On Solving the Linear Programming Problem Approximately

Nimrod Megiddo*

Abstract. This paper studies the complexity of some approximate solutions of linear programming problems with real coefficients.

1. Introduction

The general linear programming problem is to maximize a linear function over a set defined by linear inequalities and equations. There are many equivalent ways to represent instances of the linear programming problem. For example, consider the symmetric form:

$$
\begin{aligned}
&\text{Maximize} \quad \boldsymbol{c}^T\boldsymbol{x} \\
(Sym(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})) \qquad &\text{subject to} \quad \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b} \\
&\qquad\qquad\quad\ \ \boldsymbol{x} \ge \boldsymbol{0} \ .
\end{aligned}
$$

The dual is then

$$
\begin{aligned}
&\text{Minimize} \quad \boldsymbol{b}^T\boldsymbol{y} \\
&\text{subject to} \quad \boldsymbol{A}^T\boldsymbol{y} \ge \boldsymbol{c} \\
&\qquad\qquad\quad\ \ \boldsymbol{y} \ge \boldsymbol{0} \ .
\end{aligned}
$$

Intuitively, two representations are equivalent if there is an easy way to transform solutions of one to solutions of the other and vice versa. We first mention some of the well-known equivalences. First, any set of linear inequalities and linear equations can be reduced to a set of linear equations with nonnegativity constraints or to a set of inequality and nonnegativity constraints. Also, any linear programming problem can be reduced to a linear programming problem with a nonempty set of solutions by using artificial variables. Moreover, any linear programming problem can be reduced to a problem of finding a solution to a system of linear inequalities (by combining the constraints of

---

*IBM Research Division, Almaden Research Center, San Jose, CA 95120-6099, and School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel

1

the primal and dual and adding the inequality $\boldsymbol{c}^T\boldsymbol{x} \geq \boldsymbol{b}^T\boldsymbol{y}$) or else concluding that the system has no solution. By the duality theorem, if we put the problem in the combined primal-dual form, every problem can be reduced to a problem which is either infeasible or feasible and bounded (but not unbounded). It is interesting to note that over any ordered field, every linear programming problemcan be reduced to one which is both feasible and bounded. This is done as follows. Suppose the problem is given in the symmetric form $Sym(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Consider the following:

$$\begin{aligned}
\text{Minimize} \quad & \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y} + t \\
\text{subject to} \quad & \boldsymbol{A}\boldsymbol{x} - te \leq \boldsymbol{b} \\
& \boldsymbol{A}^T\boldsymbol{y} + te \geq \boldsymbol{c} \\
& \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{y} \geq 0 \\
& \boldsymbol{x}, \boldsymbol{y}, t \geq 0 \ ,
\end{aligned}$$

where $\boldsymbol{e}$ denotes a vector of 1's. It is easy to verify that the optimal value of the latter is zero if and only if the former has an optimal solution. In this case the latter provides optimal solutions for the former and its dual.

The equivalences mentioned above are valid as long as exact computation is feasible. In practice one usually works with finite precision and hence obtains results which are only "approximately true". However, the meaning of the last sentence depends on the particular representation of the practical problem. Indeed, a good approximate solution for one representation of the problem may transform into a very bad approximate solution for another "equivalent" representation of the same problem.

When two people talk about approximate solutions, they often think of different notions of approximation. It is quite likely though that they refer to one of the following:

(i) A feasible point (i.e., one which satisfies all the constraints in the exact sense) and is close in a certain metric to an optimal point.

(ii) A feasible solution whose objective function value is close to the optimal value.

(iii) A point, not necessarily feasible, close to an optimal solution.

(iv) A point which approximately satisfies every constraint, and whose objective function value is close to the optimal value.

(v) A point close to the feasible domain, whose objective function value is close to the optimal value (called the "weak optimization problem" in [7]).

(vi) A basis where the simplex algorithm (using exact arithmetic) terminates, but the numerical values of variables are only approximate.

(vii) A basis where the simplex algorithm terminates due to a prescribed tolerance.

The choice of the right definition depends very much on the practical situation. In fact, practical considerations dictate which constraints must be satisfied and which may

be approximately satisfied. In other words, the tolerance may be different for different constraints.

It is not known whether the $m \times n$ linear programming problem with real data can be solved in a polynomial number of arithmetic operations and comparisons in terms of m and n. (We will refer to this notion of complexity as strongly polynomial time, even though the usual definition of this concept requires also polynomial time in the usual sense.) Thus, another natural question is whether for any $\epsilon > 0$, any of the ϵ-approximation problems can be solved in a polynomial number of operations in terms of m, n and $- \log \epsilon$. To deal with this question, we first have to define what we mean by an "ϵ-approximate" solution. In particular, such a definition should not make the second question trivially equivalent to the first one. Consider, for example, the concept suggested in (i) above. Thus, for the problem: maximize $c^T x$ subject to $Ax \leq b$ (assuming its maximum V^* exists) an ϵ-approximate solution is a point x such that $Ax \leq b$ and $c^T x \geq V^* - \epsilon$. This definition is not satisfactory since it is not clear whether an ϵ-approximation algorithm is required to decide the existence of an x such that $Ax \leq b$, and the boundedness of the function $c^T x$ on the feasible domain. If indeed it is required to decide these questions then in the worst-case sense this approximation problem is trivially equivalent to the exact problem.

The consequences of the ellipsoid algorithm with respect to approximation problems on convex sets are studied in [7]. It is not clear whether these results can be applied to achieve the type of results we seek here. The reason is that over the real numbers it seems difficult to obtain estimates of the radii of a circumscribing sphere and an inscribed sphere. The main complexity result on convex minimization in [7] (Theorem 2.2.15) assumes the convex set is given with estimates of such radii. Our main interest here is the question of what is a reasonable sense of approximation when the algorithm fails to classify the instance correctly as feasible, unbounded, etc.

