

Potential Reduction Algorithms *

Kurt M. Anstreicher
Department of Management Sciences
University of Iowa
Iowa City, IA 52242, USA

1 Introduction

Potential reduction algorithms have a distinguished role in the area of interior point methods for mathematical programming. Karmarkar's [44] algorithm for linear programming, whose announcement in 1984 initiated a torrent of research into interior point methods, used three key ingredients: a non-standard linear programming formulation, projective transformations, and a potential function with which to measure the progress of the algorithm. It was quickly shown that the non-standard formulation could be avoided, and eventually algorithms were developed that eliminated the projective transformations, but retained the use of a potential function. It is then fair to say that the only really essential element of Karmarkar's analysis was the potential function. Further modifications to Karmarkar's original potential function gave rise to potential reduction algorithms having the state-of-the-art theoretical complexity of $O(\sqrt{n}L)$ iterations, to solve a standard form linear program with n variables, and integer data with total bit size L . In the classical optimization literature, potential reduction algorithms are most closely related to Huard's [39] "method of centres," see also Fiacco and McCormick [21, Section 7.2]. However, Karmarkar's use of a potential function to facilitate a *complexity*, as opposed to *convergence* analysis, was completely novel.

The purpose of this article is to give a comprehensive survey of potential reduction algorithms for linear programming. (In the final section we will also briefly describe the extension of potential reduction algorithms to more general problems.) The major algorithms that are discussed, and analyzed, are Karmarkar's algorithm, the affine potential reduction method, and the primal-dual potential reduction algorithm. The different algorithms are all

*This paper appears as Chapter 4 in *Interior Point Methods of Mathematical Programming*, T. Terlaky, ed., Kluwer, 1996.

described using simple, consistent notation in order to facilitate a comparison between them. Before discussing any algorithms we provide (in the next section) the basic complexity arguments based on the primal, and primal–dual potential functions. In the last section we describe various modifications and extensions of the algorithms.

Todd [78] has already written an excellent survey of potential reduction algorithms. Compared to [78], this is a more introductory article that covers less material. For the reader interested in a more technical discussion of the topics covered here, with a greater emphasis on research issues and new extensions, we highly recommend [78]. For a discussion of path–following methods, the other major class of polynomial–time interior point algorithms, we highly recommend the survey paper of Gonzaga [37].

Notation. We use very standard notation throughout. Subscripts denote components of a vector, and superscripts denote iteration numbers. For a vector $x \in \mathbf{R}^n$, we use X to denote the $n \times n$ diagonal matrix having $X_{ii} = x_i$, $i = 1, \dots, n$; similar notation is used for s and S , v and V , and so on. We use e to denote a vector of variable dimension, with each component equal to one. We use $\|\cdot\|$ to denote the two–norm, $\|x\| = \|x\|_2$, and $\|\cdot\|_\infty$ to denote the infinity–norm.

2 Potential Functions for Linear Programming

Consider a standard form linear program, and its dual:

$$\begin{array}{ll} \text{LP :} & \min \quad c^T x \\ & Ax = b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \text{LD :} & \max \quad b^T y \\ & A^T y + s = c \\ & s \geq 0, \end{array}$$

where A is an $m \times n$ matrix. We assume without loss of generality that the rows of A are linearly independent. We also assume that the set of optimal solutions for LP is nonempty and bounded, and let z^* denote the optimal objective value in LP and LD. The *primal potential function* for LP is then

$$f(x, z) = q \ln(c^T x - z) - \sum_{i=1}^n \ln(x_i),$$

where $x > 0$ is a point in the (relative) interior of LP, $z \leq z^*$ is a lower bound on the optimal objective value, and $q \geq n$. Given an initial interior point x^0 , and lower bound z^0 , a potential reduction method based on $f(\cdot, \cdot)$ obtains a sequence (x^k, z^k) , $k \geq 0$ of interior points and lower bounds such that $f(x^k, z^k) \rightarrow -\infty$. The usual approach to analyzing such an algorithm is to show that on each iteration k it is possible to reduce $f(\cdot, \cdot)$ by some

uniform, positive amount δ . Note that for any $x > 0$,

$$\sum_{i=1}^n \ln(x_i) \leq n \ln\left(\frac{e^T x}{n}\right),$$

by the arithmetic–geometric mean inequality. If we assume that a decrease of at least δ occurs on each iteration, then after k iterations we immediately obtain

$$\ln(c^T x^k - z^k) \leq \frac{f(x^0, z^0)}{q} - \frac{k\delta}{q} + \frac{n}{q} \ln\left(\frac{e^T x^k}{n}\right). \quad (1)$$

Clearly then if the solution sequence $\{x^k\}$ is bounded, the “gap” $c^T x^k - z^k$ will be driven to zero. We will next translate this observation into a precise complexity result for LP.

The usual complexity model for LP (see for example [65]) assumes that the data in LP is integral, and characterizes the performance of an algorithm in terms of the dimensions m and n , and the number of bits L required to encode the problem instance in binary. (The quantity L is commonly referred to as the *size* of LP.) A complete complexity analysis should bound the number of digits required in all computations carried out by the algorithm, but we ignore this issue here and consider only the number of arithmetic operations performed, and not the sizes of the numbers involved. We will use the well-known fact (see [65]) that if $c^T x - z \leq 2^{-2L}$ for a feasible solution x and lower bound z , then x can be “rounded” to an exact optimal solution of LP in $O(m^2 n)$ operations. It is also well known that if LP has an optimal solution value z^* , then $-2^{O(L)} \leq z^* \leq 2^{O(L)}$.

To start, we assume that we are given an initial interior solution x^0 , and lower bound z^0 , such that $f(x^0, z^0) \leq O(qL)$. Later we will discuss the “initialization” problem of finding such a pair (x^0, z^0) .

Theorem 2.1 *Assume that the set of optimal solutions of LP is nonempty and bounded. Suppose that $f(x^0, z^0) \leq O(qL)$, and $f(\cdot, \cdot)$ is reduced by δ on each iteration. Then after $k = O(qL/\delta)$ iterations, $c^T x^k - z^k \leq 2^{-2L}$.*

Proof: We will show that $\ln(e^T x^k/n) = O(L)$ for all $k \geq O(qL/\delta)$, and therefore the theorem immediately follows from (1). For each iteration k define scalars

$$\lambda_1^k = \frac{n}{e^T x^k}, \quad \lambda_2^k = \left(\frac{n}{e^T x^k}\right)^{1-\frac{n}{q}} \exp\left[\frac{f(x^0, z^0)}{q} - \frac{k\delta}{q}\right],$$

and let $\xi^k = n x^k / e^T x^k = \lambda_1^k x^k$, so that $e^T \xi^k = n$. Exponentiating (1) then results in

$$c^T \xi^k \leq \lambda_1^k z^k + \lambda_2^k.$$

It follows that for every $k \geq 0$, $(\xi^k, \lambda_1^k, \lambda_2^k)$ is a feasible solution for the linear programming problem:

$$\begin{aligned} \min \quad & \lambda_1 + \lambda_2 \\ & A\xi - \lambda_1 b = 0 \\ & e^T \xi = n \\ & c^T \xi - \lambda_1 z_{\max} - \lambda_2 \leq 0 \\ & \xi \geq 0, \lambda_1 \geq 0, \lambda_2 \geq 0, \end{aligned} \tag{2}$$

where $z_{\max} = 2^{O(L)}$ is an upper bound for z^* . Since the set of optimal solutions of LP is nonempty and bounded, the optimal objective value in (2) is strictly positive. Moreover the size of (2) is $O(L)$, and therefore the optimal objective value is at least $2^{-O(L)}$ (see [65]). However, after $k = O(qL/\delta)$ iterations we must have either $e^T x^k \leq n$, or $\lambda_2^k < 2^{-O(L)}$. It follows that for all $k \geq O(qL/\delta)$, $\lambda_1^k \geq 2^{-O(L)}$, and therefore $\ln(e^T x^k/n) \leq O(L)$, as claimed. \square

To provide a complete complexity result for LP we still need to deal with the issue of satisfying the assumptions of Theorem 2.1. This is quite simple, at least from a theoretical standpoint. For an arbitrary problem LP, with no assumptions whatsoever, consider the augmented problem:

$$\begin{aligned} \text{MLP :} \quad \min \quad & \hat{c}^T \hat{x} \\ & \hat{A} \hat{x} = b \\ & e^T \hat{x} \leq M \\ & \hat{x} \geq 0, \end{aligned}$$

where $\hat{x} \in \mathbf{R}^{n+1}$, and

$$\hat{A} = (A, b - Ae), \quad \hat{c} = \begin{pmatrix} c \\ M \end{pmatrix}.$$

It is then very well known (see for example [65]) that MLP is equivalent to LP for $M = 2^{O(L)}$, in that x^* with $e^T x^* < M$ is an optimal solution for LP if and only if $\hat{x}_i = x_i^*$, $i = 1, \dots, n$, $\hat{x}_{n+1} = 0$ is an optimal solution for MLP. (If the optimal solution to MLP has $\hat{x}_{n+1} > 0$ then LP is infeasible. If the optimal solution to MLP has $e^T \hat{x} = M$ then either LP is unbounded, or LP has an unbounded set of optimal solutions, and these cases can be distinguished by doubling M and solving MLP again.) The primal potential function can then be defined for MLP instead of LP, and it is easy to verify that for $z^0 = -2^{O(L)}$, $\hat{x}^0 = e$, the assumptions of Theorem 2.1 are satisfied.

In addition to potential reduction methods based on $f(\cdot, \cdot)$, we will consider algorithms that utilize the *primal-dual potential function* for LP:

$$F(x, s) = q \ln(x^T s) - \sum_{i=1}^n \ln(x_i) - \sum_{i=1}^n \ln(s_i),$$

where $q > n$, $x > 0$ is feasible for LP, and $s > 0$ is feasible for LD. (By the latter we mean that there is a $y \in \mathbf{R}^m$ so that $A^T y + s = c$.) Note that for any such x and s ,

$$\begin{aligned} F(x, s) &= (q - n) \ln(x^T s) - \sum_{i=1}^n \ln\left(\frac{x_i s_i}{x^T s}\right) \\ &\geq (q - n) \ln(x^T s) + n \ln(n), \end{aligned} \tag{3}$$

by the arithmetic–geometric mean inequality, where we are using the fact that $e^T(XSe/x^T s) = 1$. A potential reduction algorithm based on $F(\cdot, \cdot)$ generates a sequence of primal and dual solutions (x^k, s^k) so that $F(x^k, s^k) \rightarrow -\infty$. We will now give a complexity result for such an algorithm under the assumption that $F(\cdot, \cdot)$ is reduced by some uniform amount δ on each iteration k . The proof of this result is extremely simple, due to the form of $F(\cdot, \cdot)$.

