MATHEMATICS OF OPERATIONS RESEARCH
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 0364-765X | EISSN 1526-5471 | 00 | 0000 | 0001

INFORMS
DOI 10.1287/xxxx.0000.0000
© 0000 INFORMS

Random projections for linear programming

Ky Vu
ITCSC, Chinese University of Hong Kong, vukhacky@gmail.com, www.lix.polytechnique.fr/∼vu

Pierre-Louis Poirion
Huawei Research Center, Paris, France, kiwisensei@gmail.com, sites.google.com/site/plpoirion86/

Leo Liberti
CNRS LIX, Ecole Polytechnique, 91128 Palaiseau, France, liberti@lix.polytechnique.fr, www.lix.polytechnique.fr/∼liberti

Random projections are random linear maps, sampled from appropriate distributions, that approximately preserve certain geometrical invariants so that the approximation improves as the dimension of the space grows. The well-known Johnson-Lindenstrauss lemma states that there are random matrices with surprisingly few rows that approximately preserve pairwise Euclidean distances among a set of points. This is commonly used to speed up algorithms based on Euclidean distances. We prove that these matrices also preserve other quantities, such as the distance to a cone. We exploit this result to devise a probabilistic algorithm to solve linear programs approximately. We show that this algorithm can approximately solve very large randomly generated LP instances. We also showcase its application to an error correction coding problem.

Key words: Johnson-Lindenstrauss lemma, concentration of measure, dimension reduction
MSC2000 subject classification: Primary: 90C05; secondary: 60D05, 52A23
OR/MS subject classification: Primary: programming: linear theory; secondary: mathematics: systems solutions
History: Submitted July 26, 2017

1. Introduction A deep and surprising result, called the *Johnson-Lindenstrauss Lemma* (JLL) [15], states that a set of high dimensional points can be projected to a much lower dimensional space while keeping Euclidean distances approximately the same. The JLL was previously exploited in purely Euclidean distance based algorithms, such as k-means [6] and k nearest neighbours [14]. The JLL has rarely been employed in mathematical optimization. The few occurrences are related to reasonably natural cases such as linear regression [25], where the error minimization is encoded by means of a Euclidean norm. One reason for this is that the very proof of the JLL exploits rotational invariance, naturally exhibited by sets of distances, but which feasible sets commonly occurring in Linear Programming (LP), such as orthants, obviously do not. In this paper we lay the theoretical foundations of solving LPs approximately using random projections, and showcase their usefulness in practice. More precisely, we address LPs in standard form

$$P \equiv \min\{c^\top x \mid Ax = b \wedge x \in \mathbb{R}^n_+\}, \tag{1}$$

where A is an $m \times n$ matrix. For each $i \leq m$ we let A^i be the i-th row of A, and for each $j \leq n$ we let A_j be the j-th column of A. If I is a set of row indices, we indicate the submatrix of A consisting of those rows by A^I; if J is a set of column indices, we indicate the submatrix of A consisting of those columns by A_J. We let $\mathsf{cone}(A)$ be the cone spanned by the column vectors A_j (for $j \le n$), and $\mathsf{conv}(A)$ be the convex hull of the column vectors A_j (for $j \le n$). We denote by $v(P)$ the optimal objective function value of the problem P, and by $\mathcal{F}(P)$ its feasible region. Note that determining whether $\mathcal{F}(P) \neq \varnothing$ is exactly the same problem as determining whether $b \in \mathsf{cone}(A)$. Throughout this paper, all norms will be Euclidean unless specified otherwise.

We often assume that c, b and all the column vectors of A have unit Euclidean norm. This assumption does not lose generality: let $\tilde{b}$ and $\tilde{A}$ be b, A after scaling all columns to unit norm. If $\tilde{x}$ is the optimal solution of the LP with constraints $\tilde{A}x = \tilde{b}$, we can retrieve the optimal solution x^* of Eq. (1) as follows:

$$\forall j \in \{1, \ldots, n\}, \quad x^*_j = \frac{\|b\| \tilde{x}_j}{\|A_j\|}.$$

The vector c can simply be replaced by the scaled vector $\tilde{c} = c/\|c\|$: the optimum will not change, and the optimal objective function value will simply by scaled by $1/\|c\|$.

A *random projector* is a $k \times m$ matrix T, sampled from appropriate distributions (more details on this below), which preserves certain geometrical properties of sets of points in $\mathbb{R}^m$. We denote by

$$P_T \equiv \min\{c^\top x \mid TAx = Tb \wedge x \in \mathbb{R}^n_+\} \tag{2}$$

the randomly projected version of P. Our main result (Thm. 4) states that we can construct a random projector T, with $k \ll n$, such that, for some given $\varepsilon > 0$, we have $|v(P) - v(P_T)| \le \varepsilon$ with arbitrarily high probability (w.a.h.p.). By this we mean that the probability of the concerned event is $1 - f(k)$ where $f(k)$ tends to zero extremely fast as k tends to infinity (typically, the function f is $O(e^{-k})$). Moreover, for fixed ε, k turns out to be $O(\ln n)$. Since the complexity of solving LPs depends on both m and n, a logarithmic reduction on m (even as a function of n) is very appealing.

So far so good; unfortunately, there are some bad news too. First, we prove that the optimum of the projected problem P_T is infeasible w.r.t. $\mathcal{F}(P)$ (the original region) with probability 1 (Prop. 3), which appears to severely limit the usefulness of Thm. 4 — we address the retrieval of an approximate solution of the original problem in Sect. 5. Second, sampling T and performing matrix multiplications $T(A, b)$ is time consuming, since T is a dense matrix. Third, even though the original LP is sparse, the projected LP is dense as a result of T being dense, which means that solving it has an added computational cost. The ill effects of density on CPU time can be moderated in computational experiments by using sparse random projectors (see e.g. [1]). Last, but not least, we have no idea how to estimate, much less compute, the multiplicative constant in the term $O(\ln n)$. We know that the term $\frac{1}{\varepsilon^2}$, which is large if we want the approximation to be tight, plays a role; but there are other universal constants that also play a role. We also know that the probability of the event $|v(P) - v(P_T)| \le \varepsilon$ approaches 1 as $1 - O(e^{-k})$. All this suggests that any practical usefulness of this methodology will come from very large and/or very dense instances.

As we shall indeed see in Sect. 7, our experience suggests that errors decrease as sizes increase. This is true for our randomly generated test set, but also in regard to our coding application.

We have also started testing random projections on LPs coming from various other applications. Though these results are not conclusive enough to be presented (yet), we had some positive experience with: (i) quantile regression [17]; (ii) large and dense diet problems with unit costs [9]; (iii) LPs arising from the basis pursuit methodology for compressed sensing [8].

1.1. Differences with existing literature

Randomized dimension reduction techniques are widely used in the analysis of large data sets, but much less so in Mathematical Programming (MP). Specifically, in the field of LP we are aware of three main results [7, 25, 12]. We set compressed sensing [7] aside, as strictly speaking this is not a solution or reformulation method, but rather a theoretical analysis which explains why ℓ_1-norm minimization of the error of an underdetermined linear system is an excellent proxy for reconstructing sparse solutions. Although we are only citing the paper [7] for compressed sensing, this line of work gave rise to a very large number of papers by many different authors. We shall see in Sect. 8 that compressed sensing, in the setting of our coding application, can be "further compressed" using our methodology.

In [25], it is shown how matrix sketching (which is strongly related to random projections) can help decrease the dimensionality of some convex quadratic minimization over an arbitrary convex set $\mathcal{C}$ from a given $\mathbb{R}^m$ to $\mathbb{R}^k$ for some $k \leq m$. Sect. 3-4 below emphasize some of the differences of the present work with [25]; [25, Eq. (28) §3.4], for example, encodes the problem of deciding whether zero is in the convex hull of the columns of a given matrix B. Unlike our development, the analysis provided in [25] requires the projected dimension k to be bounded below by a function of several parameters before any probability estimation can be made. Another remarkable difference is that the framework described in [25] *requires* a convex purely quadratic objective function: to encode a linear objective $c^\top x$ using a quadratic, the most direct way involves the introduction of a new scalar variable y, and then rewriting $\min c^\top x$ as $\min y^2$ subject to $y \geq c^\top x$ and $y \geq 0$. This reformulation, however, makes the application of the method impossible, since the quadratic form $y^2 = y(1)y$ is represented by a 1×1 matrix, which has dimensions that obviously cannot further reduced by sketching. Lastly, in [25] we find that the projected dimension k is of the order of magnitude of the Gaussian width W of $\mathcal{C}$. To require $k \ll m$, this implies working with convex sets $\mathcal{C}$ having small Gaussian width. By contrast, our technique optimizes over orthants, which have (large) Gaussian width $O(n)$.

The paper [12] proposes a randomized dimensionality reduction based on PAC learning [3]: from a small training set, it is possible to forecast some properties of large data sets while keeping the error low. This is exploited in LPs with very few variables and huge numbers of inequality constraints: it is found that this number can be greatly reduced while keeping the optimality error bounded. In order to have PAC learning assumptions work, the authors focus on application cases which have a specific structure, i.e. there is an order on the constraints which makes their slope vary in a controlled way (an example is given by the piecewise linear approximation of a two-dimensional closed convex curve: one can take many tangents, but few of these suffice to give almost the same approximation). The prominent difference with the method proposed in this paper is that we make no such assumption.

1.2. Contents

The rest of the paper is organized as follows. Section 2 reports the basic concepts about the JLL. In Sect. 3 we show that random projections approximately preserve LP feasibility with high probability. The proof of our main theorem is offered in Sect. 4, where we argue that random projections also preserve LP optimality with high probability. In Sect. 5 we address the limitation referred to above, and provide a method to work out the solution of the original LP given the solution of the projected LP. In Sect. 6 we make some remarks about computational complexity. Sect. 7 reports some computational results, and Sect. 8 showcases an application to error correcting codes.

2. The Johnson-Lindenstrauss lemma

The JLL is stated as follows:

Theorem 1 **(Johnson-Lindenstrauss Lemma [15]).** *Given $\varepsilon \in (0,1)$ and an $m \times n$ matrix A, there exists a $k \times m$ matrix T such that:*

$$\forall 1 \leq i < j \leq n \quad (1-\varepsilon)\|A_i - A_j\| \leq \|TA_i - TA_j\| \leq (1+\varepsilon)\|A_i - A_j\|, \tag{3}$$

where k is $O(\varepsilon^{-2} \ln n)$.

Thus, all sets of n points can be projected to a subspace having dimension logarithmic in n (and, surprisingly, independent of the original number m of dimensions), such that no distance is distorted by more than $1 + 2\varepsilon$. The JLL can be established as a consequence of a general property (see Lemma 1 below) of *sub-gaussian* random mappings $T = \frac{1}{\sqrt{k}}U$ [19]. Some of the most popular choices for U are:

Choices of random projection:
1. orthogonal projections on a random k-dimensional linear subspace of $\mathbb{R}^m$ [15];
2. random $k \times m$ matrices with each entry independently drawn from the standard normal distribution $N(0,1)$ [14];
3. random $k \times m$ matrices with each entry independently taking values $+1$ and -1, each with probability $\frac{1}{2}$ [1];
4. random $k \times m$ matrices with entries independently taking values $+1$, 0, -1, respectively with probability $\frac{1}{6}$, $\frac{2}{3}$, $\frac{1}{6}$ [1] (we call this the *Achlioptas random projector*).

Other, sparser projectors have been proposed in [1, 10, 16, 2]. In this paper we just limit our attention to the normally distributed $T \sim N(0,1/k)$ (where $1/k$ is the variance, not the standard deviation) and its discrete approximation in Item 4 above. Our reasons for ignoring this issue is that we believe that the rue bottleneck lies the unknown "large constants" referred to above. The matrix product operation (on which the choice of random projector would have the greatest impact) is one of the most common in scientific computing, and many ways are known to optimize and streamline it. In our computational experiments (Sect. 7-8) we use the Achlioptas projector and the most obvious matrix product implementation.

Note that all the random projectors we consider have zero mean. This is necessary in order to ensure that our randomized algorithms will yield the result we want in expectation. This also explains why we consider LPs in standard rather than canonical form: we cannot apply the random projection to the inequality system $Ax \le b$ to yield $TAx \le Tb$: this is almost always false, since the signs of the components of the matrix T are distributed uniformly.

The JLL can be derived from a more fundamental result [20].

Lemma 1 **(Random projection lemma)**. *For all $\varepsilon \in (0,1)$ and all vectors $y \in \mathbb{R}^m$, let T be a $k \times m$ random projector from one of the choices (1-4) above , then*

$$\mathsf{Prob}(\,(1-\varepsilon)\|y\| \le \|Ty\| \le (1+\varepsilon)\|y\|\,) \ge 1 - 2e^{-\mathcal{C}\varepsilon^2 k} \tag{4}$$

for some constant $\mathcal{C} > 0$ (independent of m, k, ε).

