KARMARKAR'S ALGORITHM: A VIEW FROM NONLINEAR PROGRAMMING

by

M.J.D. Powell

Department of Applied Mathematics and Theoretical Physics,
University of Cambridge, Cambridge CB3 9EW, England.

No. 53 November, 1989

Karmarkar's algorithm: a view from nonlinear programming

M.J.D. Powell[1]

Abstract Karmarkar's algorithm for linear programming has become a highly active field of research, because it is claimed to be supremely efficient for the solution of very large calculations, because it has polynomial-time complexity, and because its theoretical properties are interesting. We describe and study the algorithm in the usual way that employs projective transformations and that requires the linear programming problem to be expressed in a standard form, the only inequality constraints being simple bounds on the variables. We then eliminate the dependence on the transformations analytically, which gives the form of the algorithm that can be viewed as a barrier function method from nonlinear programming. In this case the directions of the changes to the variables are solutions of quadratic programming calculations that have no general inequality constraints. By using some of the equalities to eliminate variables, we find a way of applying the algorithm directly to linear programming problems in general form. Thus, except for the addition of at most two new variables that make all but one of the constraints homogeneous, there is no need to increase the original number of variables, even when there are very many constraints. We apply this procedure to a two variable problem with an infinite number of constraints that are derived from tangents to the unit circle. We find that convergence occurs to a point that, unfortunately, is not the solution of the calculation. In finite cases, however, our way of treating general linear constraints directly does preserve all the convergence properties of the standard form of Karmarkar's algorithm.

November, 1989

[1] Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge CB3 9EW, England.

1. Introduction

A linear programming problem is the minimization of a linear function of real variables subject to linear constraints on the values of the variables, which may include equality conditions as well as inequalities. We let n be the number of variables, $x \in \mathcal{R}^n$ be the vector of variables, $c^T x$ be the objective function where c is a constant vector in $\mathcal{R}^n$, and $\mathcal{S}$ be the set of points in $\mathcal{R}^n$ that satisfy the linear constraints. We assume that $\mathcal{S}$ is nonempty and bounded, and, due to linearity, it is a convex polytope. We define a vertex to be a point of $\mathcal{S}$ that is on the boundaries of n linearly independent constraints and an edge to be a straight line segment in $\mathcal{S}$ that is contained in the intersection of $n-1$ boundaries of linearly independent constraints. Thus the two end points of each edge are vertices. Thinking geometrically, it should be clear that the least value of $c^T x$ subject to $x \in \mathcal{S}$ can always be achieved at a vertex, and that sometimes there are many optimal vectors of variables.

The most widely used algorithm for solving linear programming calculations is the simplex method. The main operations of this algorithm can be viewed as moves from vertex to vertex along edges of the polytope, each edge being chosen so that the move reduces the objective function. Thus cycling does not occur, and termination is a consequence of the finiteness of the total number of vertices. Usually this algorithm is very efficient, but our geometrical interpretation shows that many vertices may have to be visited on the way to the solution even for small values of n. Therefore some iterative algorithms have been developed that adjust x by taking straight line steps within the interior of $\mathcal{S}$, and usually they avoid vertices except in the limit when convergence occurs to the required solution. Such algorithms are called "interior point methods".

We address the most successful of these methods, namely Karmarkar's algorithm. It hit the front page of the New York Times in 1984 because of the stunning improvements in efficiency over the simplex method that were claimed by the author. Many researchers found these claims unbelievable even after trying the algorithm in practice, but they were unable to refute the claims because many crucial details of the implementation of the algorithm are not given in the original paper (Karmarkar, 1984). This intriguing situation developed into the most active field of study throughout mathematical programming, and now the consensus is that the algorithm is much faster than the simplex method in many calculations when n is very large. In any case the ideas behind the algorithm are interesting and we are going to consider them.