Theorem 2.2 *Suppose that $F(x^0, s^0) \leq O((q - n)L)$, and $F(\cdot, \cdot)$ is reduced by δ on each iteration. Then after $k = O((q - n)L/\delta)$ iterations, $(x^k)^T s^k \leq 2^{-2L}$.*

Proof: Using (3) we obtain

$$\ln((x^k)^T s^k) \leq \frac{F(x^k, s^k)}{q - n} \leq \frac{F(x^0, s^0) - k\delta}{q - n},$$

from which the theorem easily follows. \square

Note that the existence of an interior point for the dual problem LD implies that the set of optimal solutions for LP is bounded, so the boundedness assumption that was explicit in Theorem 2.1 is implicit in Theorem 2.2. To provide a complete complexity result for LP based on the reduction of $F(\cdot, \cdot)$ we must deal with the initialization problem of finding (x^0, s^0) that satisfy the assumptions of Theorem 2.2. This can be done using an augmented problem that is very similar to the problem MLP described above, but the analysis is somewhat more complex than for the primal case, and is omitted here. We refer the interested reader to [13, Section 5] for the details of such an initialization.

Remarks. The primal potential function was introduced by Karmarkar [44]. The exponentiated, or “multiplicative” form of the potential function was used by Iri and Imai [41], and was further studied by Imai [40]. The use of general values for q was suggested by Gonzaga [33]. The primal–dual potential function was introduced by Todd and Ye [80], and (in multiplicative form) Tanabe [70]. See Ye, Todd, and Mizuno [91] and Jansen, Roos and Terlaky [42] for alternative “homogeneous self–dual” approaches to the initialization problem.

3 Karmarkar's Algorithm

In this section we describe Karmarkar's projective algorithm for LP. The original algorithm, as presented in [44], was based on a linear program in a non-standard special form. The "standard form" version we describe here was independently devised in [3], [20], [28], [32], [67], and [89]. Let x^k , $k \geq 0$, be a feasible interior point for LP, and $z^k \leq z^*$ a valid lower bound. Our goal is to generate a new interior point x^{k+1} , and lower bound z^{k+1} , so that the primal potential function $f(\cdot, \cdot)$, with $q = n + 1$, is decreased by an amount $\delta = \Omega(1)$. From Theorem 2.1, such a decrease immediately provides an $O(nL)$ iteration algorithm for LP.

Consider a new linear programming problem, with variables $\bar{x} \in \mathbf{R}^{n+1}$:

$$\begin{aligned} \overline{\text{HLP}}: \quad \min \quad & \bar{c}^T \bar{x} \\ & \bar{A} \bar{x} = 0 \\ & d^T \bar{x} = 1 \\ & \bar{x} \geq 0, \end{aligned}$$

where

$$\bar{A} = (AX^k, -b), \quad \bar{c} = \begin{pmatrix} X^k c \\ 0 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

One can think of obtaining $\overline{\text{HLP}}$ from LP by applying a transformation of variables:

$$\bar{x} = \begin{pmatrix} (X^k)^{-1}x \\ 1 \end{pmatrix},$$

and then using the additional variable \bar{x}_{n+1} to "homogenize" the original equality constraints of LP. Clearly $\overline{\text{HLP}}$ is equivalent to LP, and $\bar{x} = e$ is feasible in $\overline{\text{LP}}$. The derivation of a step in LP is based on the transformed problem $\overline{\text{LP}}$. First we consider the issue of updating the lower bound. For any matrix B , let P_B denote the orthogonal projection onto the nullspace of B . In the case that B has independent rows, we then have $P_B = I - B^T(BB^T)^{-1}B$.

Lemma 3.1 (Todd and Burrell [79]) *Suppose that $z \in \mathbf{R}$ satisfies $P_{\bar{A}}(\bar{c} - zd) \geq 0$. Then $z \leq z^*$.*

Proof: The dual of $\overline{\text{HLP}}$ is:

$$\begin{aligned} \overline{\text{HLD}}: \quad \max \quad & z \\ & \bar{A}^T y + dz \leq \bar{c}. \end{aligned}$$

But $P_{\bar{A}}(\bar{c} - zd) = (\bar{c} - zd) - \bar{A}^T y(z)$ for some $y(z) \in \mathbf{R}^m$, so $P_{\bar{A}}(\bar{c} - zd) \geq 0$ implies that $(y(z), z)$ is feasible in $\overline{\text{HLD}}$. Then $z \leq z^*$, since LP and $\overline{\text{HLP}}$ have the same optimal objective value. \square

Using Lemma 3.1 the lower bound z^k can be updated as follows. Let $\mathcal{Z}^k = \{z \geq z^k \mid P_{\bar{A}}(\bar{c} - zd) \geq 0\}$, and define z^{k+1} to be:

$$z^{k+1} = \begin{cases} \max\{z \in \mathcal{Z}^k\} & \text{if } \mathcal{Z}^k \neq \emptyset, \\ z^k & \text{otherwise.} \end{cases}$$

Then $\bar{z} = z^{k+1} \leq z^*$, by Lemma 3.1, and moreover by construction we have $P_{\bar{A}}(\bar{c} - \bar{z}d) \not\geq 0$. Now let

$$\Delta\bar{x} = P_{\left[\begin{smallmatrix} \bar{A} \\ e^T \end{smallmatrix}\right]}(\bar{c} - \bar{z}d) = P_{e^T} P_{\bar{A}}(\bar{c} - \bar{z}d) = P_{\bar{A}}(\bar{c} - \bar{z}d) - \frac{(\bar{c} - \bar{z}d)^T e}{n+1} e,$$

where we are using the fact that $\bar{A}e = 0$. Since $P_{\bar{A}}(\bar{c} - \bar{z}d) \not\geq 0$, we then immediately have

$$\|\Delta\bar{x}\| \geq \|\Delta\bar{x}\|_\infty \geq \frac{(\bar{c} - \bar{z}d)^T e}{n+1}. \quad (4)$$

The next point, in the transformed variables \bar{x} , will be of the form

$$\bar{x}' = e - \alpha \frac{\Delta\bar{x}}{\|\Delta\bar{x}\|}, \quad (5)$$

where $\alpha > 0$ is a steplength yet to be decided. Note that the resulting \bar{x}' will satisfy the equality constraints $\bar{A}\bar{x} = 0$ of $\overline{\text{HLP}}$, but in general will fail to satisfy $d^T \bar{x}' = 1$. In order to obtain a new point x^{k+1} which is feasible for LP, we employ a *projective* transformation

$$x^{k+1} = \frac{X^k \bar{x}'}{d^T \bar{x}'}. \quad (6)$$

Substituting (6) into the definition of $f(\cdot, \cdot)$, with $q = n+1$, for α sufficiently small we obtain

$$\begin{aligned} & f(x^{k+1}, z^{k+1}) - f(x^k, z^{k+1}) \\ &= (n+1) \ln \left(\frac{c^T x^{k+1} - z^{k+1}}{c^T x^k - z^{k+1}} \right) - \sum_{i=1}^n \ln \left(\frac{x_i^{k+1}}{x_i^k} \right) \\ &= (n+1) \ln \left(\frac{(\bar{c} - \bar{z}d)^T \bar{x}' / \bar{x}'_{n+1}}{(\bar{c} - \bar{z}d)^T e} \right) - \sum_{i=1}^n \ln \left(\frac{\bar{x}'_i}{\bar{x}'_{n+1}} \right) \\ &= (n+1) \ln \left(\frac{(\bar{c} - \bar{z}d)^T (e - \alpha \Delta\bar{x} / \|\Delta\bar{x}\|)}{(\bar{c} - \bar{z}d)^T e} \right) - \sum_{i=1}^{n+1} \ln(\bar{x}'_i) \\ &= (n+1) \ln \left(1 - \frac{\alpha \|\Delta\bar{x}\|}{(\bar{c} - \bar{z}d)^T e} \right) - \sum_{i=1}^{n+1} \ln \left(1 - \alpha \frac{\Delta\bar{x}_i}{\|\Delta\bar{x}\|} \right) \\ &\leq -\alpha - \sum_{i=1}^{n+1} \ln \left(1 - \alpha \frac{\Delta\bar{x}_i}{\|\Delta\bar{x}\|} \right), \end{aligned} \quad (7)$$

where the inequality uses (4), and the fact that $\ln(1 - t) \leq -t$ for any $0 \leq t < 1$.

To obtain a bound on the potential decrease for Karmarkar's algorithm we need to obtain a bound for (7). One approach is to use the following well-known inequality.

Lemma 3.2 *Let $u \in \mathbf{R}^n$, $\|u\|_\infty \leq 1$. Then*

$$\sum_{i=1}^n \ln(1 + u_i) \geq e^T u - \frac{\|u\|^2}{2(1 - \|u\|_\infty)} .$$

Proof: For each $i = 1, \dots, n$ the Taylor series expansion for $\ln(1 + u_i)$ results in

$$\ln(1 + u_i) = \sum_{j=1}^{\infty} \frac{(-1)^{j+1} u_i^j}{j} \geq u_i - \frac{1}{2} \sum_{j=2}^{\infty} |u_i|^j = u_i - \frac{u_i^2}{2(1 - |u_i|)} . \quad (8)$$

The proof is completed by summing (8), and using $|u_i| \leq \|u\|_\infty$ for each i . \square

Theorem 3.3 *On every iteration $k \geq 0$ of Karmarkar's algorithm, the steplength α may be chosen so that $f(x^k, z^k) - f(x^{k+1}, z^{k+1}) \geq .25$.*

Proof: We have

$$\begin{aligned} f(x^k, z^k) - f(x^{k+1}, z^{k+1}) &\geq f(x^k, z^{k+1}) - f(x^{k+1}, z^{k+1}) \\ &\geq \alpha + \sum_{i=1}^{n+1} \ln \left(1 - \alpha \frac{\Delta \bar{x}_i}{\|\Delta \bar{x}\|} \right) \\ &\geq \alpha - \frac{\alpha^2}{2(1 - \alpha)}, \end{aligned} \quad (9)$$

where the first inequality uses $z^{k+1} \geq z^k$, the second uses (7), and the third uses Lemma 3.2 and the fact that $e^T \Delta \bar{x} = 0$. The proof is completed by substituting $\alpha = .5$ into (9). \square

An important feature of Karmarkar's algorithm is that in practice, an approximate linesearch in the steplength α can be performed to maximize the potential decrease on each step. Such a linesearch typically obtains steplengths, and potential decreases, that are much larger than the $\Omega(1)$ values that appear in the worst-case analysis above.

Remarks. There are many papers that consider different aspects of Karmarkar's algorithm. One line of investigation concerns the potential decrease assured in Theorem 3.3. The decrease of .25 proved here can easily be improved to $1 - \ln(2) \approx .31$ by sharper approximation of the logarithmic barrier

terms. Muramatsu and Tsuchiya [59] show that using a “fixed fraction to the boundary” step, based on the “affine” direction $P_{\bar{A}}(\bar{c} - \bar{z}d)$, a decrease of about .41 is always possible. Anstreicher [4] and McDiarmid [48] independently proved that with exact linesearch of the potential function a decrease of approximately .7215 is always possible, and this bound is tight. Another interesting topic is the derivation of a *lower bound* for the worst-case complexity of the algorithm. Anstreicher [8] shows that using exact linesearch of the potential function, the algorithm may produce an $O(1)$ reduction in $f(\cdot, \cdot)$ on every iteration, and may require $\Omega(\ln(n/\epsilon))$ iterations to reduce the gap $c^T x^k - z^k$ to a factor $\epsilon < 1$ of its initial value. Ji and Ye [43] elaborate further the analysis of [8]. Powell [66] shows that the iterates of Karmarkar’s algorithm, with exact linesearch, may visit the neighborhoods of $\Omega(n)$ extreme points of the feasible region.

