers for $(1-\epsilon)$-approximately solving packing LPs in the Real RAM model[3] (see [PS85] for an introduction to the Real RAM model). Specifically, the run time is $\widetilde{O}(N/\epsilon^c)$, where N is the number of non-zeros of the input and $c > 0$ is an absolute constant. We use $\widetilde{O}(\cdot)$ to hide $\mathrm{poly}\log(N/\epsilon)$ factors. Although algorithms for solving general LPs, such as interior point method and ellipsoid method, can be applied to packing LPs with a $\mathrm{poly}\log(1/\epsilon)$ dependence on the run time, these methods usually have large dependence on input size and thus are not suitable for large-scale instances.

Small dependency on $1/\epsilon$ is meaningful in two aspects. First, the dependence on $1/\epsilon$ is a natural measure of the efficiency of iterative methods, where a $1/\epsilon^c$ dependence indicates that to get one more bit of accuracy, the algorithm needs to do 2^c times more work. Second, as packing and covering LPs are often used as subroutines in algorithm design (e.g., bipartite matching, set cover [LN93], scheduling [PST95], multicommodity flow [GK07, Mad10, AHK12], etc.), the dependence on $1/\epsilon$ determines the complexity of the overall algorithm. Sometimes, ϵ is chosen to get the best trade-off among multiple parts of the overall algorithm. One example is bipartite matching [Wan17]. Given a bipartite graph with n vertices and m edges, computing its maximum matching can be formulated as a packing LP with $O(m)$ non-zeros. Suppose one can $(1-\epsilon)$-approximately solve this packing LP in time $O(m/\epsilon^c)$ for some constant c. To turn this approximate solution to a maximum matching, one needs to compute $O(\epsilon n)$ augmenting paths in time $O(m \cdot \epsilon n)$. Setting $\epsilon = n^{-1/(1+c)}$, the total run time is $O(mn^{c/(1+c)})$. Faster approximate solvers for packing LPs would imply better run time for this algorithm.

Designing nearly linear time approximate solvers for packing and covering LPs was initiated by the seminal work of Luby and Nisan [LN93], which proposed an algorithm with run time $\widetilde{O}(N/\epsilon^4)$. The run time dependence on $1/\epsilon$ was then improved by subsequent works including [You01, You14, KY14, AZO15b, AZO15a, WRM16, MRWZ16, CQ18], etc. The best known sequential algorithm has expected run time $\widetilde{O}(N/\epsilon)$ [AZO15a, WRM16].

Therefore, a natural question is: Can we design a $(1-\epsilon)$-approximate solver for packing LPs with run time $\widetilde{O}(N/\epsilon^{o(1)})$? In this paper, we give a negative answer, conditioned on that linear equations cannot be solved fast.

Solving linear equations is important in numerical linear algebra, and is a central tool in computer science, statistics, economics, physics, and engineering. It is used as a subroutine of interior point method when applied to solving a general LP. Using interior point method, solving a polynomially well-conditioned LP can be reduced to solving $\widetilde{O}(\sqrt{\mathrm{rank}(\boldsymbol{A})})$ linear equations of the same size as the input [LS14], where $\boldsymbol{A}$ is the LP coefficient matrix. Improvements in linear equation solvers would directly imply improvements for LP solvers in some regimes, and hence would imply improvements for a large class of convex optimization problems.

Our results. We give a conditional lower bound, by relating solving a packing LP to solving a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. Specially, given an $n \times n$ matrix $\boldsymbol{A}$ with full rank, an n-dimensional column vector $\boldsymbol{b}$, an error parameter $\epsilon \geq 0$, the goal is to compute a vector $\boldsymbol{x}$ such that $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2 \leq \epsilon \|\boldsymbol{b}\|_2$.

We introduce (several variants of) the Linear Equation Time Hypothesis LTH: Firstly, $\mathrm{LTH}_{1.5}^{2.372}$ is the assumption that an arbitrary linear equation instance $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an $n \times n$ matrix with condition number of $\boldsymbol{A}^\top \boldsymbol{A}$ at most $O(n^{2 \cdot 1.5})$ cannot be solved to error $\epsilon \leq n^{-15}$ faster than time $\widetilde{O}(n^{2.372})$ in the Real RAM model[4] $\mathrm{LTH}_{1.5}^{2.372}$ is a major open question in numerical linear algebra and scientific computing since falsifying it implies we can solve linear equations faster than current best known bounds on matrix multiplication. The exponent ω is defined as the number s.t. matrix multiplication can be computed in time bounded by $O(n^\omega)$ in the Real RAM model, and the current best bound is $\omega \leq 2.372\ldots$ Again, in the Real RAM model, is known that one can exactly solve $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ by simply inverting $\boldsymbol{A}$ and then multiplying $\boldsymbol{A}^{-1}\boldsymbol{b}$. We show that in the Real RAM model no algorithm can solve Packing LPs with N non-zero entries in time $\widetilde{O}(N/\epsilon^{0.067})$ unless $\mathrm{LTH}_{1.5}^{2.372}$ is false.

More generally, we define LTH_k^γ as the assumption that an arbitrary linear equation instance $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ is an $n \times n$ matrix with condition number of $\boldsymbol{A}^\top \boldsymbol{A}$ at most $O(n^{2k})$, cannot be solved in time $\widetilde{O}(n^\gamma)$ in

[3] Although we prove results in the Real RAM model, our condition number assumptions ensure that our results can be translated to fixed point arithmetic with $(\log n)^{O(1)}$ bits per number.

[4] Note that in the Real RAM model, using matrix multiplication-based matrix inversion, one can in fact solve a non-singular linear equation exactly in time $\widetilde{O}(n^\omega)$. However, even in fixed point arithmetic, given our assumptions about $\boldsymbol{A}$ and the desired accuracy of the solution, the linear equation can be solved to the desired accuracy in $\widetilde{O}(n^\omega)$ arithmetic operations on numbers with $(\log n)^{O(1)}$ bits each [DDH07].

the Real RAM model. We show that in the Real RAM model, no algorithm can solve Packing LPs with N non-zero entries in time $\widetilde{O}(N/\epsilon^{\alpha})$ unless $\text{LTH}_k^{2+\alpha+3\alpha k}$ is false. By instantiating our result for different parameter values, we can state our first main result informally as

THEOREM 1.1. *[Informal Statement] In the Real RAM model,*

- *unless linear equations in matrices with condition number $O(n^{10})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.01}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.0003})$.*
- *unless linear equations in matrices with condition number $O(n^{1.5})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.372}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.067})$.*

An arguably even more important question in numerical linear algebra and scientific computing is how quickly sparse linear systems can be solved. We define SLTH_k^{γ} as the assumption that an arbitrary linear equation instance $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$ cannot be solved to high accuracy faster than time $\widetilde{O}(\text{nnz}(\boldsymbol{A})^{\gamma})$, when $\boldsymbol{A}$ is an $m \times n$ matrix s.t. $\boldsymbol{A}^\top\boldsymbol{A}$ has condition number n^{2k} Conjugate Gradient [5] rules out SLTH_k^{1+k}, but falsifying $\text{SLTH}_k^{1+0.99k}$ would give the first improvement over Conjugate Gradient for this problem since 1952 [HS52].

We show that no algorithm can solve packing LPs with N non-zero entries in time $\widetilde{O}(N/\epsilon^{\alpha})$ unless $\text{SLTH}_k^{1+1.5\alpha+3\alpha k}$ is false. This has interesting consequences for sparse matrices, e.g. $\text{nnz}(\boldsymbol{A}) \approx n\log n$ with small condition number, e.g. at most $n^{1.5}$. Both Conjugate Gradient and matrix-inversion by matrix multiplication (even with $\omega = 2$) are consistent with $\text{SLTH}_{1.5}^{1.99}$, and no algorithm can solve Packing LPs with N non-zero entries in time $\widetilde{O}(N/\epsilon^{0.165})$ unless $\text{SLTH}_{1.5}^{1.99}$ is false. We can summarize this second main result informally as

THEOREM 1.2. *[Informal Statement] In the Real RAM model, unless linear equations $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$ can be solved to high accuracy faster than time $\widetilde{O}(\text{nnz}(\boldsymbol{A})\sqrt{\text{condition number of } \boldsymbol{A}^\top\boldsymbol{A}})$ when $\boldsymbol{A}^\top\boldsymbol{A}$ has condition number $\Theta(n^3)$, then no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.165})$.*

[5] To understand the condition number dependence, see Section 3 Running time of Conjugate Gradient scales like $\sqrt{\text{condition number of } \boldsymbol{A}^\top\boldsymbol{A}}$, which by our definition of condition number becomes scaling like condition number of $\boldsymbol{A}$. .

Under the assumptions of the informal theorem statement, the Conjugate Gradient algorithm has a worst case running time of $\widetilde{O}(\text{nnz}(\boldsymbol{A})\sqrt{\text{condition number of } \boldsymbol{A}^\top\boldsymbol{A}})$, so the informal theorem also tells us that Conjugate Gradient is suboptimal in this regime if sufficiently good packing LP solvers exist.

While we prove results in the Real RAM model, our condition number assumptions ensure that our results can be translated to fixed point arithmetic with $(\log n)^{O(1)}$ bits per number under appropriate assumptions (see Remark 3.1).

Instead of exploring consequences of having access to a packing LP solver with N non-zero entries in time $\widetilde{O}(N/\epsilon^{\alpha})$ for small constant α, we could also look at consequences of having a $\widetilde{O}(N\log(1/\epsilon))$ solver. Using our reductions, one can then show that general LPs, with polynomially bounded condition numbers and bit complexity, can be solved in $\widetilde{O}(N\log(1/\epsilon))$ time (see Appendix D).

Our reduction. We will first reduce a linear system instance to a linear program instance with non-negative variables, by standard techniques. Given $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$, solving $\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ is equivalent to solving the two equations: $\boldsymbol{A}^\top\boldsymbol{z} = \boldsymbol{A}^\top\boldsymbol{b}, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$, where $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in \mathbb{R}^n$ are variables. Note these two linear equations are always feasible. Let $\boldsymbol{x}^+, \boldsymbol{x}^-, \boldsymbol{z}^+, \boldsymbol{z}^- \in \mathbb{R}^n_{\geq 0}$. We replace $\boldsymbol{x}$ by $\boldsymbol{x}^+ - \boldsymbol{x}^-$ and replace $\boldsymbol{z}$ by $\boldsymbol{z}^+ - \boldsymbol{z}^-$, and write an equality constraint as two inequality constraints:

$$\begin{aligned}
\boldsymbol{A}^\top(\boldsymbol{z}^+ - \boldsymbol{z}^-) &\leq \boldsymbol{A}^\top\boldsymbol{b} \\
-\boldsymbol{A}^\top(\boldsymbol{z}^+ - \boldsymbol{z}^-) &\leq -\boldsymbol{A}^\top\boldsymbol{b} \\
\boldsymbol{A}(\boldsymbol{x}^+ - \boldsymbol{x}^-) - (\boldsymbol{z}^+ - \boldsymbol{z}^-) &\leq \boldsymbol{0} \\
-\boldsymbol{A}(\boldsymbol{x}^+ - \boldsymbol{x}^-) + (\boldsymbol{z}^+ - \boldsymbol{z}^-) &\leq \boldsymbol{0}
\end{aligned}$$

For notation simplicity, we will write the above linear inequalities in its matrix form: $\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{p}$, where $\boldsymbol{y} \in \mathbb{R}^{4n}_{\geq 0}$ contains non-negative variables. Wlog, we can scale $\boldsymbol{C}, \boldsymbol{p}$ so that all entries of $\boldsymbol{C}$ are between -1 and 1, and $\|\boldsymbol{p}\|_2 = 1$.

The key of our reduction is to reduce the linear program $\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{p}$ with non-negative variables to a packing LP which has both non-negative variables and non-negative coefficients. We will convert entries of $\boldsymbol{C}, \boldsymbol{p}$ to non-negative entries by a bounding box constraint. Specifically, we introduce a new non-negative variable $y_0 \in \mathbb{R}_{\geq 0}$, and a new constraint

$$\mathbf{1}^\top\boldsymbol{y} + y_0 = U, \tag{1.2}$$

where $\mathbf{1}$ is the all-one vector, and U is a sufficiently large number so that the linear program $\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{p}, \mathbf{1}^\top\boldsymbol{y} \leq U$

is feasible. One can use binary search to find a value $U \geq 1$ which satisfies the above conditions and is upper bounded by $\text{poly}(n, \kappa)$, where κ is the condition number of $\boldsymbol{A}$. We then add the equality (1.2) to each row of $\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{p}$, and we get

$$(\boldsymbol{C} + \mathbf{1}\mathbf{1}^\top)\boldsymbol{y} + y_0\mathbf{1} \leq \boldsymbol{p} + U\mathbf{1}.$$

Since the entries of $\boldsymbol{C}, \boldsymbol{p}$ are between -1 and 1, the above inequality has non-negative coefficients. Finally, we turn the equality constraint (1.2) to an inequality constraint and set the objective to maximizing $\mathbf{1}^\top\boldsymbol{y}+y_0$:

$$\begin{aligned} \max \quad & \mathbf{1}^\top\boldsymbol{y} + y_0 \\ \text{s.t.} \quad & (\boldsymbol{C} + \mathbf{1}\mathbf{1}^\top)\boldsymbol{y} + y_0\mathbf{1} \leq \boldsymbol{p} + U\mathbf{1} \\ & \mathbf{1}^\top\boldsymbol{y} + y_0 \leq U \\ & y_0 \geq 0, \boldsymbol{y} \geq \mathbf{0} \end{aligned} \tag{1.3}$$

The above LP has size $O(n)$ and the reduction can be implemented in time $O(n)$. In addition, its optimal value is U and its optimal solution $\boldsymbol{y}$ is a feasible solution to the linear system $\boldsymbol{C}\boldsymbol{y} = \boldsymbol{p}$.