My reason for presenting a paper on this subject is unusual. Instead of describing some of my own research, I agreed to speak about Karmarkar's algorithm at the IMA Silver Jubilee Conference in order to force myself to study this popular subject. After reading about one per cent of all relevant publications in order to grasp the main ideas, I decided to relate these ideas to my knowledge of nonlinear programming instead of perusing more papers, not having enough time for both of these activities, and now I agree with Gill, Murray, Saunders, Tomlin and Wright (1986) that the relations to barrier function methods are of fundamental importance. Thus we derive a version of Karmarkar's algorithm that handles general inequality constraints directly, which is not taken from the papers that I read. Applying this version to a semi-infinite programming problem in only two variables, we find that the algorithm is not always efficient when there are very many constraints.

The paper that was most helpful to my studies is a report by Gonzaga (1988), which he kindly provided when I requested some information on an excellent talk that he presented at the 1989 SIAM meeting on Optimization. It explains very well the geometric properties of Karmarkar's algorithm. A brief introduction to these properties is given by Strang (1987), who emphasises that the projective transformations of the algorithm provide "room to move" when one seeks changes to the variables that reduce the objective function. I learnt several useful technical details from Todd and Burrell (1986) and from Gill *et al* (1986) which are mentioned later. The other papers that I read are of less relevance to the material that follows, but certainly I would have been helped greatly by the work of Tomlin (1987) and Shanno (1988) if I had chosen to discuss numerical comparisons to the simplex method and modified projections that are easy to compute.

A basic form of Karmarkar's algorithm is described in Section 2, and we note the role of the potential function and the "restrictive assumption" that the optimal value of the objective function is zero. Section 3 presents the convergence theorem that delights experts in complexity theory because it shows that linear programming problems can be solved in polynomial time. Further, it is explained that the "restrictive assumption" can be replaced by a lower bound on c^Tx. Section 4 describes the "big simplification", if one is a nonlinear programmer, because here the projections of each iteration are replaced by algebra that corresponds implicitly to the projections. Extensions of the algorithm to general linear inequality constraints are studied in Section 5. Here we find that there is no need to express a general linear programming problem in standard form before applying Karmarkar's algorithm, which often avoids large increases in the number of variables. This technique is demonstrated in Section 6 by the example that has been mentioned that has an infinite number of constraints. Finally, Section 7 includes a discussion of the given procedures and numerical results.

2. The basic algorithm

Throughout Sections 2–4, the feasible region $\mathcal{S}$ is the set

$$\mathcal{S} = \{x \mid x \geq 0,\ A^T x = 0,\ a_0^T x = 1\}, \tag{2.1}$$

where A is a given $n \times m$ matrix with linearly independent columns and a_0 is a given vector in $\mathcal{R}^n$. Because there are only n inequality constraints, this form is not very suitable for the geometric interpretation of Section 1, but in fact expression (2.1) does not lose generality in theory. Indeed, any general inequality constraint, $w_i^T x \geq b_i$ say, can be expressed as the equation $\tilde{x}_i = w_i^T x - b_i$ and the simple bound $\tilde{x}_i \geq 0$, where $\tilde{x}_i$ is a new variable of the calculation, so the only inequalities are nonnegativity conditions on the components of an augmented x. Further, we can ensure that at most one constraint has a nonzero right hand side by forming linear combinations of equality constraints if necessary. Alternatively, the conditions $W^T x = b$, for example, can be replaced by $W^T x - \beta b = 0$ and $\beta = 1$, where β is a new nonnegative variable, so all the constraints become homogeneous in the new vector of variables except for $\beta = 1$.

It has been mentioned already that we require $\mathcal{S}$ to be nonempty and bounded, and that in this case the linear programming problem has at least one solution, x^* say. Strengthening this assumption, we suppose that a vector $x^{(1)}$ is available that is in $\mathcal{S}$, that has strictly positive components, and that is not optimal, which implies the inequality

$$c^T x^{(1)} > c^T x^*. \tag{2.2}$$

The notation $x^{(1)}$ is used, because Karmarkar's algorithm requires such a point in order to begin an iterative procedure that calculates $x^{(k+1)}$ from $x^{(k)}$ for $k = 1, 2, 3, \ldots$. Our assumptions provide the following fundamental properties of the linear programming problem.

