INCORPORATING CONDITION MEASURES INTO THE COMPLEXITY THEORY OF LINEAR PROGRAMMING

James Renegar

1. Introduction

1.1 This work is an attempt, among other things, to begin developing a complexity theory in which problem instance data is allowed to consist of real, even irrational, numbers and yet computations are of finite precision.

Complexity theory generally assumes that the exact data specifying a problem instance is used by algorithms. The efficiency of an algorithm is judged relative to the "size" of the input. For the Turing model of computation, size refers to the bit-length of the input, which is required to consist of integers (or rational numbers separated into numerators and denominators).

We replace customary measures of size with "condition measures". These measures reflect the amount of data accuracy necessary to achieve the desired computational goal. The measures are similar in spirit, and closely related, to condition numbers.

1.2 To introduce concepts gradually we begin by discussing the most basic decision problem in linear programming, that of determining if a system of constraints is consistent. Our main technical results do not concern this problem.

Let $Ax \leq b,\ x \geq 0$ be the system of interest where A is an $m \times n$ matrix whose coefficients are real numbers. The system is represented by the "data vector" $d := (A, b) \in \mathbb{R}^{mn+m}$; think of the coefficients of A and b as being strung into a long vector. We refer to $\mathbb{R}^{mn+m}$ as "data space"; each vector in data space represents a problem instance.

For an instance $d' = (A', b')$ let

$$\text{Soln}(d') := \{x; A'x \leq b',\ x \geq 0\}.$$

In attempting to determine if the instance d is a consistent system of constraints we assume algorithms will be provided with rational approximate data $\bar{d} = (\bar{A}, \bar{b})$ and an upper bound $\bar{\delta}$ on its error, i.e., $||d - \bar{d}||_\infty < \bar{\delta}$. One can think, for instance, as the approximate data consisting of truncated decimal expansions of the actual real number data.

*Research supported by IBM and by NSF Grant #CCR-9103285.

In working only with the approximate data $\bar{d}$ and the upper bound $\bar{\delta}$, an algorithm will not be able to distinguish the actual instance d from any other instance within distance $\bar{\delta}$ of $\bar{d}$. Thus, to make a decision about the actual instance d, the decision must be correct for all instances within distance $\bar{\delta}$ of $\bar{d}$. With this as motivation, we define the "condition measure of instance d with respect to the decision problem" as follows:

If $\text{Soln}(d) \neq \phi$ define

$$C(d) := ||d||_\infty / \sup\{\delta; ||d' - d||_\infty < \delta \Rightarrow \text{Soln}(d') \neq \emptyset\} \tag{1.1}$$

Replace "$\neq$" with "$=$" if $\text{Soln}(d) = \emptyset$.

Observe that $1/C(d)$ is the minimal relative perturbation size required to obtain a system from d whose answer for the decision problem is different than the answer for d. Roughly speaking, $\log C(d)$ relative bits of data accuracy are necessary to reach a decision.

Note that $0 \leq C(d) \leq \infty$.

Instance d is "ill-posed for the decision problem" if $C(d) = \infty$; it is "ill-conditioned" if $C(d)$ is large.

Note that $C(d)$ is invariant under positive scaling $d \mapsto td$ just as the decision problem is invariant; the reader might find it useful to assume $||d||_\infty = 1$ in what follows, or what is essentially the same, assume $||\bar{d}||_\infty = 1$.

Now we discuss what we want of an algorithm. We consider algorithms with input and output as follows:

Input: $\bar{d} = (\bar{A}, \bar{b}), \bar{\delta}$

Output: One of the following statements:

A) "Consistent"

B) "Inconsistent"

C) "Decision Deferred"

There are three properties we want such an algorithm to possess.

I.) Correctness. The algorithm should never make an incorrect decision for the actual problem instance d. As the algorithm must be applicable to any instance (i.e. d may vary from one application to the next), if the algorithm replies "Consistent" then correctness requires that all systems within distance $\bar{\delta}$ of the input $\bar{d}$ be consistent. Similarly, if the algorithm replies "Inconsistent".

II.) Computational Efficiency. There are various ways to define this. In this paper, we take the approach of traditional complexity theory: Requiring the input $\bar{d}, \bar{\delta}$ to consist only of rational numbers, we say the algorithm is *computationally efficient* if it terminates within polynomial-time as measured in terms of the bit-length of the input.

III.) Data Efficiency. We want the algorithm to make a decision using nearly minimal data precision. We say that the algorithm is *data efficient* if there exist a positive constant E and a polynomial $p(m, n)$, both independent of the actual instance d and input $(\bar{d}, \bar{\delta})$, such that the algorithm makes a decision if

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{p(m,n)C(d)^E}, \tag{1.2}$$

i.e., the algorithm makes a decision when provided with E times the number of relative bits of accuracy necessary to make a decision (plus a number of bits growing only like the logarithm of the dimensions of the instance).

We say that an algorithm for the decision problem is *fully efficient* if it is correct, computationally efficient and data efficient.

Important Note. It is conceivable that when a more formal framework is developed to encompass a broader class of problems it may be necessary to replace (1.2) with something like

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{p_1(m,n)C(d)^{p_2(m,n)}}, \tag{1.3}$$

i.e., a decision is made when provided with $p_2(m,n)$ times the number of bits of relative accuracy necessary to make a decision.

Is there an algorithm for the decision problem which is fully efficient? The answer is "yes" as is shown in Section 3. In fact, for constraints of the form $Ax \leq b, x \geq 0$ as we are considering, the construction and analysis of such an algorithm are deceptively trivial, assuming the algorithm can call on a polynomial-time algorithm for LP as a subroutine. In terms of (1.2) we have $E = 1, p(m,n) \equiv 2$.

(In all of our algorithm constructions we rely on a polynomial-time LP algorithm as a subroutine; any such algorithm is adequate. We treat the polynomial-time LP algorithm as a "black box".)

If one removes the non-negativity constraints, considering systems $Ax \leq b$, it is not so easy to argue the existence of a fully efficient algorithm for the decision problem. However, Vera [7] has constructed and analyzed one, obtaining $E = 3$ in (1.2).

A foremost goal in this type of complexity theory is to keep E in (1.2) as small as possible, subject to the condition of polynomially-bounded running time in terms of the bit-length of $\bar{d}, \bar{\delta}$.

It is important to understand that once one has a good algorithm for one form of constraints it does not immediately yield a good algorithm for other forms. This is in contrast to traditional complexity theory. For example, in traditional complexity theory one can replace the single-variable, single-equation system $3x = 6$ with the equivalent two-constant system $3x \leq 6, 3x \geq 6$; such transformations roughly preserve bit-length. However, when developing a complexity theory based on condition measures, such transformations are inadequate. The first system is "well-posed" with respect to the decision problem; small perturbations preserve consistency. The second system is ill-posed; arbitrarily small perturbations can destroy consistency.

Judging the efficiency of algorithms relative to condition measures introduces demands on algorithms not required in traditional complexity theory, but the converse is also true. Judged relative to condition measures, algorithms are required to perform few operations in deciding that an instance consistent (inconsistent) if the instance is far from being inconsistent (consistent). However, algorithms are not even required to make a decision for ill-posed instances, regardless of how accurate the data is. The

reason why is the requirement that input $\bar{d}, \bar{\delta}$ satisfy the strict inequality $||\bar{d} - d||_\infty < \bar{\delta}$. If we replaced "$<$" with "$\leq$", the results of this paper would be unaffected, but the character of the general theory would not be. The relation "$\leq$" would force our theory to be strictly more stringent than traditional complexity theory because it would require that any rational data instance be solved in polynomial-time; just input $\bar{d} = d, \bar{\delta} = 0$. A strict inequality leaves rational data vectors d undistinguished from irrational ones. A strict inequality leaves open the possibility of efficient algorithms, when judged in terms of condition measures, for problems which are NP-hard in the sense of traditional complexity theory; this possibility is not addressed by this paper.