It can be proved easily that JLL is a consequence of Lemma 1 by setting $y = A_i - A_j$ for all pairs of (i,j) and then applying the union bound. Moreover, Lemma 1 shows that the probability of finding a good T is very high for large enough values of k. Indeed, from Lemma 1, the probability that Eq. (3) holds for all $i \ne j \le n$ is at least

$$1 - 2\binom{n}{2}e^{-\mathcal{C}\varepsilon^2 k} = 1 - n(n-1)e^{-\mathcal{C}\varepsilon^2 k}. \tag{5}$$

Therefore, if we want this probability to be larger than, say 99.9%, we simply choose any k such that $\frac{1}{1000n(n-1)} > e^{-\mathcal{C}\varepsilon^2 k}$. This means k can be chosen to be $k = \lceil \frac{\ln(1000)+2\ln(n)}{\mathcal{C}\varepsilon^2} \rceil$, which is $O(\,\varepsilon^{-2}(\ln(n) + 3.5)\,)$.

Note that the distributions from which T is sampled are such that the the average of $\|Ty\|$ over T is equal to $\|y\|$. Lemma 1 is a *concentration of measure* result, and it states that the probability of a single sampling of T yielding a value of $\|Ty\|$ very close to its mean approaches 1 as fast as a negative exponential of k approaches zero.

We shall also need a squared version of the random projection lemma [11].

Lemma 2 **(Random projection lemma, squared version)**. *For all $\varepsilon \in (0,1)$ and all vectors $y \in \mathbb{R}^m$, let T be a $k \times m$ random projector from one of the choices (1-4) above, then*

$$\mathsf{Prob}(\,(1-\varepsilon)\|y\|^2 \le \|Ty\|^2 \le (1+\varepsilon)\|y\|^2\,) \ge 1 - 2e^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k} \tag{6}$$

for some constant $\mathcal{C} > 0$ (independent of m, k, ε).

Another relevant result about the JLL is the preservation of angles (or scalar product) with high probability. Indeed, given any $x, y \in \mathbb{R}^n$, and T a $k \times m$ random projector from one of the choices (1-4) above, by applying Lemma 2 on two vectors $x+y$, $x-y$ and using the union bound, we have

$$\begin{aligned} |\langle Tx, Ty\rangle - \langle x, y\rangle| &= \tfrac{1}{4}\Big|\|T(x+y)\|^2 - \|T(x-y)\|^2 - \|x+y\|^2 + \|x-y\|^2\Big| \\ &\le \tfrac{1}{4}\Big|\|T(x+y)\|^2 - \|x+y\|^2\Big| + \tfrac{1}{4}\Big|\|T(x-y)\|^2 - \|x-y\|^2\Big| \\ &\le \tfrac{\varepsilon}{4}(\|x+y\|^2 + \|x-y\|^2) = \tfrac{\varepsilon}{2}(\|x\|^2 + \|y\|^2), \end{aligned} \tag{7}$$

with probability at least $1 - 4e^{-\mathcal{C}\varepsilon^2 k}$. We can strengthen this further to obtain the following useful result.

Proposition 1. *Let $T : \mathbb{R}^m \to \mathbb{R}^k$ be a $k \times m$ random projector from one of the choices (1-4) above and let $0 < \varepsilon < 1$. Then there is a universal constant $\mathcal{C}$ such that, for any $x, y \in \mathbb{R}^n$:*

$$-\varepsilon\|x\|\,\|y\| \le \langle Tx, Ty\rangle - \langle x, y\rangle \le \varepsilon\|x\|\,\|y\|$$

with probability at least $1 - 4e^{-\mathcal{C}\varepsilon^2 k}$.

Proof . Apply Eq. (7) with x replaced by $\frac{x}{\|x\|}$ and y replaced by $\frac{y}{\|y\|}$. This yields $|\langle Tx, Ty\rangle - \langle x, y\rangle| \le \varepsilon$; now we can multiply both sides by $\|x\|\,\|y\|$ to obtain the desired result. □

Lemma 2 and Proposition 1 will be used extensively throughout the paper in order to estimate the distance between a vector and a linear combination of other vectors. In particular, we can bound $\|Tx - \sum_i \lambda_i Ty_i\|^2$ by first expanding it and then approximating each $\langle Tx, Ty_i\rangle$ by $\langle x, y_i\rangle$ and $\langle Ty_i, Ty_j\rangle$ by $\langle y_i, y_j\rangle$. We leverage over the weights λ_i to get the desired estimation. In the rest of the paper, we will refer to a $k \times m$ random matrix from one of the choices (1-4) in Section 2 as a "random projector".

We remark that Lemma 1 is central to all random projection theory, and will be used again later in Sect. 3. Prop. 1 will be used later on in Sect. 4. If $Ax = b \wedge x \ge 0$ is infeasible, then b is not in the cone spanned by the columns of A. By proving angle preservation w.a.h.p., Prop. 1 is key to understanding why, when we pre-multiply $Ax = b$ by a random projector, the resulting system is still infeasible w.a.h.p.

3. Preserving LP feasibility

Consider the Linear Feasibility Problem (LFP)

$$\mathcal{F} = \mathcal{F}(P) \equiv \{x \in \mathbb{R}^n_+ \mid Ax = b\}$$

and its randomly projected version

$$T\mathcal{F} = \mathcal{F}(P_T) \equiv \{x \in \mathbb{R}^n_+ \mid TAx = Tb\}.$$

In this section we prove that $F \neq \varnothing$ if and only if $T\mathcal{F} \neq \varnothing$ w.a.h.p.

We remark that, for any $k \times m$ matrix T, any feasible solution for $\mathcal{F}$ is also a feasible solution for $T\mathcal{F}$ by linearity. So the real issue is proving that if $\mathcal{F}$ is infeasible then $T\mathcal{F}$ is also infeasible w.a.h.p. This is where we exploit the fact that T is a random projector. More precisely, we prove the following statements about linear infeasibility w.a.h.p.:

1. a nonzero vector is randomly projected to a nonzero vector;
2. if x is not feasible in $\mathcal{F}$, then it is not feasible in $T\mathcal{F}$;
3. if x is not feasible in $\mathcal{F}$ for all x in a finite set X, then the same follows for $T\mathcal{F}$;
4. if b is not in the convex hull of A, then Tb is not in the convex hull of TA.
5. if b is not in the cone of A, then Tb is not in the cone of TA.

The first result is actually a corollary of Lemma 1. We denote by E^{c} the complement of an event E.

COROLLARY 1. *Let T be a $k \times m$ random projector and $y \in \mathbb{R}^m$ with $y \neq 0$. Then we have*

$$\mathsf{Prob}(Ty \neq 0) \geq 1 - 2e^{-\mathcal{C}k}. \tag{8}$$

for some constant $\mathcal{C} > 0$ (independent of n, k).

Proof. For any $\varepsilon \in (0,1)$, we define the following events:

$$\begin{aligned} \mathcal{A} &= \big\{ Ty \neq 0 \big\} \\ \mathcal{B} &= \big\{ (1-\varepsilon)\|y\| \leq \|Ty\| \leq (1+\varepsilon)\|y\| \big\}. \end{aligned}$$

By Lemma 1 it follows that $\mathsf{Prob}(\mathcal{B}) \geq 1 - 2e^{-\mathcal{C}\varepsilon^2 k}$ for some constant $\mathcal{C} > 0$ independent of m, k, ε. On the other hand, $\mathcal{A}^{\mathsf{c}} \cap \mathcal{B} = \emptyset$, since otherwise, for any $\varepsilon \in (0,1)$ there is a mapping T_1 such that $T_1(y) = 0$ and $(1-\varepsilon)\|y\| \leq \|T_1(y)\|$, which altogether imply that $y = 0$ (a contradiction). Therefore, $\mathcal{B} \subseteq \mathcal{A}$, and we have $\mathsf{Prob}(\mathcal{A}) \geq \mathsf{Prob}(\mathcal{B}) \geq 1 - 2e^{-\mathcal{C}\varepsilon^2 k}$. This holds for all $0 < \varepsilon < 1$, so $\mathsf{Prob}(\mathcal{A}) \geq 1 - 2e^{\mathcal{C}k}$. □

The following theorem settles points 2-3 above.

THEOREM 2. *Let T be a $k \times m$ random projector and $\mathcal{F} \equiv \{x \geq 0 \mid Ax = b\}$ with A an $m \times n$ matrix. Then for any $x \in \mathbb{R}^n$, we have:*

(i) If $b = \sum\limits_{j=1}^{n} x_j A_j$ then $Tb = \sum\limits_{j=1}^{n} x_j T A_j$;

(ii) If $b \neq \sum_{j=1}^{n} x_j A_j$ then $\mathsf{Prob}\left[Tb \neq \sum_{j=1}^{n} x_j T A_j \right] \geq 1 - 2e^{-\mathcal{C}k}$;

(iii) If $b \neq \sum_{j=1}^{n} x_j A_j$ for all $x \in X \subseteq \mathbb{R}^n$, where $|X|$ is finite, then

$$\mathsf{Prob}\left[\forall x \in X \ Tb \neq \sum_{j=1}^{n} x_j T A_j \right] \geq 1 - 2|X|e^{-\mathcal{C}k};$$

for some constant $\mathcal{C} > 0$ (independent of n, k).

Proof. Point (i) follows by linearity of T, and (ii) by applying Cor. 1 to $Ax - b$. For (iii), the union bound on (ii) yields:

$$\begin{aligned} \mathsf{Prob}\left[\forall x \in X \ Tb \neq \sum_{j=1}^{n} x_j T A_j \right] &= \mathsf{Prob}\left[\bigcap_{x \in X} \Big\{ Tb \neq \sum_{j=1}^{n} x_j T A_j \Big\} \right] \\ = 1 - \mathsf{Prob}\left[\bigcup_{x \in X} \Big\{ Tb \neq \sum_{j=1}^{n} x_j T A_j \Big\}^{\mathsf{c}} \right] &\geq 1 - \sum_{x \in X} \mathsf{Prob}\left[\Big\{ Tb \neq \sum_{j=1}^{n} x_j T A_j \Big\}^{\mathsf{c}} \right] \\ \text{[by (ii)]} \qquad &\geq 1 - \sum_{x \in X} 2e^{-\mathcal{C}k} = 1 - 2|X|e^{-\mathcal{C}k}, \end{aligned}$$

as claimed. □

Thm. 2 can be used to project certain types of integer programs. It also gives us an indication to why estimating the probability that $Tb \notin \mathsf{cone}(A)$ is not straightforward. This event can be written as an intersection of uncountably many events $\{Tb \neq \sum_{j=1}^{n} x_j T A_j\}$ where $x \in \mathbb{R}^n_+$. Even if each of these occurs w.a.h.p., their intersection might still be small. As these events are dependent, however, we shall show that there is hope yet.

3.1. Convex hull feasibility Next, we show that if the distance between a point and a closed set is positive, it remains positive with high probability after applying a random projection. We consider the convex hull membership problem: given vectors $b, A_1, \ldots, A_n \in \mathbb{R}^m$, decide whether $b \in \mathsf{conv}(\{A_1, \ldots, A_n\})$. Although we do not use this result directly in the following, we believe it is of independent interest.

We have the following result:

Proposition 2. *Given $A_1, \ldots, A_n \in \mathbb{R}^m$, let $C = \mathsf{conv}(\{A_1, \ldots, A_n\})$, $b \in \mathbb{R}^m$ such that $b \notin C$, $d = \min_{x \in C} \|b - x\|$ and $D = \max_{1 \le j \le n} \|b - A_j\|$. Let $T : \mathbb{R}^m \to \mathbb{R}^k$ be a random projector. Then*

$$\mathsf{Prob}\big[Tb \notin TC\big] \ge 1 - 2n^2 e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k} \tag{9}$$

for some constant $\mathcal{C}$ (independent of m, n, k, d, D) and $\varepsilon < \frac{d^2}{D^2}$.

This proposition is based on the fact that any vector in C can be represented as a convex combination of A_j for $j \in \{1, \ldots, n\}$. Since the distance between Tb and all TA_j is positive with high probability and the total weight, i.e. $\sum_{j=1}^n \lambda_j$, is always 1, we can bound the distance between Tb and all vectors in TC.