In Section 2 we give some preliminaries and discuss the difficulties involved in classifying the problem. In Section 3 we discuss approximate solutions based on satisfying a termination criterion within some tolerance. In Section 4 we discuss a notion of approximation which is based on solving a perturbed instance exactly. Section 5 gives an analysis of complexity for various notions of complexity.

2. Preliminaries

We pointed out in the introduction that the practical situation usually dictates the right notion of approximate solution. For a theoretical discussion it is often convenient to consider the problem in the symmetric form $Sym(A, b, c)$ (see Section 1). Traditionally, an exact algorithm for this problem (for example, the simplex method) is supposed to provide the user with information as follows. It has to classify the problem into one of the following three categories:

(i) Infeasible (the domain X defined by $\boldsymbol{Ax} \leq \boldsymbol{b}$ and $\boldsymbol{x} \geq \boldsymbol{0}$ is empty).

(ii) Feasible and bounded (there exists a maximizer of $\boldsymbol{c}^T\boldsymbol{x}$ over X).

(iii) Feasible and unbounded (the function $\boldsymbol{c}^T\boldsymbol{x}$ is unbounded over X).

In case (ii) the algorithm has to provide an optimal solution. It may also be required in case (iii) to provide a ray contained in X along which $\boldsymbol{c}^T\boldsymbol{x}$ tends to infinity. A nice property of the simplex method is that it also solves the dual problem:

$$\text{Minimize } \boldsymbol{b}^T\boldsymbol{y}$$
$$\text{subject to } \boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{c}$$
$$\boldsymbol{y} \geq \boldsymbol{0} \; .$$

Thus, besides providing such a ray in case (iii), it also provides in case (ii) an optimal solution to the dual, and in case (i) a "certificate" in the form of a ray of a related problem.

In fact, the (exact) simplex method always computes a *basis* which provides the required information. Specifically, it provides a representation of the problem (by a suitable linear transformation of the space) from which the classification and the numerical values of both the primal and the dual variables are transparent. Thus, the simplex method classifies problems into one of *four* categories (even though the commonly used variants do not distinguish between IF and II):

(i) FF: primal feasible, dual feasible,

(ii) FI: primal feasible, dual infeasible (that is, unbounded primal),

(iii) IF: primal infeasible, dual feasible (here the dual is unbounded), and

(iv) II: primal infeasible, dual infeasible.

It is quite common to include this classification in the requirements from an exact algorithm for the general linear programmingproblem. We refer to it later as the classification problem of linear programs.

An approximation algorithm should be expected sometimes to fail in classifying the input into the categories FF, FI, IF and II. Interestingly, the existence of a strongly polynomial algorithm for the classification problem implies the existence of one for the problem itself (see p. 445 in [1]).

So far we have discussed the subject of approximation under the assumption that the result should be "close" to the true one. However, a different approach can sometimes be useful. We may allow the algorithm to be totally wrong in a small number of cases. This approach is appropriate when the output space of the algorithm is discrete and has no natural metric associated with it. For example, consider the following trivial problem: given two numbers α, β, recognize whether $\alpha > \beta$ or $\alpha \leq \beta$. Suppose the comparison of

α to β can be performed with arbitrary finite precision. Thus, for any given $\epsilon > 0$, we can recognize either that

$$\alpha \leq \beta + \epsilon$$

or that

$$\alpha \geq \beta - \epsilon \ .$$

The algorithm reports $\alpha \leq \beta$ in the first case and $\alpha > \beta$ in the second one. Thus, the algorithm gives the correct answer if

$$|\beta - \alpha| > \epsilon$$

but may fail otherwise. The grey area is the set of pairs (α, β) such that $|\alpha - \beta| \leq \epsilon$. Of course, the smaller ϵ the smaller the grey area. Thus, the measure of the grey area reflects the quality of the approximation.

The problem of the preceding paragraph can be cast as a linear programmingproblem:

$$\text{Maximize } x_1$$
$$\text{subject to } \alpha x_1 \ \leq \ x_2 \ \leq \ \beta x_1 \ .$$

Here the point $(0, 0)$ is feasible for any α and β. The problem is unbounded if and only if $\alpha \leq \beta$. This suggests that the grey area approach would be suitable for the classification problem of linear programs.

A general linear programming problem in standard form:

$$(SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})) \qquad \begin{array}{l} \text{Maximize } \ \boldsymbol{c}^T \boldsymbol{x} \\ \text{subject to } \ \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ \qquad\qquad\ \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

is determined by $\boldsymbol{A} \in R^{m \times n}$, $\boldsymbol{b} \in R^m$ and $\boldsymbol{c} \in R^n$. There is a one-to-one correspondence between problems of order $m \times n$ and points of $\Omega = R^{mn+m+n}$. The classification corresponds to a partition of Ω into four sets: FF, FI, IF and II as discussed above. For example, IF is the set of triples $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ that determine infeasible primal problems whose dual problems are feasible (and hence unbounded).

Let Ω' denote the union of the boundaries of these four sets. Obviously, an approximation algorithm (for the classification problem) may fail if the input $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is close to Ω'. For example, if an instance is close to the common boundary of FF and FI, but far from the boundaries of IF and II, then an approximation algorithm is expected to recognize that the problem is feasible, but is expected to fail in deciding whether it is bounded. Interestingly, there are more "pathological" cases. In fact, the intersection of all four boundaries, which we denote by Ω_4, is not empty. Thus, given an instance close to the intersection of the four boundaries, an approximation algorithm may not be able

to recognize anything in terms of the above classification. This observation is obvious in view of the invariance of the classification under multiplication of columns and rows by positive scalars. Thus, the neighborhood of the origin is obviously pathological in this sense. The difficulties with the origin can be avoided by scaling rows and columns. However, it is easy to construct other examples with similar characteristics.