Anstreicher [3] and Steger [67] describe a “ball update” alternative to Todd and Burrell’s [79] lower bound methodology. Shaw and Goldfarb [69] show that with a weakened version of the ball update, and short steps ($\alpha < 1$), the projective algorithm can be viewed as a path following method and has a complexity of $O(\sqrt{n}L)$ iterations. Anstreicher [3] describes a modification of the algorithm that assures monotonicity of the objective values $\{c^T x^k\}$. Anstreicher [11] describes a stronger monotonicity modification, and obtains a complexity of $O(\sqrt{n}L)$ iterations using the weakened ball updates, and steplengths based on the primal-dual potential function $F(\cdot, \cdot)$. Goldfarb and Mehrotra [30],[31] modify the projective algorithm to allow for the use of inexact computation of the search direction $\Delta\bar{x}$. Todd [71] considers the computation of lower bounds, and the search direction, for problems with special structure. Todd [72] and Ye [84] describe the construction of “dual ellipsoids” that contain all dual optimal solutions. In principle this procedure could be used to eliminate variables as the algorithm iterates, but Anstreicher [7] describes why the process fails in the presence of degeneracy. Todd [74] and Anstreicher and Watteyne [14] describe alternatives to the usual search direction obtained via decomposition, and projection onto a simplex, respectively. Computational results for Karmarkar’s algorithm are reported in [14], and by Todd [73].

Asic et al. [2] consider the the asymptotic behavior of the iterates in Karmarkar’s algorithm using short step ($\alpha < 1$), while Megiddo and Shub [49] and Monteiro [53] examine properties of the continuous trajectories associated with the algorithm. Bayer and Lagarias [15] explore connections between Karmarkar’s algorithm and Newton’s method, Gill et al. [29] describe relationships between Karmarkar’s algorithm and logarithmic barrier methods, and Mitchell and Todd [51] relate Karmarkar’s method to the primal affine scaling algorithm. Freund [23], Gonzaga [35], and Mitchell and Todd [52] consider the projective algorithm for more general problem for-

mulations than that of LP. See also Freund [26] for a very general discussion of the use of projective transformations.

4 The Affine Potential Reduction Algorithm

Although Karmarkar’s algorithm caused a revolution in mathematical programming, there are some aspects of the method that are less than ideal. For example projective transformations have rarely been employed in the optimization literature, and the use of the projective transformation (6) is not particularly intuitive. In addition, the $O(nL)$ iteration complexity bound for the algorithm was eventually bettered by “path-following” methods for linear programming (see for example [37]), which achieve a complexity of $O(\sqrt{n}L)$ iterations.

It turns out that both of the above issues can be addressed by a method that is quite similar to Karmarkar’s algorithm, but which avoids the use of a projective transformation on each step. Given a feasible interior point x^k , $k \geq 0$, consider a transformed problem:

$$\begin{aligned} \overline{\text{LP}}: \quad \min \quad & \bar{c}^T x \\ & \bar{A}\bar{x} = b \\ & \bar{x} \geq 0, \end{aligned}$$

where now $\bar{A} = AX^k$ and $\bar{c} = X^k c$. Let $\overline{\text{LD}}$ denote the dual of $\overline{\text{LP}}$. One can think of obtaining $\overline{\text{LP}}$ from LP by applying a simple re-scaling of the variables of the form

$$\bar{x} = (X^k)^{-1}x. \tag{10}$$

Clearly $\overline{\text{LP}}$ is equivalent to LP, and $\bar{x} = e$ is feasible in $\overline{\text{LP}}$. As in Karmarkar’s algorithm, the derivation of a step in LP is based on the transformed problem $\overline{\text{LP}}$. Define a transformed potential function

$$\bar{f}(\bar{x}, z) = q \ln(\bar{c}^T \bar{x} - z) - \sum_{i=1}^n \ln(\bar{x}_i).$$

Note that if x and \bar{x} are related by (10), then $f(x, z)$ and $\bar{f}(\bar{x}, z)$ differ by a constant which depends only on x^k . As a result, it suffices to analyze the decrease in $\bar{f}(\cdot, \cdot)$ starting at $\bar{x} = e$, $z = z^k$. To this end, let $\Delta\bar{x}$ be the projection of the gradient of $\bar{f}(e, z^k)$ onto the nullspace of \bar{A} :

$$\Delta\bar{x} = P_{\bar{A}}[\nabla_{\bar{x}}\bar{f}(e, z^k)]^T = P_{\bar{A}}\left(\frac{q}{\bar{c}^T e - z^k}\bar{c} - e\right). \tag{11}$$

Re-arranging (11), it follows that there is a $y' \in \mathbf{R}^m$ so that

$$\bar{c} = \bar{A}^T y' + \frac{\bar{c}^T e - z^k}{q} (e + \Delta \bar{x}). \quad (12)$$

Lemma 4.1 *Let $q = n + \sqrt{n}$, and suppose that $\|\Delta \bar{x}\| \leq \eta < 1$. Then $z^{k+1} = b^T y'$ satisfies $z^k < z^{k+1} \leq z^*$, and $f(x^k, z^k) - f(x^k, z^{k+1}) \geq (1 - \eta)\sqrt{n}$.*

Proof: Clearly $e + \Delta \bar{x} > 0$, so (12) implies that y' is feasible for the dual of $\overline{\text{LP}}$, and therefore $b^T y' \leq z^*$. In addition,

$$\bar{c}^T e - b^T y' = \left(\frac{\bar{c}^T e - z^k}{q} \right) e^T (e + \Delta \bar{x}) \leq \frac{n + \eta\sqrt{n}}{q} (\bar{c}^T e - z^k), \quad (13)$$

implying $z^{k+1} > z^k$. Finally,

$$\begin{aligned} f(x^k, z^{k+1}) - f(x^k, z^k) &= \bar{f}(e, z^{k+1}) - \bar{f}(e, z^k) \\ &= q \ln \left(\frac{\bar{c}^T e - z^{k+1}}{\bar{c}^T e - z^k} \right) \leq q \ln \left(\frac{n + \eta\sqrt{n}}{q} \right) \\ &= q \ln \left(1 - \frac{(1 - \eta)\sqrt{n}}{q} \right) \leq -(1 - \eta)\sqrt{n}, \end{aligned}$$

where the last inequality uses $\ln(1 - t) \leq -t$ for $t < 1$. \square

Let $0 < \eta < 1$ be a fixed constant, independent of n . By Lemma 4.1, if $\|\Delta \bar{x}\| < \eta$, then the lower bound can be updated to a new value z^{k+1} , such that the potential function $f(\cdot, \cdot)$ is reduced by $\Omega(\sqrt{n})$. Consider next the situation when $\|\Delta \bar{x}\| \geq \eta$. In this case we take a step in the transformed problem of the form:

$$\bar{x}' = e - \alpha \frac{\Delta \bar{x}}{\|\Delta \bar{x}\|}, \quad (14)$$

where $\alpha > 0$ is a steplength yet to be decided. Following such a step, a new point x^{k+1} is defined by $x^{k+1} = X^k \bar{x}'$.

Lemma 4.2 *Let $q = n + \sqrt{n}$, and suppose that $\|\Delta \bar{x}\| \geq \eta > 0$. Then there is a steplength α so that $f(x^k, z^k) - f(x^{k+1}, z^k) \geq (1 + \eta) - \sqrt{1 + 2\eta} > 0$.*

Proof: We have

$$\begin{aligned} & f(x^{k+1}, z^k) - f(x^k, z^k) \\ &= \bar{f}(\bar{x}', z^k) - \bar{f}(e, z^k) \\ &= q \ln \left(\frac{\bar{c}^T (e - \alpha \Delta \bar{x} / \|\Delta \bar{x}\|) - z^k}{\bar{c}^T e - z^k} \right) - \sum_{i=1}^n \ln \left(1 - \alpha \frac{\Delta \bar{x}_i}{\|\Delta \bar{x}\|} \right) \end{aligned}$$

$$\begin{aligned}
&\leq q \ln \left(1 - \frac{\alpha \bar{c}^T \Delta \bar{x} / \|\Delta \bar{x}\|}{\bar{c}^T e - z^k} \right) + \alpha \frac{e^T \Delta \bar{x}}{\|\Delta \bar{x}\|} + \frac{\alpha^2}{2(1-\alpha)} \\
&\leq -\alpha \left(\frac{q}{\bar{c}^T e - z^k} \bar{c} - e \right)^T \frac{\Delta \bar{x}}{\|\Delta \bar{x}\|} + \frac{\alpha^2}{2(1-\alpha)} \\
&= -\alpha \|\Delta \bar{x}\| + \frac{\alpha^2}{2(1-\alpha)} \\
&\leq -\alpha \eta + \frac{\alpha^2}{2(1-\alpha)}, \tag{15}
\end{aligned}$$

where the first inequality uses Lemma 3.2, and the second uses $\ln(1-t) \leq -t$ for $t < 1$. A straightforward calculus exercise shows that (15) is minimized at $\alpha = 1 - 1/\sqrt{1+2\eta}$, and substitution of this value into (15) completes the proof. \square

Taken together, Lemmas 4.1 and 4.2 immediately imply that for $q = n + \sqrt{n}$, an $\Omega(1)$ decrease in $f(\cdot, \cdot)$ is always possible. As a result, the affine potential reduction algorithm is an $O(nL)$ iteration method for LP. However, there is a striking asymmetry between Lemmas 4.1 and 4.2, since the former shows that in fact an $\Omega(\sqrt{n})$ decrease occurs on steps where the lower bound is updated. In fact the affine potential reduction method, exactly as described above, can be shown to be an $O(\sqrt{n}L)$ iteration algorithm by analyzing the algorithm using the symmetric primal-dual potential function $F(\cdot, \cdot)$, instead of the primal potential function $f(\cdot, \cdot)$.

Suppose that $x^k > 0$ and $s^k > 0$ are feasible for LP and LD, respectively. Consider a linear transformation of the dual variables

$$\bar{s} = X^k s. \tag{16}$$

Then for any $x > 0$ and $s > 0$, feasible in LP and LD, respectively, \bar{x} and \bar{s} from (10) and (16) are feasible in $\overline{\text{LP}}$ and $\overline{\text{LD}}$, respectively, and moreover $F(x, s) = F(\bar{x}, \bar{s})$. As a result, it suffices to analyze the descent in $F(\cdot, \cdot)$ starting at $\bar{x} = e$, $\bar{s} = \bar{s}^k = X^k s^k$. Let $\Delta \bar{x}$ be as in (11), for $z^k = b^T y^k$, where $A^T y^k + s^k = c$. If $\|\Delta \bar{x}\| \geq \eta$, we continue to take a step as in (14), and let $x^{k+1} = X^k \bar{x}'$.