Proof . Let S_ε be the event that both

$$(1 - \varepsilon)\|x - y\|^2 \le \|T(x - y)\|^2 \le (1 + \varepsilon)\|x - y\|^2$$

and

$$(1 - \varepsilon)\|x + y\|^2 \le \|T(x + y)\|^2 \le (1 + \varepsilon)\|x + y\|^2$$

hold for all $x, y \in \{0, b - A_1, \ldots, b - A_n\}$. Assume S_ε occurs. Then for all real $\lambda_j \ge 0$ with $\sum\limits_{j=1}^n \lambda_j = 1$, we have:

$$\begin{aligned}
&\|Tb - \sum_{j=1}^n \lambda_j T A_j\|^2 = \|\sum_{j=1}^n \lambda_j T(b - A_j)\|^2 \quad \text{(by linearity of } T \text{ and } \textstyle\sum_j \lambda_j = 1) \\
&= \sum_{j=1}^n \lambda_j^2 \|T(b - A_j)\|^2 + 2 \sum_{1 \le i < j \le n} \lambda_i \lambda_j \langle T(b - A_i), T(b - A_j)\rangle \\
&= \sum_{j=1}^n \lambda_j^2 \|T(b - A_j)\|^2 + \frac{1}{2} \sum_{1 \le i < j \le n} \lambda_i \lambda_j \Big(\|T(b - A_i + b - A_j)\|^2 - \|T(A_i - A_j)\|^2 \Big).
\end{aligned} \tag{10}$$

Here the last equality follows from the fact that $\langle x, y\rangle = \frac{1}{4}(\|x + y\|^2 - \|x - y\|^2)$ for all vectors x, y. Moreover, since S_ε occurs, we have

$$\|T(b - A_j)\|^2 \ge (1 - \varepsilon)\|b - A_j\|^2$$

and

$$\|T(b - A_i + b - A_j)\|^2 - \|T(A_i - A_j)\|^2 \ge (1 - \varepsilon)\big\|b - A_i + b - A_j\big\|^2 - (1 + \varepsilon)\|A_i - A_j\|^2$$

for all $1 \le i < j \le n$. Therefore, the RHS in (10) is greater than or equal to

$$\begin{aligned}
&(1 - \varepsilon) \sum_{j=1}^n \lambda_j^2 \|b - A_j\|^2 + \frac{1}{2} \sum_{1 \le i < j \le n} \lambda_i \lambda_j \Big((1 - \varepsilon)\big\|b - A_i + b - A_j\big\|^2 - (1 + \varepsilon)\|A_i - A_j\|^2 \Big) \\
&= \|b - \sum_{j=1}^n \lambda_j A_j\|^2 - \varepsilon \Big(\sum_{j=1}^n \lambda_j^2 \|b - A_j\|^2 + \frac{1}{2} \sum_{1 \le i < j \le n} \lambda_i \lambda_j (\|b - A_i + b - A_j\|^2 + \|A_i - A_j\|^2) \Big) \\
&= \|b - \sum_{j=1}^n \lambda_j A_j\|^2 - \varepsilon \Big(\sum_{j=1}^n \lambda_j^2 \|b - A_j\|^2 + \sum_{1 \le i < j \le n} \lambda_i \lambda_j (\|b - A_i\|^2 + \|b - A_j\|^2) \Big).
\end{aligned}$$

From the definitions of d and D, we have $\|b - \sum_{j=1}^n \lambda_j A_j\|^2 \geq d^2$ and $\|b - A_i\| \leq D^2$ for all $1 \leq i \leq n$. Therefore:

$$\|Tb - \sum_{j=1}^{n} \lambda_j TA_j\|^2 \geq d^2 - \varepsilon D^2 \Big(\sum_{j=1}^{n} \lambda_j^2 + 2 \sum_{1 \leq i < j \leq n} \lambda_i \lambda_j \Big) = d^2 - \varepsilon D^2 \Big(\sum_{j=1}^{n} \lambda_j \Big)^2 = d^2 - \varepsilon D^2 > 0$$

due to the fact that $\sum_{j=1}^n \lambda_j = 1$ and the choice of $\varepsilon < \frac{d^2}{D^2}$.

Now, since $\|Tb - \sum\limits_{j=1}^{n} \lambda_j TA_j\|^2 > 0$ for all choices of $\lambda \geq 0$ with $\sum_{j=1}^n \lambda_j = 1$, it follows that $Tb \notin \mathsf{conv}(\{TA_1, \ldots, TA_n\})$.

In summary, if S_ε occurs, then $Tb \notin \mathsf{conv}(\{TA_1, \ldots, TA_n\})$. Thus, by Lemma 2 and the union bound,

$$\mathsf{Prob}(Tb \notin TC) \geq \mathsf{Prob}(S_\varepsilon) \geq 1 - 2\big(n + 2\binom{n}{2}\big) e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k} = 1 - 2n^2 e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$$

for some constant $\mathcal{C} > 0$. □

As an interesting aside, we remark that this proof can also be extended to show that disjoint polytopes project to disjoint polytopes with high probability.

3.2. Cone feasibility

We now deal with the last (and most relevant) result: if b is not in the cone of the columns of A, then Tb is not in the cone of the columns of TA w.a.h.p. We first define the A-norm of $x \in \mathsf{cone}(A)$ as

$$\|x\|_A = \min \Big\{ \sum_{j=1}^{n} \lambda_j \;\Big|\; \lambda \geq 0 \wedge x = \sum_{j=1}^{n} \lambda_j A_j \Big\}.$$

For each $x \in \mathsf{cone}(A)$, we say that $\lambda \in \mathbb{R}^n_+$ yields a *minimal A-representation* of x if and only if $\sum\limits_{j=1}^{n} \lambda_j = \|x\|_A$. We define $\mu_A = \max\{\|x\|_A \mid x \in \mathsf{cone}(A) \wedge \|x\| \leq 1\}$; then, for all $x \in \mathsf{cone}(A)$, we have

$$\|x\| \leq \|x\|_A \leq \mu_A \|x\|.$$

In particular $\mu_A \geq 1$. Note that μ_A serves as a measure of worst-case distortion when we move from Euclidean to $\|\cdot\|_A$ norm.

For the next result, we assume we are given an estimate of a lower bound Δ to $d = \min\limits_{x \in C} \|b - x\|$, and also (without loss of generality) that b and the column vectors of A have unit Euclidean norm.

Theorem 3. *Given an $m \times n$ matrix A and $b \in \mathbb{R}^m$ s.t. $b \notin \mathsf{cone}(A)$. Then for any $0 < \varepsilon < \frac{\Delta^2}{\mu_A^2 + 2\mu_A\sqrt{1-\Delta^2} + 1}$ and any $k \times m$ random projector T (such as one in Section 2), we have*

$$\mathsf{Prob}(Tb \notin \mathsf{cone}(TA)) \geq 1 - 2(n+1)(n+2)e^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k} \tag{11}$$

for some constant $\mathcal{C}$ (independent of m, n, k, Δ).

The proof to this theorem (below) is based on the fact that any vector x in the cone generated by A can be represented as a nonnegative combination of A_j for $j \in \{1, \ldots, n\}$. Since the distance between Tb and all TA_j is positive with high probability, so if we can bound the total weight, i.e. $\sum_{j=1}^n \lambda_j$, it is possible to bound the distance between Tb and all vectors in TC. However, vectors in the cone might have many different such representations. Therefore we choose a minimal representation, which we obtain by minimizing the corresponding total weight. This motives our definition of A-norm. With this definition, we can then estimate

$$\|Tb - Tx\|^2 \geq \|b - x\|^2 - \varepsilon(1 + \|x\|_A^2)$$

with high probability. The problem is that $\|x\|_A$ can go to infinity when $\|x\| \to \infty$, therefore the left-hand side might be negative. To overcome this difficulty, we argue that $\|b-x\|^2$ can also be scaled up when $\|x\| \to \infty$, so that it always dominates $\varepsilon(1+\|x\|_A^2)$. This is the fact we prove later in the claim that

$$\|b-x\|^2 \geq \|x\|^2 - 2\|x\|\|p\| + 1,$$

where p is the projection of b to the cone.

Proof. For any ε chosen as in the theorem statement, let S_ε be the event that both

$$(1-\varepsilon)\|x-y\|^2 \leq \|T(x-y)\|^2 \leq (1+\varepsilon)\|x-y\|^2$$

and

$$(1-\varepsilon)\|x+y\|^2 \leq \|T(x+y)\|^2 \leq (1+\varepsilon)\|x+y\|^2$$

hold for all $x, y \in \{0, b, A_1, \ldots, A_n\}$. By Lemma 1, we have

$$\mathsf{Prob}(S_\varepsilon) \geq 1 - 4\binom{n+2}{2} e^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k} = 1 - 2(n+1)(n+2)e^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k}$$

for some constant $\mathcal{C}$ (independent of m, n, k, d). We will prove that if S_ε occurs, then we have $Tb \notin \mathsf{cone}\{TA_1, \ldots, TA_n\}$. Assume that S_ε occurs. Consider an arbitrary $x \in \mathsf{cone}\{A_1, \ldots, A_n\}$ and let $\sum\limits_{j=1}^{n} \lambda_j A_j$ be a minimal A-representation of x. Then we have:

$$\begin{aligned}
&\|Tb - Tx\|^2 = \|Tb - \sum_{j=1}^{n} \lambda_j TA_j\|^2 \\
&= \|Tb\|^2 + \sum_{j=1}^{n} \lambda_j^2 \|TA_j\|^2 - 2\sum_{j=1}^{n} \lambda_j \langle Tb, TA_j\rangle + 2\sum_{1\leq i<j\leq n} \lambda_i\lambda_i \langle TA_i, TA_j\rangle \\
&= \|Tb\|^2 + \sum_{j=1}^{n} \lambda_j^2 \|TA_j\|^2 + \sum_{j=1}^{n} \frac{\lambda_j}{2}(\|T(b-A_j)\|^2 - \|T(b+A_j)\|^2) + \sum_{1\leq i<j\leq n} \frac{\lambda_i\lambda_j}{2}(\|T(A_i+A_j)\|^2 - \|T(A_i-A_j)\|^2)
\end{aligned} \tag{12}$$

Here the last equality follows by the fact that $\langle x, y\rangle = \frac{1}{4}(\|x+y\|^2 - \|x-y\|^2)$ for all vectors x, y. Moreover, since S_ε occurs, we have

$$\|Tb\|^2 \geq (1-\varepsilon)\|b\|^2, \qquad \|TA_j\|^2 \geq (1-\varepsilon)\|A_j\|^2 \quad \text{for all } 1 \leq j \leq n$$

and

$$\begin{aligned}
\|T(b-A_j)\|^2 - \|T(b+A_j)\|^2 &\geq (1-\varepsilon)\big\|b-A_j\big\|^2 - (1+\varepsilon)\|b+A_j\|^2 \\
\|T(A_i+A_j)\|^2 - \|T(A_i-A_j)\|^2 &\geq (1-\varepsilon)\|A_i+A_j\|^2 - (1+\varepsilon)\|A_i-A_j\|^2
\end{aligned}$$

for all $1 \leq i < j \leq n$. Therefore, the RHS in (12) is greater than or equal to

$$\begin{aligned}
(1-\varepsilon)\|b\|^2 + (1-\varepsilon)\sum_{j=1}^{n} \lambda_j^2 \|A_j\|^2 + \sum_{j=1}^{n} \frac{\lambda_j}{2}((1-\varepsilon)\|b-A_j\|^2 - (1+\varepsilon)\|b+A_j\|^2) \\
+ \sum_{1\leq i<j\leq n} \frac{\lambda_i\lambda_j}{2}((1-\varepsilon)\|A_i+A_j\|^2 - (1+\varepsilon)\|A_i-A_j\|^2).
\end{aligned} \tag{13}$$

Since we have assumed that $\|b\| = \|A_1\| = \ldots \|A_n\| = 1$, it can then be rewritten as

$$\begin{aligned}
&\|b - \sum_{j=1}^{n} \lambda_j A_j\|^2 - \varepsilon\left(1 + \sum_{j=1}^{n} \lambda_j^2 + 2\sum_{i=j}^{n} \lambda_j + 2\sum_{j\neq i} \lambda_i\lambda_j\right) \\
&= \|b - \sum_{j=1}^{n} \lambda_j A_j\|^2 - \varepsilon\big(1 + \sum_{j=1}^{n} \lambda_j\big)^2 \\
&= \|b - x\|^2 - \varepsilon\big(1 + \|x\|_A\big)^2 \qquad \text{(by the definition of } A\text{-norm).}
\end{aligned}$$

In summary, we have proved that, when the event S_ε occurs, then

$$\|Tb - Tx\|^2 \geq \|b - x\|^2 - \varepsilon\big(1 + \|x\|_A\big)^2. \tag{14}$$

Denote by $\alpha = \|x\|$ and let p be the orthogonal projection of b to $\mathsf{cone}\{A_1, \dots, A_n\}$, which means $\|b - p\| = \min\{\|b - x\| \mid x \in \mathsf{cone}\{A_1, \dots, A_n\}\}$. We will need to use the following claim:

Claim. For all b, x, α, p given above, we have $\|b - x\|^2 \geq \alpha^2 - 2\alpha\|p\| + 1$.
By this claim (proved later), from inequality (14), we have:

$$\begin{aligned}\|Tb - Tx\|^2 &\geq \alpha^2 - 2\alpha\|p\| + 1 - \varepsilon\big(1 + \|x\|_A\big)^2 \\ &\geq \alpha^2 - 2\alpha\|p\| + 1 - \varepsilon\big(1 + \mu_A\alpha\big)^2 \quad (\text{since } \|x\|_A \leq \mu_A\|x\|) \\ &= \big(1 - \varepsilon\mu_A^2\big)\alpha^2 - 2\big(\|p\| + \varepsilon\mu_A\big)\alpha + (1 - \varepsilon).\end{aligned}$$

The last expression can be viewed as a quadratic function with respect to α. We will prove this function is positive for all $\alpha \in \mathbb{R}$. This is equivalent to[1]

$$\begin{aligned}&\big(\|p\| + \varepsilon\mu_A\big)^2 - \big(1 - \varepsilon\mu_A^2\big)(1 - \varepsilon) < 0 \\ \Leftrightarrow\ &\big(\mu_A^2 + 2\|p\|\mu_A + 1\big)\varepsilon < 1 - \|p\|^2 \\ \Leftrightarrow\ &\varepsilon < \frac{1 - \|p\|^2}{\mu_A^2 + 2\|p\|\mu_A + 1} = \frac{d^2}{\mu_A^2 + 2\|p\|\mu_A + 1},\end{aligned}$$

which holds for the choice of ε as in the hypothesis. In conclusion, if the event S_ε occurs, then $\|Tb - Tx\|^2 > 0$ for all $x \in \mathsf{cone}\{A_1, \dots, A_n\}$, i.e. $Tx \notin \mathsf{cone}\{TA_1, \dots, TA_n\}$. Thus,

$$\mathsf{Prob}(Tb \notin TC) \geq \mathsf{Prob}(S_\varepsilon) \geq 1 - 2(n+1)(n+2)e^{-c(\varepsilon^2 - \varepsilon^3)k}$$

as claimed. The result follows since $\|p\|_2^2 + d^2 = 1$ by Pythagoras' theorem, and $\Delta \leq d$.

Proof of the claim that $\|b - x\|^2 \geq \alpha^2 - 2\alpha\|p\| + 1$*:*
If $x = 0$ then the claim is trivially true, since $\|b - x\|^2 = \|b\|^2 = 1 = \alpha^2 - 2\alpha\|p\| + 1$. Hence we assume $x \neq 0$. First consider the case $p \neq 0$. By Pythagoras' theorem, we must have $d^2 = 1 - \|p\|^2$. We denote by $z = \frac{\|p\|}{\alpha}x$, then $\|z\| = \|p\|$. Set $\delta = \frac{\alpha}{\|p\|}$, we have

$$\begin{aligned}\|b - x\|^2 &= \|b - \delta z\|^2 \\ &= (1 - \delta)\|b\|^2 + (\delta^2 - \delta)\|z\|^2 + \delta\|b - z\|^2 \\ &= (1 - \delta) + (\delta^2 - \delta)\|p\|^2 + \delta\|b - z\|^2 \\ &\geq (1 - \delta) + (\delta^2 - \delta)\|p\|^2 + \delta d^2 \\ &= (1 - \delta) + (\delta^2 - \delta)\|p\|^2 + \delta(1 - \|p\|^2) \\ &= \delta^2\|p\|^2 - 2\delta\|p\|^2 + 1 = \alpha^2 - 2\alpha\|p\| + 1.\end{aligned}$$

Next, we consider the case $p = 0$. In this case we have $b^T(x) \leq 0$ for all $x \in \mathsf{cone}\{A_1, \dots, A_n\}$. Indeed, for an arbitrary $\delta > 0$,

$$0 \leq \frac{1}{\delta}(\|b - \delta x\|^2 - 1) = \frac{1}{\delta}(1 + \delta^2\|x\|^2 - 2\delta b^T x - 1) = \delta\|x\|^2 - 2b^T x$$

which tends to $-2b^T x$ when $\delta \to 0^+$. Therefore

$$\|b - x\|^2 = 1 - 2b^T x + \|x\|^2 \geq \|x\|^2 + 1 = \alpha^2 - 2\alpha\|p\| + 1,$$

which proves the claim. □

Since cone membership is the same as LP feasibility, Thm. 3 establishes that LFPs can be randomly projected accurately w.a.h.p.

[1] Here we use the fact that a quadratic function $ax^2 + bx + c > 0$ for all $x \in \mathbb{R}$ if and only if $a > 0$ and $b^2 - 4ac < 0$.

4. Preserving optimality In this section we show that, if the projected dimension k is large enough, $v(P) \approx v(P_T)$ w.o.p (Thm. 4). We assume all along, and without loss of generality, that b, c and the columns of A have unit Euclidean norms.

The proof of Thm. 4 is divided into two main parts.

- In the first part, we write $v(P) \approx v(P_T)$ formally as "given $\delta > 0$ there is a random projector T such that $v(P) - \delta \leq v(P_T) \leq v(P)$ w.a.h.p.", formalize some infeasible LFPs which encode $v(P) - \delta$ and $v(P_T)$, and emphasize their relationship.
- In the second part, we formally argue the "overwhelming probability" by means of an $\varepsilon > 0$ (in function of δ) which ensures that the probability of $v(P) - \delta \leq v(P_T)$ approaches 1 fast enough (as a function of ε). This ε refers to the projected (infeasible) LFP of the first part, but for technical reasons we cannot simply "inherit it" from Thm. 3. Instead, from the cone of the infeasible LFP we carefully construct a new *pointed* cone which allows us to carry out a projected separation argument based on inner product preservation (Prop. 1).

Our proof assumes that the feasible region of P is non-empty and bounded. Specifically, we assume that a constant $\theta > 0$ is given such that that there exists an optimal solution x^* of P (see Eq. (1)) satisfying

$$\sum_{j=1}^{n} x_j^* < \theta. \tag{15}$$

For the sake of simplicity (and without loss of generality), we assume further that $\theta \geq 1$. This assumption is used to control the excessive flatness of the involved cones, which is required in the projected separation argument.

4.1. A cone transformation operation Before introducing Thm. 4 and its proof, we explain how to construct a pointed cone from the cone of the LFP in such a way as to preserve a certain membership property.

Given a polyhedral cone

$$\mathcal{K} = \left\{ \sum_{j \leq n} x_j C_j \;\middle|\; x \in \mathbb{R}^n_+ \right\}$$

in which $C_1, \ldots, C_n$ are column vectors of an $m \times n$ matrix C, in other words $\mathcal{K} = \mathsf{cone}(C)$. For any $u \in \mathbb{R}^m$, we consider the following transformation $\phi_{u,\theta}$, defined by:

$$\phi_{u,\theta}(\mathcal{K}) := \left\{ \sum_{j=1}^{n} x_j \left(C_j - \frac{1}{\theta} u \right) \;\middle|\; x \in \mathbb{R}^n_+ \right\}.$$

In particular, $\phi_{u,\theta}$ moves the origin in the direction u by a step $1/\theta$ (see Figure 1). For θ defined in Eq. (15), we also consider the following set

$$\mathcal{K}_\theta = \left\{ \sum_{j=1}^{n} x_j C_j \;\middle|\; x \in \mathbb{R}^n_+ \wedge \sum_{j=1}^{n} x_j < \theta \right\}.$$

$\mathcal{K}_\theta$ can be seen as a set truncated from $\mathcal{K}$ (in particular, it is not a cone anymore). We shall show that $\phi_{u,\theta}$ preserves the membership of the vector u in the "truncated cone" $\mathcal{K}_\theta$.

Lemma 3. *For any $u \in \mathbb{R}^m$, we have $u \in \mathcal{K}_\theta$ if and only if $u \in \phi_{u,\theta}(\mathcal{K})$.*

Proof. First of all, let denote by $t = 1 - \frac{1}{\theta} \sum_{j=1}^{n} x_j$.

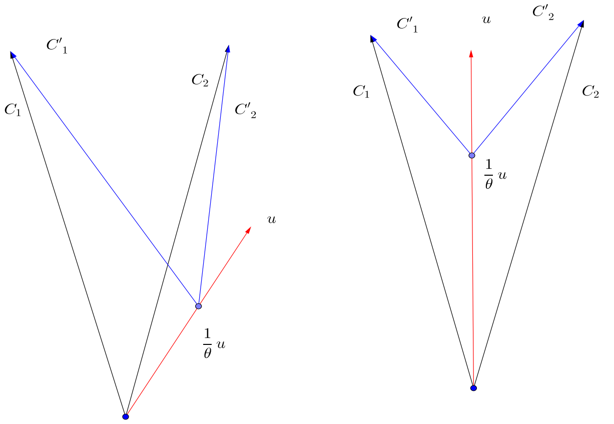

FIGURE 1. The effect of ϕ_u when u does not belong to the cone (left) and when it does (right).

($\Rightarrow$) If $u \in \mathcal{K}_\theta$, then there exists $x \in \mathbb{R}^n_+$ such that $u = \sum\limits_{j=1}^n x_j C_j$ and $\sum\limits_{j=1}^n x_j < \theta$. Then u can be written as $\sum\limits_{j=1}^n x'_j\big(C_j - \frac{1}{\theta}u\big)$ with $x' = \frac{1}{t}x$. Indeed,

$$\begin{aligned}\sum_{j=1}^n x'_j\big(C_j - \frac{1}{\theta}u\big) &= \frac{1}{t}\sum_{j=1}^n x_j\big(C_j - \frac{1}{\theta}u\big)\\ &= \frac{1}{t}\sum_{j=1}^n x_j C_j - \frac{1}{t}\big(\sum_{j=1}^n \frac{1}{\theta}x_j\big)u\\ &= \frac{1}{t}u - \frac{1}{t}\big(\sum_{j=1}^n \frac{1}{\theta}x_j\big)u\\ &= \frac{1}{t}\big(1 - \frac{1}{\theta}\sum_{j=1}^n x_j\big)u\\ &= u \qquad \text{(by definition of } t\text{)}.\end{aligned}$$

Moreover, due to the assumption that $\sum\limits_{j=1}^n x_j < \theta$, we have $x' \geq 0$. It follows that $u \in \phi_{u,\theta}(\mathcal{K})$.

($\Leftarrow$) If $u \in \phi_{u,\theta}(\mathcal{K})$, then there exists $x \in \mathbb{R}^n_+$ such that $u = \sum\limits_{j=1}^n x_j\big(C_j - \frac{1}{\theta}u\big)$. It is equivalent to $\big(1 + \frac{1}{\theta}\sum\limits_{j=1}^n x_j\big)u = \sum\limits_{j=1}^n x_j C_j$. Thus u can also be written as $\sum\limits_{j=1}^n x'_j C_j$, where $x'_j = \frac{x_j}{1+\frac{1}{\theta}\sum\limits_{i=1}^n x_i}$. Note that $\sum\limits_{j=1}^n x'_j < \theta$ because

$$\sum_{j=1}^n x'_j = \frac{\sum\limits_{j=1}^n x_j}{1 + \frac{1}{\theta}\sum\limits_{j=1}^n x_j} < \theta,$$

which implies that $u \in K_\theta$. □

Note that this result is still valid when the transformation $\phi_{u,\theta}$ is only applied to a subset of columns of C. Given any vector u and an index set $J \subseteq \{1,\ldots,n\}$, we define $\forall j \leq n$:

$$C_j^{Ju} = \begin{cases} C_j - \frac{1}{\theta}u & \text{if } j \in J \\ C_j & \text{otherwise.} \end{cases}$$

We extend $\phi_{u,\theta}$ to

$$\phi_{u,\theta}^J(\mathcal{K}) = \left\{ \sum_{j=1}^n x_j C_j^{Ju} \,\middle|\, x \in \mathbb{R}_+^n \right\} = \mathsf{cone}(C_j^{Ju} \mid 1 \leq j \leq n), \tag{16}$$

and define

$$\mathcal{K}_\theta^J = \left\{ \sum_{j=1}^n x_j C_j \,\middle|\, x \in \mathbb{R}_+^n \wedge \sum_{j\in J} x_j < \theta \right\}.$$

The following corollary can be proved in the same way as Lemma 3, in which $\phi_{u,\theta}$ is replaced by $\phi_{u,\theta}^J$.

Corollary 2. *For any vector $u \in \mathbb{R}^m$ and any index set $J \subseteq \{1,\ldots,n\}$, we have $u \in \mathcal{K}_\theta^J$ if and only if $u \in \phi_{u,\theta}^J(\mathcal{K})$.*

4.2. The main theorem Given an LFP instance $Ax = b \wedge x \geq 0$, where A is an $m \times n$ matrix and T is a $k \times m$ random projector. By Thm. 3, we know that,

$$\exists x \geq 0\,(Ax = b) \quad \Leftrightarrow \quad \exists x \geq 0\,(TAx = Tb)$$

w.a.h.p. We remark that this also holds for a $(k+h) \times m$ random projector of the form

$$\begin{pmatrix} I_h & 0 \\ \multicolumn{2}{c}{T} \end{pmatrix},$$

where T is a $k \times m$ random matrix. This allows us to claim the feasibility equivalence w.a.h.p. even when we only want to project a subset of rows of A. In the following, we will use this observation to handle constraints and objective function separately. In particular, we only project the constraints while keeping objective function unchanged.