Suppose we are given a $(1-\epsilon)$-approximate solution $\boldsymbol{y}$ of the packing LP (1.3). We can turn it to an approximate solution of $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ with error $\epsilon' = \text{poly}(\epsilon, n, \kappa)$. This already implies that if one can solve a packing LP with error ϵ in time $O(n^2 \log(1/\epsilon))$, then one can solve a linear system with error ϵ' in time $O(n^2 \log(n\kappa/\epsilon'))$, which falsifies the hypothesis LTH_{10}^2.

We further improve the error dependence between the linear system instance and the packing LP instance, by by using iterative refinement for linear equations. Specifically, one could solve a system of linear equations with error ϵ', by iteratively solving a sequence of $O(\log(1/\epsilon'))$ linear systems with error 0.1. To compute an approximate solution of $\min_{\boldsymbol{x}} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2$ with error ϵ', instead of solving one packing LP with a small error, we will solve $O(\log(1/\epsilon'))$ packing LPs with a relatively larger error independent of ϵ'. By a careful calculation, we prove Theorem 1.1.

We prove Theorem 1.2 by slightly modifying the above reduction to preserve the sparsity of $\boldsymbol{C}$: for each row of $\boldsymbol{C}\boldsymbol{y} \leq \boldsymbol{p}$, we add a bounding box constraint containing only the variables with non-zero coefficients in this row.

Discussion of our reduction. One should take a moment to reflect on the elements of our construction. Linear equation solvers allow for "accuracy amplification" via iterative refinement. This turns a constant error algorithm into a high accuracy algorithm, provided one is careful to adopt the right notion of error[6]. This in turn allows us to get a type of hardness amplification when reducing systems of linear equations to packing LPs. The most important aspect of our approach is that we are able to turn crude packing LP solvers into crude linear equation solvers of the form that allow of this "accuracy amplification", and hence we are able to derive surprisingly strong consequences from the existence of packing LP solvers with running time scaling as $\text{poly}(1/\epsilon)$.

This should be contrasted with other approaches to hardness amplification in fine-grained complexity, such as the Distributed PCP framework of [ARW17], which was used for showing hardness of approximation for subquadratic time algorithms for problems including finding a maximum inner product pair among a set of vectors. It is not clear how to apply this in a setting where algorithms with highly subquadratic running times in the input size are known to exist. We hope that our approach of showing hardness using a reduction from *constant error* linear equation solving will prove fruitful for other questions in fine-grained hardness of approximation.

1.1 Previous Works Previous works on packing and covering LPs can be divided into width-dependent solvers and width-independent solvers. In this context, width is the product of OPT and the largest entry in $\boldsymbol{A}$. The dependence of width usually comes from multiplicative weights update (e.g., see [PST95, AHK12]). In general, width-dependent solvers can be very slow when the width is large.

The line of width-independent nearly linear time solvers for packing and covering LPs was initiated by the seminal work of Luby and Nisan [LN93], which gave a $\widetilde{O}(N/\epsilon^4)$ time algorithm for a packing or covering LP with N non-zeros. Their algorithm can be made parallel with $\widetilde{O}(1/\epsilon^4)$ time and $\widetilde{O}(N/\epsilon^4)$ total work. Subsequent work along this line of research mostly focus on getting better dependence on $1/\epsilon$ (for example, refer to [You01, You14, KY14, AZO15b, AZO15a, WRM16, MRWZ16, CQ18], etc). Currently, the fastest parallel algorithm for packing and covering LPs runs in $\widetilde{O}(1/\epsilon^2)$ time and $\widetilde{O}(N/\epsilon^2)$ work ([AZO15b, MRWZ16]). The fastest sequential algorithm has expected running time $\widetilde{O}(N/\epsilon)$ and with at least a constant probability solves a packing or covering LP with multiplicative error ϵ

[6]This well-known phenomenon is discussed in Appendix B. Note a subtle point: because the crude linear equation solver we build from a packing LP solver is not guaranteed to act as a linear operator, we make sure it achieves multiplicative error strictly less than 1 to before we use it for iterative refinement. This is in contrast to the more common preconditioning of linear systems by linear operators which allows for "accuracy amplification" even when the preconditioning, corresponding to our crude solver, has much higher error.

([AZO15a, WRM16]).

From the complexity perspective, the main focuses of previous works are on fast parallel solvers. Trevisan and Xhafa [TX98] showed that, *exactly* solving packing LPs is P-complete, that is, there is no fast parallel exact solvers for packing LP unless P = NC. However, their result does not rule out any fast algorithms for *approximately* solving packing LPs. Both exact and approximate solvers for general LPs were proved to be P-complete [DLR79, Ser91, Meg92].

Exactly solvers for the worst cases of many combinatorial problems, geometric problems can be slow. Itai [Ita78] showed that LPs are polynomially equivalent to the following problems: Linear equalities with / without bounded coefficients, Homologous flow, and 2-commodity flow problem. Later, Dobkin and Reiss [DR80] gave more LP-complete problems in computational geometry, such as determining intersection of hyperplanes, finding extreme point, and so on. Allowing some error usually significantly improves running time.

When solving general LPs in the Real RAM model, different results are optimal in different regimes. All known polynomial time algorithms require bounds on bit complexity of the input. By [CLS19], if $\omega \leq 2 + 1/6$ (and another condition on the so-called dual exponent is satisfied), then polynomially well-conditioned LPs can be solved in $\widetilde{O}(n^{2+1/6})$ time using matrix inverse maintenance-based techniques, and using currently known matrix multiplication time, they get a bound of $\widetilde{O}(n^{2.372\cdots})$, matching the current bounds on ω. Solving a polynomially well-conditioned LP can be reduced to solving $\widetilde{O}(\sqrt{\mathrm{rank}(\boldsymbol{A})})$ linear equations of the same size as the input [LS14], where $\boldsymbol{A}$ is the LP coefficient matrix.

Fine-grained complexity [WW10] has motivated the study of conditional lower bounds related to matrix multiplication, and linear equation solving. For example, [MNS+17] showed that, fast algorithms for high accuracy spectrum approximation problems, such as logarithm of matrix determinant, trace of matrix inverse, and trace of matrix exponential, would imply triangle detection algorithms for general graphs running in faster than the state of art matrix multiplication time. In addition, [KZ17] proved that, if one can solve some structured linear equations fast, then one can solve general linear equations fast.

1.2 Organization of the Remaining Paper In Section 2, we give some notations and discuss approximately solving linear equations. In Section 3, we formally define Linear Equation Time Hypotheses, and state our main results for lower bounds of packing LPs. In Section 4, we prove the main results.

2 Preliminaries

2.1 Notations We use $\|\cdot\|_2$ to denote the Euclidean norm on vectors and the spectral norm on matrices. When $\boldsymbol{M}$ is an $n \times n$ positive semidefinite matrix, we define the $\boldsymbol{M}$-norm on a vector $\boldsymbol{x} \in \mathbb{R}^n$ by $\|\boldsymbol{x}\|_{\boldsymbol{M}} \stackrel{\text{def}}{=} \sqrt{\boldsymbol{x}^\top \boldsymbol{M} \boldsymbol{x}}$. Let $\mathrm{nnz}(\boldsymbol{A})$ denote the number of non-zero entries in a matrix $\boldsymbol{A}$. We define $\|\boldsymbol{A}\|_{\max} = \max_{i,j} |\boldsymbol{A}_{ij}|$.

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{c} \in \mathbb{R}^m$ for some m, n, we call the tuple $(\boldsymbol{A}, \boldsymbol{c})$ a linear system. Given matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, let $\mathrm{im}(\boldsymbol{A})$ denote the image of $\boldsymbol{A}$, and let $\boldsymbol{\Pi}_{\boldsymbol{A}} \stackrel{\text{def}}{=} \boldsymbol{A}(\boldsymbol{A}\boldsymbol{A}^\top)^\dagger \boldsymbol{A}^\top$, i.e. the orthogonal projection onto $\mathrm{im}(\boldsymbol{A})$. Note that $\boldsymbol{\Pi}_{\boldsymbol{A}} = \boldsymbol{\Pi}_{\boldsymbol{A}}^\top$ and $\boldsymbol{\Pi}_{\boldsymbol{A}} = \boldsymbol{\Pi}_{\boldsymbol{A}}^2$. We define the maximum singular value $\sigma_{\max}(\boldsymbol{A})$ in the usual way as $\sigma_{\max}(\boldsymbol{A}) = \max_{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}} \sqrt{\frac{\boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}}$. We define the minimum singular value as $\sigma_{\min}(\boldsymbol{A}) = \min_{\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x} \neq \boldsymbol{0}} \sqrt{\frac{\boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}}$.

We define the condition number of $\boldsymbol{A}$ as $\kappa(\boldsymbol{A}) = \frac{\sigma_{\max}(\boldsymbol{A})}{\sigma_{\min}(\boldsymbol{A})}$.

2.2 Approximately Solving A Linear Equation In this section we formally define the notions of approximate solutions to linear systems that we work with throughout this paper. Suppose $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\mathrm{rank}(A) = n$. In general, the linear equation $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ may not have a solution. A reasonable generalization of solving the linear equation is then to solve the problem

$$\arg\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \tag{2.4}$$

Definining a useful notion of an *approximate* solution to the above optimization problem requires some care. We use the following definition. [Linear Equation Approximation Problem, LEA] Given linear system $(\boldsymbol{A}, \boldsymbol{b})$, where $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$, and given a scalar $0 \leq \epsilon \leq 1$, we refer to the LEA problem for the triple $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ as the problem of finding $\boldsymbol{x} \in \mathbb{R}^n$ s.t.

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2 \leq \epsilon \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2,$$

and we say that such an $\boldsymbol{x}$ is a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

This definition of LEA is widely used in the literature on solving systems of linear equations. When $\boldsymbol{b}$ is in the image of $\boldsymbol{A}$, this definition is equivalent to obtaining a solution that is within a $1 + \epsilon$ multiplicative factor of the optimum value of Problem (2.4), but these notions are not equivalent when $\boldsymbol{b}$ is not contained in the image of $\boldsymbol{A}$. In Appendix A, we give more background on this notion of approximately solving linear equations for readers unfamiliar with the subject. Below, we state an important fact from this appendix, which should give

the reader an impression of why the LEA definition is natural. In particular, the definition is equivalent to several other convenient notions of error.

FACT 2.1. *The vector $\boldsymbol{x}$ being a solution to* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ *is equivalent to*

$$\|\boldsymbol{x} - \boldsymbol{x}^*\|^2_{\boldsymbol{A}^T\boldsymbol{A}} \leq \epsilon \|\boldsymbol{x}^*\|^2_{\boldsymbol{A}^T\boldsymbol{A}} = \epsilon \left\|\boldsymbol{A}^\top \boldsymbol{b}\right\|^2_{\left(\boldsymbol{A}^T\boldsymbol{A}\right)^{-1}}$$

Our notion of LEA is useful to work with because an algorithm with this guarantee can self-amplify its accuracy

LEMMA 2.1. *If we have a solver for* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ *that works for arbitrary $\boldsymbol{b}$, then we can obtain a solver for* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ *by iterating it $O(\log(1/\epsilon))$ times.*

This Lemma is well-known, but for completeness we prove it in Appendix B.

3 Linear Equation Hardness Assumptions and Consequences

In this section we cover the state of the art for solving systems of linear equations, and define natural hardness assumptions for solving systems of linear equations. We then state our main results on reductions from linear equation problems to solving Packing LPs and derive some important consequences.

Linear Equations in the Real RAM model. Since [Str69], it has been known that the inverse a square, full rank, $n \times n$ matrix A can be computed using $\widetilde{O}(n^\omega)$ time in the Real RAM model where ω is the *matrix multiplication constant*: the exponent of the fastest algorithm for multiplying two $n \times n$ matrices, again in the Real RAM model. Currently the best known upper bound on the exponent is $\omega < 2.372...$ [LG14]. In general, to output the product of two $n \times n$ matrices requires specifying n^2 entries of the output, so $\omega \geq 2$. It is, however, known that matrix multiplication requires $n^2 \log n$ operations in an arithmetic circuit model [Raz02].

In the Real RAM model, it is also known that if one can compute the inverse of a square $n \times n$ matrix in time $O(n^c)$, then one can compute the product of two square $n \times n$ matrices in $\widetilde{O}(n^c)$ time [Mat]. Thus, up to logarithmic factors, running times for matrix multiplication and matrix inversion must be the same in in this model.