Lemma 4.3 *Let $q = n + \sqrt{n}$, and let $\Delta \bar{x}$ be as in (11), with $z^k = b^T y^k$. Suppose that $\|\Delta \bar{x}\| \geq \eta > 0$. Then there is a steplength α so that $F(x^k, s^k) - F(x^{k+1}, s^k) \geq (1 + \eta) - \sqrt{1 + 2\eta} > 0$.*

Proof: The proof is identical to that of Lemma 4.2, using the fact that for any \bar{x} , $\bar{x}^T \bar{s}^k = \bar{c}^T \bar{x} - z^k$. \square

Next we turn to the case of $\|\Delta \bar{x}\| \leq \eta$. As before, we will use the fact that (12) provides a feasible solution for $\overline{\text{LD}}$. Define

$$\bar{s}' = \frac{\bar{c}^T e - z^k}{q} (e + \Delta \bar{x}) = \frac{e^T \bar{s}^k}{q} (e + \Delta \bar{x}). \tag{17}$$

We now require an analysis of the step from \bar{s}^k to \bar{s}' that includes the effect of the dual barrier terms in $F(\cdot, \cdot)$.

Theorem 4.4 *Suppose that $\|\Delta\bar{x}\| \leq \eta$. Let \bar{s}' be as in (17), and let $s^{k+1} = (X^k)^{-1}\bar{s}'$. Then $F(x^k, s^k) - F(x^k, s^{k+1}) \geq (1 - 2\eta)/(2 - 2\eta)$.*

Proof: We have

$$\begin{aligned}
& F(x^k, s^{k+1}) - F(x^k, s^k) \\
&= F(e, \bar{s}') - F(e, \bar{s}^k) \\
&= q \ln \left(\frac{e^T \bar{s}'}{e^T \bar{s}^k} \right) - \sum_{i=1}^n \ln(\bar{s}'_i) + \sum_{i=1}^n \ln(\bar{s}^k_i) \\
&= q \ln \left(\frac{n + e^T \Delta\bar{x}}{q} \right) - n \ln \left(\frac{e^T \bar{s}^k}{q} \right) - \sum_{i=1}^n \ln(1 + \Delta\bar{x}_i) + \sum_{i=1}^n \ln(\bar{s}^k_i),
\end{aligned}$$

where the last equality uses (17). Note that

$$\sum_{i=1}^n \ln(\bar{s}^k_i) \leq n \ln(e^T \bar{s}^k / n), \tag{18}$$

by the arithmetic–geometric mean inequality. Moreover, Lemma 3.2 implies that

$$-\sum_{i=1}^n \ln(1 + \Delta\bar{x}_i) \leq -e^T \Delta\bar{x} + \frac{\eta^2}{2(1 - \eta)}. \tag{19}$$

Using (18) and (19), we obtain $F(x^k, s^{k+1}) - F(x^k, s^k)$

$$\begin{aligned}
& \leq q \ln \left(\frac{n + e^T \Delta\bar{x}}{q} \right) - n \ln \left(\frac{e^T \bar{s}^k}{q} \right) - e^T \Delta\bar{x} + \frac{\eta^2}{2(1 - \eta)} + n \ln \left(\frac{e^T \bar{s}^k}{n} \right) \\
&= (q - n) \ln \left(\frac{n + e^T \Delta\bar{x}}{q} \right) + n \ln \left(\frac{n + e^T \Delta\bar{x}}{n} \right) - e^T \Delta\bar{x} + \frac{\eta^2}{2(1 - \eta)} \\
&= \sqrt{n} \ln \left(1 - \frac{\sqrt{n} - e^T \Delta\bar{x}}{q} \right) + n \ln \left(1 + \frac{e^T \Delta\bar{x}}{n} \right) - e^T \Delta\bar{x} + \frac{\eta^2}{2(1 - \eta)} \\
& \leq \frac{-n(1 - \eta)}{q} + \frac{\eta^2}{2(1 - \eta)},
\end{aligned}$$

where the last inequality uses $\ln(1 + t) \leq t$ for $t > -1$ (twice), and the fact that $\|\Delta\bar{x}\| \leq \eta$. The proof is completed by noting that $q = n + \sqrt{n} \leq 2n$ for $n \geq 1$. \square

Lemma 4.3, and Theorems 4.4 and 2.2, imply that the affine potential reduction algorithm, using $q = n + \sqrt{n}$ and $\eta < .5$, is an $O(\sqrt{n}L)$ algorithm

for LP. As with Karmarkar’s algorithm, in practice a linesearch in α can also be used to improve the decrease in $F(\cdot, \cdot)$ on primal steps.

Remarks. The affine potential reduction method based on $f(\cdot, \cdot)$ was proposed by Gonzaga [33], who assumed that $z^0 = z^*$. The lower bound logic based on (12) was suggested in [33], and fully developed by Freund [24]. Independently, Ye [85] devised the analysis based on $F(\cdot, \cdot)$, which reduces the complexity of the algorithm to $O(\sqrt{n}L)$ iterations. Ye [83] also describes an alternative $O(\sqrt{n}L)$ iteration algorithm that uses $F(\cdot, \cdot)$, but employs projective transformations as in Karmarkar’s algorithm.

The lower bound, or dual variable, update based on (12) can be modified in several different ways. For example, in [24] the lower bound is increased to a value z^{k+1} so that following the bound update it is always the case that $\|\Delta\bar{x}\| \geq \eta$. As a result, updates of the lower bound (or dual solution) are immediately followed by primal steps. Gonzaga [36] considers a general procedure for the construction of lower bounds, and Mitchell [50] relates the construction in [36] to earlier results of Todd [72].

Anstreicher [10] describes a monotonicity modification for the affine potential reduction algorithm, and Ye [86] analyzes a variant that allows for column generation. Monteiro [58] considers the behavior of the continuous trajectories associated with the algorithm. Todd [77] describes analogs of potential reduction methods for semi-infinite linear programming. Anstreicher [12] devises an algorithm which is similar to the affine potential reduction for LD, but which employs a *volumetric* potential function

$$q \ln(z - b^T y) - \frac{1}{2} \ln(\det(AS^{-2}A^T)),$$

where $s = c - A^T y > 0$, $q = O(m)$, and $z > z^*$. The resulting algorithm has a complexity of $O(m\sqrt{n}L)$ iterations. Using a potential function that combines the volumetric barrier with the usual logarithmic barrier, the algorithm’s complexity is reduced to $O(\sqrt{mn}L)$ iterations.

5 The Primal–Dual Algorithm

In the analysis of the previous section, the use of the primal–dual potential function $F(\cdot, \cdot)$, with $q = n + \sqrt{n}$, results in a comparable potential reduction on primal and dual steps, and improves the complexity of the affine potential reduction algorithm to $O(\sqrt{n}L)$ iterations. The algorithm’s treatment of primal versus dual variables is still very asymmetric, however. In this section we describe a different potential reduction method based on $F(\cdot, \cdot)$ which treats the primal and dual variables in a completely symmetric fashion. This “primal–dual” algorithm is due to Kojima, Mizuno, and Yoshise [47]. Our

derivation here differs somewhat from that in [47], as we wish to emphasize the connection with the primal algorithm of the previous section.

Let $q = n + \sqrt{n}$, and let x^k and s^k be feasible interior solutions of LP and LD, respectively. Consider a change of variables

$$\begin{aligned}\bar{x} &= (X^k)^{-1/2}(S^k)^{1/2}x \\ \bar{s} &= (X^k)^{1/2}(S^k)^{-1/2}s.\end{aligned}\tag{20}$$

Then for any x feasible in LP, \bar{x} from (20) is feasible for a rescaled problem $\overline{\text{LP}}$ defined as in the previous section, but using the primal–dual scaling matrix $(X^k)^{1/2}(S^k)^{-1/2}$ in place of X^k . Similarly if s is feasible for LD, then \bar{s} is feasible in $\overline{\text{LD}}$, the dual of $\overline{\text{LP}}$. Moreover, $F(x, s) = F(\bar{x}, \bar{s})$. Note that the transformation (20) maps both x^k and s^k to the vector $v = (X^k)^{1/2}(S^k)^{1/2}e$. As a result, it suffices to consider the reduction in $F(\cdot, \cdot)$ starting at $\bar{x} = \bar{s} = v$. Note that

$$[\nabla_x F(v, v)]^T = [\nabla_s F(v, v)]^T = \frac{q}{\|v\|^2}v - V^{-1}e.$$

We define directions

$$\Delta\bar{x} = P_{\bar{A}}\left(\frac{q}{\|v\|^2}v - V^{-1}e\right), \quad \Delta\bar{s} = (I - P_{\bar{A}})\left(\frac{q}{\|v\|^2}v - V^{-1}e\right),\tag{21}$$

where $\bar{A} = A(X^k)^{1/2}(S^k)^{-1/2}$. Consider simultaneous primal and dual steps of the form:

$$\begin{aligned}\bar{x}' &= v - \frac{\alpha}{\gamma}\Delta\bar{x} = V(e - \frac{\alpha}{\gamma}V^{-1}\Delta\bar{x}), \\ \bar{s}' &= v - \frac{\alpha}{\gamma}\Delta\bar{s} = V(e - \frac{\alpha}{\gamma}V^{-1}\Delta\bar{s}),\end{aligned}\tag{22}$$

where $\gamma = \sqrt{\|V^{-1}\Delta\bar{x}\|^2 + \|V^{-1}\Delta\bar{s}\|^2}$, and $\alpha > 0$ is a steplength yet to be decided. We then have $F(\bar{x}', \bar{s}') - F(v, v)$

$$\begin{aligned}&= q \ln\left(\frac{(v - \alpha\Delta\bar{x}/\gamma)^T(v - \alpha\Delta\bar{s}/\gamma)}{\|v\|^2}\right) - \sum_{i=1}^n \ln\left(\frac{\bar{x}'_i}{v_i}\right) - \sum_{i=1}^n \ln\left(\frac{\bar{s}'_i}{v_i}\right) \\ &= q \ln\left(1 - \frac{\alpha v^T(\Delta\bar{x} + \Delta\bar{s})}{\gamma\|v\|^2}\right) - \sum_{i=1}^n \ln\left(1 - \frac{\alpha\Delta\bar{x}_i}{\gamma v_i}\right) - \sum_{i=1}^n \ln\left(1 - \frac{\alpha\Delta\bar{s}_i}{\gamma v_i}\right),\end{aligned}$$

where we are using the fact that $\Delta\bar{x}^T\Delta\bar{s} = 0$. Applying Lemma 3.2, and the fact that $\ln(1 - t) \leq -t$ for $t < 1$, for α sufficiently small we obtain

$$\begin{aligned}F(\bar{x}', \bar{s}') - F(v, v) &\leq \frac{-\alpha q v^T(\Delta\bar{x} + \Delta\bar{s})}{\gamma\|v\|^2} + \frac{\alpha e^T V^{-1}(\Delta\bar{x} + \Delta\bar{s})}{\gamma} + \frac{\alpha^2}{2(1 - \alpha)} \\ &= \frac{-\alpha}{\gamma} \left\| \frac{q}{\|v\|^2}v - V^{-1}e \right\|^2 + \frac{\alpha^2}{2(1 - \alpha)},\end{aligned}\tag{23}$$

where the equality uses the fact that $\Delta\bar{x} + \Delta\bar{s} = (q/\|v\|^2)v - V^{-1}e$. Now let $v_{\min} = \min_i\{v_i\}$. Then

$$\begin{aligned}
\gamma^2 &= \|V^{-1}\Delta\bar{x}\|^2 + \|V^{-1}\Delta\bar{s}\|^2 \\
&\leq \frac{1}{v_{\min}^2}(\|\Delta\bar{x}\|^2 + \|\Delta\bar{s}\|^2) \\
&= \frac{1}{v_{\min}^2}\|\Delta\bar{x} + \Delta\bar{s}\|^2 \\
&= \frac{1}{v_{\min}^2}\left\|\frac{q}{\|v\|^2}v - V^{-1}e\right\|^2. \tag{24}
\end{aligned}$$