Lemma 1 Let $\bar{\mathcal{S}}_0$ be the closed set

$$\bar{\mathcal{S}}_0 = \{x \mid x \geq 0,\ A^T x = 0\}. \tag{2.3}$$

Every element of $\bar{\mathcal{S}}_0$ satisfies the inequality

$$a_0^T x \geq \|x\|_2 / M, \quad x \in \bar{\mathcal{S}}_0, \tag{2.4}$$

where M is the constant

$$M = \max\{\|x\|_2 \mid x \in \mathcal{S}\}. \tag{2.5}$$

Moreover, every solution x^* has at least one and at most n–1 zero components.

Proof Because inequality (2.4) is trivial when $x=0$, we let x be a nonzero element of $\bar{\mathcal{S}}_0$. The value of $a_0^T x$ is nonzero, because if we had $a_0^T x = 0$ then we would also have $(x^{(1)}+\lambda x) \in \mathcal{S}$ for all $\lambda \geq 0$, which would contradict the boundedness of $\mathcal{S}$. Further, $a_0^T x$ is positive if x is a multiple of $x^{(1)}$ because $a_0^T x^{(1)}=1$ and both x and $x^{(1)}$ have no negative components. Otherwise, if $a_0^T x$ were negative, then the vector $v = x-(a_0^T x)\, x^{(1)}$ would be a nonzero element of $\bar{\mathcal{S}}_0$ that satisfied $a_0^T v = 0$, which is the previous contradiction. Therefore $a_0^T x$ is positive, and, since inequality (2.4) is homogeneous in x, we can restrict attention to the vectors of $\bar{\mathcal{S}}_0$ that are also in $\mathcal{S}$. In this case the bound (2.4) is a consequence of $a_0^T x=1$ and the definition (2.5), which establishes the first half of the lemma. The second half is true because, if x^* had no zero components, then the point $v = x^*+\lambda(x^{(1)}-x^*)$ would be in $\mathcal{S}$ for some negative values of λ, giving the contradiction $c^T v < c^T x^*$, while the condition $a_0^T x^* = 1$ does not allow all the components of x^* to be zero. □

We now consider the procedure that calculates $x^{(k+1)}$ from $x^{(k)}$ when $x^{(k)}>0$ and $x^{(k)} \in \mathcal{S}$. Several authors, in particular Strang (1987), motivate the procedure by taking the view that, if some of the components of $x^{(k)}$ are very small, then the corresponding nonnegativity conditions cramp the choice of $x^{(k+1)}$, so it becomes difficult to achieve a substantial reduction in the objective function. Therefore we seek a change of variables, $\hat{x}=T(x)$ from x-space to $\hat{x}$-space say, that maps $x^{(k)}$ and $\mathcal{S}$ into $\hat{x}^{(k)}$ and $\hat{\mathcal{S}}^{(k)}$ respectively, such that $\hat{x}^{(k)}$ is not close to the boundaries of the inequality constraints of $\hat{\mathcal{S}}^{(k)}$. We pick a search direction $\hat{d}^{(k)}$ in $\hat{\mathcal{S}}^{(k)}$, and move along it from $\hat{x}^{(k)}$ to $\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)}$, where $\alpha^{(k)}$ is the step-length of the line search. The new point $x^{(k+1)}$ is obtained by applying an inverse of the transformation to $\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)}$.