In the Real RAM model, in the general case, given a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^n$, the fastest known algorithm for solving a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, i.e. finding $\boldsymbol{x}$ that satisfies the equation is based on simply computing $\boldsymbol{A}^{-1}$ and applying this matrix to $\boldsymbol{b}$ to get $\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$. Consequently, the best known running time in this model for solving general linear equations is $\widetilde{O}(n^\omega)$. However, it is not known whether a faster algorithm could exist, and this is a major open question in numerical linear algebra and scientific computing, and resolving it would have wide impact across theoretical computer science.

Linear Equations in fixed point arithmetic. In fixed point arithmetic, when we operate on integer inputs, an $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ instance of LEA can be solved in $\widetilde{O}(n^\omega)$ arithmetic operations on numbers represented with $(\log((\|\boldsymbol{A}\| + \|\boldsymbol{b}\|)/\epsilon) \log n)^{O(1)}$ bits each, if the input matrix has $n^{O(1)}$ condition number [DDH07]. A crucial point is that the polynomially bounded condition number ensures that the output and intermediate calculations can be approximately represented using few bits.

Linear Equations in Real RAM with polynomially conditioned inputs. Our analysis is based on inputs with polynomially bounded condition numbers, but we assume the Real RAM model of computation for simplicity.

REMARK 3.1. *Because we study* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ *instances with polynomially bounded condition numbers, and because we rely on simple, numerically stable reductions it would be straightforward to extend our analysis to the fixed point arithmetic model assuming integer inputs and $(\|\boldsymbol{A}\| + \|\boldsymbol{b}\|)/\epsilon = poly(n)$. Then using $\log^{O(1)} n$ bits per number for our computations should suffice.*

The current, long-standing, state of the art for solving linear equations in both the Real RAM model and in fixed point arithmetic on inputs with polynomially bounded condition number motivates our Linear Equation Time Hypothesis, which formalizes the hypothesis that even moderately well-conditioned linear equations cannot be solved quickly. We state a parameterized family of hypotheses, and then discuss the significance of the hypotheses for different parameter values.

DEFINITION 3.1. (LINEAR EQUATION TIME HYPOTHESIS: LTH^γ_k) *In the Real RAM model, in the worst case, when $\boldsymbol{A}$ is a full rank $n \times n$ matrix and has condition number most $\kappa(\boldsymbol{A}) \leq n^k$, the problem* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ *with $\epsilon = n^{-10k}$* cannot *be solved in $\widetilde{O}(n^\gamma)$ time.*

Note that the particular exponent for the bound on ϵ is not important. Also note that in general, solving linear equations requires reading the whole input, so the running time must be at least $\Omega(n^2)$. Hence, LTH^γ_k is true for $\gamma < 2$, at least for $k = \Omega(1)$.

For sparse linear equations in matrices with bounded condition number, a different class of algorithms give a better running time. The best known

general result in this case is based on using Conjugate Gradient Descent (or Chebyshev Iterations, or an accelerated first order method such as Nesterov's algorithm) [Saa03], and these solve $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ in time[7] $\text{nnz}(A)\kappa(\boldsymbol{A})\log(1/\epsilon)$ in the Real RAM model, or if we adapt a different formulation of the LEA for positive definite matrices, the time would be $\text{nnz}(A)\sqrt{\kappa(\boldsymbol{A})}\log(1/\epsilon)$, (when appropriately phrased for $\boldsymbol{A}$ being square and positive definite).

Conjugate gradient can also be shown in the Real RAM model to converge to an exact solution in n iterations, giving a running time of order $n \cdot \text{nnz}(\boldsymbol{A})$. However, this is highly misleading, because achieving this type of behavior in a *floating* point arithmetic model requires about n bits per number, as opposed to $(\log((\|\boldsymbol{A}\| + \|\boldsymbol{b}\|)/\epsilon)\log n)^{O(1)}$ bits in *floating* point arithmetic to achieve the condition number dependent behavior [MMS18].

For our formulation of LEA, no better algorithm than CG is known, and improving on the $\text{nnz}(A)\kappa(\boldsymbol{A})\log(1/\epsilon)$ running time is a major open problem. This motivates the following parameterized family of hardness assumptions.

DEFINITION 3.2. (SPARSE LINEAR EQUATION TIME HYPOTHESIS: SLTH_k^γ) *In the Real RAM model, in the worst case, when $\boldsymbol{A}$ is an $m \times n$ matrix with $m \geq n$, rank$(\boldsymbol{A}) = n$, and $\kappa(\boldsymbol{A}) \leq (\text{nnz}(A))^k$, $\text{LEA}(\boldsymbol{A}, \gamma, \epsilon)$ with $\epsilon = n^{-10k}$* cannot *be solved in time $\widetilde{O}(\text{nnz}(\boldsymbol{A})^\gamma)$.*

Note that Conjugate Gradient shows that SLTH_k^{1+k} is false. However, there is essentially no evidence against $\text{SLTH}_k^{1+0.99k}$ and falsifying this hardness assumption would constitute major progress in solving sparse systems of linear equations. Note that if we instead formulated LEA for Positive Definite matrices and with a different notion of error, Conjugate Gradient would falsify $\text{SLTH}_k^{1+0.5k}$ and falsifying $\text{SLTH}_k^{1+0.49k}$ would be the right bar for making progress.

We state our main results for lower bounds of both dense and sparse packing LP instances in the following. The statements about running times all refer to the Real RAM model.

THEOREM 3.1. (DENSE REDUCTION) *If we can $(1-\epsilon)$-approximately solve a packing LP of size $O(n) \times O(n)$ in time $\widetilde{O}(n^\beta/\epsilon^\alpha)$, then given a condition number upper bound $K_{\boldsymbol{A}} \geq \kappa(\boldsymbol{A})$, we can solve $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ for an $n \times n$ matrix in time*

$$\widetilde{O}(n^\beta \left(nK_{\boldsymbol{A}}^3\right)^\alpha).$$

[7]Note that applying CG to $\boldsymbol{A}^\top\boldsymbol{A}$ gives a condition number dependence of the form $\sqrt{\kappa(\boldsymbol{A})^2} = \kappa(\boldsymbol{A})$.

THEOREM 3.2. (SPARSE REDUCTION) *If we can $(1-\epsilon)$-approximately solve a packing LP that has N non-zeros in time $\widetilde{O}(N/\epsilon^\alpha)$, for any constant α, then given a condition number upper bound $K_{\boldsymbol{A}} \geq \kappa(\boldsymbol{A})$, we can solve $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ for an $m \times n$ matrix $\boldsymbol{A}$ in time*

$$\widetilde{O}\left(\text{nnz}(\boldsymbol{A})\left(m^{3/2}K_{\boldsymbol{A}}^3\right)^\alpha\right).$$

REMARK 3.2. *One could replace $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ in Theorem 3.1 and 3.2 by $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon')$ for any $\epsilon' = 1/\text{poly}(n)$. According to Lemma 2.1, the run times of solving $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon')$ and $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ only differ by $O(\log n)$, which will be hidden in $\widetilde{O}(\cdot)$.*

By connecting these two theorems with our Linear Equation Time Hypotheses defined as above, we get the following corollaries.

COROLLARY 3.1. *If we can $(1-\epsilon)$-approximately solve a packing LP of size $n \times n$ in time $O(n^\beta/\epsilon^\alpha)$, then for any constant $k > 0$, then $\text{LTH}_k^{\beta+\alpha+3\alpha k}$ is false.*

Proof. By Theorem 3.1, we get an algorithm for $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ with $K_{\boldsymbol{A}} = n^k$, which by Theorem 2.1, we can turn into an algorithm for $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ with $\epsilon = n^{-10k}$ using $\log(1/\epsilon) = O(k \log n)$ iterations of the $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ solver with different $\boldsymbol{b}$. Altogether, the running time is $\widetilde{O}(k \log n \cdot n^\beta \left(nK_{\boldsymbol{A}}^3\right)^\alpha)$, which for any constant k is

$$\widetilde{O}\left(n^{\beta+\alpha+3k\alpha}\right).$$

This falsifies $\text{LTH}_k^{\beta+\alpha+3\alpha k}$. ∎

Restricting our attention to matrices $\boldsymbol{A}$ with condition number of $\boldsymbol{A}^\top\boldsymbol{A}$ at most $O(n^{10})$, we can state our central corollary of this as:

THEOREM 3.3. *[Informal Statement] In the Real RAM model,*

- *unless linear equations in matrices with condition number $O(n^{10})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.01}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.0003})$.*
- *unless linear equations in matrices with condition number $O(n^{1.5})$ can be solved to ϵ accuracy faster than $\widetilde{O}(n^{2.372}\log(1/\epsilon))$, no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.067})$.*

Proof. From our $O(N\epsilon^{-0.0003})$ time packing LP solver, by Theorem 3.1 and Lemma 2.1, noting that $N \leq n^2$ and $2 + 0.0003 \cdot (3 \cdot 10 + 1)) \leq 2.01$ immediately

get our $\widetilde{O}(n^{2.01}\log(1/\epsilon))$ solver for $\text{LEA}(\boldsymbol{A},\boldsymbol{b},\epsilon)$ with $K_{\boldsymbol{A}} = O(n^{10})$. The second statement follows similarly by checking the appropriate parameters. ∎

COROLLARY 3.2. *If we can $(1-\epsilon)$-approximately solve a packing LP that has N non-zeros in time $\widetilde{O}(N/\epsilon^{\alpha})$, then $\text{SLTH}_k^{1+1.5\alpha+3k\alpha}$ is false.*

Proof. The proof is similar to that of Corollary 3.1, but it uses the reduction from Theorem 3.2 instead, and that given our assumptions on $\boldsymbol{A}$,

$$\text{nnz}(\boldsymbol{A})\left(m^{3/2}K_{\boldsymbol{A}}^3\right)^{\alpha} \leq \left(\text{nnz}(\boldsymbol{A})^{1+1.5\alpha+3k\alpha}\right)$$

∎

This has interesting consequences for sparse matrices, e.g. $\text{nnz}(\boldsymbol{A}) \approx n\log n$ with small condition number, e.g. at most $n^{1.5}$. Both Conjugate Gradient and matrix-inversion by matrix multiplication (even with $\omega = 2$) are consistent with $\text{SLTH}_{1.5}^{1.99}$, and no algorithm can solve Packing LPs with N non-zero entries in time $\widetilde{O}(N/\epsilon^{-0.165})$ unless $\text{SLTH}_{1.5}^{1.99}$ is false. We can summarize this consequence as

THEOREM 3.4. *[Informal Statement] In the Real RAM model, unless linear equations $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$ can be solved to high accuracy faster than time $\widetilde{O}(\text{nnz}(\boldsymbol{A})\sqrt{\text{condition number of } \boldsymbol{A}^\top\boldsymbol{A}})$ when $\boldsymbol{A}^\top\boldsymbol{A}$ has condition number $\Theta(n^3)$, then no algorithm $(1-\epsilon)$-approximately solves packing LPs in time $\widetilde{O}(N\epsilon^{-0.165})$.*

Proof. Similar to the proof of our first main theorem, we combine Lemma 2.1 and Corollary 3.2, and note that for $\alpha = 0.165$ and $k = 1.5$, we have $1+1.5\alpha+3k\alpha \leq 1.99$, so a packing LP solver with the stated parameters would outperform Conjugate Gradient. ∎

REMARK 3.3. *Suppose that there is likely to be very large gap between $\kappa(\boldsymbol{A})$ and a known upper bound $K_{\boldsymbol{A}}$. Then we might want to get reductions that depend on $\kappa(\boldsymbol{A})$ instead of $K_{\boldsymbol{A}}$. We can essentially do this, and reduce the dependence on $K_{\boldsymbol{A}}$ to logarithmic. For example, in the proof of Theorem 3.1, we constructed a solver for $\text{LEA}(\boldsymbol{A},\boldsymbol{b},\epsilon)$ with running time $\widetilde{O}(n^{\beta}\left(nK_{\boldsymbol{A}}^3\right)^{\alpha}\log(1/\epsilon))$. and we can turn this into a solver with running time $\widetilde{O}(n^{\beta}\left(n\kappa(\boldsymbol{A})^3\right)^{\alpha}\log(K_{\boldsymbol{A}}/\epsilon))$. The same type of improvement can be obtained in the sparse case. We explain these modified reductions in Appendix C.*

4 Reduction

In this section we discuss how to reduce solving linear systems to solving packing LPs. We introduce two reductions from linear equation problems to packing LP problems. Using the two reductions, we will prove Theorems 3.1 and 3.2 later in this section.

Normalization. Given a linear equation approximation problem instance $\text{LEA}(\boldsymbol{A},\boldsymbol{b},\epsilon)$ as defined in Definition 2.2, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m, 0 \leq \epsilon \leq 1$. We will assume that $m \geq n$, $\sigma_{\min}(\boldsymbol{A}) > 0$, and $\kappa(\boldsymbol{A}) \leq K_A$ for some known K_A. Furthermore, we will assume $\boldsymbol{A}^\top\boldsymbol{b} \neq 0$, otherwise $\boldsymbol{x} = 0$ is a solution to $\text{LEA}(\boldsymbol{A},\boldsymbol{b},\epsilon)$, in which case $\text{LEA}(\boldsymbol{A},\boldsymbol{b},\epsilon)$ is solved in time $O(mn)$.