Using (24) in (23), we obtain

$$F(\bar{x}', \bar{s}') - F(v, v) \leq -\alpha v_{\min} \left\|\frac{q}{\|v\|^2}v - V^{-1}e\right\| + \frac{\alpha^2}{2(1-\alpha)}. \tag{25}$$

To obtain an estimate for the decrease in $F(\cdot, \cdot)$ for the primal-dual algorithm we require a bound for the linear term in (25). Such a bound is provided by the following lemma.

Lemma 5.1 [47, Lemma 2.5] *Let $v \in \mathbf{R}^n$, $v > 0$, and $q = n + \sqrt{n}$. Then*

$$v_{\min} \left\|\frac{q}{\|v\|^2}v - V^{-1}e\right\| \geq \frac{\sqrt{3}}{2}.$$

Proof: We have

$$\begin{aligned}
\left\|V^{-1}e - \frac{q}{\|v\|^2}v\right\|^2 &= \left\|V^{-1}e - \frac{n}{\|v\|^2}v - \frac{\sqrt{n}}{\|v\|^2}v\right\|^2 \\
&= \left\|V^{-1}e - \frac{n}{\|v\|^2}v\right\|^2 + \left\|\frac{\sqrt{n}}{\|v\|^2}v\right\|^2 \\
&\geq \left(\frac{1}{v_{\min}} - \frac{n}{\|v\|^2}v_{\min}\right)^2 + \frac{n}{\|v\|^2} \\
&= \frac{1}{v_{\min}^2} - \frac{n}{\|v\|^2} + \frac{n^2 v_{\min}^2}{\|v\|^4} \\
&= \frac{3}{4v_{\min}^2} + \left(\frac{1}{2v_{\min}} - \frac{nv_{\min}}{\|v\|^2}\right)^2 \\
&\geq \frac{3}{4v_{\min}^2},
\end{aligned}$$

where the second equality uses the fact that $v^T[V^{-1}e - (n/\|v\|^2)v] = 0$. \square

Theorem 5.2 *Let $q = n + \sqrt{n}$, and consider the primal-dual steps defined as in (22). Let $x^{k+1} = (X^k)^{1/2}(S^k)^{-1/2}\bar{x}'$, $s^{k+1} = (X^k)^{-1/2}(S^k)^{1/2}\bar{s}'$, Then there is a steplength α so that $F(x^k, s^k) - F(x^{k+1}, s^{k+1}) \geq .16$.*

Proof: From (25) and Lemma 5.1 we have

$$F(x^{k+1}, s^{k+1}) - F(x^k, s^k) = F(\bar{x}', \bar{s}') - F(v, v) \leq -\frac{3}{4}\alpha + \frac{\alpha^2}{2(1-\alpha)}. \quad (26)$$

The proof is completed by substituting $\alpha = .37$ into (26). \square

Remarks. Todd and Ye [80], who introduce the primal–dual potential function $F(\cdot, \cdot)$, devise an interesting primal–dual potential reduction algorithm that may be considered to be a precursor to the algorithm of this section. The method of [80] uses projective transformations, like Karmarkar’s algorithm, and attains a complexity of $O(\sqrt{n}L)$ iterations. Unfortunately the iterates are constrained to lie in a neighborhood of the central path, making the algorithm similar to a path following method, and precluding the use of linesearches to increase the descent in $F(\cdot, \cdot)$ on each step. Gonzaga and Todd [38] describe a “primal or dual” potential reduction method based on $F(\cdot, \cdot)$ which achieves symmetry between the primal and dual variables in a fundamentally different way from the algorithm of this section. In [38], the algorithm takes either a primal step as in (14), or a dual step which is based on a projected gradient step in the transformed dual variables \bar{s} , after a scaling of the form $\bar{s} = (S^k)^{-1}s$ which maps s^k to e . It is shown that for $q = n + \sqrt{n}$, one of these two steps must produce an $\Omega(1)$ decrease in $F(\cdot, \cdot)$. Mizuno and Nagasawa [56], and Tunçel [81] consider variants of the primal–dual potential reduction algorithm that use the primal–dual affine scaling direction. Ye et al. [90] consider modifications of the primal–dual algorithm based on varying the value of the parameter q in the system used to derive the primal–dual directions.

6 Enhancements and Extensions

In this section we describe several modifications of the potential reduction methods described in the previous sections that enhance the theoretical complexity, and/or practical performance, of the algorithms. We also describe extensions of the algorithms to problems more general than LP.

6.1 Partial Updating

For each of the algorithms described above, the dominant computational task on each iteration is the formation, and factorization, of an $m \times m$ matrix, requiring $O(m^2n)$ operations using standard linear algebra. The remaining work per iteration is all $O(mn)$. As a result, the total complexity for Karmarkar’s algorithm is $O(m^2n^2L)$ operations, and the total complexity for both the affine and primal–dual potential reduction algorithms (using $q = n + \sqrt{n}$) is $O(m^2n^{1.5}L)$ operations.

The above total complexity bounds can be improved using a technique known as partial updating. Consider Karmarkar’s algorithm, or the affine potential reduction method. Then the matrix to be formed and factorized on each iteration is of the form $A(X^k)^2A^T$. The idea of partial updating is to instead maintain a factorization of a matrix $A(\tilde{X}^k)^2A^T$, where $\tilde{x}^k > 0$ satisfies

$$\frac{1}{\rho} \leq \frac{\tilde{x}_i^k}{x_i^k} \leq \rho, \quad i = 1, \dots, n, \quad (27)$$

and $\rho > 1$ is a $O(1)$ constant. The computations required on each step are then modified to use the factorization of $A(\tilde{X}^k)^2A^T$, instead of a factorization of $A(X^k)^2A^T$. Following a step from x^k to x^{k+1} , the algorithm first sets $\tilde{x}^{k+1} = \tilde{x}^k$, and then “updates” any indices i which fail to satisfy (27), for $k = k+1$. Each such update produces a rank-one change in $A(\tilde{X}^{k+1})^2A^T$, requiring an update of the factorization of $A(\tilde{X}^{k+1})^2A^T$ that can be performed in $O(m^2)$ operations. See for example Shanno [68] for details of updating a Cholesky factorization. Karmarkar [44], who introduced the technique, shows that when his algorithm uses partial updating the number of iterations is still $O(nL)$ but the total number of updates required on all iterations is only $O(n^{1.5}L)$. As a result, the complexity of Karmarkar’s algorithm using partial updating is reduced to $O(n^{1.5}(m^2)L + n(mn)L) = O(m^{1.5}n^2L)$. In the interior point literature the distinction between m and n is often ignored, in which case partial updating provides a factor-of- \sqrt{n} complexity improvement.

We will not present the details of potential reduction algorithms that incorporate partial updating, but we will describe some results on the topic. A serious shortcoming of Karmarkar’s original analysis of partial updating is that the complexity improvement requires that the algorithm take short steps ($\alpha < 1$), instead of performing a linesearch of the potential function. This restriction makes the technique hopelessly impractical. Anstreicher [6] shows that with a simple safeguard, a linesearch can be performed when using partial updating, while still retaining the complexity improvement. Ye [85] describes a partial updating version of the affine potential reduction algorithm that reduces the total complexity to $O(m^{1.5}n^{1.5}L)$ operations. However, the analysis of [85], like that in [44], requires that the algorithm take short steps. Anstreicher and Bosch [13] adapt the safeguarded linesearch of [6] to the affine potential reduction algorithm, resulting in an $O(m^{1.5}n^{1.5}L)$ algorithm that can use linesearch to improve the reduction in $F(\cdot, \cdot)$ on each iteration. Other partial updating variants of the affine potential reduction method are devised by Bosch [16], and Mizuno [53], [54].

Partial updating can also be applied to primal–dual algorithms, which are based on a primal–dual scaling matrix of the form $(X^k)^{1/2}(S^k)^{-1/2}$. Bosch and Anstreicher [17] devise an $O(m^{1.5}n^{1.5}L)$ partial updating variant of the

primal–dual potential reduction algorithm of [47], that allows for safeguarded linesearch of $F(\cdot, \cdot)$ using unequal primal and dual steplengths.

Although partial updating is important from the standpoint of theoretical complexity, the technique has not been used very much in practice. The reason for this is quite simple. The complexity improvement from partial updating is based on worst–case decrease in the potential function, and reducing the number of updates per iteration from n to an average of $O(\sqrt{n})$. However, in practice algorithms typically achieve potential decreases that are much better than the worst–case bounds, using long steps that would result in $O(n)$ updates per iteration. The additional “overhead” required to implement partial updating then makes the technique uncompetitive. Shanno [68] and Bosch and Anstreicher [18] present computational results using partial updating. In [18] it is shown that for certain problem structures partial updating can actually enhance the practical performance of the affine potential reduction algorithm.

6.2 Long Steps

For each of the algorithms considered above, the steps (in (5), (14), and (22)) are parameterized using a two–norm steplength. In practice a potential reduction algorithm can (and generally will) use a steplength having $\alpha > 1$, but the performance on such a “long” step cannot be theoretically analyzed. One way to analyze such long steps, and in so doing perhaps get more insight into the typical behavior of a potential reduction algorithm, is to parameterize the step in terms of an infinity–norm, as opposed to two–norm, steplength.