Proposition 2.1. *The instance*

$(*)$
$$\begin{aligned}
\text{Maximize} \quad & x_1 - x_2 \\
\text{subject to} \quad x_1 - x_2 \ & \leq \ -1 \\
-x_1 + x_2 \ & \leq \ -1 \\
x_1, x_2 \ & \geq \ 0 \ .
\end{aligned}$$

belongs to Ω_4.

Proof: It is easy to see that $(*)$ itself is in IF. Using small perturbations, one can move from $(*)$ to instances in any of the other three classes. More precisely, if only c_1 is slightly increased then we can get instances in II. If only A_{21} is slightly decreased then we get instances in FF. Finally, if only A_{11} is slightly increased then an instance in FI is obtained. ∎

We note that most of the numerical difficulties with solving linear programming problems are due the fact that many such problems are ill-posed. It is well-known in numerical analysis (see, e.g., [3]) that near-singularities in the matrix $\boldsymbol{A}$ can cause problems. However, in this paper we also discuss intrinsic aspects of approximate solutions which arise even when exact arithmetic is used. For example, if one is interested only in an approximate solution, what should be a good termination criterion? Becasue of such questions we have to deal with perturbations of the vectors $\boldsymbol{b}$ and $\boldsymbol{c}$ and not only of the matrix $\boldsymbol{A}$.

Due to the classification aspect of the problem, we clearly cannot always measure the quality of the approximation by the distance between the exact solution and the approximate one (neither in terms of the solution vector nor in terms of the objective function value). Thus a different approach to approximation may be proposed for a general situation, where there is some natural metric on the input space, but there does not seem to exist one for the output space. The following definition is similar to backward analysis of errors in numerical analysis [9].

Definition 2.2. Let $M = (S, d)$ be a metric space and let f be a mapping from S into some set T which does not necessarily have any metric associated with it. A mapping $g : S \rightarrow T$ is called *an ϵ-approximation to f* if for every $\boldsymbol{x} \in S$, there exists an $\boldsymbol{x}' \in S$ such that $d(\boldsymbol{x}, \boldsymbol{x}') < \epsilon$ and $g(\boldsymbol{x}) = f(\boldsymbol{x}')$.

In linear programming the output space (including the classification) does have a metric structure. Besides the classification information, there are also numerical values associated with the variables. One might propose for the linear programming problem the following approach to approximation by posing the following problem:

Problem 2.3. Given the problem $SF(A, b, c)$ and $\epsilon > 0$, name a class $S \in \{FF, FI, IF, II\}$ and assign numerical values to the variables so that the following condition is satisfied: there exists an instance (A', b', c') in S for which the numerical values are correct within an error of ϵ, such that $\|(A, b, c) - (A', b', c')\|_\infty < \epsilon$.

The approach represented by Problem 2.3 takes care of pathological cases where some other approaches fail. Consider, for comparison, a different notion of approximate solution of systems of inequalities reflected in the following problem:

Problem 2.4. Given A, b and $\epsilon > 0$, either give an x such that $Ax \leq b + \epsilon e$ or conclude that there is no x such that $Ax \leq b - \epsilon e$.

The approach represented by Problem 2.4 seems to be a natural generalization of the obvious approximate comparison of two real numbers. Its weakness is apparent in the following example. Consider the problem

$$\alpha x_1 - x_2 \leq -1$$
$$-x_1 + x_2 \leq -1 \ .$$

If $\alpha = 1$ then, obviously, for every $\epsilon < 1$ the system

$$x_1 - x_2 \leq -1 + \epsilon$$
$$-x_1 + x_2 \leq -1 + \epsilon$$

is infeasible. However, for any $\alpha \neq 1$ the system is feasible for every ϵ. Thus, in order to solve Problem 2.4 we have to know whether $\alpha = 1$. This example can easily be generalized so that in order to solve Problem 2.4 one has to know whether a certain matrix is singular. The latter involves some numerical difficulties in practice.

The weakness of the concept of Problem 2.4 is that it considers perturbations of the given problem only in a limited and quite arbitrary way. In Problem 2.3 we allow perturbations in all directions. We note that for certain classes of linear programming problems (e.g., the min-cost flow problem) certain coefficients have the values 1 or 0 throughout the class. In such cases we would allow only perturbations within the subject class. Thus, if the concept represented by Problem 2.3 were to apply to a min-cost flow problem $SF(A, b, c)$ then we would require that $A' = A$.

3. Tolerance-based approximation

In this section we discuss the issue of approximation as it arises in the context of the simplex method. There are two ways to look at the question. First, imagine we run the simplex algorithm using exact arithmetic but have to compute only "ϵ-approximate" solutions. Thus, rather than running the algorithm to the end, we seek to apply some stopping rule that guarantees our output to be ϵ-approximate. The interesting problem is of course to devise such stopping rules for various concepts of approximation. Another way to look at the question is to realize that on a machine we usually have numerical errors and thus we almost always have to specify some "tolerance" within which we accept our results. It is important to know the implications of using a certain tolerance with regard to the results.

Consider the problem $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. In the Appendix we review some properties of basic solutions and how the simplex method uses them (see the Appendix for the notation).