Of course T depends on $x^{(k)}$, and at first sight the transformation is nonlinear, because we require "room to move" about $\hat{x}^{(k)}$ without $\hat{\mathcal{S}}^{(k)}$ being a severe distortion of $\mathcal{S}$ when some of the components of $x^{(k)}$ are very small. We let $X^{(k)}$ be the $n \times n$ diagonal matrix whose diagonal elements are the (positive) components of $x^{(k)}$, and for each k the transformation is the formula

$$\hat{x} = T(x) = n\, X^{(k)-1} x / (e^T X^{(k)-1} x), \quad x \in \mathcal{S}, \tag{2.6}$$

where e is the vector in $\mathcal{R}^n$ whose components are all one. The denominator is positive because every x in $\mathcal{S}$ has no negative and at least one positive components. Thus the inequality constraints $x \geq 0$ become $\hat{x} \geq 0$, and $x^{(k)}$ is mapped into $\hat{x}^{(k)} = e$, which is well away from the inequality constraint boundaries as required. Further, the equality conditions $A^T x=0$ become the equations $\hat{A}^{(k)T}\hat{x}=0$, where $\hat{A}^{(k)}$ is the matrix

$$\hat{A}^{(k)} = X^{(k)} A, \tag{2.7}$$

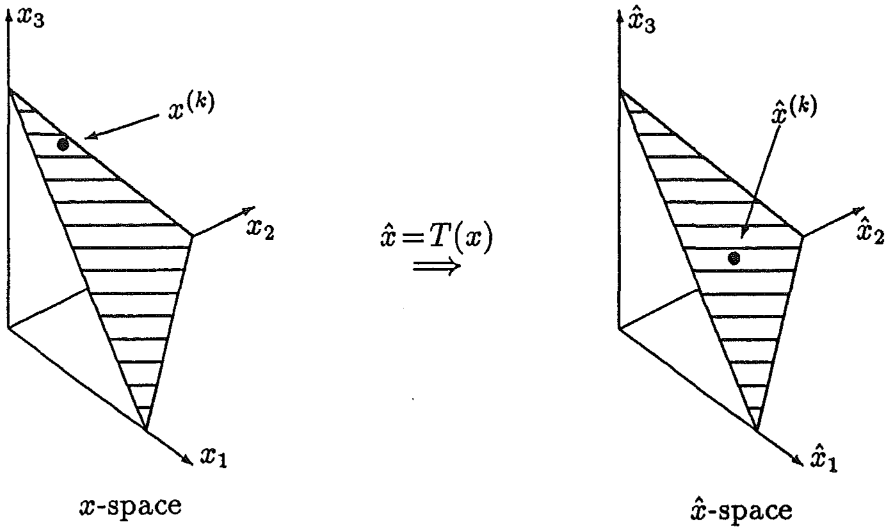

Figure 1: A transformation in three dimensions

which depends on the symmetry of $X^{(k)}$. Further, the inhomogeneous condition $a_0^Tx=1$ is irrelevant to $\hat{x}$, because the transformation (2.6) is independent of the scaling of x, but we see that $\hat{x}$ satisfies $e^T\hat{x}=n$. Therefore $\hat{\mathcal{S}}^{(k)}$ is the set

$$\hat{\mathcal{S}}^{(k)} = \{\hat{x} \mid \hat{x}\geq 0,\ \hat{A}^{(k)T}\hat{x}=0,\ e^T\hat{x}=n\}. \tag{2.8}$$

We let the inverse transformation be the formula

$$x = T^{-1}(\hat{x}) = X^{(k)}\hat{x}/a_0^TX^{(k)}\hat{x}, \quad \hat{x}\in\hat{\mathcal{S}}^{(k)}, \tag{2.9}$$

the denominator being positive because $X^{(k)}\hat{x}$ is a nonzero vector in the set (2.3) for all $\hat{x}^{(k)}\in\hat{\mathcal{S}}^{(k)}$. This denominator gives the condition $a_0^Tx=1$. Thus the transformation and its inverse provide a one-to-one correspondence between $\mathcal{S}$ and $\hat{\mathcal{S}}^{(k)}$. In particular, if the only equality constraint is the inhomogeneous equation

$$x_1+x_2+\cdots+x_n = n, \tag{2.10}$$

then $a_0=e/n$ and $\hat{\mathcal{S}}^{(k)}$ is the same as $\mathcal{S}$ for all k. Figure 1 shows this transformation when $n=3$, the feasible regions being shaded.