Consider for example Karmarkar’s algorithm. Instead of the step as in (5), define a step of the form

$$\bar{x}' = e - \alpha \frac{\Delta \bar{x}}{\|\Delta \bar{x}\|_\infty}. \quad (28)$$

Proceeding as in the derivation of (7), we then obtain

$$\begin{aligned} & f(x^{k+1}, z^{k+1}) - f(x^k, z^{k+1}) \\ &= (n+1) \ln \left(1 - \frac{\alpha \|\Delta \bar{x}\|^2 / \|\Delta \bar{x}\|_\infty}{(\bar{c} - \bar{z}d)^T e} \right) - \sum_{i=1}^{n+1} \ln \left(1 - \alpha \frac{\Delta \bar{x}_i}{\|\Delta \bar{x}\|_\infty} \right) \\ &\leq (n+1) \ln \left(1 - \frac{\alpha (\|\Delta \bar{x}\| / \|\Delta \bar{x}\|_\infty)^2 \|\Delta \bar{x}\|_\infty}{(\bar{c} - \bar{z}d)^T e} \right) + \frac{\alpha^2 (\|\Delta \bar{x}\| / \|\Delta \bar{x}\|_\infty)^2}{2(1-\alpha)} \\ &\leq \left(-\alpha + \frac{\alpha^2}{2(1-\alpha)} \right) \frac{\|\Delta \bar{x}\|^2}{\|\Delta \bar{x}\|_\infty^2}, \end{aligned} \quad (29)$$

where the first inequality uses Lemma 3.2 and the fact that $e^T \Delta \bar{x} = 0$, and the second uses (4) and $\ln(1-t) \leq -t$ for $t < 1$. As in Theorem 3.3, (29)

shows that an $\Omega(1)$ decrease in $f(\cdot, \cdot)$ is always possible. However, (29) also indicates that the decrease on a step of Karmarkar's algorithm will typically be *much* greater. In particular, $\|\Delta\bar{x}\|^2/\|\Delta\bar{x}\|_\infty^2$ is typically $\Omega(n/\ln(n))$ (as first observed by Nemirovskii [60]), implying that the algorithm can obtain a potential decrease of $\Omega(n/\ln(n))$ on a single step. From Theorem 2.1, this magnitude of potential decrease per step results in an $O(\ln(n)L)$ iteration algorithm, in accord with the observation that in practice the convergence of the algorithm is independent (or nearly independent) of n .

Nesterov and Todd [62] suggest a similar "long step" analysis for the affine potential reduction algorithm based on $f(\cdot, \cdot)$, with $q = 2n$. Let $\Delta\bar{x}$ be as in (11), and suppose that $\|\Delta\bar{x}\|_\infty \leq \eta < 1$. Let $z^{k+1} = b^T y'$, where y' is as in (12). It then follows easily that

$$\bar{c}^T e - z^{k+1} \leq \frac{1+\eta}{2}(\bar{c}^T e - z^k), \quad (30)$$

and also that

$$f(x^k, z^{k+1}) - f(x^k, z^k) = q \ln \left(\frac{\bar{c}^T e - z^{k+1}}{\bar{c}^T e - z^k} \right) \leq -(1-\eta)n.$$

Thus updates of the lower bound now produce an $\Omega(n)$ decrease in $f(\cdot, \cdot)$. Next consider the situation where $\|\Delta\bar{x}\|_\infty > \eta$. Instead of using the step as in (14), define

$$\bar{x}' = e - \alpha \frac{\Delta\bar{x}}{\|\Delta\bar{x}\|_\infty}.$$

Proceeding as in the proof of Lemma 4.2, we obtain

$$\begin{aligned} & f(x^{k+1}, z^k) - f(x^k, z^k) \\ & \leq -\alpha \left(\frac{q}{\bar{c}^T e - z^k} \bar{c} - e \right)^T \frac{\Delta\bar{x}}{\|\Delta\bar{x}\|_\infty} + \frac{\alpha^2 (\|\Delta\bar{x}\|/\|\Delta\bar{x}\|_\infty)^2}{2(1-\alpha)} \\ & = \left(-\alpha \|\Delta\bar{x}\|_\infty + \frac{\alpha^2}{2(1-\alpha)} \right) \frac{\|\Delta\bar{x}\|^2}{\|\Delta\bar{x}\|_\infty^2} \\ & \leq \left(-\alpha\eta + \frac{\alpha^2}{2(1-\alpha)} \right) \frac{\|\Delta\bar{x}\|^2}{\|\Delta\bar{x}\|_\infty^2}. \end{aligned} \quad (31)$$

As in the case of Karmarkar's algorithm, (31) assures an $\Omega(1)$ decrease in $f(\cdot, z^k)$, but indicates that a much larger decrease will typically occur.

If one considers the affine potential reduction algorithm using $F(\cdot, \cdot)$, with $q = 2n$, then the situation on primal steps, with $\|\Delta\bar{x}\|_\infty \geq \eta$, is exactly as above. For dual steps, the effect on $F(\cdot, \cdot)$ can easily be analyzed as in the proof of Theorem 4.4. The final result is that on a dual step, where $\|\Delta\bar{x}\|_\infty \leq \eta$, $F(x^k, s^k) - F(x^k, s^{k+1}) \geq n(1-2\eta)/(2-2\eta)$, a decrease of

exactly n times the bound of Theorem 4.4. However, with $q = 2n$ there is essentially no reason to measure progress of the algorithm using $F(\cdot, \cdot)$.

For a more extensive discussion of the use of “long steps” in potential reduction methods see Nesterov [61], Nesterov and Todd [62], and Todd [78]. The latter also describes a “long step” analysis for the primal–dual potential reduction algorithm.

6.3 Large–step dual updates

The affine potential reduction method based on $F(\cdot, \cdot)$, with $q = n + \sqrt{n}$, was considered a major breakthrough in interior point methods. Previous $O(\sqrt{n}L)$ iteration methods were all of the short–step path–following variety (see for example [37]), where iterates were constrained to remain within a small neighborhood of the central trajectory. The affine potential reduction method, on the other hand, placed no explicit restrictions on the iterates, and offered the possibility of an $O(\sqrt{n}L)$ algorithm that might perform well in practice. Unfortunately the algorithm does *not* perform well in practice with $q = n + \sqrt{n}$. An explanation for this phenomenon was provided by Gonzaga [34]. With $q = n + \sqrt{n}$, dual updates are performed when $\|\Delta\bar{x}\| \leq \eta < 1$. The result of such an update is a “small–step” reduction in the gap; in fact (13) indicates that the gap is reduced by a factor which is no smaller than $1 - (1 + \eta)/\sqrt{n}$. On the other hand the algorithm takes primal steps when $\|\Delta\bar{x}\| > \eta$, and in this case the “worst–case” reduction in $F(\cdot, \cdot)$ is only $\Omega(1)$. One might hope that the use of a linesearch on the primal and dual steps could improve the performance of the algorithm, but in practice (with $q = n + \sqrt{n}$) this improvement is minimal.

It turns out that it is possible to retain the $O(\sqrt{n}L)$ iteration complexity of the affine potential reduction algorithm while using “larger–step” dual updates. Consider $q = n + \nu\sqrt{n}$, where $\nu = O(1)$. The analysis of descent in $F(\cdot, \cdot)$ for primal and dual steps is then almost identical to the analysis with $\nu = 1$, and the bounds provided by Lemma 4.3 and Theorem 4.4 continue to hold. By Theorem 2.2, the algorithm remains an $O(\sqrt{n}L)$ iteration algorithm. However, the dual update will now result in

$$\frac{c^T x^k - z^{k+1}}{c^T x^k - z^k} \leq \frac{n + \eta\sqrt{n}}{n + \nu\sqrt{n}},$$

so large values of ν produce a better gap reduction on dual steps. In addition, following such a step one will tend to have a larger value for $\|\Delta\bar{x}\|$, resulting in a primal step with better potential decrease. This is the rationale behind the “large step dual update” of [34], although Gonzaga describes the dual update somewhat differently from the way we describe it here, and bases his complexity analysis on $f(\cdot, \cdot)$ rather than $F(\cdot, \cdot)$.

A “truly large” dual step update, with an $\Omega(1)$ reduction in the gap, is provided by using $q = 2n$. In this case the algorithm can also be analyzed using an infinity–norm parameterization of the primal step, as described above. Thus $q = 2n$ produces truly–large–step dual updates, and allows for long primal steps, leading to a very substantial improvement in the practical performance of the algorithm.

6.4 Infeasible–Start Methods

The potential reduction algorithms described above all require an initial primal feasible $x^0 > 0$, and possibly an initial dual feasible s^0 . As described in Section 2, it is possible to devise an augmented problem like MLP which has an initial feasible solution. However, the large value of the parameter M makes the use of MLP computationally unattractive.

Several approaches have been developed to allow potential reduction algorithms to operate on problems that do not have a known feasible interior point, without the use of M as in MLP. Phase I – Phase II algorithms use a formulation similar to MLP, but without the explicit use of the M objective coefficient. Consider a problem:

$$\begin{aligned} \min \quad & \hat{c}^T \hat{x} \\ & \hat{A} \hat{x} = b \\ & d^T \hat{x} = 0 \\ & \hat{x} \geq 0, \end{aligned} \tag{32}$$

where $\hat{x} \in \mathbf{R}^{n+1}$,

$$\hat{A} = (A, b - Ax^0), \quad \hat{c} = \begin{pmatrix} c \\ 0 \end{pmatrix}, \quad d = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

and $x^0 > 0$. It is *not* assumed that $Ax^0 = b$. Clearly (32) is equivalent to LP, and \hat{x}^0 given by $\hat{x}_i^0 = x_i^0$, $i = 1, \dots, n$, $\hat{x}_{n+1}^0 = 1$ is feasible for all of the constraints of (32) *except* the constraint $d^T \hat{x} = 0$. The approach of a Phase I – Phase II potential reduction algorithm is to simultaneously decrease the usual primal potential function $f(\cdot, \cdot)$ based on (32), and also decrease a “Phase I” potential function:

$$\hat{f}(\hat{x}) = q \ln(d^T \hat{x}) - \sum_{i=1}^{n+1} \ln(\hat{x}_i).$$

Algorithms of this type based on Karmarkar’s algorithm, using $q = n + 1$, were devised by Anstreicher [4], and Todd [75]. Methods based on the affine potential reduction algorithm, using $f(\cdot, \cdot)$ and $q \geq n + \sqrt{n}$, can be found in Anstreicher [9], and Todd [76]. It should be noted that even with $q = n + \sqrt{n}$,

the latter algorithms cannot use $F(\cdot, \cdot)$ to improve the complexity of these methods to $O(\sqrt{n}L)$ iterations.

DeGhellinck and Vial [20] describe a variant of Karmarkar’s algorithm, based on parameterized feasibility problems, that does not require an initial feasible point. When initialized with a feasible point, the method of [20] is essentially the “standard form” variant of Karmarkar’s method, as described in Section 3. Fraley [22] considers an improvement of the lower bound procedure in [20] when the initial point is not feasible. Freund [27] describes a Phase I – Phase II affine potential reduction algorithm that uses a single potential function, and enforces a “balance” between the Phase I and Phase II objectives through an added constraint.

Freund [25] uses a “shifted barrier” approach to allow for the initialization of a potential reduction algorithm with an infeasible point. In [25] it is assumed that $Ax^0 = b$, but that x^0 may have negative components. The usual potential function $f(\cdot, \cdot)$ is replaced by a function of the form

$$q \ln(c^T x - z) - \sum_{i=1}^n \ln(x_i + h_i(c^T x - z)),$$

where $q = n + \sqrt{n}$, and $h > 0$ is a “shift” vector such that $x^0 + (c^T x^0 - z^0)h > 0$. Similarly $F(\cdot, \cdot)$ is replaced with a potential function that includes the shifted primal barrier terms. Algorithms based on these perturbed potential functions have complexities of $O(nL)$ or $O(\sqrt{n}L)$ iterations, under various assumptions regarding the dual feasible region.