In practice, one usually works with some "tolerance" $\delta > 0$, so that any number $a \leq \delta$ is accepted as nonpositive and any $a \geq -\delta$ is accepted as nonnegative. This suggests another approach to approximation, which may be called the "tolerance" approach:

Definition 3.1. A basic solution $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{B}) = (\boldsymbol{x}_B, \boldsymbol{x}_N)$ (where $\boldsymbol{x}_B = \boldsymbol{B}^{-1}\boldsymbol{b}$ and $\boldsymbol{x}_N = \boldsymbol{0}$) is *optimal with tolerance* δ for $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ if for every j,

$$x_j \geq -\delta \quad \text{and} \quad c_j \leq \boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_j + \delta \ .$$

Obviously, such a solution is not necessarily feasible. It is only "approximately feasible" in the sense that the equality constraints are satisfied while the nonnegativity constraints are approximately satisfied. Analogously, the dual vector $\boldsymbol{y}(\boldsymbol{B})$ is approximately feasible in the dual problem. Moreover, the vectors $\boldsymbol{x}(\boldsymbol{B})$ and $\boldsymbol{y}(\boldsymbol{B})$ satisfy the complementary slackness condition:

$$x_j(\boldsymbol{y}^T \boldsymbol{A}_j - c_j) = 0$$

which is necessary for optimality. This implies that both vectors yield the same objective function value in their respective problem:

$$\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{c}^T \boldsymbol{B}^{-1} \boldsymbol{b} = \boldsymbol{y}^T \boldsymbol{b} \ .$$

Note that the problem may also be feasible and unbounded but still an optimal solution with tolerance δ may exist. One can also define a notion of an unbounded ray with tolerance δ.

4. Perturbation-based approximation

It is interesting to observe that from a solution that is optimal with small tolerance we can easily obtain an exact solution to an instance which is close to the given one. More precisely, we have the following proposition.

Proposition 4.1. *Suppose $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{B})$ is a basic solution which is optimal with tolerance δ. Let $\boldsymbol{x}'$ be defined by $x'_j = x_j + \delta$ for j associated with $\boldsymbol{B}$ and $x_j = 0$ otherwise. Also, let $\boldsymbol{c}'$ be defined by $c'_j = c_j$ for j associated with $\boldsymbol{B}$ and $c'_j = c_j - \delta$ otherwise, and let $\boldsymbol{b}' = \boldsymbol{b} + \delta\boldsymbol{B}\boldsymbol{e}$. Under these conditions, $\boldsymbol{x}'$ is an optimal solution for the problem $SF(\boldsymbol{A}, \boldsymbol{b}', \boldsymbol{c}')$, and $\boldsymbol{y} = \boldsymbol{B}^{-T}\boldsymbol{c}_B$ is optimal for its dual problem.*

Proof: We have

$$\boldsymbol{x}'_B = \boldsymbol{B}^{-1}\boldsymbol{b}' = \boldsymbol{x}_B + \delta\boldsymbol{e} \geq \boldsymbol{0} \ ,$$

$$\boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{c}'$$

and $\boldsymbol{x}'$ and $\boldsymbol{y}$ satisfy the complementary slackness conditions in $SF(\boldsymbol{A}, \boldsymbol{b}', \boldsymbol{c}')$. ∎

In simpler words, we have

Corollary 4.2. *If $\boldsymbol{x}$ is optimal with tolerance δ for $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ then there exist $\boldsymbol{b}'$ and $\boldsymbol{c}'$ and an optimal solution $\boldsymbol{x}'$ for $SF(\boldsymbol{A}, \boldsymbol{b}', \boldsymbol{c}')$ such that*

$$\|\boldsymbol{x}' - \boldsymbol{x}\|_\infty \ , \ \|\boldsymbol{c}' - \boldsymbol{c}\|_\infty \quad \leq \quad \delta \ ,$$

and

$$\|\boldsymbol{b}' - \boldsymbol{b}\|_\infty \quad \leq \quad \min\{\delta\|\boldsymbol{A}\|_\infty, mT\delta\}$$

where $\|\boldsymbol{A}\|_\infty$ is the usual operator norm, corrsponding to the vector supremum norm $\|\cdot\|$, T is the maximum absolute value of any entry in $\boldsymbol{A}$ and m is the number of rows of $\boldsymbol{A}$.

A analogous proposition can be proven with respect to the unbounded case:

Proposition 4.3. *Suppose $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{B})$ is a basic solution such that*

$$x_j \geq -\delta$$

and for some k not associated with $\boldsymbol{B}$,

$$c_k > \boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{A}_k$$

and

$$\boldsymbol{B}^{-1}\boldsymbol{A}_k \geq -\delta\boldsymbol{e} \ .$$

Let $\boldsymbol{c}'$ be the same as $\boldsymbol{c}$ except that $c'_k = c_k + \delta\boldsymbol{c}_B^T\boldsymbol{e}$, and let $\boldsymbol{b}' = \boldsymbol{b} + \delta\boldsymbol{B}\boldsymbol{e}$. Also let $\boldsymbol{A}'$ be the same as $\boldsymbol{A}$ except that $\boldsymbol{A}'_k = \boldsymbol{A}_k + \delta\boldsymbol{B}\boldsymbol{e}$. Under these conditions, the problem $SF(\boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}')$ is unbounded.

Proof: In the new problem the basis $\boldsymbol{B}$ certifies unboundedness since

$$\boldsymbol{B}^{-1}\boldsymbol{b}' \geq \boldsymbol{0} \ ,$$

$$c_k' > (c')_B^T \boldsymbol{B}^{-1} \boldsymbol{A}_k'$$

and

$$\boldsymbol{B}^{-1}\boldsymbol{A}_k' \geq \boldsymbol{0} \ .$$

∎

Corollary 4.4. *If $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is concluded within tolerance δ to be unbounded, then there exist $\boldsymbol{A}'$, $\boldsymbol{b}'$ and $\boldsymbol{c}'$ such that $SF(\boldsymbol{A}', \boldsymbol{b}', \boldsymbol{c}')$ is unbounded,*

$$\|\boldsymbol{A}' - \boldsymbol{A}\|_1 \ \leq \ \delta \|\boldsymbol{A}\|_\infty$$

(where $\|\boldsymbol{A}' - \boldsymbol{A}\|_1$ is the operator norm corresponding to the vector norm $\|\cdot\|_1$,

$$\|\boldsymbol{b}' - \boldsymbol{b}\|_\infty \ \leq \ \delta \|\boldsymbol{A}\|_\infty \ ,$$

and

$$\|\boldsymbol{c}' - \boldsymbol{c}\|_\infty \ \leq \ \delta \|\boldsymbol{c}\|_1 \ .$$