We denote $\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{A}^T\boldsymbol{b}$, and WLOG we assume $\boldsymbol{A},\boldsymbol{b}$ are normalized so that

$$\|\boldsymbol{A}\|_{\max} \stackrel{\text{def}}{=} \max_{i,j}|\boldsymbol{A}_{i,j}| = 1, \qquad \|\boldsymbol{p}\|_2 = 1.$$

This normalization can be done in $\text{nnz}(\boldsymbol{A})$ time, and given any ϵ-approximate solution to the normalized linear system, we can scale the solution to get an ϵ-approximate solution to the original linear system. Moreover, the condition number $\kappa(\boldsymbol{A})$ is not changed by the normalization. Furthermore, as $\|\boldsymbol{A}\|_{\max} = 1$, we must have $\sigma_{\max}(\boldsymbol{A}) \geq 1$, which gives the following claim.

CLAIM 4.1. $\sigma_{\min}^{-1}(\boldsymbol{A}) \leq \kappa(\boldsymbol{A})$.

We will present two reductions from linear systems to packing LPs. Both reductions use the same high-level idea, but one reduction always constructs a packing LP that is dense, while the other preserves the sparsity of the given linear system instance. Formally, given a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $\boldsymbol{A}$ has $\text{nnz}(A)$ non-zeros and $\kappa(\boldsymbol{A}) \leq K_A$ for some known $K_A = \text{poly}(n)$; and a scalar parameter U which we will discuss shortly, we have a function $\text{PLP}_{\text{dense}}(\boldsymbol{A},\boldsymbol{b},U)$ that constructs a dense packing LP of size $O(m)\times O(m)$, and a function $\text{PLP}_{\text{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ that constructs a packing LP of size $O(m)\times O(m)$ and $O(\text{nnz}(\boldsymbol{A}))$ non-zeros.

In both the dense reduction and sparse reduction, we first rewrite solving the linear system as finding a feasible solution to a LP which we denote by $\text{LP}(\boldsymbol{A},\boldsymbol{b})$ as follows. Recall $\boldsymbol{p} \stackrel{\text{def}}{=} \boldsymbol{A}^T\boldsymbol{b}$, and solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ can be rewritten as $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{p}$, which in turn can be written as finding $\boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{z} \in \mathbb{R}^m$ such that

$$\begin{aligned} \boldsymbol{A}^T\boldsymbol{z} &= \boldsymbol{p} \\ \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{z} \end{aligned} \tag{4.5}$$

Note that even when $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ does not have an exact solution, the linear system (4.5) always has an exact solution, and we can bound the norm of the exact solution.

CLAIM 4.2. *There exists* $\boldsymbol{x}^*, \boldsymbol{z}^*$ *satisfying the linear system* (4.5)*, and*

$$\|\boldsymbol{z}^*\|_1 \le \sqrt{m}\,\|\Pi_A \boldsymbol{b}\|_2\,, \qquad \|\boldsymbol{x}^*\|_1 \le \sqrt{n}\sigma_{\min}^{-1}(\boldsymbol{A})\,\|\Pi_A \boldsymbol{b}\|_2\,,$$

moreover, we know

$$\|\Pi_A \boldsymbol{b}\|_2 \in \left[\sigma_{\max}^{-1}(\boldsymbol{A})\,\|\boldsymbol{p}\|_2\,, \sigma_{\min}^{-1}(\boldsymbol{A})\,\|\boldsymbol{p}\|_2\right].$$

Proof. Consider the solution where $\boldsymbol{z}^* = \Pi_A \boldsymbol{b} = \boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{p}$ and $\boldsymbol{x}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$. It is straightforward to check $\boldsymbol{z}^*, \boldsymbol{x}^*$ satisfy (4.5). Moreover,

$$\|\boldsymbol{z}^*\|_1 \le \sqrt{m}\,\|\boldsymbol{z}^*\|_2 = \sqrt{m}\,\|\Pi_A \boldsymbol{b}\|_2\,,$$

and

$$\|\boldsymbol{x}^*\|_1 \le \sqrt{n}\,\|\boldsymbol{x}^*\|_2 \le \sqrt{n}\sigma_{\min}^{-1}(\boldsymbol{A})\,\|\Pi_A \boldsymbol{b}\|_2$$

where the last inequality follows from $\boldsymbol{A}\boldsymbol{x}^* = \Pi_A \boldsymbol{b}$.

Consider the SVD of $\boldsymbol{A} = \sum_i \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$. Then $\|\Pi_A \boldsymbol{b}\|_2^2 = \sum_i (\boldsymbol{u}_i^\top \boldsymbol{b})^2$ and $\|\boldsymbol{p}\|_2^2 = \left\|\boldsymbol{A}^\top \boldsymbol{b}\right\|_2^2 = \sum_i \sigma_i^2 (\boldsymbol{u}_i^\top \boldsymbol{b})^2$. Thus,

$$\sigma_{\max}^{-1}(\boldsymbol{A}) \le \frac{\|\Pi_A \boldsymbol{b}\|_2}{\|\boldsymbol{p}\|_2} \le \sigma_{\min}^{-1}(\boldsymbol{A}).$$

□

In both of our reductions $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ and $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A}, \boldsymbol{b}, U)$, the parameter U is supposed to be a tight upper-bound of $\|\boldsymbol{z}^*\|_1 + \|\boldsymbol{x}^*\|_1$. Claim 4.2 implies that it suffices to pick $U = 2K_A^2\sqrt{m}$.

From the linear system (4.5), we construct a feasibility linear program $\mathrm{LP}(\boldsymbol{A}, \boldsymbol{b})$ as follows. We want all variables to be non-negative, and we can do this in a fairly standard way by creating two non-negative variables per $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$:

$$\boldsymbol{x}_j = \boldsymbol{x}_j^{(+)} - \boldsymbol{x}_j^{(-)} \tag{4.6}$$

$$\boldsymbol{z}_i = \boldsymbol{z}_i^{(+)} - \boldsymbol{z}_i^{(-)} \tag{4.7}$$

Let $\boldsymbol{x}^{(+)}$, $\boldsymbol{x}^{(-)}$, $\boldsymbol{z}^{(+)}$, and $\boldsymbol{z}^{(-)}$ be vectors whose entries are $\boldsymbol{x}_j^{(+)}$, $\boldsymbol{x}_j^{(-)}$, $\boldsymbol{z}_i^{(+)}$, $\boldsymbol{z}_i^{(-)}$ respectively. $\mathrm{LP}(\boldsymbol{A}, \boldsymbol{b})$ is the feasibility LP of finding a solution satisfying

$$\begin{aligned}
\boldsymbol{A}^T\boldsymbol{z}^{(+)} - \boldsymbol{A}^T\boldsymbol{z}^{(-)} &\le \boldsymbol{p} \\
-\boldsymbol{A}^T\boldsymbol{z}^{(+)} + \boldsymbol{A}^T\boldsymbol{z}^{(-)} &\le -\boldsymbol{p} \\
\boldsymbol{A}\boldsymbol{x}^{(+)} - \boldsymbol{A}\boldsymbol{x}^{(-)} - (\boldsymbol{z}^{(+)} - \boldsymbol{z}^{(-)}) &\le 0 \\
-\left(\boldsymbol{A}\boldsymbol{x}^{(+)} - \boldsymbol{A}\boldsymbol{x}^{(-)}\right) + \left(\boldsymbol{z}^{(+)} - \boldsymbol{z}^{(-)}\right) &\le 0 \\
\boldsymbol{x}^{(+)}, \boldsymbol{x}^{(-)}, \boldsymbol{z}^{(+)}, \boldsymbol{z}^{(-)} &\ge 0
\end{aligned}$$

The two reductions $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ and $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A}, \boldsymbol{b}, U)$ proceed differently to write $\mathrm{LP}(\boldsymbol{A}, \boldsymbol{b})$ into packing LPs, and we present the remaining steps of the two reduction and the analysis in the rest of the section.

4.1 $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ To turn $\mathrm{LP}(\boldsymbol{A}, \boldsymbol{b})$ into a packing LP, we add one non-negative slack variable $s \ge 0$, and denote

$$\alpha_{\mathrm{sum}} \stackrel{\mathrm{def}}{=} s + \sum_{1\le j\le n}\left(\boldsymbol{x}_j^{(+)} + \boldsymbol{x}_j^{(-)}\right) + \sum_{1\le i\le m}\left(\boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)}\right).$$

Note α_{sum} is merely a short-hand for the sum of all the variables instead of a new variable. For each existing constraint of $\mathrm{LP}(\boldsymbol{A}, \boldsymbol{b})$ except the positivity constraints on single variables, we add α_{sum} and U to the LHS and RHS of the constraint respectively, and we also add

$$\alpha_{\mathrm{sum}} \le U \tag{4.8}$$

as an additional constraint. This completes the construction of $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$'s constraints, and if written explicitly are Equation (4.9).

The objective of $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ is to maximize α_{sum}, that is

$$\max s + \sum_{1\le j\le n}\left(\boldsymbol{x}_j^{(+)} + \boldsymbol{x}_j^{(-)}\right) + \sum_{1\le i\le m}\left(\boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)}\right).$$

LEMMA 4.1. *Given* $U = 2K_A^2\sqrt{m} > 1$ *is an upper bound of the* ℓ_1 *norm of some solution of the linear system* (4.5)*,* $PLP_{dense}(\boldsymbol{A}, \boldsymbol{b}, U)$ *is a packing LP of size* $O(m) \times O(m)$*, and* $PLP_{dense}(\boldsymbol{A}, \boldsymbol{b}, U)$ *has optimum at* U*.*

Proof. The size of $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ is obvious. For it to be a packing LP, we need to show

1. All variables are non-negative: This holds by construction.
2. All coefficients are non-negative: This is true since $\|\boldsymbol{A}\|_{\max} \le 1$, so by adding 1 to every coefficient, all coefficients become non-negative.
3. All constants on the RHS of the constraints are non-negative: This is true since $\|\boldsymbol{p}\|_2 = 1$, so $\|\boldsymbol{p}\|_\infty \le 1$, and we add $U \ge 1$ to the RHS of each constraint.
4. All coefficients in the objective function are non-negative: This is true since all coefficients are 1 in the objective.

If $U \ge \|\boldsymbol{x}^*\|_1 + \|\boldsymbol{z}^*\|_1$ for some $\boldsymbol{x}^*$ and $\boldsymbol{z}^*$ satisfying the linear system (4.5), consider the following solution to $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$ where we set $\boldsymbol{x}^{(+)}$ and $\boldsymbol{x}^{(-)}$ based on the signs of $\boldsymbol{x}^*$

$$\left(\boldsymbol{x}_i^{(+)}, \boldsymbol{x}_i^{(-)}\right) := \begin{cases} (\boldsymbol{x}^*{}_i, 0) & \text{if } \boldsymbol{x}^*{}_i \ge 0, \\ (0, -\boldsymbol{x}^*{}_i) & \text{if } \boldsymbol{x}^*{}_i \le 0, \end{cases}$$

$$
\begin{aligned}
s + \sum_{1\le i\le m} (1+\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(+)} + (1-\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(-)} + \sum_{1\le j'\le n} \boldsymbol{x}_{j'}^{(+)} + \boldsymbol{x}_{j'}^{(-)} &\le \boldsymbol{p}_j + U \qquad \forall_{1\le j\le n}\\
s + \sum_{1\le i\le m} (1-\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(+)} + (1+\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(-)} + \sum_{1\le j'\le n} \boldsymbol{x}_{j'}^{(+)} + \boldsymbol{x}_{j'}^{(-)} &\le -\boldsymbol{p}_j + U \qquad \forall_{1\le j\le n}\\
s + \sum_{1\le j\le n} (1+\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(+)} + (1-\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(-)} + 2\boldsymbol{z}_i^{(-)} + \sum_{1\le i'\le n, i'\ne i} \boldsymbol{z}_{i'}^{(+)} + \boldsymbol{z}_{i'}^{(-)} &\le U \qquad \forall_{1\le i\le m}\\
s + \sum_{1\le j\le n} (1-\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(+)} + (1+\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(-)} + 2\boldsymbol{z}_i^{(+)} + \sum_{1\le i'\le n, i'\ne i} \boldsymbol{z}_{i'}^{(+)} + \boldsymbol{z}_{i'}^{(-)} &\le U \qquad \forall_{1\le i\le m}\\
s + \sum_{1\le j\le n} \left(\boldsymbol{x}_j^{(+)} + \boldsymbol{x}_j^{(-)}\right) + \sum_{1\le i\le m} \left(\boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)}\right) &\le U\\
s, \boldsymbol{x}^{(+)}, \boldsymbol{x}^{(-)}, \boldsymbol{z}^{(+)}, \boldsymbol{z}^{(-)} &\ge 0
\end{aligned}
\tag{4.9}
$$

and similarly for $\boldsymbol{z}^{(+)}$ and $\boldsymbol{z}^{(-)}$:

$$
\left(\boldsymbol{z}_i^{(+)}, \boldsymbol{z}_i^{(-)}\right) := \begin{cases} (\boldsymbol{z}^*{}_i, 0) & \text{if } \boldsymbol{z}^*{}_i \ge 0,\\ (0, -\boldsymbol{z}^*{}_i) & \text{if } \boldsymbol{z}^*{}_i \le 0, \end{cases}
$$

For the added slack variable s, as $U \ge \|\boldsymbol{x}^*\|_1 + \|\boldsymbol{z}^*\|_1$, we can set

$$
\begin{aligned}
s &\stackrel{\text{def}}{=} U - \|\boldsymbol{x}^*\|_1 + \|\boldsymbol{z}^*\|_1\\
&= U - \left(\sum_{1\le j\le n} \left(\boldsymbol{x}_j^{(+)} + \boldsymbol{x}_j^{(-)}\right) + \sum_{1\le i\le m} \left(\boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)}\right) \right).
\end{aligned}
$$

It is straightforward to see this solution is feasible, and gives objective value U, which must be the optimal due to the constraint (4.8). ∎

In the following lemma, we translate the error bound between the original linear system and $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$.