In practice, primal–dual “infeasible–interior–point” methods have been used very successfully to solve linear programs from infeasible starting points. For a given iterate $x^k > 0$, $s^k > 0$, these algorithms obtain search directions Δx and Δs by solving a system of the form:

$$\begin{aligned} A\Delta x &= b - Ax^k \\ A^T \Delta y + \Delta s &= c - A^T y^k - s^k \\ S^k \Delta x + X^k \Delta s &= \gamma \mu^k e - X^k S^k e, \end{aligned} \tag{33}$$

where $0 \leq \gamma \leq 1$, and $\mu^k = (x^k)^T s^k / n$. (The use of $\gamma = 0$ results in the “primal–dual affine scaling,” or “predictor” step, while $\gamma = 1$ gives a “centering,” or “corrector” step.) The next point is of the form

$$x^{k+1} = x^k + \alpha \Delta x, \quad s^{k+1} = s^k + \alpha \Delta s,$$

for a step parameter $\alpha \leq 1$. Most algorithms based on (33) are of the path–following, or predictor–corrector variety. However, Mizuno, Kojima and Todd [55] devise a potential reduction algorithm that uses directions from (33).

6.5 Linear Complementarity Problems

The Linear Complementarity Problem is:

$$\begin{aligned} \text{LCP :} \quad & s - Mx = q \\ & s \geq 0, x \geq 0, x^T s = 0, \end{aligned}$$

where M is an $n \times n$ matrix, and $q \in \mathbf{R}^n$. It is well known that for appropriate choices of M (see for example [19]), LCP can be used to represent linear programming, convex quadratic programming, matrix games, and other problems. Many primal–dual algorithms for LP can be extended to LCP, under the assumption that M is a positive semidefinite (but not necessarily symmetric) matrix. In particular, the primal–dual potential reduction algorithm of Section 5 was originally devised as a method for LCP, and retains a complexity of $O(\sqrt{n}L)$ iterations so long as M is positive semidefinite. See Kojima, Mizuno, and Yoshise [47] for details.

The theory of LCP depends very heavily on the membership of M in various classes of matrices (for example, positive semidefinite matrices). Kojima et al. [45] discuss the application of interior point algorithms, including primal–dual potential reduction methods, to LCP problems with different types of M . Kojima, Megiddo, and Ye [46] analyze a potential–reduction algorithm in the case that M is a P–matrix (that is, a matrix with positive principal minors), for which a solution to LCP always exists (see [19]). Ye [87] analyzes a potential reduction algorithm that obtains an approximate stationary point of a general LCP, and Ye [88] considers a potential reduction method for the related problem of approximating a Karush–Kuhn–Tucker point of a general quadratic programming problem. The last three references show that the potential reduction framework can be used to analyze algorithms that are not polynomial–time methods.

6.6 Linear Programming over Cones

Nesterov and Nemirovskii [64] consider a “conic” extension of the usual linear programming problem of the form

$$\begin{aligned} \text{CLP :} \quad & \min \langle c, x \rangle \\ & Ax = b \\ & x \in K, \end{aligned}$$

where x is in a finite–dimensional real vector space \mathbf{X} , c is in the dual space \mathbf{X}^* , b is in a finite–dimensional real vector space \mathbf{Y} , A is a linear mapping from \mathbf{X} to \mathbf{Y} , and K is a closed, convex, and pointed cone in \mathbf{X} . A dual

problem for CLP is then

$$\begin{aligned} \text{CLD :} \quad & \min \langle b, y \rangle \\ & A^*y + s = c \\ & s \in K^*, \end{aligned}$$

where $A^* : \mathbf{Y}^* \rightarrow \mathbf{X}^*$ is the adjoint of A , $y \in \mathbf{Y}$, and $K^* \subset \mathbf{X}^*$ is the dual cone

$$K^* = \{s \in \mathbf{X}^* \mid \langle x, s \rangle \geq 0 \ \forall x \in K\}.$$

Strong duality holds between CLP and CLD if, for example, CLP and CLD both have feasible solutions which are interior to the cones K and K^* , respectively. See [64] for more extensive duality results for these problems. Note that if $\mathbf{X} = \mathbf{R}^n$, $\mathbf{Y} = \mathbf{R}^m$, and $K = \mathbf{R}_+^n$, the nonnegative orthant, then CLP is simply LP. It is shown in [64] that CLP actually provides a formulation for general convex programming.

In [64, Chapter 4] it is shown that Karmarkar's algorithm, and the affine potential reduction algorithm, can be extended to problems of the form CLP so long as the cone K possesses a ϑ -logarithmically-homogeneous barrier. The exact definition of such a barrier, and its properties, are beyond the scope of this article. We note here only that the complexities of algorithms for CLP depend on the parameter ϑ . For the usual LP problem, $-\sum_{i=1}^n \ln(x_i)$ is an n -logarithmically-homogeneous barrier for \mathbf{R}_+^n . Another important special case takes \mathbf{X} to be the space of $n \times n$ symmetric matrices, and K the cone of symmetric positive semidefinite matrices. For this case $\langle x, s \rangle = \text{tr}(xs)$, where $\text{tr}(\cdot)$ denotes the trace of a matrix, $K^* = K$, and the barrier $-\ln(\det(x))$ is an n -logarithmically-homogeneous barrier for K . Problems of the latter type are now commonly referred to as *semidefinite programming* problems, and have a number of significant applications in combinatorial optimization, control theory, and elsewhere. See Vandenberghe and Boyd [82] for an excellent survey of semidefinite programming applications, and algorithms.

Todd [78] gives a much more extensive discussion of Nesterov and Nemirovskii's [64] generalization of potential reduction algorithms to CLP. The extension of a potential reduction algorithm (specifically Ye's [83] projective potential reduction method) to semidefinite programming was independently obtained by Alizadeh [1]. Nesterov and Todd [62], [63] obtain an extension of the primal-dual potential reduction method to problems of the form CLP where K and its barrier are *self-scaled*; see also [78] for a summary of these results.

Acknowledgement

I would like to thank Rob Freund, Tamas Terlaky, Mike Todd, and Yinyu Ye for their comments on a draft of this article.

References

- [1] F. Alizadeh, “Interior point methods in semidefinite programming with applications to combinatorial optimization,” *SIAM J. Opt.* 5 (1995) 13-51.
- [2] M.D. Asic, V.V. Kovacevic-Vujcic, and M.D. Radosavljevic-Nikolic, “A note on limiting behavior of the projective and the affine rescaling algorithms, *Contemporary Mathematics* 114 (1990) 151-157.
- [3] K.M. Anstreicher, “A monotonic projective algorithm for fractional linear programming,” *Algorithmica* 1 (1986) 483-498.
- [4] K.M. Anstreicher, “The worst-case step in Karmarkar’s algorithm,” *Math. Oper. Res.* 14 (1989) 294-302.
- [5] K.M. Anstreicher, “A combined phase I–phase II projective algorithm for linear programming,” *Math. Prog.* 43 (1989) 209-223.
- [6] K.M. Anstreicher, “A standard form variant, and safeguarded line-search, for the modified Karmarkar algorithm,” *Math. Prog.* 47 (1990) 337-351.
- [7] K.M. Anstreicher, “Dual ellipsoids and degeneracy in the projective algorithm for linear programming,” *Contemporary Mathematics* 114 (1990) 141-149.
- [8] K.M. Anstreicher, “On the performance of Karmarkar’s algorithm over a sequence of iterations,” *SIAM J. Opt.* 1 (1991) 22-29.
- [9] K.M. Anstreicher, “A combined phase I – phase II scaled potential algorithm for linear programming,” *Math. Prog.* 52 (1991) 429-439.
- [10] K.M. Anstreicher, “On monotonicity in the scaled potential algorithm for linear programming,” *Linear Algebra Appl.* 152 (1991) 223-232.
- [11] K.M. Anstreicher, “Strict monotonicity and improved complexity in the standard form projective algorithm for linear programming,” *Math. Prog.* 62 (1993) 517-535.
- [12] K.M. Anstreicher, “Large step volumetric potential reduction algorithms for linear programming,” to appear in *Annals of O.R.* (1996).
- [13] K.M. Anstreicher and R.A. Bosch, “Long steps in an $O(n^3L)$ algorithm for linear programming,” *Math. Prog.* 54 (1992) 251-265.
- [14] K.M. Anstreicher and P. Watteyne, “A family of search directions for Karmarkar’s algorithm,” *Operations Research* 41 (1993), 759-767.

- [15] D. Bayer and J.C. Lagarias, “Karmarkar’s linear programming algorithm and Newton’s method,” *Math. Prog.* 50 (1991) 291-330.
- [16] R.A. Bosch, “On Mizuno’s rank one updating algorithm for linear programming,” *SIAM J. Opt.* 3 (1993) 861-867.
- [17] R.A. Bosch and K.M. Anstreicher, “On partial updating in a potential reduction linear programming algorithm of Kojima, Mizuno, and Yoshise,” *Algorithmica* 9 (1993) 184-197.
- [18] R.A. Bosch and K.M. Anstreicher, “A partial updating algorithm for linear programs with many more variables than constraints,” *Optimization Methods and Software* 4 (1995) 243-257.
- [19] R.W. Cottle, J.-S. Pang, and R.E. Stone, *The Linear Complementarity Problem* (Academic Press, Boston, 1992).
- [20] G. de Ghellinck and J.-Ph. Vial, “A polynomial Newton method for linear programming,” *Algorithmica* 1 (1986) 425-453.
- [21] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming, Sequential Unconstrained Minimization Techniques*, (John Wiley, New York, 1968); reprinted as *Classics in Applied Mathematics Vol. 4*, (SIAM, Philadelphia, 1990).
- [22] C. Fraley, “Linear updates for a single-phase projective method,” *O.R. Letters* 9 (1990) 169-174.
- [23] R.M. Freund, “An analog of Karmarkar’s algorithm for inequality constrained linear programs, with a ‘new’ class of projective transformations for centering a polytope,” *O.R. Letters* 7 (1988) 9-14.
- [24] R.M. Freund, “Polynomial-time algorithms for linear programming based only on primal scaling and projected gradients of a potential function,” *Math. Prog.* 51 (1991) 203-222.
- [25] R.M. Freund, “A potential-function reduction algorithm for solving a linear program directly from an infeasible ‘warm start’,” *Math. Prog.* 52 (1991) 441-466.
- [26] R.M. Freund, “Projective transformations for interior-point algorithms, and a superlinearly convergent algorithm for the w-center problem,” *Math. Prog.* 58 (1993) 385-414.
- [27] R.M. Freund, “A potential reduction algorithm with user-specified phase I-phase II balance for solving a linear program from an infeasible warm start,” *SIAM J. Opt.* 5 (1995) 247-268.