5.   Complexity questions

We start this section with yet another variant of an approximation problem. Again, we consider approximation concepts which avoid the difficulties involved in the classification problem. Suppose the exact problem is given in the dual form:

$(DF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}))$
$$\text{Minimize } \boldsymbol{c}^T \boldsymbol{x}$$
$$\text{subject to } \boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$$

where the output has to be one of the following:

(i) a point $\boldsymbol{x}^*$ which minimizes $\boldsymbol{c}^T \boldsymbol{x}$ subject to $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$,

(ii) a point $\boldsymbol{x}^*$ and a scalar $t^* > 0$ which minimize the value of t subject to $\boldsymbol{A}\boldsymbol{x} + t\boldsymbol{e} \geq \boldsymbol{b}$ (in which case the problem is infeasible), or

(iii) vectors $\boldsymbol{x}$ and $\boldsymbol{u}$ such that $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$, $\boldsymbol{c}^T \boldsymbol{u} < 0$ and $\boldsymbol{A}\boldsymbol{u} \geq \boldsymbol{0}$ (in which case the problem is unbounded).

The above motivates the definition of the following approximation problem:

Problem 5.1. Denote the optimal value of a given problem $DF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ by V^* (allowing $V^* = \pm\infty$) and let t^* denote the minimum of t subject to $\boldsymbol{A}\boldsymbol{x} + t\boldsymbol{e} \geq \boldsymbol{b}$. Given a number $\epsilon > 0$, output one of the following:

(i) a point $\boldsymbol{x}$ such that $\boldsymbol{c}^T\boldsymbol{x} \le V^* + \epsilon$ and $\boldsymbol{Ax} \ge \boldsymbol{b} - \epsilon\boldsymbol{e}$,

(ii) a point $\boldsymbol{x}$ and a scalar t, $-\epsilon \le t \le t^* + \epsilon$ such that $\boldsymbol{Ax} + t\boldsymbol{e} \ge \boldsymbol{b} - \epsilon\boldsymbol{e}$, or

(iii) vectors $\boldsymbol{x}$ and $\boldsymbol{u}$ such that $\boldsymbol{Ax} \ge \boldsymbol{b} - \epsilon\boldsymbol{e}$, $\boldsymbol{c}^T\boldsymbol{u} < \epsilon$ and $\boldsymbol{Au} \ge -\epsilon\boldsymbol{e}$.

It is interesting to look at the question of the existence of an algorithm for the approximation problem, where the number of operations is expressed in terms of ϵ as well.

Proposition 5.2. *Over any ordered field, if Problem 5.1 can be solved in $f(m, n, \epsilon)$ field operations (including comparisons) then it can be solved in $g(m, n) = O(f(m, n, 1))$ operations.*

Proof: Suppose $\mathcal{A}$ is an algorithm for Problem 5.1 which runs in $f(m, n, \epsilon)$ field operations. Given $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$ and $\epsilon > 0$, let

$$\bar{\boldsymbol{A}} = \epsilon^{-1}\boldsymbol{A} \quad , \quad \bar{\boldsymbol{b}} = \epsilon^{-1}\boldsymbol{b} \quad , \quad \bar{\boldsymbol{c}} = \epsilon^{-1}\boldsymbol{c} \ .$$

The instance $DF(\bar{\boldsymbol{A}}, \bar{\boldsymbol{b}}, \bar{\boldsymbol{c}})$ is equivalent to $DF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Moreover, a valid output for $DF(\bar{\boldsymbol{A}}, \bar{\boldsymbol{b}}, \bar{\boldsymbol{c}})$ with precision $\epsilon = 1$ is also a valid output for $DF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ with the prescribed precision ϵ. ∎

Corollary 5.3. *If there exists a polynomial $f(m, n, \epsilon)$ such that Problem 5.1 with rational data can be solved in $f(m, n, \epsilon)$ arithmetic operations, then the exact problem with rational data can be solved in a polynomial number of operations.*

Proof: For a problem with rational data it is easy to determine a value ϵ such that an exact solution can be computed from a solution of Problem 5.1 in a polynomial number of operations. Thus, the problem can be scaled so that $\epsilon = 1$ suffices for determining an exact solution. ∎

In view of Proposition 5.2 it is reasonable to ask whether Problem 5.1 can be solved in a polynomial number of operations in terms of m, n and $\log \frac{R}{\epsilon}$, where $R = R(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) > 0$ is some quantity such that for any positive scalar λ

$$R(\lambda\boldsymbol{A}, \lambda\boldsymbol{b}, \lambda\boldsymbol{c}) = \lambda R(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$$

(e.g., R equals the maximum absolute value of any input coefficients.) Consider first the feasibility problem:

$$(FB(\boldsymbol{A}, \boldsymbol{b})) \qquad \begin{array}{l} \text{Minimize } t \\ \text{subject to } \boldsymbol{Ax} + t\boldsymbol{e} \ge \boldsymbol{b} \\ \qquad\qquad t \ge -1 \ , \end{array}$$

and the associated approximation problem:

Problem 5.4. Given $FB(\boldsymbol{A}, \boldsymbol{b})$ and $\epsilon > 0$, find $\boldsymbol{x}$ and t such that $\boldsymbol{Ax} + t\boldsymbol{e} \geq \boldsymbol{b}$ and $t < t^* + \epsilon$ (where t^* is the minimum of $FB(\boldsymbol{A}, \boldsymbol{b})$).

Before stating the next proposition, recall that the set of solutions of the system $\boldsymbol{Ax} \geq \boldsymbol{b}$ is bounded for every $\boldsymbol{b}$ if and only the rows of $\boldsymbol{A}$ span the space R^n in nonnegative linear combinations.