If we add the constraint $\sum\limits_{j=1}^n x_j \leq \theta$ to the problem P_T (defined in Eq. (2)), we obtain the following:

$$P_{T,\theta} \equiv \min \left\{ c^\top x \,\middle|\, TAx = Tb \wedge \sum_{j=1}^n x_j \leq \theta \wedge x \in \mathbb{R}_+^n \right\}. \tag{17}$$

So we come to our main theorem, which asserts that the optimal objective value of P can be well-approximated by that of $P_{T,\theta}$.

Theorem 4. *Assume $\mathcal{F}(P)$ is bounded and non-empty. Let y^* be an optimal dual solution of P of minimal Euclidean norm. Given $0 < \delta \leq |v(P)|$, we have*

$$v(P) - \delta \leq v(P_{T,\theta}) \leq v(P), \tag{18}$$

with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$, where $\varepsilon = O(\frac{\delta}{\theta^2 \|y^*\|})$.

First, we will informally explain the idea of the proof. Since $v(P)$ is the optimal objective value of problem P, for any positive δ, the problem

$$Ax = b \wedge x \geq 0 \wedge c^\top x \leq v(P) - \delta.$$

is infeasible (because we can not obtain a lower objective value than $v(P)$). That problem can now be projected in such a way that it remains infeasible w.a.h.p. By rewriting this original problem in the standard form as

$$\begin{pmatrix} c^\top & 1 \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ s \end{pmatrix} = \begin{pmatrix} v(P) - \delta \\ b \end{pmatrix}, \text{ where } \begin{pmatrix} x \\ s \end{pmatrix} \geq 0, \tag{19}$$

and applying a random projection of the form

$$\left(\begin{array}{c|c} 1 & 0 \dots 0 \\ \hline 0 & \\ \dots & T \\ 0 & \end{array}\right), \text{ where } T \text{ is a } k \times m \text{ random projector,}$$

we will obtain the following problem, which is supposed to be infeasible w.a.h.p.

$$\left.\begin{aligned} cx + s &= v(P) - \delta \\ TAx &= Tb \\ s &\geq 0 \\ x &\geq 0 \end{aligned}\right\}. \tag{20}$$

The main idea is that, the prior information about the optimal solution x^* (i.e. the condition $\sum_{j=1}^n x_j^* \leq \theta$), can now be added into this new projected problem. This does not change its feasibility, but later can be used to transform the corresponding cone into the one which is easier to deal with. Therefore, w.a.h.p., the problem

$$\left.\begin{aligned} cx &\leq v(P) - \delta \\ TAx &= Tb \\ \sum_{j=1}^n x_j &\leq \theta \\ x &\geq 0 \end{aligned}\right\} \tag{21}$$

is infeasible. Hence we deduce that $cx \geq v(P) - \delta$ holds w.a.h.p. for any feasible solution x of the problem $P_{T,\theta}$, and that proves the LHS of Eq. (18). For the RHS, the proof is trivial since P_T is a relaxation of P with the same objective function. We now turn to the formal proof.

Proof. Let

$$\tilde{A} = \begin{pmatrix} c^\top & 1 \\ A & 0 \end{pmatrix}, \tilde{x} = \begin{pmatrix} x \\ s \end{pmatrix} \text{ and } \tilde{b} = \begin{pmatrix} v(P) - \delta \\ b \end{pmatrix}$$

Furthermore, let

$$\tilde{T} = \left(\begin{array}{c|c} 1 & 0 \dots 0 \\ \hline 0 & \\ \dots & T \\ 0 & \end{array}\right), \text{ where } T \text{ is a } k \times m \text{ random projector.}$$

In the rest of the proof, we prove that $\tilde{b} \notin \mathsf{cone}(\tilde{A})$ if and only if $T\tilde{b} \notin \mathsf{cone}(T\tilde{A})$ w.a.h.p.

Let J be the index set of the first n columns of $\tilde{A}$. Consider the transformation $\phi^J_{\tilde{b},\theta'}$ as defined above, using a step $\frac{1}{\theta'}$ instead of $\frac{1}{\theta}$, in which $\theta' \in (\theta, \theta+1)$. We define the following matrix:

$$A' = \left(\tilde{A}_1 - \tfrac{1}{\theta'}\tilde{b} \ \cdots \ \tilde{A}_n - \tfrac{1}{\theta'}\tilde{b} \ \tilde{A}_{n+1} \right)$$

Since Eq. (19) is infeasible, it is easy to verify that the system:

$$\left.\begin{aligned} \tilde{A}\tilde{x} &= \tilde{b} \\ \sum_{j=1}^{n} \tilde{x}_j &< \theta' \\ \tilde{x} &\geq 0 \end{aligned}\right\} \tag{22}$$

is also infeasible. It is equivalent to

$$\tilde{b} \notin \left\{ \sum_{j=1}^{n} \tilde{x}_j \tilde{A}_j \;\middle|\; \tilde{x} \in \mathbb{R}^n_+ \wedge \sum_{j\in J} \tilde{x}_j < \theta' \right\}.$$

Then, by Cor. 2, it follows that $\tilde{b} \notin \mathsf{cone}(A')$.

Let $y^* \in \mathbb{R}^m$ be an optimal dual solution of P of minimal Euclidean norm. By the strong duality theorem, we have $y^* A \leq c$ and $y^* b = v(P)$. We define

$$\tilde{y} = \begin{pmatrix} 1 \\ -y^* \end{pmatrix}.$$

We will prove that $\tilde{y} A' > 0$ and $\tilde{y}\tilde{b} < 0$. Indeed, since $\tilde{y}\tilde{A} = \begin{pmatrix} 1 \\ -y^* \end{pmatrix}^{\top} \begin{pmatrix} c^\top & 1 \\ A & 0 \end{pmatrix} = \begin{pmatrix} c - y^* A \\ 1 \end{pmatrix} \geq 0$ and $\tilde{y}\tilde{b} = v(P) - \delta - y^* b = -\delta < 0$, then we have

$$\tilde{y} A' = \begin{pmatrix} c - y^* A + \frac{\delta}{\theta'} \\ 1 \end{pmatrix} \geq \frac{\delta}{\theta'}\mathbf{1} \geq \frac{\delta}{\theta+1}\mathbf{1} \text{ and } \tilde{y}\tilde{b} = -\delta \tag{23}$$

(where $\mathbf{1}$ is the all-one vector), which proves the claim.

Now we can apply the scalar product preservation property. By Proposition 1 and the union bound, we have that

$$\forall j \leq n \quad |\, ((\tilde{T}\tilde{y})(\tilde{T}A') - \tilde{y}A')_j \,| \leq \varepsilon\eta \tag{24}$$

$$|\, (\tilde{T}\tilde{y})(\tilde{T}\tilde{b}) - \tilde{y}\tilde{b} \,| \leq \varepsilon\eta \tag{25}$$

hold with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k}$. Here, η is the normalization constant (to scale vectors to unit norm)

$$\eta = \max\left\{ \|\tilde{y}\|\,\|\tilde{b}\|, \max_{1\leq j\leq n} \|\tilde{y}\|\,\|{A_j}'\| \right\},$$

in which we can easily estimate $\eta = O(\theta\|y^*\|)$ (the proof is given at the end). Let us now fix $\varepsilon = \frac{\delta}{2(\theta+1)\eta}$. It is easy to see that

$$\varepsilon = \frac{\delta}{2(\theta+1)\eta} = O(\frac{\delta}{\theta^2\|y^*\|}).$$

Then with this choice of ε, by (23), (24) and (25), we have, with probability at least p,

$$\begin{aligned} (\tilde{T}\tilde{y})(\tilde{T}A') &\geq \tilde{y}A' - \varepsilon\eta\mathbf{1} \geq \left(\frac{\delta}{\theta+1} - \varepsilon\eta \right)\mathbf{1} \geq 0 \\ (\tilde{T}\tilde{y})(\tilde{T}\tilde{b}) &\leq \tilde{y}\tilde{b} + \varepsilon\eta \leq -\delta + \varepsilon\eta < 0, \end{aligned}$$

which then implies that the problem

$$\begin{aligned} \tilde{T}A'\tilde{x} &= \tilde{T}\tilde{b} \\ \tilde{x} &\geq 0 \end{aligned}$$

is infeasible (by Farkas' Lemma). By definition, $\tilde{T}A'\tilde{x} = \tilde{T}\tilde{A}\tilde{x} - \frac{1}{\theta'}\sum\limits_{j=1}^{n} x_j \tilde{T}\tilde{b}$, which implies that the system

$$\left.\begin{aligned} \tilde{T}\tilde{A}\tilde{x} &= \tilde{T}\tilde{b} \\ \sum_{j=1}^{n} \tilde{x}_j &< \theta' \\ \tilde{x} &\geq 0 \end{aligned}\right\}$$

is also infeasible with probability at least p (the proof is similar to that of Corollary 2). Therefore, with probability at least p, the following optimization problem:

$$\inf\left\{ c^\top x \;\middle|\; TAx = Tb \wedge \sum_{j=1}^{n} x_j < \theta' \wedge x \in \mathbb{R}^n_+ \right\}.$$

has its optimal value greater than $v(P) - \delta$. Since $\theta' > \theta$, it follows that with probability at least p, we have $v(P_{T,\theta}) \geq v(P) - \delta$, as claimed. The proof is done.

Proof of the claim that $\eta = O(\theta\|y^*\|)$: We have

$$\begin{aligned} \|\tilde{b}\|^2 &= \|b\|^2 + (v(P)-\delta)^2 \qquad \text{(by the definition of } \tilde{b}\text{)} \\ &\leq \|b\|^2 + 2(v(P))^2 + 2\delta^2 \qquad \text{(using the inequality } (x-y)^2 \leq 2x^2 + 2y^2 \text{ for all } x, y.) \\ &\leq \|b\|^2 + 4(v(P))^2 \qquad \text{(by assumption that } |\delta| \leq |v(P)|\text{)} \\ &= 1 + 4|c^\top x^*| \\ &\leq 1 + 4\|c\|_\infty \, \|x^*\|_1 \quad \text{(by Hölder inequality)} \\ &\leq 1 + 4\theta \quad \text{(since } \|c\|_\infty \leq \|c\|_2 = 1 \text{ and } \textstyle\sum x_i^* \leq \theta\text{)} \\ &\leq 5\theta \quad \text{(by the assumption that } \theta \geq 1\text{)}. \end{aligned}$$

Therefore, we conclude that

$$\eta = \max\left\{ \|\tilde{y}\| \, \|\tilde{b}\|, \; \max_{1\leq j\leq n} \; \|\tilde{y}\| \, \|A_j{}'\| \right\} = O(\theta \, \|y^*\|)$$

□

5. Solution retrieval

In this section we explain how to retrieve an approximation $\tilde{x}$ of the optimal solution x^* of problem P. Let $\delta > 0$, by Theorem 4, we can build a vector $x' \in \mathbb{R}^n_+$ such that $v(P) - \delta \leq cx' \leq v(P)$ and $TAx' = Tb$ for some $k \times m$ projection matrix T.

5.1. Infeasibility of projected solutions

We first prove that $Ax' \neq b$ almost surely, which means that the projected problem directly gives us an approximate optimal objective function value, but not the optimum itself. Let $0 \leq \nu \leq \delta$ such that $v(P_T) = v(P) - \nu$.

Let $\tilde{A} = \begin{pmatrix} c \\ A \end{pmatrix}$, $\tilde{b} = \begin{pmatrix} v(P) - \nu \\ b \end{pmatrix}$, and $\tilde{T} = \begin{pmatrix} 1 \\ T \end{pmatrix}$. We assume here that the projected solution x' (s.t. $cx' = v(P) - \nu$) is found uniformly in the projected solution set $F' = \{x \in \mathbb{R}^n_+ \mid \tilde{T}\tilde{A}x = \tilde{T}\tilde{b}\}$. We denote $F = \{x \in \mathbb{R}^n_+ \mid \tilde{A}x = \tilde{b}\}$.

Proposition 3. *Assume that* $\mathsf{cone}(A)$ *is full dimensional in* $\mathbb{R}^m$ *and that any optimal solution of* P *has at least* m *non-zero components. Let* x' *be uniformly chosen in* F'. *Then, almost surely,* $\tilde{A}x' = \tilde{b}$ *does not hold.*

Proof. If $\nu > 0$ then obviously $\tilde{A}x' = \tilde{b}$ does not hold, because otherwise, it would contradict the minimality of $v(P)$. Hence we assume in the rest of the proof that $\nu = 0$, i.e, the value of the projected problem is the same than the value of the original one.