1.3 We move to the next level of difficulty, constructing an approximate solution to a system of constraints when one exists. Again we let $d = (A, b)$ denote the data vector of the actual instance, $\bar{d} = (\bar{A}, \bar{b})$ denote rational approximate data, and $\bar{\delta}$ denote a rational upper bound on the data error, $||\bar{d} - d||_\infty < \bar{\delta}$.

We consider algorithms with input and output as follows:

Input: $\bar{d} = (\bar{A}, \bar{b}),\ \bar{\delta}$
Output: One of the following statements:
A) "Consistent.
Approximate solution: $\bar{x}$.
Error bound: $\bar{\epsilon}$."
B) "Inconsistent."
C) "Decision Deferred."

The approximate solution $\bar{x}$ and error bound $\bar{\epsilon}$ are computed by the algorithm. If the algorithm replies statement A then it is asserting that there exists $x \in \text{Soln}(d)$ satisfying $||x - \bar{x}||_\infty \leq \bar{\epsilon}$.

We allow $\bar{\epsilon} = \infty$; we assume the algorithm (Turing machine) has a distinguished symbol for ∞.

What do we want of such algorithm?

I.) Correctness. Besides the aspects of correctness previously discussed, correctness requires that if the algorithm replies statement A, then

$$||d' - \bar{d}||_\infty < \bar{\delta} \Rightarrow \exists\ x' \in \text{Soln}(d') \text{ s.t. } ||x' - \bar{x}||_\infty \leq \bar{\epsilon},$$

that is, $\bar{x}$ is an approximate solution for all instances d' within error $\bar{\delta}$ of $\bar{d}$.

II.) Computational Efficiency. The algorithm terminates within polynomial-time as measured in terms of the bit-length of the input $\bar{d}, \bar{\delta}$.

III.) Data Efficiency. The next few paragraphs are devoted to a discussion of this.

Data efficiency is more involved here, primarily because we are not simply dealing with a yes-no answer, but also because the tasks are becoming layered; first there is the task of deciding consistency; second there is the task of computing $\bar{x}$ and $\bar{\epsilon}$ if the system is determined to be consistent.

We associate a condition measure with each task layer. For the first layer, that of deciding consistency, the condition measure is the same as before, $C(d)$. The first requirement of data efficiency is that the algorithm reply statement A or B whenever $\bar{\delta}$ satisfies (1.2), where E and $p(m,n)$ are again instance- and input-independent.

The second layer of tasks is that of computing $\bar{x}$ and $\bar{\epsilon}$ (possibly ∞) if consistency has been determined. The condition measure associated with this should reflect the finest solution accuracy one could hope for with the given data accuracy. There are various non-equivalent ways to formalize this, at least one of which is natural for algorithm analysis.

For instance d and all $\epsilon \geq 0$, define

$$C(d,\epsilon) := ||d||_\infty / \sup\{\delta; \exists \tilde{x} \text{ s.t. } ||d'-d||_\infty < \delta \Rightarrow \exists x' \in \text{Soln}(d') \text{ s.t. } ||x'-\tilde{x}||_\infty \leq \epsilon\}. \tag{1.4}$$

Thus, $1/C(d,\epsilon)$ represents the largest relative inaccuracy in the data with which one could hope to compute a point $\tilde{x}$ guaranteed to be within error ϵ of a solution for d, i.e., $\log C(d,\epsilon)$ relative bits of accuracy are necessary.

Note that $C(d) \leq C(d,\epsilon)$.

Besides requiring an algorithm to decide consistency efficiently, we also require for all $\epsilon \geq 0$ that

$$\frac{\bar{\delta}}{||d||_\infty} < \frac{1}{p(m,n)C(d,\epsilon)^E} \Rightarrow [\text{Algorithm replies } A \text{ where } \bar{\epsilon} \leq \epsilon],$$

i.e., the algorithm requires at most E times the number of relative bits of accuracy required to compute an ϵ-approximate solution.

In summary:

III. Data Efficiency. There exist positive constants E_i and polynomials $p_i(m,n)$, $i = 1,2$, all instance- and input-independent, such that:

* The algorithm replies A or B if

$$\frac{\bar{\delta}}{||d||_\infty} < \frac{1}{p_1(m,n)C(d)^{E_1}}. \tag{1.5}$$

* For all $\epsilon \geq 0$,

$$\frac{\bar{\delta}}{||d||_\infty} < \frac{1}{p_2(m,n)C(d,\epsilon)^{E_2}} \Rightarrow [\text{Algorithm replies } A \text{ where } \bar{\epsilon} \leq \epsilon], \tag{1.6}$$

Again we say that an algorithm is *fully efficient* if it is correct, computationally efficient and data efficient.

There do exist fully efficient algorithms in this context. It is again a deceptively trivial matter to construct and analyze such an algorithm for constraints of the form $Ax \leq b, x \geq 0$. We do so in Section 3, obtaining $E_1 = E_2 = 1$, $p_1(m,n) \equiv p_2(m,n) \equiv 2$.

It is not so easy to argue the existence of fully efficient algorithms for other forms of constraint. Vera [7] has done so.

Some readers must wonder how our so-called condition measures relate to condition numbers. Roughly, one can think of the limit

$$\limsup_{\epsilon \downarrow 0} \ \epsilon C(d, \epsilon)$$

as a condition number for instance d. In the context of linear equations, one can easily verify that an analogous limit gives the usual condition number. However, the above limit is not necessarily close to condition numbers for linear inequalities as defined, say, by Mangasarian [2]; the main difference stems from the fact that $C(d, \epsilon)$ is highly dependent on both A and b in $d = (A, b)$ whereas condition numbers in the literature as represented by [3] are assigned to A by considering the worst-case b; in the context of square systems of linear equations the two approaches are roughly equivalent, but in the context of linear inequalities they are not. Both approaches have their merits.

Condition numbers are defined asymptotically; by contrast, our "condition measures" are global.

1.4 Now we consider linear programming proper, our main focus. We restrict attention to problems with constraints of the form $Ax \leq b, x \geq 0$. In this context such constraints ease the analysis but do not make it trivial.

So consider LP's of the form

$$\begin{aligned} \max \quad & c^T x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

The data vector is $d = (A, b, c)$. Approximate data, assumed to be rational, is denoted by $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$, and $\bar{\delta}$ again denotes an upper bound on the error, $||\bar{d} - d||_\infty < \bar{\delta}$.

For an LP instance $d' = (A', b', c')$, let $\mathrm{Opt}(d')$ denote the optimal solution set and let $\mathrm{Feas}(d')$ denote the feasible region, i.e., $\mathrm{Feas}(d') := \{x; A'x \leq b', x \geq 0\}$. Let $\mathrm{DualFeas}(d')$ denote the feasible region of the dual LP.

We consider algorithms with input and output as follows:

Input: $\bar{d} = (\bar{A}, \bar{b}, \bar{c}),\ \bar{\delta}$
Output: One of the following statements:
A) "There is an optimal solution.
Approximation: $\bar{x}$
Error bound: $\bar{\epsilon}$."
B) "Unbounded optimal solution."
C) "Infeasible."
D) "Feasible, but decision on the existence of an optimal solution is deferred."
E) "All decisions deferred."