LEMMA 4.2. *Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m, 0 \le \epsilon' \le 1$, where $m \ge n$, and $\boldsymbol{A}^\top \boldsymbol{b} \ne 0$, and $\kappa(\boldsymbol{A})$ is upper bounded by some known number K_A. Let*

$$
U \stackrel{\text{def}}{=} 2K_A^2\sqrt{m},\ \epsilon \stackrel{\text{def}}{=} \frac{\epsilon'}{2K_A\sqrt{m}U}
$$

Suppose $\boldsymbol{x}^{(+)}, \boldsymbol{x}^{(-)}, \boldsymbol{z}^{(+)}, \boldsymbol{z}^{(-)}, s$ is a feasible solution of $PLP_{dense}(\boldsymbol{A}, \boldsymbol{b}, U)$ with objective value at least $(1-\epsilon)U$, then $\boldsymbol{x} = \boldsymbol{x}^{(+)} - \boldsymbol{x}^{(-)}$ is a solution to $\mathrm{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon')$, that is,

$$
\|\boldsymbol{A}\boldsymbol{x} - \Pi_A \boldsymbol{b}\|_2 \le \epsilon' \|\Pi_A \boldsymbol{b}\|_2 .
$$

Proof. By Claim 4.2, U is an upper bound of the ℓ_1 norm of some solution of the linear system (4.5), that is, U satisfies the condition of Lemma 4.1. By Claim 4.2,

$$
\|\Pi_A \boldsymbol{b}\|_2 \ge \frac{1}{\sigma_{\max}(\boldsymbol{A})} \tag{4.10}
$$

Let $\boldsymbol{x} = \boldsymbol{x}^{(+)} - \boldsymbol{x}^{(-)}, \boldsymbol{z} = \boldsymbol{z}^{(+)} - \boldsymbol{z}^{(-)}$. We want to bound the error in $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}, \boldsymbol{A}^T\boldsymbol{z} = \boldsymbol{p}$. Consider the i-th equality constraint in the linear system: $\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{z}_i$, we have a corresponding pair of constraints in $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A}, \boldsymbol{b}, U)$

$$
\begin{aligned}
\alpha_{\mathrm{sum}} - \left(\boldsymbol{z}_i^{(+)} - \boldsymbol{z}_i^{(-)}\right) + \sum_{1\le j\le n} \boldsymbol{A}_{ij} \left(\boldsymbol{x}_j^{(+)} - \boldsymbol{x}_j^{(-)}\right) &\le U\\
\alpha_{\mathrm{sum}} + \left(\boldsymbol{z}_i^{(+)} - \boldsymbol{z}_i^{(-)}\right) - \sum_{1\le j\le n} \boldsymbol{A}_{ij} \left(\boldsymbol{x}_j^{(+)} - \boldsymbol{x}_j^{(-)}\right) &\le U
\end{aligned}
$$

Since we have a feasible solution, we know from the above two constraints that

$$
|\boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{z}_i| \le U - \alpha_{\mathrm{sum}}.
$$

The same argument holds for any linear constraint $(\boldsymbol{A}^T)_j\boldsymbol{z} = \boldsymbol{p}_j$, so we have

$$
\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_\infty, \left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{p}\right\|_\infty \le U - \alpha_{\mathrm{sum}},
$$

and by the approximation guarantee the RHS is at most $\epsilon \cdot U$.

Now we bound the error of the solution $\boldsymbol{x}$ in terms of the linear system

$$
\begin{aligned}
&\|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{b}\|_2\\
\le &\|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{z}\|_2 + \|\Pi_A\boldsymbol{z} - \Pi_A\boldsymbol{b}\|_2\\
\le &\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_2 + \sigma_{\min}^{-1}(A) \left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{A}^T\boldsymbol{b}\right\|_2\\
\le &\sqrt{m}\,\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_\infty + \sqrt{n}\sigma_{\min}^{-1}(A) \left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{A}^T\boldsymbol{b}\right\|_\infty\\
\le &\epsilon\kappa(\boldsymbol{A})(\sqrt{m} + \sqrt{n})U.
\end{aligned}
\tag{4.11}
$$

In the second inequality, we have $\|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{z}\|_2 \le \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_2$ because $\boldsymbol{A}\boldsymbol{x}$ is in the span of columns of

$\boldsymbol{A}$, so projecting $\boldsymbol{z}$ to the column span of $\boldsymbol{A}$ makes the distance to $\boldsymbol{Ax}$ smaller. Also in the second inequality, we use

$$\begin{aligned}
&\|\Pi_A \boldsymbol{z} - \Pi_A \boldsymbol{b}\|_2 \\
=&\left\|\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\boldsymbol{z}-\boldsymbol{b})\right\| \\
=&\sqrt{(\boldsymbol{z}-\boldsymbol{b})^T\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\boldsymbol{z}-\boldsymbol{b})} \\
=&\sqrt{(\boldsymbol{z}-\boldsymbol{b})^T\boldsymbol{A}(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T(\boldsymbol{z}-\boldsymbol{b})} \\
\leq&\sigma_{\min}^{-1}(A)\left\|\boldsymbol{A}^T\boldsymbol{z}-\boldsymbol{A}^T\boldsymbol{b}\right\|_2.
\end{aligned}$$

The last inequality is due to Claim 4.1. Plugging Equation (4.10) into Equation (4.11):

$$\begin{aligned}
\|\boldsymbol{Ax}-\Pi_A\boldsymbol{b}\|_2 &\leq \epsilon\kappa(\boldsymbol{A})(\sqrt{m}+\sqrt{n})U\cdot\|\Pi_A\boldsymbol{b}\|_2 \\
&\leq 2\epsilon K_A\sqrt{m}U\cdot\|\Pi_A\boldsymbol{b}\|_2.
\end{aligned}$$

By the setting of ϵ, we have $\|\boldsymbol{Ax}-\Pi_A\boldsymbol{b}\|_2 \leq \epsilon'\|\Pi_A\boldsymbol{b}\|$. ∎

Proof. [Proof of Theorem 3.1] Given an arbitrary linear equation approximation problem instance $\textsc{lea}(\boldsymbol{A},\boldsymbol{b},0.1)$ where $\boldsymbol{A}\in\mathbb{R}^{n\times n},\boldsymbol{b}\in\mathbb{R}^n$ and $\kappa(\boldsymbol{A})\leq K_A$ for some known K_A. We first check whether $\boldsymbol{A}^\top\boldsymbol{b}=0$. If $\boldsymbol{A}^\top\boldsymbol{b}=0$, then $\boldsymbol{x}=0$ is a solution to $\textsc{lea}(\boldsymbol{A},\boldsymbol{b},0.1)$. Otherwise, we will construct $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A},\boldsymbol{b},U)$ with $U=K_A^2\sqrt{n}$ and solve it up to multiplicative error $\epsilon=\frac{1}{100K_A\sqrt{n}U}$. By Lemma 4.2, we will get a solution to $\textsc{lea}(\boldsymbol{A},\boldsymbol{b},0.1)$.

Suppose we can solve the packing LP $\mathrm{PLP}_{\mathrm{dense}}(\boldsymbol{A},\boldsymbol{b},U)$, whose size is $O(n)\times O(n)$, up to multiplicative error ϵ in time $\widetilde{O}(n^\beta/\epsilon^\alpha)$, then we can solve the $\textsc{lea}(\boldsymbol{A},\boldsymbol{b},0.1)$ in time $\widetilde{O}(n^\beta(K_A^3n)^\alpha)$. ∎

4.2 $\mathbf{PLP_{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ The sparsity preserving reduction follows the same approach as the dense reduction. However, instead of adding the same α_{sum} and U to the LHS and RHS of every inequality of $\mathrm{LP}(\boldsymbol{A},\boldsymbol{b})$ to make the coefficients non-negative, we will pin-point only the non-zero coefficients to preserve sparsity. We construct $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ starting from $\mathrm{LP}(\boldsymbol{A},\boldsymbol{b})$ as follows.

For each $j\in[1,n]$, look at the pair of constraints in $\mathrm{LP}(\boldsymbol{A},\boldsymbol{b})$ corresponding to the j-th row of $\boldsymbol{A}^T\boldsymbol{z}=\boldsymbol{p}$ in the linear system (4.5):

$$\begin{aligned}
(\boldsymbol{A}^T)_j\boldsymbol{z}^{(+)}-(\boldsymbol{A}^T)_j\boldsymbol{z}^{(-)}&\leq\boldsymbol{p}_j \\
-(\boldsymbol{A}^T)_j\boldsymbol{z}^{(+)}+(\boldsymbol{A}^T)_j\boldsymbol{z}^{(-)}&\leq-\boldsymbol{p}_j
\end{aligned}$$

We add a non-negative slack variable $\boldsymbol{s}_j^{(p)}$ that serves as the slack for *both* of these constraints. Furthermore, we denote

$$\alpha_j^{(p)}\stackrel{\mathrm{def}}{=}\boldsymbol{s}_j^{(p)}+\sum_{i:A_{ij}\neq0}\boldsymbol{z}_i^{(+)}+\boldsymbol{z}_i^{(-)},$$

and again note $\alpha_j^{(p)}$ is merely a shorthand rather than a new variable. We add $\alpha_j^{(p)}$ and U to the LHS and RHS respectively of both the constraints above. Since $\|\boldsymbol{A}\|_{\max}\leq1$, all the coefficients in these constraints become non-negative. We then add to $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ a new inequality

$$\alpha_j^{(p)}\leq U.$$

Similarly, for each $i\in[1,m]$, we consider the pair of constraints corresponding to the i-th row in $\boldsymbol{Ax}=\boldsymbol{z}$

$$\begin{aligned}
\boldsymbol{A}_i\boldsymbol{x}^{(+)}-\boldsymbol{A}_i\boldsymbol{x}^{(-)}-(\boldsymbol{z}_i^{(+)}-\boldsymbol{z}_i^{(-)})&\leq0 \\
-\left(\boldsymbol{A}_i\boldsymbol{x}^{(+)}-\boldsymbol{A}_i\boldsymbol{x}^{(-)}\right)+\left(\boldsymbol{z}_i^{(+)}-\boldsymbol{z}_i^{(-)}\right)&\leq0
\end{aligned}$$

We add a non-negative slack variable $\boldsymbol{s}_i^{(0)}$ for both of these constraints, and denote

$$\alpha_i^{(0)}:=\boldsymbol{s}_i^{(0)}+\boldsymbol{z}_i^{(+)}+\boldsymbol{z}_i^{(-)}+\sum_{1\leq j\leq n:\boldsymbol{A}_{ij}\neq0}\boldsymbol{x}_j^{(+)}+\boldsymbol{x}_j^{(-)}.$$

We add $\alpha_i^{(0)}$ and U to the LHS and RHS of the pair of constraints, and add $\alpha_i^{(0)}\leq U$ as a new constraint.

We can write the constraints of $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ explicitly as Equation (4.12).

Finally, the objective of $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$ is to maximize $\sum_{1\leq j\leq n}\alpha_j^{(p)}+\sum_{1\leq i\leq m}\alpha_i^{(0)}$.

LEMMA 4.3. *Given $U=2K_A^2\sqrt{m}>1$ is an upper bound of the ℓ_1 norm of some solution of the linear system (4.5), $PLP_{sparse}(\boldsymbol{A},\boldsymbol{b},U)$ constructs a packing LP with $O(\mathrm{nnz}(\boldsymbol{A}))$ non-zeros and $PLP_{sparse}(\boldsymbol{A},\boldsymbol{b},U)$ has optimal value $(m+n)U$.*

The proof is a straightforward but tedious adaptation of the proof of Lemma 4.1, so we omit it.

Now we translate the error bound between the original linear system and $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A},\boldsymbol{b},U)$.