In order to aim at a contradiction, we assume that

$$\mathsf{Prob}(x' \in F) = \mathsf{p} > 0.$$

For each $\epsilon \in \mathsf{ker}(T)$, let

$$F_\epsilon = \{x \geq 0 \mid \tilde{A}x - \tilde{b} = \epsilon\} \cap F'.$$

We will prove that there exists $d > 0$ and a family $\mathcal{V}$ of infinitely many $\epsilon \in \mathsf{ker}(\tilde{T})$ such that $\mathsf{Prob}(x' \in F_\epsilon) \geq d > 0$. Since $(F_\epsilon)_{\epsilon \in \mathcal{V}}$ is a family of disjoint sets, we deduce that $\mathsf{Prob}\left(x' \in \bigcup\limits_{\epsilon \in \mathcal{V}} F_v\right) \geq \sum\limits_{\epsilon \in \mathcal{V}} d = +\infty$, leading to a contradiction.

Claim: $\tilde{b}$ belongs to the relative interior of a facet of the $m+1$ dimensional cone, $\mathsf{cone}(\tilde{A})$.

Proof of claim. Notice first that if $\tilde{b}$ belongs to the relative interior of $\mathsf{cone}(\tilde{A})$ then we can find a feasible solution for P with a smaller cost. Hence $\tilde{b}$ belongs to a face of dimension at most m. Assume now, to aim at a contradiction, that $\tilde{b}$ belongs to the relative interior of a face of dimension $d \leq m-1$ of $\mathsf{cone}(\tilde{A})$. Then, we could write $\tilde{b}$ as a positive sum of d extreme rays, $\tilde{A}_j,\ j \in J$. Hence there exists an optimal solution x^* of P with d non-negative components. Since $d < m$ there is a contradiction.

Hence 0 belongs to a facet of $\{\tilde{A}x - \tilde{b} \mid x \geq 0\}$, and since $\mathsf{dim}(\mathsf{ker}(\tilde{T})) \geq 2$ (w.l.o.g.), then there exists a segment $[-u, u]$ (for $\|u\|$ small enough) that is contained in the intersection $\mathsf{ker}(\tilde{T}) \cap \{\tilde{A}x - \tilde{b} \mid x \geq 0\}$.

Let $\tilde{A}_j,\ j \in J$ be the rays of $\mathsf{cone}(\tilde{A})$ that belong to the same facet of $\mathsf{cone}(\tilde{A})$ as $\tilde{b}$. There exists $\bar{x} \geq 0$ such that $A\bar{x} = b$ and $\bar{x}_j > 0,\ \forall j \in J$ (because $\tilde{b}$ belongs to the relative interior of this facet). Since $[-u, u]$ belongs to this facet, there exits $\hat{x} \in \mathbb{R}^n$ such that $A\hat{x} = -u$ and such that $\hat{x}_j = 0,\ \forall j \notin J$. We can hence compute $\bar{N} > 0$ large enough such that $2\hat{x} \leq \bar{N}\bar{x}$.

For all $N \geq \bar{N}$ and for all $x \in F$, we denote $x'_N = \frac{\bar{x}+x}{2} - \frac{1}{N}\hat{x}$. Then we have $\tilde{A}x'_N = \tilde{b} - \frac{1}{N}\tilde{A}\hat{x} = \tilde{b} + \frac{u}{N}$ and $x'_N = \frac{x}{2} + (\frac{\bar{x}}{2} - \frac{\hat{x}}{N}) \geq 0$. Therefore,

$$\frac{\bar{x}+F}{2} - \frac{1}{N}\hat{x} \subseteq F_{\frac{u}{N}}$$

which implies that, for all $N \geq \bar{N}$,

$$\mathsf{Prob}(x' \in F_{\frac{u}{N}}) = \mu(F_{\frac{u}{N}}) \geq \mu(\frac{\bar{x}+F}{2}) \geq \alpha\mu(F) = \alpha\mathsf{p} > 0$$

for some constant $\alpha > 0$, where μ is a uniform measure on F'. □

5.2. Approximate solution retrieval

Let us consider y^* to be an optimal solution of the following dual problem:

$$D \equiv \max\{b^\top y \mid y^\top A \leq c \wedge y \in \mathbb{R}^m\} \tag{26}$$

and let y_T be an optimal solution of the dual of the projected problem:

$$D_T \equiv \max\{(Tb)^\top y \mid y^\top TA \leq c \wedge y \in \mathbb{R}^k\}. \tag{27}$$

Let define $y_{\mathsf{prox}} = T^\top y_T$. It is easy to see that y_{prox} is also a feasible solution for the dual problem D in (26).

In this section we will assume that the vector $b \in \mathbb{R}^m$ belongs to the relative interior of the normal cone at some vertex of the dual polyhedron. Under this assumption, the dual solution y^* is uniquely determined.

Let $C_t(y^*)$ be the tangent cone of the dual polyhedron $\mathcal{F}(D) \equiv \{y \in \mathbb{R}^m \,|\, y^\top A \le c\}$ at y^*, which is defined as

$$C_t(y^*) = \;\mathsf{closure}\left(\left\{d : \exists \lambda > 0 \text{ such that } x + \lambda d \in \mathcal{F}(D)\right\}\right)$$

In other words, $C_t(y^*)$ is the closure of the set of all feasible directions of the dual polyhedron $\mathcal{F}(D)$ at y^*. Moreover, it is a convex cone generated by a set of vectors $v^i = y^i - y^*$ where y^i are the neighboring vertices of y^* for $i \le p$. Notice that by the previous hypothesis, we have:

$$b^\top v^i < 0 \qquad \text{for all } i \le p.$$

For each $1 \le i \le p$, let α_i denote the angle between the vectors $-b$ and v^i. Let denote by

$$\alpha^* \in \operatorname*{arg\,min}_{\alpha_i,\ldots,\alpha_p} \cos(\alpha_i)$$

We first prove the following lemma, which states that y_{prox} is approximately close to y^*.

Lemma 4. *For any $\varepsilon > 0$, there is a constant $\mathcal{C}$ such that:*

$$\|y^* - y_{\mathsf{prox}}\|_2 \le \frac{\mathcal{C}\theta^2\varepsilon}{\cos(\alpha^*)\|b\|_2}\|y^*\|_2 \tag{28}$$

with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k}$

Proof. By definition, y_{prox} is also a feasible solution for the dual problem D. Furthermore, by Theorem 4, there is a constant $\mathcal{C}$ such that:

$$b^\top y_{\mathsf{prox}} \ge b^\top y^* - \mathcal{C}\theta^2\varepsilon\|y^*\|_2 \tag{29}$$

with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2-\varepsilon^3)k}$.

Since $y_{\mathsf{prox}} - y^*$ belongs to the tangent cone $C_t(y^*)$, there exists non-negative scalars λ_i (for $i \le p$) such that $y_{\mathsf{prox}} - y^* = \sum\limits_{i=1}^{p} \lambda_i v^i$. Hence

$$\|y^* - y_{\mathsf{prox}}\|_2 = \|\sum_{i=1}^{p} \lambda_i v^i\|_2 \le \sum_{i=1}^{p} \lambda_i \|v^i\|_2.$$

By equation (29), we have also

$$\mathcal{C}\theta^2\varepsilon\|y^*\|_2 \ge b^\top(y^* - y_{\mathsf{prox}}) = \sum_{i=1}^{p} \lambda_i(-b^\top v^i) \quad (\text{we recall that } -b^\top v^i > 0 \text{ for all } i)\ .$$

Let us consider the following LP:

$$\left.\begin{array}{l} \max \sum\limits_{i=1}^{p} \lambda_i \|v^i\|_2 \\ \sum\limits_{i=1}^{p} \lambda_i(-b^\top v^i) \le \mathcal{C}\theta^2\varepsilon\|y^*\|_2 \\ \lambda \ge 0. \end{array}\right\} \tag{30}$$

The LP above is a simple continuous knapsack problem whose solution can be computed easily by a greedy algorithm: let j be such that $\frac{\|v^j\|_2}{-b^\top v^j} \ge \frac{\|v^i\|_2}{-b^\top v^i}$ for all $i \in \{1,\ldots,p\}$, then

$$\frac{\|v^j\|_2}{-b^\top v^j}\mathcal{C}\theta^2\varepsilon\|y^*\|_2 = \frac{1}{\cos(\alpha^*)\|b\|_2}\mathcal{C}\theta^2\varepsilon\|y^*\|_2$$

is the optimal value of (30). The lemma is proved. □

We consider the following algorithm which retrieves an approximate solution for the original LP from an optimal basis of the projected problem.

Algorithm 1 Retrieving an approximate solution of P

```
Let $y_T$ be the associated basic dual solution of the projected dual problem $(D_T)$.
Define $y_{\mathsf{prox}} = T^\top y_T$
for all $1 \le j \le n$ do
    $z_j := \frac{c_j - A_j^\top y_{\mathsf{prox}}}{\|A_j\|_2}$
Let $\mathcal{B}$ be the set of indices $j$ corresponding to the $m$ smallest values of $z_j$.
return $x := A_{\mathcal{B}}^{-1} b$.
```

Notice that, for all $1 \le j \le n$, $z_j := \frac{c_j - A_j^\top y_{\mathsf{prox}}}{\|A_j\|_2}$ is the distance between y_{prox} and the hyperplane defined by $A_j^\top y = c_j$. Hence, Algorithm 1 searches for the m facets of the dual polyhedron that are the closest to y_{prox} and return the corresponding basis.

Let $\mathcal{B}^*$ be the optimal basis. We consider the shortest distance from y^* to any hyperplane $A_j^\top y = c_j$ for $j \notin \mathcal{B}^*$:

$$d^* = \min_{j \notin \mathcal{B}^*} \frac{c_j - A_j^\top y^*}{\|A_j\|_2}$$

Proposition 4. *Assume that the LP problem P satisfies the following two assumptions:*

(a) there is no degenerated vertex in the dual polyhedron.

(b) the vector $b \in \mathbb{R}^m$ belongs to the relative interior of the normal cone at some vertex of the dual polyhedron.

If

$$\frac{\mathcal{C}\theta^2\varepsilon}{\cos(\alpha^*)\|b\|_2}\|y^*\|_2 < \frac{d^*}{2},$$

where $\mathcal{C}$ is the universal constant in Lemma 4, then with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$, the Algorithm 1 returns an optimal basis solution.

Proof. By Lemma 4, we have that with probability at least $p = 1 - 4ne^{-\mathcal{C}(\varepsilon^2 - \varepsilon^3)k}$,

$$\|y^* - y_{\mathsf{prox}}\|_2 \le \frac{\mathcal{C}\theta^2\varepsilon}{\cos(\alpha^*)\|b\|_2}\|y^*\|_2$$

Let $\mathcal{B}^*$ be the optimal basis. Since $\|y^* - y_{\mathsf{prox}}\|_2 < \frac{d^*}{2}$, We deduce that for all $j \in \mathcal{B}^*$, $z_j \le \|y^* - y_{\mathsf{prox}}\|_2 < \frac{d^*}{2}$.

Now, let us consider $j \notin \mathcal{B}^*$. We have $z_j \ge \frac{d}{2}$, otherwise y^* would be at a distance less than d^* from $A_j^\top y = c_j$. Since y^* is non-degenerated we have $d^* > 0$. This ends the proof. □

Note that both assumptions (a)-(b) in Prop. 4 hold almost surely for random instances.

6. Computational complexity

The main aim of this paper is that of proving that random projections can be applied to the given LP P with some probabilistic bounds on feasibility and optimality errors. The projected LP P_T can be solved by any method, e.g. simplex or interior point. Formally, we envisage the following the solution methodology:

1. sample a random projection matrix T;

2. perform the multiplication $T(A, b)$;
3. solve P_T (Eq. (2));
4. retrieve a solution for P,

where (A, b) is the $m \times (n+1)$ matrix consisting of A with the column b appended.

A very coarse computational complexity estimation is as follows: we assume computing each component of T takes $O(1)$, so computing T is $O(km)$. The best practical algorithm for serial matrix multiplication is only very slightly better than the naive algorithm, which takes $O(kmn) = O(mn \log n)$, but more efficient parallel and distributed algorithms exist. For solution retrieval, Alg. 1 runs in time $O(km + mn + n \log n + m^2) = O(n(m + \log n))$. The complexity $O(mn \log n)$ of matrix multiplication therefore dominates the complexity of sampling.

The last step, solution retrieval, is essentially dominated by taking the inverse of the $m \times m$ matrix $A_{\mathcal{B}}$ in Alg. 1, which we can assume to have complexity $O(m^3)$.