As before we allow $\bar{\epsilon} = \infty$. If the algorithm replies statement A then it is asserting that there exists $x \in \text{Opt}(d)$ such that $||x - \bar{x}||_\infty \leq \epsilon$.

We now have three layers of tasks: (1) decide primal feasibility; (2) if primal feasible then decide dual feasibility; (3) if both primal and dual feasible, then compute $\bar{x}$ and $\bar{\epsilon}$. With each task layer we have a condition measure:

(1) The same as the value we have been denoting $C(d)$, i.e., if $\text{Feas}(d) \neq \emptyset$ then define

$$C_P(d) := ||d||_\infty / \sup\{\delta; ||d' - d||_\infty < \delta \Rightarrow \text{Feas}(d') \neq \emptyset\}. \tag{1.7}$$

Replace "$\neq$" with "$=$" if $\text{Feas}(d) = \emptyset$.

(2)

$$C_{PD}(d) := \max\{C_P(d), C_D(d)\} \tag{1.8}$$

where $C_D(d)$ is defined as $C_P(d)$ is, but with $\text{DualFeas}(d)$ instead of $\text{Feas}(d)$.

(3) For all $\epsilon \geq 0$,

$$C(d, \epsilon) := ||d||_\infty / \sup\{\delta; \exists \tilde{x} \text{ s.t. } ||d' - d||_\infty < \delta \Rightarrow \exists x' \in \text{Opt}(d') \text{ s.t. } ||x' - \tilde{x}||_\infty \leq \epsilon\}. \tag{1.9}$$

Note that $C_P(d) \leq C_{PD}(d) \leq C(d, \epsilon)$; the condition measures are monotonic in the task number.

A *fully efficient* algorithm is one possessing the following three properties:

I. Correctness. The statement replied must be valid for all instances within distance $\bar{\delta}$ of $\bar{d}$.

II. Computational Efficiency. Polynomial-time in the input bit-length.

III. Data Efficiency. There exist positive constants E_i and polynomials $p_i(m, n)$, $i = 1, 2, 3$, all instance- and input-independent, such that:

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{p_1(m,n) C_P(d)^{E_1}} \Rightarrow [\text{Algorithm replies A, B, C or D}] \tag{1.10}$$

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{p_2(m,n) C_{PD}(d)^{E_2}} \Rightarrow [\text{Algorithm replies A, B or C}] \tag{1.11}$$

For all $\epsilon \geq 0$,

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{p_3(m,n) C(d,\epsilon)^{E_3}} \Rightarrow [\text{Algorithm replies A where } \bar{\epsilon} \leq \epsilon.] \tag{1.12}$$

In Section 4 we construct and analyze a fully efficient algorithm. Regarding (1.12), we obtain

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{Kn C_{PD}(d)^3 C(d,\epsilon)^3} \Rightarrow [\text{Algorithm replies A where } \bar{\epsilon} \leq \epsilon.] \tag{1.13}$$

K denoting a constant. Strictly speaking we thus obtain $E_3 = 6$, but in a sense, $E_3 \approx 3$, at least when $C(d, \epsilon)$ is large relative to $C_{PD}(d)$, as it will be as $\epsilon \downarrow 0$ if d has a unique optimal solution and $C_{PD}(d) < \infty$.

Remarks. Requirement (1.12) is stringent, perhaps too much so even for linear programming; it creates technical headaches when Opt(d) has positive circumradius (i.e., Opt(d) is a positive-dimensional facet) and ϵ is only slightly larger than the circumradius of Opt(d).

Our algorithm measures the exact ℓ_∞-radius of certain polytopes specified by rational constraints. This can be done in polynomial time; by contrast, it is NP-hard to measure the ℓ_2-radius of polytopes (Bodlaender, Gritzmann, Klee and Van Leeuwen [2]).

It is the author's opinion that the particular norm should not be of extreme importance in a complexity theory based on condition measures. Perhaps the most natural way to remove the dependence is to replace the right-side of (1.12) with

$$[\text{Algorithm replies A where } \bar{\epsilon} \leq p_4(m, n)\epsilon], \tag{1.14}$$

where $p_4(m, n)$ is yet another polynomial. This alleviates the technical headaches mentioned above. For our algorithm we obtain

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{1}{KnC_{PD}(d)^3 C(d, \epsilon)^2} \Rightarrow [\text{Algorithm replies A where } \bar{\epsilon} \leq 2\epsilon.] \tag{1.15}$$

Comparing (1.13) and (1.15), we save a factor of $C(d, \epsilon)$ in the denominator on the left only at the expense of a factor of 2 on the right. However, to see what we lose with (1.15) consider an LP instance d such that Opt(d) is of radius $\tilde{\epsilon} > 0$ and $C_{PD}(d) < \infty$; then $C(d, \epsilon) = \infty \Leftrightarrow \epsilon \leq \tilde{\epsilon}$. Note (1.15) does not require the algorithm be able to compute ϵ-approximate optimal solutions when $\tilde{\epsilon} < \epsilon < 2\tilde{\epsilon}$, whereas (1.13) does.

A final remark: in a more formal theory pertaining to a broader class of problem one might want to replace the exponents E_i with polynomials.

In Section 5 we consider the problem of computing a feasible point whose objective value is nearly optimal, again assuming constraints are of the form $Ax \leq b$, $x \geq 0$. As the reader might expect, continuity of the optimal objective value under data perturbations makes this problem much easier than that of approximating optimal solutions.

Vera [7] has "extended" all of our results to other forms of LP's. Although he relies on some of our ideas, the other forms of LP's present many complications requiring additional ideas; his work is a significant step beyond ours. Readers might be interested to know that he finds analytical centers to be particularly useful.

Questions concerning the stability of linear programming solutions have been studied for many years, although not in the context of complexity theory; c.f., Ashmanov[1] and Robinson[5]. For example, it is well known that $C_{PD}(d)$ is finite if and only if the optimal solution sets of both the instance d and its dual are bounded. Also of related interest are the regularization techniques of Tikhonov and his followers; c.f., Tikhonov, Ryutin and Agayan[6].

In closing the introduction I wish to thank a referee for her/his careful reading of the manuscript and many thoughtful suggestions.

2. Relations Between Measures of Condition and Solution Size, Etc.

In this section we establish a few simple, but crucial, relations between condition measures and sizes of solutions, etc. These relations are similar in spirit, and similar in role, to the following much used relation in traditional complexity theory: if an L-bit linear programming problem has a feasible point (optimal solution) then it has one satisfying $||x||_\infty \leq 2^L$.

The relations established in this section generalize substantially as is shown in Renegar [4]. However, we now only consider problem instances $d = (A, b, c)$ *of the form*

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

Fixing m and n (the number of constraints and variables), let Pri$\emptyset$ denote the set of primal infeasible LP's (represented as data vectors), and let Dual$\emptyset$ denote the set of dual infeasible LP's.

For $d = (A, b, c)$ let $\text{dis}(d, \text{Pri}\emptyset)$ denote the ℓ_∞-distance from d to the set Pri$\emptyset$; define $\text{dis}(d, \text{Dual}\emptyset)$ analogously.

Let d^* denote the LP which is dual to d, i.e., d^* is as follows:

$$\begin{aligned} \min \quad & b^T y \\ s.t. \quad & A^T y \geq c \\ & y \geq 0. \end{aligned}$$

Let $k(d)$ denote the optimal objective value of d; if d is unbounded define $k(d) = \infty$; if d is primal infeasible define $k(d) = -\infty$.