We stated that "at first sight the transformation is nonlinear" because the denominators that cause the nonlinearities can be deleted. Specifically, we can take the view that $\mathcal{S}$ is not the set of points (2.1), but instead each element of $\mathcal{S}$ is a straight line in the positive orthant of $\mathcal{R}^n$ that begins at the origin. Specifically, we associate with every $x\in\mathcal{S}$ the half-line $\{\lambda x \mid \lambda\geq 0\}$. Thus the homogeneous constraint $a_0^T x = 1$ of expression (2.1) becomes irrelevant, which is appropriate to the homogeneity of the transformation (2.6). Further, we can forget the denominator of formula (2.9), because now only the direction and not the magnitude of x is important. Thus, for each k, there is now a one-to-one correspondence between the half-lines from the origin in the set (2.3) and the points of $\hat{\mathcal{S}}^{(k)}$. In practice the vectors $\{x^{(k)} \mid k=1,2,3,\ldots\}$ that occur in the calculation are nonzero points on the appropriate half-lines that need not satisfy the normalization condition $a_0^T x^{(k)} = 1$, but of course the vector of variables that is returned to the user of the algorithm is scaled so that it is in $\mathcal{S}$. Lemma 1 shows that this scaling can always be done.

The technique that admits the objective function $\{c^T x \mid x \in \mathcal{R}^n\}$ into this structure is an essential ingredient of Karmarkar's algorithm. A "potential function" $\{V(x) \mid x \in \mathcal{S}_0\}$ is employed, where $\mathcal{S}_0$ is the set

$$\mathcal{S}_0 = \{x \mid x > 0,\ A^T x = 0\}, \tag{2.11}$$

which includes all the points $\{x^{(k)} \mid x=1,2,3,\ldots\}$ that have been mentioned. We require V to tend to its least value when the sequence $\{x^{(k)} \mid k=1,2,3,\ldots\}$ tends to a solution x^*, because every iteration provides the reduction

$$V(x^{(k+1)}) < V(x^{(k)}), \quad k=1,2,3,\ldots\ . \tag{2.12}$$

We also require the half-line structure to be preserved, which demands that $V(\lambda x)$ be independent of λ for all $x \in \mathcal{S}_0$, where λ is any positive number. Karmarkar's solution is to include the term $\log c^T x$ in $V(x)$ and to provide homogeneity by some balancing $\log x_i$ terms, where x_i is still the i-th component of x. Specifically, the potential function has the value

$$V(x) = n \log c^T x - \sum_{i=1}^{n} \log x_i, \quad x \in \mathcal{S}_0, \tag{2.13}$$

which is constant on each half-line.

A limitation of this choice is that $c^T x < 0$ is forbidden, so the solution x^* must satisfy $c^T x^* \geq 0$. Moreover, if $c^T x^*$ were strictly positive, then, because Lemma 1 shows that x^* has a zero component, the sequence $\{V(x^{(k)}) \mid k=1,2,3,\ldots\}$ would diverge to $+\infty$ if $x^{(k)} \to x^*$, which would not allow the decreases (2.12) in the potential function. Therefore we have to make the "restrictive assumption"

$$c^T x^* = 0, \tag{2.14}$$

although it is embarassing to need to know the optimal value of the objective function in advance. We will find in the next section that suitable adjustments to the procedure can be made automatically if this assumption is incorrect, but for the time being we impose the condition (2.14). It provides the following fundamental properties of the potential function.

Lemma 2 The given assumptions imply that the function (2.13) is well-defined but not bounded below. If $\{x^{(k)} \mid k=1,2,3,\ldots\}$ is a sequence of points in $\mathcal{S}_0$ that satisfies condition (2.12), and if the values of the potential function $\{V(x^{(k)}) \mid k=1,2,3,\ldots\}$ tend to $-\infty$, then $\{x^{(k)}/a_0^T x^{(k)} \mid k=1,2,3,\ldots\}$ is a well-defined sequence in $\mathcal{S}$ whose limit points are solutions of the linear programming problem.