LEMMA 4.4. *Given $\boldsymbol{A}\in\mathbb{R}^{m\times n},\boldsymbol{b}\in\mathbb{R}^m,0\leq\epsilon'\leq1$, where $m\geq n$, $\boldsymbol{A}^\top\boldsymbol{b}\neq0$ and $\kappa(\boldsymbol{A})\leq K_A$ for some known number K_A. Let*

$$U\stackrel{\mathrm{def}}{=}2K_A^2\sqrt{m},\ \epsilon\stackrel{\mathrm{def}}{=}\frac{\epsilon'}{2K_AmU}.$$

Suppose $\boldsymbol{x}^{(+)},\boldsymbol{x}^{(-)},\boldsymbol{z}^{(+)},\boldsymbol{z}^{(-)},\boldsymbol{s}^{(p)},\boldsymbol{s}^{(0)}$ is a feasible solution to $PLP_{sparse}(\boldsymbol{A},\boldsymbol{b},U)$ with objective value at least $(1-\epsilon)\cdot(m+n)U$. Then, $\boldsymbol{x}=\boldsymbol{x}^{(+)}-\boldsymbol{x}^{(-)}$ is a solution to $\textsc{lea}(\boldsymbol{A},\boldsymbol{b},\epsilon')$, that is,

$$\|\boldsymbol{Ax}-\Pi_A\boldsymbol{b}\|_2\leq\epsilon'\|\Pi_A\boldsymbol{b}\|_2.$$

$$
\begin{aligned}
\boldsymbol{s}_j^{(p)} + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} (1+\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(+)} + (1-\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(-)} &\le \boldsymbol{p}_j + U \qquad \forall_{1\le j\le n}\\
\boldsymbol{s}_j^{(p)} + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} (1-\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(+)} + (1+\boldsymbol{A}_{ij})\,\boldsymbol{z}_i^{(-)} &\le -\boldsymbol{p}_j + U \qquad \forall_{1\le j\le n}\\
\boldsymbol{s}_j^{(p)} + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)} &\le U \qquad \forall_{1\le j\le n}\\
2\boldsymbol{z}_i^{(-)} + \boldsymbol{s}_i^{(0)} + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} (1+\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(+)} + (1-\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(-)} &\le U \qquad \forall_{1\le i\le m} \qquad (4.12)\\
2\boldsymbol{z}_i^{(+)} + \boldsymbol{s}_i^{(0)} + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} (1-\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(+)} + (1+\boldsymbol{A}_{ij})\,\boldsymbol{x}_j^{(-)} &\le U \qquad \forall_{1\le i\le m}\\
\boldsymbol{z}_i^{(+)} + \boldsymbol{z}_i^{(-)} + \boldsymbol{s}_i^{(0)} + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{x}_j^{(+)} + \boldsymbol{x}_j^{(-)} &\le U \qquad \forall_{1\le i\le m}\\
\boldsymbol{x}^{(+)}, \boldsymbol{x}^{(-)}, \boldsymbol{z}^{(+)}, \boldsymbol{z}^{(-)}, \boldsymbol{s}^{(p)}, \boldsymbol{s}^{(0)} &\ge 0
\end{aligned}
$$

Proof. Note U is an upper bound of the ℓ_1 norm of some solution of the linear system (4.5), and thus U satisfies the condition of Lemma 4.3. Besides, $\|\Pi_A \boldsymbol{b}\|_2 \ge \frac{1}{\sigma_{\max}(\boldsymbol{A})}$.

Given the $(1-\epsilon)$-approx optimal solution to $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A}, \boldsymbol{b}, U)$, we consider the following solution to the linear system

$$
\begin{aligned}
\boldsymbol{x} &:= \boldsymbol{x}^{(+)} - \boldsymbol{x}^{(-)}\\
\boldsymbol{z} &:= \boldsymbol{z}^{(+)} - \boldsymbol{z}^{(-)}
\end{aligned}
$$

We want to bound the error in $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}, \boldsymbol{A}^T\boldsymbol{z} = \boldsymbol{p}$.

Consider the i-th equality constraint in the linear system $\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{z}_i$, we have a corresponding pair of constraints in $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A}, \boldsymbol{b}, U)$:

$$
\begin{aligned}
\alpha_i^{(0)} + \boldsymbol{A}_i\boldsymbol{x}^{(+)} - \boldsymbol{A}_i\boldsymbol{x}^{(-)} - (\boldsymbol{z}_i^{(+)} - \boldsymbol{z}_i^{(-)}) &\le U\\
\alpha_i^{(0)} - \left(\boldsymbol{A}_i\boldsymbol{x}^{(+)} - \boldsymbol{A}_i\boldsymbol{x}^{(-)}\right) + \left(\boldsymbol{z}_i^{(+)} - \boldsymbol{z}_i^{(-)}\right) &\le U
\end{aligned}
$$

which subtracting away $\alpha_i^{(0)}$ and U from the LHS and RHS respectively gives

$$
|(\boldsymbol{A}\boldsymbol{x})_i - \boldsymbol{z}_i| \le U - \alpha_i^{(0)}.
$$

The same argument holds for any pair of linear constraints corresponding to $(\boldsymbol{A}^T\boldsymbol{z})_j = \boldsymbol{p}_j$, and in total we have

$$
\begin{aligned}
&\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_1 + \left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{p}\right\|_1\\
&\le (m+n)\,U - \sum_j \alpha_j^{(p)} - \sum_i \alpha_i^{(0)}\\
&\le \epsilon\,(m+n)\,U.
\end{aligned}
$$

Now we bound the error of the solution $\boldsymbol{x}$ in terms of LEA

$$
\begin{aligned}
\|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{b}\|_2 &\le \|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{z}\|_2 + \|\Pi_A\boldsymbol{z} - \Pi_A\boldsymbol{b}\|_2\\
&\le \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_2 + \sigma_{\min}^{-1}(A)\left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{A}^T\boldsymbol{b}\right\|_2\\
&\le \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{z}\|_1 + \sigma_{\min}^{-1}(A)\left\|\boldsymbol{A}^T\boldsymbol{z} - \boldsymbol{A}^T\boldsymbol{b}\right\|_1\\
&\le \epsilon\sigma_{\min}^{-1}(A)(m+n)U\\
&\le \epsilon\sigma_{\min}^{-1}(A)(m+n)U \cdot \|\Pi_A\boldsymbol{b}\|_2\\
&\le 2\epsilon K_A m U \|\Pi_A\boldsymbol{b}\|_2.
\end{aligned}
$$

Setting

$$
\epsilon = \frac{\epsilon'}{2K_A m U},
$$

we have $\|\boldsymbol{A}\boldsymbol{x} - \Pi_A\boldsymbol{b}\|_2 \le \epsilon' \|\Pi_A\boldsymbol{b}\|_2$. ∎

Proof. [Proof of Theorem 3.2] The proof is similar to the proof of Theorem 3.1. If $\boldsymbol{A}^\top\boldsymbol{b} = 0$ then $\boldsymbol{x} = 0$ is a solution to $\mathrm{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$. Otherwise we solve $\mathrm{PLP}_{\mathrm{sparse}}(\boldsymbol{A}, \boldsymbol{b}, U)$. If we can ϵ-approximately solve a packing LP in time $\widetilde{O}(N\epsilon^{-\alpha})$, then we can solve $\mathrm{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ in time $\widetilde{O}(nnz(\boldsymbol{A})(m^{3/2}K_A^3)^\alpha)$. ∎

Acknowledgements

The authors thank Richard Peng for many helpful discussions related to this project, and we thank anonymous reviewers for their helpful comments.

References

[AHK12] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[ARW17] A. Abboud, A. Rubinstein, and R. Williams. Distributed pcp theorems for hardness of approximation in p. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 25–36, Oct 2017.

[AZO15a] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-linear time positive LP solver with faster convergence rate. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 229–236, 2015.

[AZO15b] Zeyuan Allen-Zhu and Lorenzo Orecchia. Using optimization to break the epsilon barrier: A faster and simpler width-independent algorithm for solving positive linear programs in parallel. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1439–1456, 2015.

[CLS19] Michael B. Cohen, Yin Tat Lee, and Zhao Song. Solving linear programs in the current matrix multiplication time. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 938–942, New York, NY, USA, 2019. ACM.

[CQ18] Chandra Chekuri and Kent Quanrud. Randomized mwu for positive lps. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 358–377. SIAM, 2018.

[DDH07] James Demmel, Ioana Dumitriu, and Olga Holtz. Fast linear algebra is stable. *Numer. Math.*, 108(1):59–91, October 2007.

[DLR79] David Dobkin, Richard J Lipton, and Steven Reiss. Linear programming is log-space hard for p. *Information Processing Letters*, 8(2):96–97, 1979.

[DR80] David P. Dobkin and Steven P. Reiss. The complexity of linear programming. *Theor. Comput. Sci.*, 11(1):1–18, 1980.

[GK07] Naveen Garg and Jochen Koenemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.

[HS52] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952.

[Ita78] Alon Itai. Two-commodity flow. *Journal of the ACM (JACM)*, 25(4):596–611, 1978.

[KY14] Christos Koufogiannakis and Neal E Young. A nearly linear-time ptas for explicit fractional packing and covering linear programs. *Algorithmica*, 70(4):648–674, 2014.

[KZ17] Rasmus Kyng and Peng Zhang. Hardness results for structured linear systems. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 684–695. IEEE, 2017.

[LG14] François Le Gall. Powers of tensors and fast matrix multiplication. In *Proceedings of the 39th international symposium on symbolic and algebraic computation*, pages 296–303. ACM, 2014.

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

[LS14] Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in o (vrank) iterations and faster algorithms for maximum flow. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 424–433. IEEE, 2014.

[Mad10] Aleksander Madry. Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 121–130. ACM, 2010.

[Mat] Notes for math 242, linear algebra, lehigh university fall 2008. Available at: https://www.lehigh.edu/ gi02/m242/08linstras.pdf.

[Meg92] Nimrod Megiddo. A note on approximate linear programming. *Inf. Process. Lett.*, 42(1):53–, April 1992.

[MMS18] Cameron Musco, Christopher Musco, and Aaron Sidford. Stability of the lanczos method for matrix function approximation. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1605–1624. Society for Industrial and Applied Mathematics, 2018.

[MNS+17] Cameron Musco, Praneeth Netrapalli, Aaron Sidford, Shashanka Ubaru, and David P Woodruff. Spectrum approximation beyond fast matrix multiplication: Algorithms and hardness. *arXiv preprint arXiv:1704.04163*, 2017.

[MRWZ16] Michael W. Mahoney, Satish Rao, Di Wang, and Peng Zhang. Approximating the solution to mixed packing and covering lps in parallel $\tilde{O}(\epsilon^{-3})$ time. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 52:1–52:14, 2016.

[PS85] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, Berlin, Heidelberg, 1985.

[PST95] Serge A Plotkin, David B Shmoys, and Éva Tardos. Fast approximation algorithms for fractional packing and covering problems. *Mathematics of Operations Research*, 20(2):257–301, 1995.

[Raz02] Ran Raz. On the complexity of matrix product. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 144–151. ACM, 2002.

[Ren95] James Renegar. Incorporating condition measures into the complexity theory of linear programming. *SIAM Journal on Optimization*, 5(3):506–524, 1995.

[Saa03] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. siam, 2003.

[Ser91] Maria Serna. Approximating linear programming is log-space complete for p. *Information Processing Letters*, 37(4):233–236, 1991.

[ST14] D. Spielman and S. Teng. Nearly linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM Journal on*

Matrix Analysis and Applications, 35(3):835–885, 2014. Available at http://arxiv.org/abs/cs/0607105.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

[TB97] Lloyd N. Trefethen and David Bau. *Numerical linear algebra*. SIAM, 1997.

[TX98] Luca Trevisan and Fatos Xhafa. The parallel complexity of positive linear programming. *Parallel Processing Letters*, 8(04):527–533, 1998.

[Wan17] Di Wang. *Fast Approximation Algorithms for Positive Linear Programs*. PhD thesis, UC Berkeley, 2017.

[WRM16] Di Wang, Satish Rao, and Michael W. Mahoney. Unified acceleration method for packing and covering problems via diameter reduction. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 50:1–50:13, 2016.

[WW10] Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 645–654. IEEE, 2010.

[You01] Neal E Young. Sequential and parallel algorithms for mixed packing and covering. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 538–546. IEEE, 2001.

[You14] Neal E Young. Nearly linear-time approximation schemes for mixed packing/covering and facility-location linear programs. *ArXiv e-prints*, 2014.

A Background on Approximately Solving Linear Equations

In this Appendix, for completeness, we give some background on notions of approximately solving systems of linear equations for readers unfamiliar with the subject. Our presentation is mostly reproduced from [KZ17] by a subset of the authors.

Recall our definition

[Linear Equation Approximation Problem, LEA] Given linear system $(\boldsymbol{A}, \boldsymbol{b})$, where $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$, and given a scalar $0 \leq \epsilon \leq 1$, we refer to the LEA problem for the triple $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ as the problem of finding $\boldsymbol{x} \in \mathbb{R}^n$ s.t.