Proposition 2.1. *Assume $d = (A, b, c)$ satisfies $Opt(d) \neq \emptyset$ and $dis(d, Dual\emptyset) > 0$. Then*

$$x \in Opt(d) \Rightarrow ||x||_1 \leq \frac{\max\{||b||_\infty, -k(d)\}}{dis(d, Dual\emptyset)}.$$

Proof. Fix an optimal solution $x \neq 0$. Let $\rho > 0$ and consider the perturbed LP

$$d + \Delta d := (A + \Delta A, b, c + \Delta c)$$

where

$$\begin{aligned} \Delta A &:= -\left(\frac{1}{||x||_1}\right) \; b \; e^T \\ \Delta c &:= \left(\frac{\max\{0, -k(d) + \rho\}}{||x||_1}\right) \; e, \end{aligned}$$

"e" denoting the vector of all ones. Note that

$$(A + \Delta A)x \leq 0$$
$$(c + \Delta c)^T x > 0.$$

Farkas' lemma implies $d + \Delta d \in \text{Dual}\emptyset$. Since

$$||\Delta d||_\infty \leq \frac{\max\{||b||_\infty, -k(d) + \rho\}}{||x||_1}$$

and since $\rho > 0$ is arbitrary, the proposition follows. □

Proposition 2.2. *Assume d^* has an optimal solution. Then every optimal solution y for d^* satisfies*

$$||y||_1 \leq \frac{\max\{||c||_\infty, k(d)\}}{dis(d, Pri\emptyset)}.$$

Proof. Analogous to the proof of Proposition 2.1. □

Proposition 2.3. *Assume $k(d)$ is finite. Then*

$$-\frac{||b||_\infty ||c||_\infty}{dis(d, Pri\emptyset)} \leq k(d) \leq \frac{||b||_\infty ||c||_\infty}{dis(d, Dual\emptyset)}$$

Proof. In proving the rightmost inequality we may assume $k(d) > 0$. Letting x denote an optimal solution for d we then have from Proposition 2.1

$$k(d) = c^T x \leq ||c||_\infty ||x||_1 \leq \frac{||b||_\infty ||c||_\infty}{\text{dis}(d, \text{Dual}\emptyset)}.$$

The leftmost inequality is established analogously, relying on Proposition 2.2. □

Lemma 2.4. *Assume $k(d)$ is finite and $\Delta d := (0, \Delta b, 0)$. Then*

$$k(d + \Delta d) - k(d) \leq ||\Delta b||_\infty \frac{\max\{||c||_\infty, k(d)\}}{dis(d, Pri\emptyset)}$$

Proof. Let y denote an optimal solution for d^*. Since y is also feasible for the dual of $d + \Delta d$, we have

$$k(d + \Delta d) \leq (b + \Delta b)^T y = k(d) + (\Delta b)^T y \leq k(d) + ||\Delta b||_\infty ||y||_1.$$

Substituting the bound of Proposition 2.2 for $||y||_1$ completes the proof. □

Proposition 2.5. *Assume $\Delta d := (\Delta A, \Delta b, \Delta c)$ and assume both $k(d)$ and $k(d+\Delta d)$ are finite. Then*

$$k(d+\Delta d) - k(d) \leq ||\Delta A||_\infty \left[\frac{\max\{||c||_\infty, k(d)\}}{dis(d, Pri\emptyset)}\right] \left[\frac{\max\{||b+\Delta b||_\infty, -k(d+\Delta d)\}}{dis(d+\Delta d, Dual\emptyset)}\right]$$
$$+ ||\Delta b||_\infty \left[\frac{\max\{||c||_\infty, k(d)\}}{dis(d, Pri\emptyset)}\right]$$
$$+ ||\Delta c||_\infty \left[\frac{\max\{||b+\Delta b||_\infty, -k(d+\Delta d)\}}{dis(d+\Delta d, Dual\emptyset)}\right]$$

Trivially, an analogous lower bound on $k(d+\Delta d) - k(d)$ is obtained by interchanging the roles of d and $d+\Delta d$.

Remark. The value $-k(d+\Delta d)$ occurring on the right side of the inequality can be replaced with $-k(d)$; this follows immediately from the inequality by considering the two cases $k(d+\Delta d) \leq k(d)$ and $k(d) \leq k(d+\Delta d)$. Similarly, the value $k(d+\Delta d)$ appearing in the analogous lower bound can be replaced with $k(d)$.

Proof. Let x denote an optimal solution for $d+\Delta d$. Let $\Delta' d := (0, \Delta' b, 0)$ where

$$\Delta' b := \Delta b - (\Delta A)x.$$

Note that x is feasible for $d+\Delta' d$ and hence $c^T x \leq k(d+\Delta' d)$. Thus

$$k(d+\Delta d) - k(d+\Delta' d) \leq (\Delta c)^T x \leq ||\Delta c||_\infty ||x||_1$$

and hence

$$k(d+\Delta d) - k(d) \leq ||\Delta c||_\infty ||x||_1 + [k(d+\Delta' d) - k(d)].$$

Noting that $||\Delta' d||_\infty \leq ||\Delta b||_\infty + ||\Delta A||_\infty ||x||_1$, the proof is now easily completed using Proposition 2.1 and Lemma 2.4. $\square$

Whenever we speak of a "dimension independent constant K", we mean that the constant does not depend on the dimensions of the LP instances being considered.

Corollary 2.6. *There exist dimension independent constants $K_1 > 0$ and $K_2 > 0$ with the following property. If d and Δd satisfy*

$$||\Delta d||_\infty < K_1 \, dis(d, Pri\emptyset \cup Dual\emptyset)$$

then

$$|k(d+\Delta d) - k(d)| < K_2 ||\Delta d||_\infty \frac{||d||_\infty \max\{||d||_\infty, |k(d)|\}}{dis(d, Pri\emptyset) \, dis(d, Dual\emptyset)}.$$

Proof. Follows immediately from Proposition 2.5, relying on the previous assertion that the value $-k(d+\Delta d)$ occurring on the right side of the inequality in Proposition 2.5 can be replaced with $-k(d)$ (similarly, the value $k(d+\Delta d)$ appearing in the analogous lower bound can be replaced with $k(d)$), and relying on the relations $\text{dis}(d, \text{Pri}\phi) \leq ||d||_\infty$, $\text{dis}(d, \text{Dual}\phi) \leq ||d||_\infty$. $\square$

3. Trivialities

In this section we consider the problem of deciding if $Ax \leq b, x \geq 0$ is consistent and, if so, of computing a solution. The form of the constraints makes this section deceptively trivial; by contrast, see Vera [4] for other forms of constraints.

Let $\bar{d} = (\bar{A}, \bar{b})$ denote approximate data and let $\bar{\delta}$ denote an upper bound or its error, i.e., $||\bar{d} - d||_\infty < \bar{\delta}$ where d is the actual data.

The triviality of this section results from the fact that there are two elements in the set of instances

$$B_\infty(\bar{d}, \bar{\delta}) := \{d'; ||d' - \bar{d}||_\infty \leq \bar{\delta}\}$$

which determine the consistency or inconsistency of all instances in the set. The two instances are

$$d_1 := (\bar{A} + \bar{\delta} e e^T, \bar{b} - \bar{\delta} e)$$
$$d_2 := (\bar{A} - \bar{\delta} e e^T, \bar{b} + \bar{\delta} e).$$

Defining

$$\text{Soln}(d') := \{x; A'x \leq b', x \geq 0\}$$

where $d' = (A', b')$, it is easily proven that

$$d' \in B_\infty(\bar{d}, \bar{\delta}) \Rightarrow \text{Soln}(d_1) \subseteq \text{Soln}(d') \subseteq \text{Soln}(d_2). \tag{3.1}$$

The fully efficient algorithms alluded to in Sections 1.2 and 1.3 follow from this implication. For example, the one alluded to in Section 1.3 is as follows:

Input: $\bar{d} = (\bar{A}, \bar{b}), \bar{\delta}$.