We focus our discussion on the most computationally costly step, i.e. that of solving the projected LP P_T. Exact polynomial-time methods for LP, such as the ellipsoid method or the interior point method, have complexity estimates ranging from $O(n^4 L)$ to $O(\frac{n^3}{\log n} L)$, where $L = \sum_{i=0}^{m} \sum_{j=0}^{n} \lceil \log(|a_{ij}| + 1) + 1 \rceil$, $a_{i0} = b_i$ for all $i \le m$, and $a_{0j} = c_j$ for all $j \le n$ [26].

Obviously, these LP complexity bounds are impacted by replacing the number m of rows in P by the corresponding number $k = O(\ln n)$ in P_T. Also note that, since $m \le n$, the complexity of solving an LP always exceeds (asymptotically) the complexity of the other steps. So the overall worst-case asymptotic complexity of our solution methodology does not change with respect to solving the original LP. On the other hand, m appears implicitly as part of L. If we assume we can write L as mL' for some L', then the complexity goes from $O(\frac{n^3}{\log n} mL')$ to $O(\frac{n^3}{\log n} (\ln n) L') = O(n^3 L')$.

The simplex method has exponential time complexity in the worst case. On the other hand, its average complexity is $O(mn^4)$ [5, Eq. (0.5.15)] in terms of the number of pivot steps, each taking $O(m^2 \tilde{L})$ in a naive implementation [24], where $\tilde{L}$ represents a factor due to the encoding length (assumed multiplicative). This yields an overall average complexity bound $O(m^3 n^4 \tilde{L})$. Replacing m by $O(\ln n)$ yields an improvement $O(n^4 (\ln n)^3 \tilde{L})$.

7. Computational results

A sizable majority of works on the applications of the JLL are theoretical in nature (with some exceptions, e.g. [27, 28]). In this section we provide some empirical evidence that our ideas show a rather solid promise of practical applicability.

We started our empirical study by considering the NetLib public LP instance library [23], but it turns out that its instances are too small and sparse to yield any CPU improvement. We therefore decided to generate and test a set of random LP instances in standard form. Our test set consists of 360 infeasible LPs and 360 feasible LPs. We considered pairs (m, n) as shown in Table 1. For each (m, n) we test constraint matrix densities in $\mathsf{dens} \in \{0.1, 0.3, 0.5, 0.7\}$. For each triplet (m, n, dens) we generate 10 instances where each component of the constraint matrix A is sampled from a uniform distribution on $[0, 1]$. The objective function vector is always $c = \mathbf{1}$. Infeasible instances are generated using Farkas' lemma: we sample a dual solution vector y such that $yA \ge 0$ and then choose b such that $by < 0$. Feasible instances are generated by sampling a primal solution vector x and letting $b = Ax$.

		m	
	500	1000	1500
	600	1200	1800
n	700	1400	2100
	800	1600	2400

Table 1. Instance sizes.

We employ Achlioptas random projectors in order to decrease the density of the projected constraint matrix. One of the foremost difficulties in using random projections in practice is that the theory behind them gives no hint as regards the "universal constants", e.g. $\mathcal{C}$ and the constant implicit in the definition of k as $O(\frac{1}{\varepsilon^2}\ln n)$. In theory, one should be able to work out appropriate values of ε and of the number σ of samplings of the random projector T for the problems at hand. In practice, following the theory will yield such small ε and large σ values that the smallest LPs where our methodology becomes efficient will be expected to have billions of rows, defying all computation on modest hardware such as today's laptops. In fact, we are defending the point of view that random projections are useful in day-to-day work involving large but not necessarily huge LPs and common hardware platforms. For such LPs, a lot of guesswork and trial-and-error is needed. In our computational results we use $k = \frac{1.8}{\varepsilon^2}\ln n$ after an indication found in [27], $\varepsilon = 0.2$ after testing some values between 0.1 and 0.3, and $\sigma = 1$ again after some testing. The choice $\sigma = 1$ implies that, occasionally, a few pairwise distances might fall outside their bounds; but enforcing *every* pairwise distance to satisfy the JLL requires excessive amounts of samplings of T. Besides, concentration of measure ensures that very few pairwise distances will be projected wrong w.a.h.p.

All results are obtained using a Julia [4] JuMP [18] script calling the CPLEX [13] barrier solver (without crossover) on four virtual cores of a dual core Intel i7-7500U CPU at 2.70GHz with 16GB RAM (we remark that Julia is a just-in-time compiled language, so aside from a small lag to initially compile the script, CPU times should be similar to compiled rather than interpreted programs). The CPLEX barrier solver is, in our opinion, the solver of choice when solving very large and possibly dense LPs — our preliminary tests with the simplex method showed repeated failures due to excessive resource usage (both CPU and RAM), and high standard deviations in evaluating the computational advantage between original and projected problems. Eliminating the crossover phase is a choice we made after some experimentation with these instances. Some preliminary results show that this choice may need to be re-evaluated when solving problems with different structures.

7.1. Infeasible instances We benchmark infeasible instances on CPU time and accuracy. The latter is expressed in terms of mismatches: i.e., an infeasible original LP that is mapped into a feasible projected LP (recall that the converse can never happen by linearity). The results are shown in Table 2. Each line is obtained as an average over the 10 instances with same $m, n,$ dens. We denote by m the number of rows, by n the number of columns, and by dens the properties of the constraint matrix A. We then report the number of rows k in the projected problem, the time orgCPU taken to solve the original LP, the time prjCPU taken to solve the projected LP, and the accuracy acc ("zero" means that no instance was incorrectly classified as feasible in the projection). While for smaller instances the proposed methodology is not competitive as regards the CPU time, the trend clearly shows that the larger the size of the orginal LP, the higher the chances of our methodology being faster, in accordance with theory. We remark that prjCPU is the sum of the times taken to sample T, to perform the matrix multiplication TA, and to solve the projected problem.

7.2. Feasible instances Feasible instances are benchmarked on CPU time as well as on three discrepancy measures to ascertain the quality of the approximated solution x^* of the projected LP. In particular, we look at feasibility with respect to both $Ax = b$ and $x \geq 0$, as well as at the optimality gap between the approximate and the guaranteed optimal objective function value. Unfortunately, we found very high errors in the application of the solution retrieval method in Alg. 1, which at this time we are only able to justify by claiming our test LPs are "too small" (but we are looking for a more detailed reason — for example, numerical errors leading to values close to zero might yield an invalid basis). We therefore also tested a different solution retrieval method

m	n	dens	k	orgCPU	prjCPU	acc
500	600	0.1	289	**2.40**	2.66	0.0
500	600	0.3	289	**2.15**	2.80	0.0
500	600	0.5	289	**2.48**	2.95	0.0
500	600	0.7	289	**2.91**	3.12	0.0
500	700	0.1	296	**2.46**	2.99	0.0
500	700	0.3	296	**2.24**	2.93	0.0
500	700	0.5	296	**2.72**	3.34	0.0
500	700	0.7	296	3.49	**3.38**	0.0
500	800	0.1	302	**2.01**	3.11	0.0
500	800	0.3	302	**2.35**	3.17	0.0
500	800	0.5	302	**2.95**	3.58	0.0
500	800	0.7	302	**3.60**	3.95	0.0
1000	1200	0.1	321	5.47	**4.50**	0.0
1000	1200	0.3	321	6.92	**5.76**	0.0
1000	1200	0.5	321	9.54	**6.87**	0.0
1000	1200	0.7	321	13.75	**7.79**	0.0
1000	1400	0.1	327	**5.34**	5.40	0.0
1000	1400	0.3	327	7.89	**6.48**	0.0
1000	1400	0.5	327	12.02	**8.47**	0.0
1000	1400	0.7	327	20.93	**9.73**	0.0
1000	1600	0.1	333	**5.64**	6.29	0.0
1000	1600	0.3	333	8.26	**8.23**	0.0
1000	1600	0.5	333	13.20	**10.15**	0.0
1000	1600	0.7	333	20.26	**13.34**	0.0
1500	1800	0.1	339	**7.40**	8.04	0.0
1500	1800	0.3	339	14.38	**10.84**	0.0
1500	1800	0.5	339	24.83	**13.97**	0.0
1500	1800	0.7	339	41.98	**19.02**	0.0
1500	2100	0.1	346	**7.98**	10.05	0.0
1500	2100	0.3	346	17.27	**12.20**	0.0
1500	2100	0.5	346	33.35	**16.27**	0.0
1500	2100	0.7	346	66.81	**19.72**	0.0
1500	2400	0.1	352	**8.52**	13.54	0.0
1500	2400	0.3	352	20.00	**17.78**	0.0
1500	2400	0.5	352	39.01	**24.75**	0.0
1500	2400	0.7	352	65.85	**31.95**	0.0

Table 2. Results on infeasible instances.

based on the pseudoinverse: it consists in replacing $A_{\mathcal{B}}x = b$ (see last line of Alg. 1) by the reduced system $A_{\mathcal{H}}^{\top} A_{\mathcal{H}} x = A_{\mathcal{H}}^{\top} b$, where $\mathcal{H}$ is a basis of the projected problem P_T (the reconstruction of the full solution from the projected basic components is heuristic). Accordingly, we present two sets of statistics for feasible instances: one labelled "1", referring to Alg. 1, and the other labelled "2", referring to the pseudoinverse variant.

The results on the feasible instances are given in Table 2. Again, each line is obtained as an average over the 10 instances with same $m, n,$ dens. The CPU time comparison takes three columns: orgCPU refers to the time taken by CPLEX to solve the original LP; prjCPU1 is the sum of the times taken to sample T, multiply T by A, solve the projected LP, and retrieve the original solution by Alg. 1; and prjCPU2 is the same as prjCPU1 but using the solution retrieval method based on the pseudoinverse. The solution quality is evaluated in the six columns feas1, feas2 (verifying feasibility with respect to $Ax = b$ using the two retrieval methods), neg1, neg2 (verifying feasibility with respect to $x \geq 0$ using the two retrieval methods), and obj1, obj2 (evaluating the optimality gap using the two retrieval methods), defined as follows:

- $\mathsf{feas} = \frac{1}{\|b\|_1} \sum_{i \leq m} |A^i x^* - b_i|$;

- neg $= \frac{1}{\|x^*\|_1} \sum\limits_{x^*_j<0} |x^*_j|$;
- obj $= \frac{|v(P)-v(P_T)|}{|v(P)|}$.

The results are presented in Table 3. Again, we see an encouraging trend showing that the CPU time for creating and solving the projected LP becomes smaller than the time taken to solve the original LP as size and density increase. According to our theoretical development, increasing size/density further will give a definite advantage to our methodology based on random projections. It is clear that feasibility w.r.t. $Ax = b$ is never a problem. On the other hand, feasibility w.r.t. non-negativity is an issue, expected with the pseudoinverse-based solution retrieval method, but not necessarily with Alg. 1. After checking it (and its implementation) multiple times, we came to two possible conclusions: (i) that our arbitrary choice of universal constants is wrong for Alg. 1, which would require larger instances than those we tested in order to work effectively; (ii) that the choice of the basis $\mathcal{B}$ in Alg. 1 is heavily affected by numerical errors, and therefore wrong. We have been unable to establish which of these reasons is most impactful, and delegate this investigation to future work. For the time being, we propose the pseudoinverse variant as the method of choice.