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2 \leq \epsilon \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2,$$

and we say that such an $\boldsymbol{x}$ is a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

This definition of a LEA instance and solution has several advantages: when $\mathrm{im}(\boldsymbol{A}) = \mathbb{R}^m$, we get $\boldsymbol{\Pi}_{\boldsymbol{A}} = \boldsymbol{I}$, and it reduces to the natural condition $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \epsilon \|\boldsymbol{b}\|_2^2$, which because $\mathrm{im}(\boldsymbol{A}) = \mathbb{R}^m$, can be satisfied for any ϵ, and for $\epsilon = 0$ tells us that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

When $\mathrm{im}(\boldsymbol{A})$ does not include all of $\mathbb{R}^m$, the vector $\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}$ is exactly the projection of $\boldsymbol{b}$ onto $\mathrm{im}(\boldsymbol{A})$, and so a solution can still be obtained for any ϵ. Further, as $(\boldsymbol{I} - \boldsymbol{\Pi}_{\boldsymbol{A}})\boldsymbol{b}$ is orthogonal to $\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}$ and $\boldsymbol{A}\boldsymbol{x}$, it follows that

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|(\boldsymbol{I} - \boldsymbol{\Pi}_{\boldsymbol{A}})\boldsymbol{b}\|_2^2 + \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2.$$

Thus, when $\boldsymbol{x}$ is a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$, then $\boldsymbol{x}$ also gives an $\epsilon^2 \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2$ additive approximation to

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 = \|(\boldsymbol{I} - \boldsymbol{\Pi}_{\boldsymbol{A}})\boldsymbol{b}\|_2^2. \tag{A.1}$$

Similarly, an $\boldsymbol{x}$ which gives an additive $\epsilon^2 \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2$ approximation to Problem (A.1) is always a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$. These observations prove the following (well-known) fact:

FACT A.1. *Let $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^m} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$, then for every $\boldsymbol{x}$,*

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \leq \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}\|_2^2 + \epsilon^2 \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2^2$$

if and only if $\boldsymbol{x}$ is a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

When the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ does not have a solution, a natural notion of solution is any minimizer of Problem (A.1). A simple calculation shows that this is equivalent to requiring that $\boldsymbol{x}$ is a solution to the linear system $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$, which always has a solution even when $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ does not. The system $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$ is referred to as the *normal equation* associated with $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ (see [TB97]).

FACT A.2. *$\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$, if and only if $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x}^* = \boldsymbol{A}^\top\boldsymbol{b}$, and this linear system always has a solution.*

This leads to a natural question: Suppose we want to approximately solve the linear system $\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{b}$. Can we choose our notion of approximation to be equivalent to that of a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$?

A second natural question is whether we can choose a notion of distance between a proposed solution $\boldsymbol{x}$ and an optimal solution $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ s.t. this distance being small is equivalent to $\boldsymbol{x}$ being a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$? The answer to both questions is yes, as demonstrated by the following facts:

FACT A.3. *Suppose $\boldsymbol{x}^* \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ then*

1. $\left\|\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top\boldsymbol{b}\right\|_{(\boldsymbol{A}^\top\boldsymbol{A})^{-1}} = \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 = \|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{A}^\top\boldsymbol{A}}$.

2. *The following statements are each equivalent to* $\boldsymbol{x}$ *being a solution to the* LEA *instance* $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$*:*

   (a) $\left\|\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b}\right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}} \leq \epsilon \left\|\boldsymbol{A}^\top \boldsymbol{b}\right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}}$ *if and only if* $\boldsymbol{x}$ *is a solution to the* LEA *instance* $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

   (b) $\|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{A}^\top \boldsymbol{A}} \leq \epsilon \|\boldsymbol{x}^*\|_{\boldsymbol{A}^\top \boldsymbol{A}}$ *if and only if* $\boldsymbol{x}$ *is a solution to the* LEA *instance* $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

   (c) $\|\boldsymbol{x} - \boldsymbol{x}^*\|_{\boldsymbol{A}^\top \boldsymbol{A}} \leq \epsilon \left\|\boldsymbol{A}^\top \boldsymbol{b}\right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}}$ *if and only if* $\boldsymbol{x}$ *is a solution to the* LEA *instance* $(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$.

Fact A.3 explains connection between our Definition 2.2, and the usual convention for measuring error in the Laplacian solver literature [ST14]. In this setting, we consider a Laplacian matrix $\boldsymbol{L}$, which can be written as $\boldsymbol{L} = \boldsymbol{A}^\top \boldsymbol{A} \in \mathbb{R}^{n \times n}$, and a vector $\boldsymbol{b}$ s.t. $\boldsymbol{\Pi}_{\boldsymbol{A}^\top \boldsymbol{A}} \boldsymbol{b} = \boldsymbol{b}$. This condition on $\boldsymbol{b}$ is easy to verify in the case of Laplacians, since for the Laplacian of a connected graph, $\boldsymbol{\Pi}_{\boldsymbol{A}^\top \boldsymbol{A}} = \boldsymbol{I} - \frac{1}{n}\boldsymbol{1}\boldsymbol{1}^\top$. Additionally, it is also equivalent to the condition that there exists $\boldsymbol{c}$ s.t. $\boldsymbol{b} = \boldsymbol{A}^\top \boldsymbol{c}$. For Laplacians it is possible to compute both $\boldsymbol{A}$ and a vector $\boldsymbol{c}$ s.t. $\boldsymbol{b} = \boldsymbol{A}^\top \boldsymbol{c}$ in time linear in $\text{nnz}(\boldsymbol{L})$. For Laplacian solvers, the approximation error of an approximate solution $\boldsymbol{x}$ is measured by the ϵ s.t. $\left\|\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^\dagger} \leq \epsilon \|\boldsymbol{b}\|_{(\boldsymbol{A}^\top \boldsymbol{A})^\dagger}$. By Fact A.3, we see that this is exactly equivalent to $\boldsymbol{x}$ being a solution to the LEA instance $(\boldsymbol{A}, \boldsymbol{c}, \epsilon)$.

B Iterative Refinement

In this Appendix, we prove the well-known Lemma 2.1 for completeness.

LEMMA 2.1. *If we have a solver for* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, 0.1)$ *that works for arbitrary* $\boldsymbol{b}$*, then we can obtain a solver for* $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ *by iterating it* $O(\log(1/\epsilon))$ *times.*

Proof. Consider the linear equation $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$. Let $\boldsymbol{M} = \boldsymbol{A}^\top \boldsymbol{A}$ and $\boldsymbol{p} = \boldsymbol{A}^\top \boldsymbol{b}$. We basically do iterative refinement involving the matrix $\boldsymbol{M}$. Note the desired solution is $\boldsymbol{x}^* = \boldsymbol{M}^{-1}\boldsymbol{p}$. (See Fact A.3). We start with defining $\boldsymbol{x}^{(0)} = \boldsymbol{0}$ and

$$\boldsymbol{x}^{(0)} = \boldsymbol{0} \text{ and } \boldsymbol{b}^{(0)} = \boldsymbol{b} \text{ and } \boldsymbol{p}^{(0)} = \boldsymbol{A}^\top \boldsymbol{b}$$

Starting from $t = 0$, we will run an $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}^{(t)}, 0.1)$ solver to produce $\boldsymbol{x}^{(t+1)}$ such that

$$\left\|\boldsymbol{x}^{(t+1)} - \boldsymbol{M}^{-1}\boldsymbol{p}^{(t)}\right\|_{\boldsymbol{M}} \leq 0.1 \left\|\boldsymbol{M}^{-1}\boldsymbol{p}^{(t)}\right\|_{\boldsymbol{M}}.$$

We then set

$$\boldsymbol{b}^{(t+1)} = \left(\boldsymbol{b}^{(t)} - \boldsymbol{A}\boldsymbol{x}^{(t+1)}\right)$$

and we let $\boldsymbol{p}^{(t+1)} = \boldsymbol{A}^\top \boldsymbol{b}^{(t+1)}$, which ensures

$$\boldsymbol{p}^{(t+1)} = \left(\boldsymbol{p}^{(t)} - \boldsymbol{M}\boldsymbol{x}^{(t+1)}\right)$$

By definition we have

$$\boldsymbol{M}^{-1}\boldsymbol{p}^{(t+1)} = \boldsymbol{M}\boldsymbol{p}^{(t)} - \boldsymbol{x}^{(t+1)},$$

and thus

$$\left\|\boldsymbol{M}^{-1}\boldsymbol{p}^{(t+1)}\right\|_{\boldsymbol{M}} \leq 0.1 \left\|\boldsymbol{M}^{-1}\boldsymbol{p}^{(t)}\right\|_{\boldsymbol{M}}.$$

For simplicity let's consider a 2-step version. Feeding in $\boldsymbol{b}^{(1)}$ as input to the solver in turn gives $\boldsymbol{x}^{(2)}$ such that

$$\begin{aligned}\left\|\boldsymbol{x}^{(2)} - \boldsymbol{M}^{-1}\boldsymbol{p}^{(1)}\right\|_{\boldsymbol{M}} &\leq 0.1 \left\|\boldsymbol{M}^{-1}\boldsymbol{p}^{(1)}\right\|_{\boldsymbol{M}} \\ &\leq 0.01 \left\|\boldsymbol{M}^{-1}\boldsymbol{p}^{(0)}\right\|_{\boldsymbol{M}}.\end{aligned}$$

Expanding the LHS gives

$$\boldsymbol{x}^{(2)} - \boldsymbol{M}^{-1}\boldsymbol{p}^{(1)} = \boldsymbol{x}^{(2)} + \boldsymbol{x}^{(1)} - \boldsymbol{M}^{-1}\boldsymbol{p},$$

which means $\boldsymbol{x}^{(1)} + \boldsymbol{x}^{(2)}$ is now a solution with error 0.01. Repeating this gives a $10\times$ smaller error after each step. ∎

C Improved Dependence on Condition Number Upper Bounds

In Section 3, we saw how our reductions from linear equations to packing LPs lead to algorithms to the $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ problem. We saw how different running times for packing LP algorithms would lead to different running times for the $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ problem. These algorithms resulting from our reductions depend polynomially on an upper bound $K_{\boldsymbol{A}}$ on the condition number of $\boldsymbol{A}$.

When there might be a large gap between $\kappa(\boldsymbol{A})$ and a known upper bound $K_{\boldsymbol{A}}$, it could be desirable to get reductions that depend on $\kappa(\boldsymbol{A})$ instead of $K_{\boldsymbol{A}}$. In this appendix, we describe how to do this, reducing the dependence on $K_{\boldsymbol{A}}$ to logarithmic. The reduction is straightforward, but we include it for completeness. In the proof of Corollary 3.1 we constructed a solver for $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ with running time $\widetilde{O}(n^\beta \left(nK_{\boldsymbol{A}}^3\right)^\alpha \log(1/\epsilon))$, assuming a packing LP solver with with running time $\widetilde{O}(n^\beta/\epsilon^\alpha)$. We can turn this into an $\text{LEA}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ solver with running time $\widetilde{O}(n^\beta \left(n\kappa(\boldsymbol{A})^3\right)^\alpha \log(K_{\boldsymbol{A}}/\epsilon))$.

One approach to doing so is to make sure we can certify approximation quality of the solution when a sufficiently good one is found. When run with a too-small (invalid) "bound" on $\kappa(\boldsymbol{A})$, the algorithms we

construct may fail, but they do not take more time than the previously derived bound. Consequently, if can check whether a solution is valid, we can then search for an adequate bound on $\kappa(\boldsymbol{A})$ using successive doublings until the algorithm succeeds.

We assume an convenient normalization of $\boldsymbol{A}$ and $\boldsymbol{b}$ (see Section 4), namely

1. $\max_{ij} |\boldsymbol{A}_{ij}| = 1$
2. $\left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 = 1$

Ultimately, our goal is to find $\boldsymbol{x}$ s.t.

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \leq \epsilon \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \tag{C.2}$$

We will show that for any ϵ', if we find a solution with

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \leq \epsilon' \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \tag{C.3}$$

then this implies

$$\left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \leq K_{\boldsymbol{A}} \sqrt{mn} \cdot \epsilon' \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \tag{C.4}$$

We can also show that

$$\left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \leq \epsilon'' \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \tag{C.5}$$

implies

$$\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \leq K_{\boldsymbol{A}} \sqrt{mn} \cdot \epsilon'' \|\boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \tag{C.6}$$

Equation (C.5) has the advantage of being a linear time checkable condition.

Now suppose we the run $\widetilde{O}(n^\beta \left(nB^3\right)^\alpha \log(1/\epsilon'))$ time algorithm for $\textsc{lea}(\boldsymbol{A}, \boldsymbol{b}, \epsilon')$, with $\epsilon' = \epsilon/(K_{\boldsymbol{A}}\sqrt{mn})^2$. If the algorithm succeeds, then Equation (C.3) is satisfied and this implies Equation (C.4). This latter condition we can check in linear time, or equivalently, we check that Equation (C.5) is satisfied with $\epsilon'' = \epsilon/(K_{\boldsymbol{A}}\sqrt{mn})$. This then implies Equation (C.6) is satisfied, and hence Equation (C.2), by our choice of ϵ' and ϵ''.

Thus, whenever if we accept whenever Equation (C.5) is satisfied, we only accept sufficiently accurate solutions. On the other hand, if the $\widetilde{O}(n^\beta \left(nB^3\right)^\alpha \log(1/\epsilon'))$ time algorithm is run with B being an upper bound on $\kappa(\boldsymbol{A})$, it will produce a solution satisfying (C.3) and hence Equations (C.5) and (C.2).

Now, finally, we run the $\widetilde{O}(n^\beta \left(nB^3\right)^\alpha \log(1/\epsilon'))$ time algorithm with successive doublings of B and each time check if Equation (C.5) is satisfied for the output. This must succeed when $B \geq \kappa(\boldsymbol{A})$, but possibly before, in which case we just get a correct solution sooner.