(1) Check consistency of d_2. If inconsistent then reply "Inconsistent" and STOP.
(2) Check consistency of d_1. If inconsistent then reply "Decision Deferred" and STOP.
(3) Compute a feasible point $\bar{x}$ for d_1. Let $\bar{\epsilon} = 0$. Reply "Consistent. Approximate Solution: $\bar{x}$. Error bound: $\bar{\epsilon}$" and STOP.

Assuming that one uses a polynomial-time algorithm for checking the consistency in steps 1 and 2, and for computing $\bar{x}$ in step 3, the claims of Section 1.1 and 1.2 follow trivially.

4. Linear Programming Proper

4.1 In this section we construct and analyze the fully efficient algorithm mentioned in Section 1.4, for LP's of the form

$$\begin{aligned} \max \quad & c^T x \\ s.t. \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

The construction of the algorithm, and the proofs of correctness and computational efficiency, are all simple. The interesting aspect is the proof that the algorithm is data efficient in approximating optimal solutions.

The "ℓ_∞-radius" of a closed set S is defined to be the smallest value r for which there exists $\bar{x}$ satisfying $S \subseteq \{x; ||x - \bar{x}||_\infty < r\}$; a corresponding $\bar{x}$ is called a "mid-point" of S; the ℓ_∞-radius may be ∞.

If S is specified as the feasible region for a system of linear inequalities with rational coefficients, then its ℓ_∞-radius and a mid-point can be computed in time polynomial in the bit-length of the coefficients, as the reader can easily verify.

Letting $\bar{d} = (\bar{A}, \bar{b}, \bar{c})$ and $\bar{\delta}$ denote input, define

$$d_1 := (\bar{A} + \bar{\delta} e e^T, \bar{b} - \bar{\delta} e, \bar{c} - \bar{\delta} e)$$
$$d_2 := (\bar{A} - \bar{\delta} e e^T, \bar{b} + \bar{\delta} e, \bar{c} + \bar{\delta} e).$$

The algorithm is as follows:
Input: $\bar{d} = (\bar{A}, \bar{b}, \bar{c}), \bar{\delta}$

(1) Check primal feasibility of d_2. If infeasible then reply "Infeasible" and STOP.
(2) Check primal feasibility of d_1. If infeasible then reply "All decisions deferred" and STOP.
(3) Check dual feasibility of d_1. If infeasible then reply "Unbounded optimal solution" and STOP.
(4) Check dual feasibility of d_2. If infeasible then reply "Feasible, but decision on the existence of an optimal solution is deferred" and STOP.
(5) Compute the ℓ_∞-radius $\bar{\epsilon}$ and a mid-point $\bar{x}$ for the feasible region of the following system:

$$\begin{aligned} A_2 x &\leq b_2 \\ c_2^T x &\geq k(d_1) \\ x &\geq 0 \end{aligned} \tag{4.1}$$

where $d_2 = (A_2, b_2, c_2)$. Reply "There is an optimal solution. Approximation: $\bar{x}$. Error bound: $\bar{\epsilon}$." STOP.

Assuming polynomial-time LP algorithms are used as subroutines, the computational efficiency of the above algorithm is immediate.

Correctness of the algorithm follows from the easily proven fact that if $d' \in B_\infty(\bar{d}, \delta)$ then

$$\text{Feas}(d_1) \subseteq \text{Feas}(d') \subseteq \text{Feas}(d_2) \tag{4.2}$$

$$\text{DualFeas}(d_2) \subseteq \text{DualFeas}(d') \subseteq \text{DualFeas}(d_1) \tag{4.3}$$

$$k(d_1) \leq k(d') \leq c_2^T x(d') \tag{4.4}$$

where $x(d')$ denotes any optimal solution of d' (assuming one exists). We leave verification of correctness as a simple exercise.

We discussed in Section 1.4 that, regarding data efficiency, there are three layers of tasks, each with an appropriate condition measure: $C_P(d), C_{PD}(d)$ and $C(d,\epsilon)$. The definition of "data efficiency" in that section addresses the task layers consecutively.

For the first task layer, that of deciding primal feasibility, the data efficiency of our algorithm is an immediate consequence of (4.2); in fact, it follows immediately that E_1 and $p_1(m,n)$ in (1.10) can be taken as the constants 1 and 2, respectively. Similarly for the second task layer, that of deciding dual feasibility, i.e., (1.11).

Finally, we come to something interesting, proving data efficiency of the algorithm in approximating an optimal solution. Fixing $\epsilon > 0$, we wish to establish (1.12). In doing so, we may assume the input $\bar{d}, \bar{\delta}$ satisfies $\bar{\delta} < 1/2C(d,\epsilon)$; it follows we may assume that upon input $\bar{d}, \bar{\delta}$ the algorithm does not terminate until step 5.

In what follows $\bar{x}$ and $\bar{\epsilon}$ refer to the approximate optimal solution and error bound computed by the algorithm upon input $\bar{d}, \bar{\delta}$. As always, d refers to the actual instance, i.e., the one $\bar{d}$ is considered to approximate.

Most of the remainder of this section is devoted to proving the following two propositions, the first of which is largely a consequence of the second. The first proposition is appropriate for the more stringent definition of "data efficiency" relying on (1.12). Either proposition is appropriate for the less stringent definition relying on (1.14), although the second proposition provides better bounds.

Recall that $C_{PD}(d) \leq C(d,\epsilon)$.

Proposition 4.1. *There is a dimension independent constant $K_3 > 0$ with the following property: For all $\epsilon \geq 0$,*

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{K_3}{nC_{PD}(d)^3 C(d,\epsilon)^3} \Rightarrow \bar{\epsilon} \leq \epsilon.$$

Proposition 4.2. *There is a dimension independent constant $K_4 > 0$ with the following property: For all $\epsilon \geq 0$ and ρ satisfying $0 < \rho \leq 1$,*

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{K_4 \rho}{nC_{PD}(d)^3 C(d,\epsilon)^2} \Rightarrow \bar{\epsilon} \leq (1+\rho)\epsilon.$$

Before proving the propositions we use them.

Theorem 4.3. *The preceding algorithm is fully efficient.*

Proof. Follows immediately from Proposition 4.1, (1.12) and the preceding discussion. □

Remark. Note that in the notation of (1.12), $E_3 = 6$ and $p(m,n) = n/K_3$; moreover, if $C(d,\epsilon)$ is large relative to $C_{PD}(d)$ then, in a sense, $E_3 = 3$. If one instead uses the weaker definition of data efficiency relying on (1.14), then Proposition 4.2 provides better values.