m	n	dens	k	orgCPU	prjCPU1	prjCPU2	feas1	feas2	neg1	neg2	obj1	obj2
500	600	0.1	289	**2.42**	9.97	6.76	0.000	0.000	0.437	**0.033**	0.079	**0.055**
500	600	0.3	289	**2.41**	10.24	7.08	0.000	0.000	0.442	**0.035**	0.029	0.027
500	600	0.5	289	**3.06**	10.53	7.37	0.000	0.000	0.444	**0.037**	0.023	0.020
500	600	0.7	289	**3.89**	10.81	7.72	0.000	0.000	0.454	**0.036**	0.042	**0.014**
500	700	0.1	296	**2.53**	10.41	7.11	0.000	0.000	0.467	**0.039**	0.246	**0.050**
500	700	0.3	296	**2.46**	10.72	7.58	0.000	0.000	0.453	**0.045**	0.068	**0.025**
500	700	0.5	296	**3.43**	11.10	7.97	0.000	0.000	0.475	**0.043**	0.065	**0.017**
500	700	0.7	296	**4.45**	11.40	8.45	0.000	0.000	0.468	**0.038**	0.028	**0.012**
500	800	0.1	302	**2.01**	10.67	7.58	0.000	0.000	0.472	**0.059**	0.102	**0.045**
500	800	0.3	302	**2.55**	11.10	8.02	0.000	0.000	0.463	**0.060**	0.053	**0.023**
500	800	0.5	302	**3.69**	11.60	8.48	0.000	0.000	0.474	**0.061**	0.068	**0.015**
500	800	0.7	302	**5.03**	12.03	9.02	0.000	0.000	0.473	**0.054**	0.044	**0.011**
1000	1200	0.1	321	**6.49**	14.03	10.04	0.000	0.000	0.466	**0.012**	**0.036**	0.067
1000	1200	0.3	321	**9.16**	15.82	11.61	0.000	0.000	0.468	**0.012**	0.054	**0.030**
1000	1200	0.5	321	14.71	17.52	**13.46**	0.000	0.000	0.487	**0.013**	0.277	**0.021**
1000	1200	0.7	321	26.89	19.45	**14.44**	0.000	0.000	0.464	**0.013**	0.092	**0.014**
1000	1400	0.1	327	**6.88**	15.54	11.50	0.000	0.000	0.484	**0.013**	0.222	**0.058**
1000	1400	0.3	327	**10.34**	17.05	12.91	0.000	0.000	0.495	**0.016**	0.411	**0.026**
1000	1400	0.5	327	22.80	19.85	**16.23**	0.000	0.000	0.488	**0.013**	0.144	**0.016**
1000	1400	0.7	327	34.73	21.64	**16.47**	0.000	0.000	0.484	**0.013**	0.111	**0.012**
1000	1600	0.1	333	**7.16**	16.98	12.93	0.000	0.000	0.487	**0.021**	0.857	**0.056**
1000	1600	0.3	333	**11.39**	20.11	15.93	0.000	0.000	0.480	**0.016**	0.102	**0.021**
1000	1600	0.5	333	25.44	22.42	**18.73**	0.000	0.000	0.486	**0.017**	0.073	**0.014**
1000	1600	0.7	333	40.84	26.31	**21.28**	0.000	0.000	0.483	**0.016**	0.066	**0.010**
1500	1800	0.1	339	**9.77**	21.64	15.68	0.000	0.000	0.479	**0.005**	0.069	0.064
1500	1800	0.3	339	20.81	26.33	**18.89**	0.000	0.000	0.477	**0.004**	0.042	**0.027**
1500	1800	0.5	339	42.95	29.95	**22.36**	0.000	0.000	0.473	**0.004**	0.054	**0.018**
1500	1800	0.7	339	74.23	35.63	**27.82**	0.000	0.000	0.472	**0.005**	0.016	0.013
1500	2100	0.1	346	**10.38**	24.78	19.02	0.000	0.000	0.485	**0.007**	0.095	**0.057**
1500	2100	0.3	346	25.74	29.22	**21.88**	0.000	0.000	0.487	**0.007**	0.156	**0.022**
1500	2100	0.5	346	52.21	34.06	**26.06**	0.000	0.000	0.483	**0.007**	0.046	**0.015**
1500	2100	0.7	346	90.18	36.81	**29.58**	0.000	0.000	0.487	**0.005**	0.064	**0.010**
1500	2400	0.1	352	**11.26**	27.90	22.12	0.000	0.000	0.485	**0.006**	0.121	**0.050**
1500	2400	0.3	352	29.85	35.97	**28.58**	0.000	0.000	0.485	**0.006**	0.134	**0.019**
1500	2400	0.5	352	61.25	42.47	**34.99**	0.000	0.000	0.489	**0.006**	0.253	**0.011**
1500	2400	0.7	352	104.58	49.98	**43.00**	0.000	0.000	0.492	**0.006**	0.126	**0.008**

Table 3. Results on feasible instances.

8. An application to error correcting codes In this section we showcase an application of our methodology to a problem of error correcting coding and decoding [21, §8.5].

A binary word w of length m can be encoded as a word z of length n (with $m < n$) such that $z = Qw$ where Q is an $n \times m$ real matrix, which we assume to have rank m. After transmission on an analogue noisy channel the other party receives $\bar{z}$. We assume $\bar{z} = z + \bar{x}$, where the transmission error $\bar{x}_j$ on the j-th character is uniformly distributed in $[-\delta, \delta]$ for some given $\delta > 0$ with some given (reasonably small) probability $\epsilon > 0$, and $\bar{x}_j = 0$ with probability $1 - \epsilon$. In other words, x is a sparse vector with density ϵ.

The decoding of $\bar{z}$ into w is carried out as follows. We find an $m \times n$ matrix A orthogonal to Q (so $AQ = 0$), we compute $b = A\bar{z}$ and note that

$$b = A\bar{z} = A(z + x) = A(Qw + x) = AQw + Ax = Ax.$$

If the system $Ax = b$ can be solved, we can find $z' = \bar{z} - x$, and recover w using the projection matrix $(Q^\top Q)^{-1} Q^\top$ followed by rounding:

$$w = \lfloor (Q^\top Q)^{-1} Q^\top z' \rceil.$$

The protocol rests on finding a sparse solution of the under-determined linear system $Ax = b$. Minimizing the number of non-zero components of a vector that also satisfies $Ax = b$ is known as "zero-norm minimization", and is **NP**-hard [22]. In a celebrated discovery later called *compressed sensing*, Candès, Rohmberg, Tao and Donoho discovered that the zero-norm is well approximated by the ℓ_1-norm. We therefore consider the following problem

$$\min\{\|x\|_1 \mid Ax = b\},$$

which can be readily reformulated to the LP

$$\min\{\sum_j s_j \mid -s \le x \le s \wedge Ax = b\}. \tag{31}$$

We propose to compare the solution of Eq. (31) with that of its randomly projected version:

$$\min\{\sum_j s_j \mid -s \le x \le s \wedge TAx = Tb\}, \tag{32}$$

where T is an Achlioptas random projector. The computational set-up for this test is similar to that of Sect. 7, except that we enable the crossover in the CPLEX barrier solver.

We compare Eq. (31) and Eq. (32) on the sentence that the Sybilla of Delphos spoke to the hapless soldier who asked her whether he would get back from the war or die in it: *Ibis redibis non morieris in bello* [Alberico delle Tre Fontane, *Chronicon*], at which the soldier rejoiced. When the wife heard her husband had actually died in the war, she contacted the Sybilla for a full refund. The Sybilla, unperturbed, pointed out that the small print in the legal terms attributed her the right of inserting commas in sentences as she saw fit, which made her prophecy into the more reality-oriented *Ibis redibis non, morieris in bello*. We test here the comma-free version: much more cryptic, ambiguous, and therefore worthy of the Sybilla.

The original sentence is encoded in ASCII-128 and then in binary without padding (1001001 1100010 1101001 1110011 1000001 1100101 1001011 1001001 1010011 1000101 1010011 1100111 0000011 0111011 0111111 0111010 0000110 1101110 1111111 0010110 1001110 0101111 0010110 1001111 0011100 0001101 0011101 1101000 0011000 1011001 0111011 0011011 0011011 11). The binary string has $m = 233$ characters, is encoded into $n = 256$ characters (assuming an error rate of 10%, typical of the Sybilla muttering incantations with a low and guttural voice), and is then projected into $k = 61$ characters. We modified the parameter of the Achlioptas projector from 1/6 down to 1/100 after verifying with many examples that this particular application is extremely robust to random projections.

While the original LP took 0.296s to solve, the projected LP only took 0.028s. The accuracy in retrieving the original text was perfect. In fact, in this application it is very hard to make mistakes in the recovery; so much so, that we could set the JLL ε at 0.3. This might be partly due to the fact that the LP in Eq. (31) does not include nonnegativity constraints, which are generally problematic because of their large Gaussian width, see Sect. 1.1.

We also tested a slightly longer word sequence from a well-known poem about aviary permanence on greek sculptures: *Once upon a midnight, dreary, while I pondered, weak and weary.* The 421 characters long binary string is encoded into 463 characters and projected into 67. The original LP took 1.332s and the projected LP took 0.064s to solve; again, the retrieval accuracy was perfect.

9. Conclusion

This paper is about the application of random projections to LP in standard form. We prove that feasibility and optimality are both approximately preserved by sub-gaussian random projections. Moreover, we show how to retrieve solutions of the original LPs from those of the projected LPs, using duality arguments. These findings make it possible to approximately solve very large scale LPs with high probability, as showcased by our computational results and application to error correcting codes.

Acknowledgments We are grateful to A.A. Ahmadi and D. Malioutov for introducing us to the Johnson-Lindenstrauss lemma, and to F. Tardella for helpful discussions. We are also grateful to two referees and an associate editor who, through accurate and constructive criticism, helped us improve this paper considerably. This research is partly supported by the SO-grid project (www.so-grid.com) funded by ADEME, by the EU Grant FP7-PEOPLE-2012-ITN No. 316647 "Mixed-Integer Nonlinear Optimization", and by the ANR "Bip:Bip" project under contract ANR-10-BINF-0003. The first author was supported by a Microsoft Research Ph.D. fellowship.

References

[1] Achlioptas D (2003) Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences* 66:671–687.

[2] Allen-Zhu Z, Gelashvili R, Micali S, Shavit N (2014) Sparse sign-consistent Johnson-Lindenstrauss matrices: Compression with neuroscience-based constraints. *Proceedings of the National Academy of Sciences* 111(47):16872–16876.

[3] Anthony M, Biggs N (1992) *Computational Learning Theory: an Introduction*. Cambridge Tracts in Theoretical Computer Science (Cambridge: Cambrige University Press).

[4] Bezanson J, Edelman A, Karpinski S, Shah V (2017) Julia: A fresh approach to numerical computing. *SIAM Review* 59(1):65–98, URL http://dx.doi.org/10.1137/141000671.

[5] Borgwardt K (1987) *The Simplex Method: a Probabilistic Analysis* (Berlin: Springer).

[6] Boutsidis C, Zouzias A, Drineas P (2010) Random projections for k-means clustering. *Advances in Neural Information Processing Systems*, 298–306 (La Jolla: NIPS Foundation).

[7] Candès E, Tao T (2005) Decoding by Linear Programming. *IEEE Transactions on Information Theory* 51(12):4203–4215.

[8] Candès E, Tao T (2008) Reflections on compressed sensing. *IEEE Information Theory Society Newsletter* 58(4):14–17.

[9] Dantzig G (1990) The Diet Problem. *Interfaces* 20(4):43–47.

[10] Dasgupta A, Kumar R, Sarlós T (2010) A sparse Johnson-Lindenstrauss transform. *Proceedings of the Symposium on the Theory Of Computing*, volume 10 of *STOC* (Cambridge: ACM).

[11] Dasgupta S, Gupta A (2002) An elementary proof of a theorem by johnson and lindenstrauss. *Random Structures and Algorithms* 22:60–65.

[12] de Farias DP, Roy BV (2004) On constraint sampling in the Linear Programming approach to approximate Dynamic Programming. *Mathematics of Operations Research* 29(3):462–478.

[13] IBM (2014) *ILOG CPLEX 12.6 User's Manual*. IBM.

[14] Indyk P, Naor A (2007) Nearest neighbor preserving embeddings. *ACM Transactions on Algorithms* 3(3):Art. 31.

[15] Johnson W, Lindenstrauss J (1984) Extensions of Lipschitz mappings into a Hilbert space. Hedlund G, ed., *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, 189–206 (Providence: American Mathematical Society).

[16] Kane D, Nelson J (2014) Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM* 61(1):4.

[17] Koenker R (2005) *Quantile regression* (Cambridge: Cambridge University Press).

[18] Lubin M, Dunning I (2015) Computing in operations research using julia. *INFORMS Journal on Computing* 27(2):238–248, URL http://dx.doi.org/10.1287/ijoc.2014.0623.

[19] Matoušek J (2008) On variants of the Johnson-Lindenstrauss lemma. *Random Structures and Algorithms* 33:142–156.

[20] Matoušek J (2013) Lecture notes on metric embeddings. Technical report, ETH Zürich.

[21] Matoušek J, Gärtner B (2007) *Understanding and using Linear Programming* (Berlin: Springer).

[22] Natarajan B (1995) Sparse approximate solutions to linear systems. *SIAM Journal of Computing* 24(2):227–234.

[23] NetLib (2015) LP instance library. http://www.netlib.org/lp/.

[24] Pan V (1985) On the complexity of a pivot step of the revised simplex algorithm. *Computers & Mathematics with Applications* 11(11):1127–1140.

[25] Pilanci M, Wainwright M (2014) Randomized sketches of convex programs with sharp guarantees. *International Symposium on Information Theory (ISIT)*, 921–925 (Piscataway: IEEE).

[26] Potra F, Wright S (2000) Interior-point methods. *Journal of Computational and Applied Mathematics* 124:281–302.

[27] Venkatasubramanian S, Wang Q (2011) The Johnson-Lindenstrauss transform: An empirical study. *Algorithm Engineering and Experiments*, volume 13 of *ALENEX*, 164–173 (Providence: SIAM).

[28] Yang J, Meng X, Mahoney M (2014) Quantile regression for large-scale applications. *SIAM Journal of Scientific Computing* 36(5):S78–S110.