- [28] D.M. Gay, “A variant of Karmarkar’s linear programming algorithm for problems in standard form,” *Math. Prog.* 37 (1987) 81-90.
- [29] P. Gill, W. Murray, M. Saunders, J. Tomlin, and M. Wright, “On projected Newton barrier methods for linear programming and an equivalence to Karmarkar’s projective method,” *Math. Prog.* 36 (1986) 183-209.
- [30] D. Goldfarb and S. Mehrotra, “Relaxed variants of Karmarkar’s algorithm for linear programs with unknown optimal objective value,” *Math. Prog.* 40 (1988), 183-195.
- [31] D. Goldfarb and S. Mehrotra, “A relaxed version of Karmarkar’s method,” *Math. Prog.* 40 (1988), 289-315.
- [32] C.C. Gonzaga, “Conical projection algorithms for linear programming,” *Math. Prog.* 43 (1989) 151-173.
- [33] C.C. Gonzaga, “Polynomial affine algorithms for linear programming,” *Math. Prog.* 49 (1991) 7-21.
- [34] C.C. Gonzaga, “Large-step path following methods for linear programming, part II: potential reduction method,” *SIAM J. Opt.* 1 (1991) 280-292.
- [35] C.C. Gonzaga, “Interior point algorithms for linear programs with inequality constraints,” *Math. Prog.* 52 (1991) 209-225.
- [36] C.C. Gonzaga, “On lower bound updates in primal potential reduction methods for linear programming,” *Math. Prog.* 52 (1991) 415-428.
- [37] C.C. Gonzaga, “Path-following methods for linear programming,” *SIAM Review* 34 (1992) 167-224.
- [38] C.C. Gonzaga and M.J. Todd, “An $O(\sqrt{n}L)$ -iteration large-step primal-dual affine algorithm for linear programming,” *SIAM J. Opt.* 2 (1992) 349-359.
- [39] P. Huard, “Resolution of mathematical programming with nonlinear constraints by the method of centres,” in *Nonlinear Programming*, J. Abadie, editor (North-Holland, Amsterdam, 1967).
- [40] H. Imai, “On the convexity of the multiplicative version of Karmarkar’s potential function,” *Math. Prog.* 40 (1988) 29-32.
- [41] M. Iri and H. Imai, “A multiplicative barrier function method for linear programming,” *Algorithmica* 1 (1986) 455-482.

- [42] B. Jansen, C. Roos, and T. Terlaky, “The theory of linear programming: skew symmetric self-dual problems and the central path,” *Optimization* 29 (1993) 225-233.
- [43] J. Ji and Y. Ye, “A complexity analysis for interior-point algorithms based on Karmarkar’s potential function,” *SIAM J. Opt.* 4 (1994) 512-520.
- [44] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” *Combinatorica* 4 (1984) 373-395.
- [45] M. Kojima, N. Megiddo, T. Noma, and A. Yoshise, “A unified approach to interior point algorithms for linear complementarity problems,” *Lecture Notes in Computer Science* 538 (Springer-Verlag, Berlin, 1991).
- [46] M. Kojima, N. Megiddo, and Y. Ye, “An interior point potential reduction algorithm for the linear complementarity problem,” *Math. Prog.* 54 (1992) 267-279.
- [47] M. Kojima, S. Mizuno, and A. Yoshise, “An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems,” *Math. Prog.* 50 (1991) 331-342.
- [48] C. McDiarmid, “On the improvement per iteration in Karmarkar’s algorithm for linear programming,” *Math. Prog.* 46 (1990) 299-320.
- [49] N. Megiddo and M. Shub, “Boundary behavior of interior point algorithms in linear programming,” *Math. Oper. Res.* 14 (1989), 97-146
- [50] J.E. Mitchell, “Updating lower bounds when using Karmarkar’s projective algorithm for linear programming,” *JOTA* 78 (1993) 127-142.
- [51] J.E. Mitchell and M.J. Todd, “On the relationship between the search directions in the affine and projective variants of Karmarkar’s linear programming algorithm,” in *Contributions to Operations Research and Economics: The Twentieth Anniversary of CORE*, B. Cornet and H. Tulkens, editors, MIT Press (Cambridge, MA, 1989) 237-250.
- [52] J.E. Mitchell and M.J. Todd, “A variant of Karmarkar’s linear programming algorithm for problems with some unrestricted variables,” *SIAM J. Matrix Anal. Appl.* 10 (1989) 30-38.
- [53] S. Mizuno, “A rank one updating algorithm for linear programming,” *The Arabian Journal for Science and Engineering* 15 (1990) 671-677.
- [54] S. Mizuno, “ $O(n^pL)$ iteration $O(n^3L)$ potential reduction algorithms for linear programming,” *Linear Algebra Appl.* 152 (1991) 155-168.

- [55] S. Mizuno, M. Kojima, and M.J. Todd, “Infeasible-interior-point primal-dual potential-reduction algorithms for linear programming,” *SIAM J. Opt.* 5 (1995) 52-67.
- [56] S. Mizuno and A. Nagasawa, “A primal-dual affine scaling potential reduction algorithm for linear programming,” *Math. Prog.* 62 (1993) 119-131.
- [57] R.D.C. Monteiro, “Convergence and boundary behavior of the projective scaling trajectories for linear programming,” *Contemporary Mathematics* 114 (1990) 213-229.
- [58] R.D.C. Monteiro, “On the continuous trajectories for a potential reduction algorithm for linear programming,” *Math. Oper. Res.* 17 (1992) 225-253.
- [59] M. Muramatsu and T. Tsuchiya, “A convergence analysis of a long-step variant of the projective scaling algorithm,” The Institute of Statistical Mathematics (Tokyo, Japan, 1993); to appear in *Math. Prog.*
- [60] A.S. Nemirovskii, “An algorithm of the Karmarkar type,” *Soviet Journal on Computers and Systems Sciences* 25 (1987) 61-74.
- [61] Y.E. Nesterov, “Long-step strategies in interior point potential-reduction algorithms,” Dept. SES-COMIN, University of Geneva (Geneva, Switzerland, 1993).
- [62] Y.E. Nesterov and M.J. Todd, “Self-scaled barriers and interior-point methods for convex programming,” Technical Report 1091, School of OR/IE, Cornell University (Ithaca, NY, 1994); to appear in *Math. Oper. Res.*.
- [63] Y.E. Nesterov and M.J. Todd, “Primal-dual interior point methods for self-scaled cones,” Technical Report 1125, School of OR/IE, Cornell University (Ithaca, NY, 1995).
- [64] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming* (SIAM, Philadelphia, 1994).
- [65] C.H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, 1982).
- [66] M.J.D. Powell, “On the number of iterations of Karmarkar’s algorithm for linear programming,” *Math. Prog.* 62 (1993) 153-197.
- [67] A.E. Steger, “An extension of Karmarkar’s algorithm for bounded linear programming problems,” M.S. Thesis, State University of New York (Stonybrook, NY, 1985).

- [68] D.F. Shanno, “Computing Karmarkar projections quickly,” *Math. Prog.* 41 (1988) 61-71.
- [69] D. Shaw and D. Goldfarb, “A path-following projective interior point method for linear programming,” *SIAM J. Opt.* 4 (1994) 65-85.
- [70] K. Tanabe, “Centered Newton method for mathematical programming,” *Lecture Notes in Control and Information Sciences* 113 (Springer-Verlag, Berlin, 1988) 197-206.
- [71] M.J. Todd, “Exploiting special structure in Karmarkar’s linear programming algorithm,” *Math. Prog.* 41 (1988) 97-113.
- [72] M.J. Todd, “Improved bounds and containing ellipsoids in Karmarkar’s linear programming algorithm,” *Mathematics of Operations Research* 13 (1988) 650-659.
- [73] M.J. Todd, “The effects of degeneracy and null and unbounded variables on variants of Karmarkar’s linear programming algorithm,” in *Large Scale Numerical Optimization*, T.F. Coleman and Y. Li, editors (SIAM, Philadelphia, 1990).
- [74] M.J. Todd, “A Dantzig–Wolfe–like variant of Karmarkar’s interior–point linear programming algorithm,” *Operations Research* 38 (1990) 1006-1018.
- [75] M.J. Todd, “On Anstreicher’s combined phase I–phase II projective algorithm for linear programming,” *Math. Prog.* 55 (1992) 1-15.
- [76] M.J. Todd, “Combining phase I and phase II in a potential reduction algorithm for linear programming,” *Math. Prog.* 59 (1993) 133-150.
- [77] M.J. Todd, “Interior–point algorithms for semi–infinite programming,” *Math. Prog.* 65 (1994) 217-245.
- [78] M.J. Todd, “Potential–reduction methods in mathematical programming,” School of IE/OR, Cornell University (Ithaca, NY, 1995); to appear in *Math. Prog.*
- [79] M.J. Todd and B.P. Burrell, “An extension of Karmarkar’s algorithm for linear programming using dual variables,” *Algorithmica* 1 (1986) 409-424.
- [80] M.J. Todd and Y. Ye, “A centered projective algorithm for linear programming,” *Math. Oper. Res.* 15 (1990) 508-529.
- [81] L. Tunçel, “Constant potential primal–dual algorithms: a framework,” *Math. Prog.* 66 (1994) 145-159.

- [82] L. Vandenberghe and S. Boyd, “Positive definite programming,” Dept. of Electrical Engineering, Stanford University (Stanford, CA, 1994); to appear in *SIAM Review*.
- [83] Y. Ye, “A class of projective transformations for linear programming,” *SIAM J. Comp.* 19 (1990) 457-466.
- [84] Y. Ye, “A ‘build down’ scheme for linear programming,” *Mathematical Programming* 46 (1990) 61-72.
- [85] Y. Ye, “An $O(n^3L)$ potential reduction algorithm for linear programming,” *Math. Prog.* 50 (1991) 239-258.
- [86] Y. Ye, “A potential reduction algorithm allowing column generation,” *SIAM J. Opt.* 2 (1992), 7-20.
- [87] Y. Ye, “A fully polynomial-time approximation algorithm for computing a stationary point of the general LCP,” *Math. Oper. Res.* 18 (1993) 334-345.
- [88] Y. Ye, “On the complexity of approximating a KKT point of quadratic programming,” Dept. of Management Sciences, University of Iowa (Iowa City, IA, 1995).
- [89] Y. Ye and M. Kojima, “Recovering optimal dual solutions in Karmarkar’s polynomial algorithm for linear programming,” *Math. Prog.* 39 (1987) 305-317.
- [90] Y. Ye, K.O. Kortanek, J.A. Kaliski, and S. Huang, “Near-boundary behavior of primal-dual potential reduction algorithms for linear programming,” *Math. Prog.* 58 (1993) 243-255.
- [91] Y. Ye, M.J. Todd, and S. Mizuno, “An $O(\sqrt{n}L)$ -iteration homogeneous and self-dual linear programming algorithm,” *Math. Oper. Res.* 19 (1994) 53-67.