Before proceeding to the proofs, we introduce simplifying notation:

$$s_{PD}(d) := \text{dis}(d_1, \text{Pri}\emptyset \cup \text{Dual}\emptyset)$$

i.e., the ℓ_∞-distance from d to the set of LP instances which are either primal or dual infeasible. Also, for all $\epsilon \geq 0$,

$$s(d,\epsilon) := \sup\{\delta; \exists \tilde{x} \text{ s.t. } ||d'-d||_\infty < \delta \Rightarrow \exists x' \in \text{Opt}(d') \text{ s.t. } ||x'-\tilde{x}||_\infty \leq \epsilon\}.$$

Note that

$$s(d,\epsilon) = ||d||_\infty / C(d,\epsilon) \tag{4.5}$$

and

$$s_{PD}(d) > 0 \Rightarrow s_{PD}(d) = ||d||_\infty / C_{PD}(d) \tag{4.6}$$

4.2 In this subsection we derive Proposition 4.1 from Proposition 4.2. The next subsection is devoted to proving Proposition 4.2.

We begin with two lemmas, the first of which is only an intermediate step to the second.

Lemma 4.4. *If* $0 < \epsilon' \leq \epsilon$ *then*

$$s(d,\epsilon) \leq \left(\frac{\epsilon}{\epsilon'}\right)\ s(d,\epsilon') + \left(\frac{\epsilon}{\epsilon'} - 1\right)\ ||d||_\infty.$$

Proof. We may assume $s(d,\epsilon') < s(d,\epsilon)$

Let δ', δ satisfy $0 \leq s(d,\epsilon') < \delta' \leq \delta < s(d,\epsilon)$. We first show it suffices to prove that

$$\delta' \leq s(d, \epsilon/\rho) \tag{4.7}$$

where

$$\rho := \frac{||d||_\infty + \delta}{||d||_\infty + \delta'}. \tag{4.8}$$

To see that this suffices, observe that (4.7) and $s(d,\epsilon') < \delta'$ imply $s(d,\epsilon') < s(d,\epsilon/\rho)$; hence, $\epsilon' < \epsilon/\rho$. Substituting from (4.8) for ρ thus yields

$$\epsilon' < \frac{||d||_\infty + \delta'}{||d||_\infty + \delta}\ \epsilon$$

and hence

$$\delta < \left(\frac{\epsilon}{\epsilon'}\right)\delta' + \left(\frac{\epsilon}{\epsilon'} - 1\right)\ ||d||_\infty.$$

Taking the limit as $\delta' \downarrow s(d,\epsilon')$, $\delta \uparrow s(d,\epsilon)$ gives the lemma.

Now to prove (4.7). Consider the set of instances

$$S := \{\widehat{d}; \widehat{d} = (A', \rho b', c') \text{ where } (A', b', c') \in B_\infty(d, \delta')\}.$$

Observe that $S \subseteq B_\infty(d, \delta)$; for if $\widehat{d} = (A', \rho b', c')$ and $d' = (A', b', c')$ then

$$\begin{aligned} ||\widehat{d} - d||_\infty &\leq ||\widehat{d} - d'||_\infty + ||d' - d||_\infty \\ &= (\rho - 1)||b'||_\infty + ||d' - d||_\infty \\ &= \frac{\delta - \delta'}{||d||_\infty + \delta'}||b'||_\infty + ||d' - d||_\infty \\ &\leq \delta - \delta' + ||d' - d||_\infty \\ &< \delta. \end{aligned}$$

Since $\delta < s(d, \epsilon)$ there thus exists $\tilde{x}$ such that

$$\widehat{d} \in S \Rightarrow \exists \widehat{x} \in \mathrm{Opt}(\widehat{d}) \text{ s.t. } ||\widehat{x} - \tilde{x}||_\infty \leq \epsilon.$$

Note that if $\widehat{d} = (A', \rho b', c')$ and $d' = (A', b', c')$ then

$$\widehat{x} \in \mathrm{Opt}(\widehat{d}) \Leftrightarrow \left(\frac{1}{\rho}\right) \widehat{x} \in \mathrm{Opt}(d').$$

It follows that

$$d' \in B_\infty(d, \delta') \Rightarrow \exists x' \in \mathrm{Opt}(d') \text{ s.t. } ||x' - \left(\frac{1}{\rho}\right) \tilde{x}||_\infty \leq \frac{\epsilon}{\rho}.$$

Hence (4.7). $\square$

Lemma 4.5. *Assume $\epsilon \geq 0$ and define*

$$\epsilon' := \frac{\epsilon}{1 + \frac{s(d,\epsilon)}{2||d||_\infty}}. \tag{4.9}$$

Then $s(d, \epsilon) \leq 3s(d, \epsilon')$.

Proof. We may assume $\epsilon > 0$. Since $\epsilon' \leq \epsilon$, Lemma 4.4 is applicable. Substituting (4.9) for ϵ' in that lemma, and rearranging yields

$$s(d, \epsilon) \leq 2 \left(1 + \frac{s(d, \epsilon)}{2||d||_\infty}\right) \; s(d, \epsilon').$$

Finally, note that $s(d, \epsilon) \leq ||d||_\infty$. $\square$

Proof of Proposition 4.1 from Proposition 4.2.

We assume $\bar{\delta}$ satisfies the assumed upper bound, that is,

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{K_3}{n} \left[\frac{s(d, \epsilon)}{||d||_\infty}\right]^3 \left[\frac{s_{PD}(d)}{||d||_\infty}\right]^3. \tag{4.10}$$

Let

$$\rho := \frac{s(d,\epsilon)}{2||d||_\infty}, \quad \epsilon' := \frac{\epsilon}{1+\rho}. \tag{4.11}$$

Lemma 4.5 shows

$$s(d,\epsilon) \leq 3s(d,\epsilon'). \tag{4.12}$$

Together, (4.10), (4.11) and (4.12) imply we may assume (by requiring K_3 to be sufficiently small)

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{K_4\rho}{n}\left[\frac{s(d,\epsilon')}{||d||_\infty}\right]^2\left[\frac{s_{PD}(d)}{||d||_\infty}\right]^3$$

where K_4 is as in Proposition 4.2; thus, from that proposition,

$$\bar{\epsilon} \leq (1+\rho)\epsilon' = \epsilon.$$

□

4.3 In this subsection we prove Proposition 4.2.

We begin with a proposition which in effect asserts that for slightly worse data error than $\bar{\delta}$ one cannot hope for much better solution accuracy than $\bar{\epsilon}$ if $s_{PD}(d)$ is not small (i.e., if $C_{PD}(d)$ is not large).

Proposition 4.6. *There exist dimension-independent positive constants K_5, K_6 with the following property: If $\bar{\delta}$ and $\Delta\delta$ are positive numbers satisfying*

$$\bar{\delta} + \Delta\delta \leq K_5 s_{PD}(d)$$

then for all $\epsilon \geq 0$,

$$\epsilon < \bar{\epsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left[\frac{||d||_\infty}{s_{PD}(d)}\right]^3 \Rightarrow 2\bar{\delta} + \Delta\delta \geq s(d,\epsilon). \tag{4.13}$$

We begin the proof of Proposition 4.6 with a lemma. The instances d_1 and d_2 referred to are those of the algorithm.

Lemma 4.7. *Assume $\Delta\delta$ is a real number satisfying*

$$0 < \Delta\delta \quad \text{and} \quad \bar{\delta} + \Delta\delta < s_{PD}(\bar{d}). \tag{4.14}$$

There exist instances d' and d'' satisfying

$$||d' - \bar{d}||_\infty \leq \bar{\delta} + \Delta\delta \tag{4.15}$$

$$||d'' - \bar{d}||_\infty \leq \bar{\delta} + \Delta\delta \tag{4.16}$$

$$dis(Opt(d'),\ Opt(d'')) \geq 2\left[\bar{\epsilon} - \frac{k(d_2) - k(d_1)}{\Delta\delta}\right] \tag{4.17}$$

where dis(S,T) is the ℓ_∞-distance between subsets S and T. (Note: (4.14) and (4.15) imply Opt$(d') \neq \emptyset$; similarly, Opt$(d'') \neq \emptyset$.)