Proposition 5.5. *If Problem 5.4 is given with real data such that the rows of $\boldsymbol{A}$ span the space R^n in nonnegative linear combinations, then it can be solved in polynomial number of arithmetic operations in terms of m, n and*

$$\rho = \log\left(\frac{b_{\max} + 1}{\epsilon}\right)$$

(assuming $b_{\max}$, the maximal b_i, is positive).

Proof: The point $\boldsymbol{x}^0 = \boldsymbol{0}$, $t^0 = b_{\max} + 1$ is in the interior of the feasible domain of $FB(\boldsymbol{A}, \boldsymbol{b})$. By our assumption, the set of optimal solutions of $FB(\boldsymbol{A}, \boldsymbol{b})$ is bounded. Several interior point algorithms are now known (e.g., [2; 10]) which can start from any interior point and reduce the value of the objective function to a value not greater than $t^* + \epsilon$ in a polynomial number of iterations in terms m, n and ρ, where each iteration takes a polynomial number of operations in terms of m and n. ∎

An obvious consequence of Proposition 5.5 is the following:

Corollary 5.6. *Suppose the rows of $\boldsymbol{A}$ spanned the whole R^n in nonnegative linear combinations. It takes a polynomial number of operations in terms of m, n and ρ to either compute a vector $\boldsymbol{x}$ such that $\boldsymbol{Ax} \geq \boldsymbol{b}$ or conclude that there is no $\boldsymbol{x}$ such that $\boldsymbol{Ax} \geq \boldsymbol{b} + \epsilon\boldsymbol{e}$.*

Proof: Run any of the polynomial interior point algorithms for a number of iterations which guarantees that $t < t^* + \epsilon$. As soon as t becomes nonpositive, stop (the current $\boldsymbol{x}$ is feasible). If at the end t is still positive then $t^* > -\epsilon$ and hence there is no $\boldsymbol{x}$ such that $\boldsymbol{Ax} - \epsilon\boldsymbol{e} \geq \boldsymbol{b}$. ∎

Another consequence with respect to optimal solutions with tolerance can be stated conveniently when the problem is in the symmetric form $Sym(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$:

$$\text{Maximize} \quad \boldsymbol{c}^T\boldsymbol{x}$$
$$\text{subject to} \quad \boldsymbol{Ax} \leq \boldsymbol{b}$$
$$\boldsymbol{x} \geq \boldsymbol{0} \,,$$

whose dual is

$$\text{Minimize} \ \ \boldsymbol{b}^T \boldsymbol{y}$$
$$\text{subject to} \ \ \boldsymbol{A}^T \boldsymbol{y} \geq \boldsymbol{c}$$
$$\boldsymbol{y} \geq \boldsymbol{0} \ .$$

Recall that by the duality theorem, a feasible solution $\boldsymbol{x}$ is optimal if and only if there exists a dual feasible solution $\boldsymbol{y}$ such that $\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{b}^T \boldsymbol{y}$. On the other hand, $\boldsymbol{c}^T \boldsymbol{x} \leq \boldsymbol{b}^T \boldsymbol{y}$ for any pair of feasible $\boldsymbol{x}$ and $\boldsymbol{y}$. This suggests the following approximation problem:

Problem 5.7. Given $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}$ and $\epsilon > 0$, either find a pair of vectors $\boldsymbol{x}, \boldsymbol{y}$ such that

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b} + \epsilon \boldsymbol{e}$$
$$\boldsymbol{A}^T \boldsymbol{y} \geq \boldsymbol{c} - \epsilon \boldsymbol{e}$$
$$\boldsymbol{b}^T \boldsymbol{y} - \boldsymbol{c}^T \boldsymbol{x} \leq \epsilon$$
$$\boldsymbol{x} \geq -\epsilon \boldsymbol{e}, \qquad \boldsymbol{y} \geq -\epsilon \boldsymbol{e}$$

or conclude that $Sym(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ does not have an optimal solution.

The notion of approximation presented in Problem 5.7 is very close to the one used in practice, where optimality criteria are applied without knowledge of proximity to the value of an optimal solution.

Note that Problem 5.7 is trivial if $\boldsymbol{c} \leq \boldsymbol{0} \leq \boldsymbol{b}$. Thus, assume

$$\gamma = \max\{-b_{\min}, c_{\max}\} > 0$$

and denote

$$\rho^* = \log\left(\frac{\gamma + 1}{\epsilon}\right) \ .$$

Proposition 5.8. *Suppose the rows of the matrix*

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{A}^T \\ -\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{c}^T & -\boldsymbol{b}^T \\ \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$$

span the space R^{m+n} in nonnegative linear combinations. Problem 5.7 can be solved in a polynomial number of operations in terms of m, n and ρ^*.

Proof: Consider the problem

$$\text{Minimize} \ \ t$$
$$\text{subject to} \ \ \boldsymbol{A}\boldsymbol{x} - t\boldsymbol{e} \leq \boldsymbol{b}$$
$$\boldsymbol{A}^T \boldsymbol{y} + t\boldsymbol{e} \geq \boldsymbol{c}$$
$$\boldsymbol{b}^T \boldsymbol{y} - \boldsymbol{c}^T \boldsymbol{x} - t \leq 0$$
$$\boldsymbol{x} + t\boldsymbol{e}, \ \boldsymbol{y} + t\boldsymbol{e} \geq \boldsymbol{0} \ .$$

13

Starting at $t = \gamma + 1$, $\boldsymbol{x} = \boldsymbol{0}$ and $\boldsymbol{y} = \boldsymbol{0}$, run a number of iterations which guarantees $t \leq t^* + \epsilon$. This number is polynomial in m, n and ρ^*, and ρ^* is trivial to compute. If $t \leq \epsilon$ then the current $\boldsymbol{x}$ and $\boldsymbol{y}$ solve Problem 5.7. Otherwise, $t^* > 0$ and $Sym(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ does not have an optimal solution. ∎