Proof Let x be any point in $\mathcal{S}_0$. Lemma 1 and the definition (2.11) imply that $x/a_0^T x$ is a well-defined point of $\mathcal{S}$ that has positive components. Therefore, by Lemma 1 again, it is not a solution of the linear programming problem, so we have $c^T(x/a_0^T x) > c^T x^* = 0$. Hence $c^T x$ is positive, making the value (2.13) well-defined.

Let $x^{(1)}$ and x^* be a point of $\mathcal{S}_0$ and a solution of the linear programming problem respectively, and let $\mathcal{I}^*$ be the set $\{i \mid x_i^* = 0\}$. Therefore $|\mathcal{I}^*|$, which is the number of zero components of x^*, satisfies $|\mathcal{I}^*| < n$. We consider the potential function on the line segment whose end points are x^* and $x^{(1)}$. The "restrictive assumption" and the definition of $\mathcal{I}^*$ imply that, for every $0 < \theta \leq 1$, we have the relation

$$\begin{aligned} V(x^* + \theta\,[x^{(1)} - x^*]) \;&\leq\; n\{\log\theta + \log(c^T x^{(1)})\} - \sum_{i\in\mathcal{I}^*}\{\log\theta + \log x_i^{(1)}\} \\ &\qquad - \sum_{i\notin\mathcal{I}^*}\log(\min[x_i^*, x_i^{(1)}]), \end{aligned} \tag{2.15}$$

which is an inequality rather than an equation only because of the final sum. Since the dependence on θ is contained in the term $(n - |\mathcal{I}^*|)\log\theta$, the limit $\theta \to 0$ establishes that V is not bounded below.

Lemma 1 shows that the sequence $\{x^{(k)}/a_0^T x^{(k)} \mid k=1,2,3,\ldots\}$, given in the statement of Lemma 2, is well-defined, and it is bounded because it is contained in $\mathcal{S}$. Further, we are told that the numbers $\{V(x^{(k)}/a_0^T x^{(k)}) = V(x^{(k)}) \mid k=1,2,3,\ldots\}$ tend to $-\infty$. In view of the definition (2.13), it follows that $c^T x^{(k)}/a_0^T x^{(k)}$ tends to zero. Therefore, by continuity, all limit points of $\{x^{(k)}/a_0^T x^{(k)} \mid k=1,2,3,\ldots\}$ are points of $\mathcal{S}$ at which the objective function is zero. Hence the restrictive assumption implies that these limit points are solutions of the linear programming problem as required. □

We now address the details of the iteration that calculates $x^{(k+1)}$ from $x^{(k)}$ when $x^{(k)} \in \mathcal{S}_0$. We make the change of variables (2.6), in order that we have room to move about the point $\hat{x}^{(k)} = e = T(x^{(k)})$, so we require the analogue of the potential function in the subset $\hat{\mathcal{S}}^{(k)}$ of $\hat{x}$-space. Specifically, we define $\hat{V}(\hat{x}) = V(x)$ if $\hat{x} = T(x)$, which is a complete definition because of the one-to-one properties of the transformation. Indeed, because equation (2.6) gives

$\hat{x}=T(X^{(k)}\hat{x})$ when $\hat{x}\in\hat{\mathcal{S}}^{(k)}$, expression (2.13) implies the value

$$\hat{V}(\hat{x}) = V(X^{(k)}\hat{x}) = n\,\log \hat{c}^{(k)T}\hat{x} - \sum_{i=1}^{n}\log \hat{x}_i - \sum_{i=1}^{n}\log x_i^{(k)}, \quad \hat{x}\in\hat{\mathcal{S}}^{(k)}, \tag{2.16}$$