Proof. First assume that $\bar{\epsilon} < \infty$; recall that $\bar{\epsilon}$ is the ℓ_∞-radius of the feasible region for (4.1), so the projection of that feasible region onto some coordinate axis is an interval of length $2\bar{\epsilon}$; assume this is so for the first coordinate axis. Let x' denote a feasible point for (4.1) whose projection is least, and let x'' denote a feasible point whose projection is greatest. So $e_1^T(x'' - x') = 2\bar{\epsilon}$ where e_1 is the first unit vector.

Let

$$d' = (A', b', c') := (A_2, b_2, c_2 - (\Delta\delta)e_1)$$
$$d'' := (A_2, b_2, c_2 + (\Delta\delta)e_1)$$

where $d_2 = (A_2, b_2, c_2)$. Since Feas(d') = Feas(d_2) we have

$$x \in \text{Feas}(d') \Rightarrow (c')^T x \leq k(d_2) - (\Delta\delta)e_1^T x. \tag{4.18}$$

Since x' is feasible for (4.1) we have $c_2^T x' \geq k(d_1)$ and hence

$$(c')^T x' \geq k(d_1) - (\Delta\delta)e_1^T x'. \tag{4.19}$$

Noting that $x' \in$ Feas(d'), together (4.18) and (4.19) yield

$$x \in \text{Opt}(d') \Rightarrow k(d_2) - (\Delta\delta)e_1^T x \geq k(d_1) - (\Delta\delta)e_1^T x',$$

that is,

$$x \in \text{Opt}(d') \Rightarrow e_1^T x \leq e_1^T x' + \frac{k(d_2) - k(d_1)}{\Delta\delta}. \tag{4.20}$$

Similarly,

$$x \in \text{Opt}(d'') \Rightarrow e_1^T x \geq e_1^T x'' - \frac{k(d_2) - k(d_1)}{\Delta\delta}. \tag{4.21}$$

¿From (4.20), (4.21) and $e_1^T(x'' - x') = 2\bar{\epsilon}$ we obtain (4.17) .

If $\bar{\epsilon} = \infty$ then the proof proceeds exactly as above except that x' and x'' are chosen so that $e_1^T(x'' - x')$ is arbitrarily large. □

Proof of Proposition 4.6. Since $s_{PD}(d) < s_{PD}(\bar{d}) + \bar{\delta}$, we may assume by choosing K_5 sufficiently small that $\Delta\delta$ satisfies the assumptions of Lemma 4.7. Hence there exist instances d' and d'' satisfying

$$d',\ d'' \in B_\infty(d, 2\bar{\delta} + \Delta\delta) \tag{4.22}$$

$$\text{dis}(\text{Opt}(d'),\ \text{Opt}(d'')) \geq 2\left[\bar{\epsilon} - \frac{k(d_2) - k(d)}{\Delta\delta} - \frac{k(d) - k(d_1)}{\Delta\delta}\right] \tag{4.23}$$

Choosing K_5 sufficiently small we may assume d and $\Delta d := d_1 - d$ satisfy the assumption of Corollary 2.6; similarly for d and $d_2 - d$. Substituting the implied bounds on $k(d_2) - k(d)$ and $k(d) - k(d_1)$ into (4.23), then substituting the bound $|k(d)| \leq ||d||_\infty^2 / s_{PD}(d)$ implied by Proposition 2.3, one obtains (using $s_{PD}(d) \leq ||d||_\infty$)

$$\mathrm{dis}(\mathrm{Opt}(d'),\ \mathrm{Opt}(d'')) \geq 2\left(\bar{\epsilon} - K_6\left(\frac{\bar{\delta}}{\Delta\delta}\right)\left[\frac{||d||_\infty}{s_{PD}(d)}\right]^3\right) \tag{4.24}$$

where K_6 is a dimension-independent constant. Together, (4.22) and (4.24) give (4.13). $\square$

The fact that the non-negativity constraints $x \geq 0$ are unaffected by data perturbations forces special attention be given the zero vector as an optimal solution; the set of instances for which $\vec{0}$ is optimal has non-empty interior. With this in mind we define

$$s_0(d) := \sup\{\delta; ||d' - d||_\infty < \delta \Rightarrow \vec{0} \in \mathrm{Opt}(d')\}.$$

If $\vec{0} \notin \mathrm{Opt}(d)$ then $s_0(d) = 0$.

It is easily seen that

$$s_0(d) = \sup\{\delta; ||d' - d||_\infty < \delta \Rightarrow \mathrm{Opt}(d') = \{\vec{0}\}\};$$

for if $\mathrm{Opt}(d') \neq \{\vec{0}\}$ then an arbitrarily slight perturbation of the objective for d' yields an instance for which $\vec{0}$ is not optimal.

Lemma 4.8. *If $\bar{\delta} < s_0(d)/2$ then $\bar{x} = \vec{0},\ \bar{\epsilon} = 0$.*

Proof. If $\bar{\delta} < s_0(d)/2$ then $\mathrm{Opt}(d_1) = \mathrm{Opt}(d_2) = \{\vec{0}\}$ and hence $k(d_1) = k(d_2) = 0$. Lemma 4.7 then implies $\bar{\epsilon} = 0$; for the lemma implies, by choosing $\Delta\delta < s_0(d) - 2\bar{\delta}$, that if $\bar{\epsilon} > 0$ then there exists d' and d'', both of distance less than $s_0(d)$ from d, and such that either $\mathrm{Opt}(d') \neq \{\vec{0}\}$ or $\mathrm{Opt}(d'') \neq \{\vec{0}\}$, contradicting the relation for $s_0(d)$ noted just prior to the statement of Lemma 4.8.

If $\bar{\epsilon} = 0$, the correctness of the algorithm implies $\bar{x}$ is optimal for all instances in the open set $\{d'; ||d' - \bar{d}||_\infty < \bar{\delta})\}$; from this it is easily argued that $\bar{x} = \vec{0}$. $\square$

One should keep in mind the relations

$$s_0(d) \leq s(d, \epsilon) \leq s_{PD}(d).$$

Lemma 4.9. *For all $\epsilon \geq 0$, $s(d, \epsilon) \leq s_0(d) + 4\epsilon n||d||_\infty$.*

Proof. We may assume $s_0(d) < s(d, \epsilon)$ and hence $s_0(d) < s_{PD}(d)$. Then it is easily seen that for each $\rho > 0$ there exists an instance $d' = (A', b', c')$ satisfying

$$||d' - d||_\infty < \rho + s_0(d)$$

and such that $\mathrm{Opt}(d')$ consists of a single point $x' \neq \vec{0}$.

Consider the instance

$$d'' := (A', b' + \left(\frac{2\epsilon + \rho}{||x'||}_{\infty}\right) A'x', c').$$

It follows from the complementary slackness conditions for optimality that $\mathrm{Opt}(d'') = \{x''\}$ where

$$x'' := x' + \left(\frac{2\epsilon + \rho}{||x'||_\infty}\right) x'.$$

Hence,

$$\mathrm{dis}(\mathrm{Opt}(d'), \mathrm{Opt}(d'')) = ||x' - x''||_\infty = 2\epsilon + \rho.$$

Consequently,

$$\begin{aligned} s(d,\epsilon) &\leq \max\{||d' - d||_\infty,\ ||d'' - d||_\infty\} \\ &\leq ||d' - d||_\infty + ||d'' - d'||_\infty \\ &\leq ||d' - d||_\infty + (2\epsilon + \rho)n||d'||_\infty \\ &< [s_0(d) + \rho] + (2\epsilon + \rho)n[||d||_\infty + s_0(d) + \rho]. \end{aligned}$$

Noting $s_0(d) \leq ||d||_\infty$, the lemma follows since $\rho > 0$ is arbitrary. $\square$

Proof of Proposition 4.2.