The running time will be dominated by the last call to the algorithm, and on that last call we must have $B < 2\kappa(\boldsymbol{A})$. Based on our choice of ϵ', we now get an overall running time of $\widetilde{O}(n^\beta \left(n\kappa(\boldsymbol{A})^3\right)^\alpha \log(K_{\boldsymbol{A}}/\epsilon))$.

All that remains is to establish the implication from Equation (C.3) to (C.4), and from (C.5) to (C.6). First, we note that with our normalization of $\boldsymbol{A}$ and $\boldsymbol{b}$, one can show $1 \leq \sigma_{\max}(\boldsymbol{A}) \leq \sqrt{mn}$. This further implies that $1/\kappa(\boldsymbol{A}) \leq \sigma_{\min}(\boldsymbol{A})$. Combining this with Fact A.3, we get

$$\begin{aligned}
\left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 &\leq \sigma_{\max} \left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}} \\
&\leq \sqrt{mn} \left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}} \\
&= \sqrt{mn} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2
\end{aligned}$$

and

$$\begin{aligned}
\left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 &\geq \sigma_{\min} \left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}} \\
&\geq \frac{1}{\kappa(\boldsymbol{A})} \left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_{(\boldsymbol{A}^\top \boldsymbol{A})^{-1}} \\
&= \frac{1}{\kappa(\boldsymbol{A})} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2
\end{aligned}$$

Also, by Claim 4.2

$$\|\Pi_A \boldsymbol{b}\|_2 \in \left[\frac{1}{\sqrt{mn}} \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2, \kappa(\boldsymbol{A}) \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \right]$$

Thus, if we assume Equation (C.3), we get

$$\begin{aligned}
\left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 &\leq \sqrt{mn} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 \\
&\leq \sqrt{mn}\epsilon' \|\Pi_A \boldsymbol{b}\|_2 \leq \epsilon' \kappa(\boldsymbol{A}) \sqrt{mn} \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2,
\end{aligned}$$

i.e. we conclude Equation (C.4) holds.

Meanwhile, if we assume Equation (C.5), we get

$$\begin{aligned}
\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{\Pi}_{\boldsymbol{A}}\boldsymbol{b}\|_2 &\leq \kappa(\boldsymbol{A}) \left\| \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \\
&\leq \kappa(\boldsymbol{A})\epsilon'' \left\| \boldsymbol{A}^\top \boldsymbol{b} \right\|_2 \\
&\leq \epsilon'' \kappa(\boldsymbol{A}) \sqrt{mn} \|\Pi_A \boldsymbol{b}\|_2 \\
&\leq \epsilon'' K_{\boldsymbol{A}} \sqrt{mn} \|\Pi_A \boldsymbol{b}\|_2
\end{aligned}$$

so Equation (C.6) holds.

Of course, we can use similar modifications to the algorithm to get an new algorithm for the sparse case of $\textsc{lea}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ assuming an $\widetilde{O}(N/\epsilon^\alpha)$ time algorithm for packing LPs with N non-zeros. This will give an algorithm for $\textsc{lea}(\boldsymbol{A}, \boldsymbol{b}, \epsilon)$ with running time $\widetilde{O}(\text{nnz}(\boldsymbol{A}) \left(m^{3/2}\kappa(\boldsymbol{A})^3\right)^\alpha \log(K_A/\epsilon))$.

D Reducing a General LP Instance to a Packing LP Instance

In this appendix, we show a reduction from a well-conditioned LP instance to a packing LP instance, by slightly modifying our reduction from Section 4. Let us first introduce some notations and definitions for a general LP.

Given $\boldsymbol{A} \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^m, \boldsymbol{c} \in \mathbb{R}^n$. We use $L = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ to denote an LP written in the canonical form:

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge 0 \end{aligned} \tag{D.7}$$

Let $\mathrm{OPT}(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ be the optimal value of LP $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Let

$$\|L\|_\infty \stackrel{\text{def}}{=} \max\{\|\boldsymbol{A}\|_{\max}, \|\boldsymbol{b}\|_\infty, \|\boldsymbol{c}\|_\infty\}.$$

where $\|\boldsymbol{A}\|_{\max} = \max_{i,j} |\boldsymbol{A}_{ij}|$.

Renegar [Ren95] introduced the notion of condition numbers to measure the complexity of solving LPs. Define the primal and dual condition numbers of a LP L as $\kappa_P(L) \stackrel{\text{def}}{=} \sup\{\delta : L + \Delta L \text{ is primal feasible if } \|\Delta L\|_\infty < \delta\}$ and $\kappa_D(L) \stackrel{\text{def}}{=} \sup\{\delta : L + \Delta L \text{ is dual feasible if } \|\Delta L\|_\infty < \delta\}$. Note κ_P and κ_D are not scaling invariant. Renegar proved the following lemma.

LEMMA D.1. ([REN95]) *Let $L = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ be an LP instance with primal optimal solution $\boldsymbol{x}^*$ and dual optimal solution $\boldsymbol{y}^*$. Then*

$$\|\boldsymbol{x}^*\|_1 \le \frac{\max\{\|\boldsymbol{b}\|_\infty, -OPT(L)\}}{\kappa_P(L)}$$
$$\|\boldsymbol{y}^*\|_1 \le \frac{\max\{\|\boldsymbol{c}\|_\infty, OPT(L)\}}{\kappa_D(L)}$$

We state our theorem in the following.

THEOREM D.1. *If we can $(1-\epsilon)$-approximately solve a packing LP with N non-zeros in time $\widetilde{O}(N\log(1/\epsilon))$, then given a feasible LP instance $L = (\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ that has N non-zeros, and polynomially bounded $\|L\|_\infty$, optimum and condition numbers, we can compute a solution $\boldsymbol{x} \ge \boldsymbol{0}$ satisfying*

$$\boldsymbol{c}^\top \boldsymbol{x} \ge OPT(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) - \epsilon', \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} + \epsilon'\boldsymbol{1} \tag{D.8}$$

in time $\widetilde{O}(N\log(1/\epsilon'))$.

Proof. Let m be the number of rows of $\boldsymbol{A}$ and n be the number of columns of $\boldsymbol{A}$.

Let $\boldsymbol{x}^*$ be a primal optimal solution of L and let $\boldsymbol{y}^*$ be a dual optimal solution of L. Since L is feasible and bounded, by the strong duality theorem, $\boldsymbol{x}^*, \boldsymbol{y}^*$ is a feasible solution to the following LP.

$$\begin{aligned} & \boldsymbol{c}^\top \boldsymbol{x} = \boldsymbol{b}^\top \boldsymbol{y} \\ & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{A}^\top \boldsymbol{y} \ge \boldsymbol{c} \\ & \boldsymbol{x}, \boldsymbol{y} \ge 0 \end{aligned} \tag{D.9}$$

Let U be a real number satisfying $U \ge \|\boldsymbol{x}^*\|_1 + \|\boldsymbol{y}^*\|_1$, which can be upper bounded by Lemma D.1. One can use binary search to find U. For simplicity let $K = \|L\|_\infty$.

Similar to our construction of $\mathrm{PLP}_{\mathrm{sparse}}$ in Section 4.2, we introduce a new variable and a new bounding box constraint for each inequality and set the objective properly, and we get the packing LP (D.10). We can check that the packing LP has $O(N)$ non-zeros and the optimum is $(n+m+1)U \stackrel{\text{def}}{=} U'$.

Let $\boldsymbol{x}, \boldsymbol{y}$ be a $(1-\epsilon)$-approximate solution to the packing LP (D.10). Then

$$\begin{aligned} & \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \le \epsilon K U' \boldsymbol{1} \\ & \boldsymbol{c} - \boldsymbol{A}^\top \boldsymbol{y} \le \epsilon K U' \boldsymbol{1} \\ & \left|\boldsymbol{c}^\top \boldsymbol{x} - \boldsymbol{b}^\top \boldsymbol{y}\right| \le \epsilon K U' \end{aligned}$$

From the third inequality above,

$$\boldsymbol{c}^\top \boldsymbol{x} \ge \boldsymbol{b}^\top \boldsymbol{y} - \epsilon K U' \ge \boldsymbol{b}^\top \tilde{\boldsymbol{y}}^* - \epsilon K U' \tag{D.11}$$

where $\tilde{\boldsymbol{y}}^*$ is an optimal solution of the following perturbed dual:

$$\begin{aligned} \min \quad & \boldsymbol{b}^\top \boldsymbol{y} \\ \text{s.t.} \quad & \boldsymbol{A}^\top \boldsymbol{y} \ge \boldsymbol{c} - \epsilon K U' \boldsymbol{1} \\ & \boldsymbol{y} \ge 0 \end{aligned}$$

Given $L(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is dual feasible and bounded, this perturbed dual linear program is feasible and bounded. Its dual is the following perturbed version of $L(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$:

$$\begin{aligned} \max \quad & \boldsymbol{c}^\top \boldsymbol{x} - \epsilon K U' \boldsymbol{1}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\ & \boldsymbol{x} \ge 0 \end{aligned}$$

Let $\tilde{\boldsymbol{x}}^*$ be an optimal solution of this perturbed LP. Then by the strong duality theorem and Equation (D.11),

$$\begin{aligned} \boldsymbol{c}^\top \boldsymbol{x} &\ge \boldsymbol{c}^\top \tilde{\boldsymbol{x}}^* - \epsilon K U' \boldsymbol{1}^\top \tilde{\boldsymbol{x}}^* - \epsilon K U' \\ &\ge \boldsymbol{c}^\top \boldsymbol{x}^* - \epsilon K U' \boldsymbol{1}^\top \boldsymbol{x}^* - \epsilon K U' \\ &\ge \boldsymbol{c}^\top \boldsymbol{x}^* - \epsilon \frac{K U' \max\{\|\boldsymbol{b}\|_\infty, -\mathrm{OPT}(L)\}}{\kappa_P(L)} - \epsilon K U' \end{aligned}$$

$$
\begin{aligned}
\max \quad & \sum_{1\le j\le n} \theta_j + \sum_{1\le i\le m} \eta_i + \zeta \\
\text{s.t.} \quad & K\gamma + \sum_{1\le j\le n:\boldsymbol{c}_j\neq 0} (K+\boldsymbol{c}_j)\boldsymbol{x}_j + \sum_{1\le i\le m:\boldsymbol{b}_i\neq 0} (K-\boldsymbol{b}_i)\boldsymbol{y}_i \le KU \\
& K\gamma + \sum_{1\le j\le n:\boldsymbol{c}_j\neq 0} (K-\boldsymbol{c}_j)\boldsymbol{x}_j + \sum_{1\le i\le m:\boldsymbol{b}_i\neq 0} (K+\boldsymbol{b}_i)\boldsymbol{y}_i \le KU \\
& \gamma + \sum_{1\le j\le n:\boldsymbol{c}_j\neq 0} \boldsymbol{x}_j + \sum_{1\le i\le m:\boldsymbol{b}_i\neq 0} \boldsymbol{y}_i \le U \\
& K\beta_i + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} (K+\boldsymbol{A}_{ij})\boldsymbol{x}_j \le \boldsymbol{b}_i + KU, \forall 1\le i\le m \\
& \beta_i + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{x}_j \le U, \forall 1\le i\le m \\
& K\alpha_j + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} (K-\boldsymbol{A}_{ij})\boldsymbol{y}_i \le -\boldsymbol{c}_j + KU, \forall 1\le j\le n \\
& \alpha_j + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{y}_i \le U, \forall 1\le j\le n \\
& \gamma \ge 0, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{y} \ge \boldsymbol{0}
\end{aligned} \tag{D.10}
$$

Here, $\theta_j \overset{\text{def}}{=} \alpha_j + \sum_{1\le i\le m:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{y}_i, \forall 1 \le j \le n$, $\eta_i \overset{\text{def}}{=} \beta_i + \sum_{1\le j\le n:\boldsymbol{A}_{ij}\neq 0} \boldsymbol{x}_j, \forall 1 \le i \le m$, and $\zeta \overset{\text{def}}{=} \gamma + \sum_{1\le j\le n:\boldsymbol{c}_j\neq 0} \boldsymbol{x}_j + \sum_{1\le i\le m:\boldsymbol{b}_i\neq 0} \boldsymbol{y}_i$. All θ_j, β_i, ζ are shorthands rather than new variables.

The last inequality is due to Lemma D.1.

To make solution $\boldsymbol{x}$ satisfy Equation (D.8), it suffices to set

$$
\epsilon \le \min\left\{ \frac{\epsilon'}{2KU'}, \frac{\epsilon'\kappa_P(L)}{2KU'\max\{\|\boldsymbol{b}\|_\infty, -\mathrm{OPT}(L)\}} \right\}
$$

If we can $(1-\epsilon)$-approximately solve the packing LP (D.10) in time $\widetilde{O}(N\log(1/\epsilon))$, then we can compute a solution $\boldsymbol{x}$ satisfying Equation (D.8) in time

$$
\widetilde{O}\left(N \log\left(\frac{KU(m+n)\max\{1, |\mathrm{OPT}(L)|\}}{\kappa_P(L)\epsilon'} \right) \right)
$$

Given all parameters in the log is polynomially bounded, the run time is $\widetilde{O}(N\log(1/\epsilon'))$. ∎