Note that a solution of Problem 5.7 does not guarantee that $\boldsymbol{c}^T \boldsymbol{x}$ is close to the optimal value V^* when the latter exists. It seems much more difficult to solve the approximation problem in the latter sense. This difficulty can be explained by considering the following practical question which arises when one applies the simplex method to the problem in standard form $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$
$$\boldsymbol{x} \geq -\delta \boldsymbol{e}$$
$$\boldsymbol{A}^T \boldsymbol{y} \geq \boldsymbol{c} - \delta \boldsymbol{e}$$
$$\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{b}^T \boldsymbol{y} \ .$$

Assuming the problem has an optimum whose value is V^*, we are interested in finding what should be the value of δ in order to ensure that

$$\boldsymbol{c}^T \boldsymbol{x} \geq V^* - \epsilon \ .$$

To answer the question, suppose $\boldsymbol{x}^*$ is an optimal solution and we get

$$\boldsymbol{c}^T \boldsymbol{x} = \boldsymbol{b}^T \boldsymbol{y} = \boldsymbol{y}^T \boldsymbol{A} \boldsymbol{x}^* \geq \boldsymbol{c}^T \boldsymbol{x}^* - \delta \boldsymbol{e}^T \boldsymbol{x}^* \ .$$

Thus, we have to estimate the quantity $\boldsymbol{e}^T \boldsymbol{x}^*$.

As is known from the analysis of the ellipsoid algorithm, the coordinates of a basic solution $\boldsymbol{x}^*$ can be bounded as follows. First, $\boldsymbol{x}^*$ satisfies an equation $\boldsymbol{B}\boldsymbol{x}^* = \boldsymbol{b}$, where $\boldsymbol{B}$ is a nonsingular square submatrix of $\boldsymbol{A}$. Thus,

$$x_i^* = \frac{\det(\boldsymbol{B}^i)}{\det(\boldsymbol{B})}$$

where $\boldsymbol{B}^i$ is the matrix obtained from $\boldsymbol{B}$ by substituting $\boldsymbol{b}$ for the i'th column. The known analyses do not use this relationship between $\boldsymbol{B}^i$ and $\boldsymbol{B}$. In the case of integer coefficients an obvious lower bound on the absolute value of a nonzero determinant is 1. Thus, a lower bound can also be obtained in the case of rational coefficients. It is not known whether over the reals the following problem can be solved in a polynomial number of arithmetic operations and comparisons:

Problem 5.9. Given a real matrix $\boldsymbol{A} \in R^{m \times n}$, compute a positive α such that for every nonsingular submatrix $\boldsymbol{B} \in R^{m \times m}$ of $\boldsymbol{A}$, $|\det(\boldsymbol{B})| \geq \alpha$.

An tight upper bound on the absolute value of a determinant in terms of the maximum value M of any entry can be obtained over the reals as follows. Suppose the columns of $\boldsymbol{B}$ are $\boldsymbol{v}^1, \cdots, \boldsymbol{v}^m$. Obviously,

$$|\det(\boldsymbol{B})| \leq \|\boldsymbol{v}^1\| \cdots \|\boldsymbol{v}^m\| \leq (\sqrt{m}\ M)^m \ .$$

At least for values of m for which there exist Hadamard matrices (i.e., matrices of orthogonal columns consisting of ± 1's), there exist matrices with determinant $(\sqrt{m}\ M)^m$.

Without exploiting the relationship between $\boldsymbol{B}^i$ and $\boldsymbol{B}$ the only claim we can prove is that

$$x_i^* \leq \frac{(\sqrt{m}\ M)^m}{\eta(\boldsymbol{A})}$$

where $\eta(\boldsymbol{A})$ is any positive lower bound on the absolute value of the determinant of any nonsingular $m \times m$ submatrix of $\boldsymbol{A}$. We note in passing that in the case of integral coefficients the bounds on x_i^* cannot be improved dramatically. For example, if

$$\boldsymbol{B} = \begin{bmatrix} 1 & & & & \\ -M & 1 & & & \\ -M & -M & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -M & -M & \cdots & -M & 1 \end{bmatrix}$$

and $\boldsymbol{b} = (M, \cdots, M)^T$ then $x_m^* = \sum_{i=1}^m M^i$. Also, if

$$\boldsymbol{B} = \begin{bmatrix} M & M & M & \cdots & M \\ -1 & M & M & \cdots & M \\ & -1 & M & \cdots & M \\ & & \ddots & \ddots & \vdots \\ & & & -1 & M \end{bmatrix}$$

and $\boldsymbol{b} = (1, 0, \cdots, 0)^T$ then $x_m^* = \frac{1}{M(M+1)^{m-1}}$.

Recall that to guarantee an ϵ-approximation in terms of the function value, we have to choose δ such that

$$\delta \leq \frac{\epsilon}{e^T \boldsymbol{x}^*}$$

hence we get the estimate

$$\delta \leq \frac{\epsilon\ \eta(\boldsymbol{A})}{m(\sqrt{m}\ T)^m}\ .$$

Another estimate can be derived by

$$e^T \boldsymbol{x}^* \leq \sqrt{m}\ \|\boldsymbol{x}^*\| = \sqrt{m}\ \|\boldsymbol{B}^{-1}\boldsymbol{b}\| \leq \frac{\sqrt{m}\ \|\boldsymbol{b}\|}{\sqrt{\lambda_{\min}(\boldsymbol{B}\boldsymbol{B}^T)}}$$

15

where $\lambda_{\min}$ denotes the least eigenvalue. Thus, what we would need is a lower bound on the least eigenvalue of any matrix of the form $\boldsymbol{B}\boldsymbol{B}^T$ where $\boldsymbol{B}$ is a basis. Note that our estimates depend on properties of an optimal basis rather one which supports the approximate solution. So, the optimal basis may be ill-conditioned, this fact being unknown to the user, and the terminal basis well-conditioned and satisfying the optimality conditions within tolerance.