We assume $\bar{\delta}$ satisfies the assumed upper bound, that is,

$$\frac{\bar{\delta}}{||d||_\infty} \leq \frac{K_4 \rho}{n} \left[\frac{s(d,\epsilon)}{||d||_\infty}\right]^2 \left[\frac{s_{PD}(d)}{||d||_\infty}\right]^3. \tag{4.25}$$

We prove the proposition assuming that K_4 is "sufficiently small"; what constitutes "sufficiently small" will become evident during the course of the proof.

We may assume $\bar{\epsilon} > 0$ and hence, by Lemma 4.8,

$$\bar{\delta} \geq \frac{s_0(d)}{2}. \tag{4.26}$$

Since

$$s(d,\epsilon) \leq s_{PD}(d) \leq ||d||_\infty \tag{4.27}$$

it follows from (4.25), (4.26) and $\rho \leq 1$ that by choosing K_4 sufficiently small we may assume

$$s_0(d) \leq \frac{s(d,\epsilon)}{2}.$$

Thus, by Lemma 4.9,

$$s(d,\epsilon) \leq 8\epsilon n||d||_\infty. \tag{4.28}$$

Also note (4.25) and $\bar{\delta} > 0$ imply

$$s(d, \epsilon) > 0. \tag{4.29}$$

Define

$$\Delta\delta := \min\{\frac{1}{2}, K_5\}\ s(d, \epsilon) - 2\bar{\delta} \tag{4.30}$$

where K_5 is as in Proposition 4.6. Note that (4.29) and (4.30) imply

$$2\bar{\delta} + \Delta\delta < s(d, \epsilon). \tag{4.31}$$

Also note that (4.25), (4.27), (4.30) and $\rho \leq 1$ imply that by choosing K_4 sufficiently small we may assume

$$\Delta\delta \geq \frac{1}{2} \min\{\frac{1}{2}, K_5\}\ s(d, \epsilon). \tag{4.32}$$

¿From (4.29) and (4.32) we have

$$\Delta\delta > 0. \tag{4.33}$$

Moreover, (4.27), (4.29) and (4.30) imply

$$\bar{\delta} + \Delta\delta \leq K_5 s_{PD}(d). \tag{4.34}$$

¿From (4.33) and (4.34) we find that Proposition 4.6 is applicable; consideration of the proposition in conjunction with (4.31) gives

$$\epsilon \geq \bar{\epsilon} - K_6 \left(\frac{\bar{\delta}}{\Delta\delta}\right) \left[\frac{||d||_\infty}{s_{PD}(d)}\right]^3. \tag{4.35}$$

Substituting (4.25) and (4.32) into (4.35) we find

$$\epsilon \geq \bar{\epsilon} - \frac{K_4 K' \rho}{n} \left[\frac{s(d, \epsilon)}{||d||_\infty}\right]$$

where

$$K' := \frac{2K_6}{\min\{\frac{1}{2}, K_5\}}.$$

Hence, by choosing K_4 sufficiently small we may assume

$$\epsilon \geq \bar{\epsilon} - \frac{\rho}{8n} \left[\frac{s(d, \epsilon)}{||d||_\infty}\right].$$

Then, by (4.28)

$$\epsilon \geq \bar{\epsilon} - \rho\epsilon,$$

completing the proof. □

5. More Simple Stuff

In this section we consider the problem of computing a feasible point whose objective value is nearly optimal assuming, as always, constraints are of the form $Ax \leq b,\ x \geq 0$.

The algorithm is identical with that of the previous section except we replace step 5 with the following.

5. Compute an optimal solution $\bar{x}$ for d_1, compute $k(d_1)$ and $k(d_2)$. Let $\bar{\epsilon} := k(d_2) - k(d_1)$. Reply "There is an optimal solution. Feasible point $\bar{x}$. Bound on the difference between the optimal value and the objective value of the feasible point: $\bar{\epsilon}$."

Assuming polynomial-time LP algorithms are used as subroutines, the computational efficiency of the algorithm is immediate: it terminates within time polynomial in the bit length of the input $\bar{d}, \bar{\delta}$.

Correctness of the algorithm is a simple exercise relying on relations (4.2), (4.3) and (4.4).

It only remains to prove the algorithm is "data efficient", a phrase which we have yet to define in this context but which the reader no doubt can infer from the previous development. The definition is the same as (1.10), (1.11) and (1.12) except for two changes. First, statement A in the algorithm referred to there should be replaced with the statement replied in step 5 above. Second, $C(d, \epsilon)$ must be redefined:

$$C(d, \epsilon) := ||d||_\infty / \sup\{\delta; \exists k \text{ s.t. } ||d' - d||_\infty < \delta \Rightarrow |k(d') - k| \leq \epsilon\}.$$

This value indicates the data accuracy necessary to approximate the optimal value to within error ϵ, but does not seem to indicate the accuracy needed to compute a feasible point whose objective value is within ϵ of the optimal value. The fact that it does so follows from the relations (4.2) and (4.4) which are very particular to constraints of the form $Ax \leq b,\ x \geq 0$. In fact

$$C(d, \epsilon) = ||d||_\infty / \sup\{\delta; \exists \tilde{x} \text{ s.t. } ||d' - d||_\infty < \delta \Rightarrow [\tilde{x} \in \text{Feas}(d') \text{ and } |k(d') - c'\tilde{x}| \leq \epsilon]\}$$

where c' refers to the objective of d'.

Relying on the relations (4.2), (4.3) and (4.4) the reader should have no difficulty verifying that the algorithm is data efficient. Once again constraints of the form $Ax \leq b,\ x \geq 0$ result in a deceptively simple proof, unlike that of the previous section.

References

[1 by S.A. Ashmanov] *Stability conditions for linear programming problems*, U.S.S.R. Comput. Maths. Math. Phys. **21** (1981), 40–49.

[2] H.L. Bodlaender, P. Gritzmann, V. Klee and J. Van Leeuwen, *The computational complexity of norm-maximization*, Combinatorica **10** (1990), 203–225.

[3] O.L. Mangasarian, *A condition number for linear inequalities and linear programs*, Proc. 6th Sympos. Oper. Res., Augsburg 7-9 September 1981, G. Bamberg and O. Opitz, eds. Verlagsgroppe Athenaum/Hain/Scriptor/Hanstein, Konigstein, 3–15.

[4] J. Renegar, *Some perturbation theory for linear programming*, to appear in Mathematical Programming.

[5] S. Robinson, *A characterization of stability in linear programming*, Operations Research **25** (1977), 435–447.

[6] A.N. Tikhonov, A.A. Ryutin and G.M. Agayan, *On a stable method of solving a linear programming problem with approximate data*, Soviet Math. Dokl **28** (1983), 494–500.

[7] J. Vera, *Ill-Posedness in Mathematical Programming and Problem Solving with Approximate Data*, Ph.D. Thesis, Cornell University, 1992 (Note: the relevant sections are available as technical reports which are being submitted for publication; contact Renegar if you are interested)..

Mathematical Sciences Department, IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853

E-mail: renegar @ orie.cornell.edu