Obviously, if any $\eta(\boldsymbol{A})$ is known then we can solve the ϵ-approximation problem in a polynomial number of operations in terms of m, n and $-\log \delta$. However, in general η is not known. It is interesting to note that for a practical solution of problems with thousands of variables, even with a sparse and well-structured problem with small integral coefficients, the required δ may be too small to be practical. It is not clear that an approximate solution based on tolerance has a value close to the optimal. In fact, the value may be far from the optimum even though the duality gap is very small, since the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are only approximately feasible.

The following example illustrates the difficulties described above. Suppose we solve a problem in standard form and the representation using the current basis is:

$$\text{Maximize } \tilde{\boldsymbol{c}}_N \boldsymbol{x}_N$$
$$\text{subject to } \boldsymbol{x}_B + \tilde{\boldsymbol{N}}\boldsymbol{x}_N = \boldsymbol{e}$$
$$\boldsymbol{x}_B, \boldsymbol{x}_N \geq \boldsymbol{0} \ ,$$

where $\tilde{\boldsymbol{N}} \in R^{m \times (n-m)}$ contains the following $m \times m$ submatrix:

$$\boldsymbol{B}' = \begin{bmatrix} 1 & & & & \\ -2 & 1 & & & \\ & -2 & 1 & & \\ & & & \ddots & \ddots & \\ & & & & -2 & 1 \end{bmatrix}$$

Suppose further that the coordinates of $\tilde{\boldsymbol{c}}_N$ are negative except those corresponding to the columns of $\boldsymbol{B}'$ which are all equal to zero except the last one which equals 10^{-20}. Suppose $m = 1000$, which is quite common in practice. Assuming 10^{-50} is considered nonpositive within tolerance, the solution $\boldsymbol{x} = \boldsymbol{e}$ is accepted as optimal, with an objective function value of 0. However, the basis $\boldsymbol{B}'$ determines a feasible solution $\boldsymbol{x}$ where $x_{B'_i} = 2^i - 1$ and the objective function value is greater then 10^{250}. Note that the current basis $\boldsymbol{B}$ is very well-conditioned. Moreover, the underlying matrix is very sparse and well-structured.

Appendix

We review some known characteristics of the simplex method for problems in standard form $SF(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Suppose, for simplicity, we run only "phase II" of the algorithm i.e.,

we start from a basic feasible solution and attempt to find an optimal one or conclude that the problem is unbounded. (It is well known that if a problem in standard form has a feasible solution then it has a basic feasible one; the "phase I" problem of finding a basic feasible solution, or concluding that none exists, can be formulated as a problem in standard form with a known basic feasible solution.) Assuming exact arithmetic, the algorithm then terminates with a feasible basis. The termination criterion is stated in terms of signs of certain entries of the "tableau". As common, let $B \in R^{m \times m}$ denote the nonsingular submatrix of A whose columns constitute the current basis, and let N denote the matrix consisting of the other columns. Let x_B and c_B denote the restrictions of the vectors x and c, respectively, to the indices corresponding to the columns of B. Let x_N and c_N denote the complementary restrictions of the vectors. The "tableau" is essentially a representation of the problem in an equivalent form:

$$\text{Maximize} \ \ (c_N^T - c_B^T B^{-1} N) x_N$$
$$\text{subject to} \ \ x_B + B^{-1} N x_N = B^{-1} b$$
$$x_B \geq 0 \ \ , \ \ x_N \geq 0 \ .$$

Given a basis B, the corresponding basic primal solution $x = x(B)$ is given by

$$x_B = B^{-1} b \ \ , \ \ x_N = 0 \ .$$

The basic dual vector $y = y(B)$ associated with B is given by the equation

$$y^T B = c_B^T \ .$$

The dual problem is:

$$\text{Minimize} \ \ y^T b$$
$$\text{subject to} \ \ y^T A \geq c^T \ .$$

Thus, $y(B)$ is feasible in the dual problem if and only if

$$c_N^T \leq c_B^T B^{-1} N \ ,$$

in which case y is optimal for the dual.

Assuming the algorithm (namely, the primal simplex method) works with exact arithmetic, for every B occurring in the process,

$$B^{-1} b \geq 0 \ .$$

The algorithm terminates in one of the following cases:

(i) $c_N^T \leq c_B^T B^{-1} N$, in which case B is optimal, or
(ii) there exists a column N_j such that $c_j > c_B^T B^{-1} N_j$ and $B^{-1} N_j \leq 0$, in which case the problem is unbounded.

In either case the basis B is said to be *terminal*. The termination criterion applies to signs of certain entries in the tableau.

17

References

[1] V. Chvátal, *Linear Programming*, W. H. Freeman, New York, 1983.

[2] D. M. Gay, "A variant of Karmarkar's linear programming algorithm for problems in standard form," *Mathematical Programming* **37** (1987) 81–90.

[3] G. H. Golub and C. Van Loan, *Matrix computations*, Johns Hopkins University Press, 1983.

[4] C. C. Gonzaga, "Polynomial affine algorithms for linear programming," *Mathematical Programming* , to appear.

[5] M. Grötschel, L. Lovász and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization," *Combinatorica* **1** (1981) 169–197; Corrigendum: *Combinatorica* **4** (1984) 291–295.

[6] L. G. Khachiyan, "A polynomial algorithm in linear programming," *Soviet. Math. Dokl.* **20** (1979) 191–194.

[7] L. Lovász, *An algorithmic theory of numbers, graphs and convexity*, SIAM, Philadelphia, 1986.

[8] N. Megiddo, "Combinatorial optimization with rational objective functions," *Mathematics of Operations Research* **4** (1979) 414–424.

[9] J. H. Wilkinson, "Error analysis of floating-point computation," *Numer. Math.* **2** (1960) 219–340.

[10] Y. Ye and M. Kojima, "Recovering optimal dual solutions in Karmarkar's polynomial algorithm for linear programming," *Mathematical Programming* **39** (1987) 305–317.