where $\hat{c}^{(k)}$ is the vector

$$\hat{c}^{(k)} = X^{(k)}c. \tag{2.17}$$

The polynomial time result, given in the next section, depends strongly on the fact that, at $\hat{x}=\hat{x}^{(k)}=e$, the second derivative matrix of the term $\{-\sum_i \log \hat{x}_i \mid \hat{x}\in\mathcal{R}^n\}$ is the unit matrix. This property makes projected steepest descent methods efficient in $\hat{x}$-space. Therefore the search direction $\hat{d}^{(k)}$ is defined by projecting the negative gradient

$$-\nabla\hat{V}(e) = -\frac{n}{\hat{c}^{(k)T}e}\hat{c}^{(k)} + e \tag{2.18}$$

into the set $\{d \mid \hat{A}^{(k)T}d=0,\ e^Td=0\}$. Thus the vector $\hat{x}^{(k)}+\alpha\hat{d}^{(k)}$ remains in $\hat{\mathcal{S}}^{(k)}$ when α is sufficiently small and positive. Forgetting the constraint $e^Td=0$ for the moment, but remembering that the columns of A are linearly independent, this procedure gives the vector

$$\hat{d}^{(k)} = [I - \hat{A}^{(k)}(\hat{A}^{(k)T}\hat{A}^{(k)})^{-1}\hat{A}^{(k)T}]\,(e - \frac{n}{\hat{c}^{(k)T}e}\hat{c}^{(k)}). \tag{2.19}$$

We were forgetful because we see that the identity

$$\hat{A}^{(k)T}e = A^TX^{(k)}e = A^Tx^{(k)} = 0 \tag{2.20}$$

implies $e^T\hat{d}^{(k)}=0$. Therefore there is no need to include the condition $e^Td=0$ explicitly in the projection that defines the search direction.

This definition provides the downhill condition

$$\hat{d}^{(k)T}\nabla\hat{V}(\hat{x}^{(k)}) \le 0, \tag{2.21}$$

and it will be shown in the next section that the inequality is strict. Therefore the reduction

$$\hat{V}(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)}) < \hat{V}(\hat{x}^{(k)}) \tag{2.22}$$

can be achieved by choosing $\alpha^{(k)}>0$. The advantage of the room to move is that the feasibility condition $(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)})\in\hat{\mathcal{S}}^{(k)}$ does not impose a *small* upper bound on the step-length. We let $\bar{\alpha}^{(k)}$ be this bound, and we let $\alpha^{(k)}$ minimize the new value of the potential function $\{\hat{V}(\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)}) \mid 0\le\alpha\le\bar{\alpha}^{(k)}\}$, which defines the step-length uniquely (Todd and Burrell, 1986). As mentioned already, the new vector of variables $x^{(k+1)}$ is set to any positive multiple of $X^{(k)}(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)})$, which completes the work of the iteration. The reduction (2.22) is equivalent to condition (2.12), because the initial definition of $\hat{V}$

implies that $V(x^{(k+1)})$ and $V(x^{(k)})$ have the values $\hat{V}(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)})$ and $\hat{V}(\hat{x}^{(k)})$ respectively.

In view of Lemma 2, we expect the sequence $\{V(x^{(k)}) \mid k=1,2,3,\ldots\}$ to tend to $-\infty$, which will be proved in Section 3. Therefore the iterations are terminated when the inequality

$$V(x^{(k)}) \leq -Ln \tag{2.23}$$

is obtained, where L is a prescribed constant. This condition and the definitions (2.5) and (2.13) imply the bound

$$n\,\log c^T\tilde{x}^{(k)} - n\,\log M \leq -Ln, \tag{2.24}$$

where $\tilde{x}^{(k)}$ is the vector $x^{(k)}/a_0^Tx^{(k)}$. Thus the final value of the objective function, namely $c^T\tilde{x}^{(k)}$, is at most Me^{-L}, which provides guidance on the choice of L.

When condition (2.23) does not hold, we require the components of $x^{(k)}$ to be positive, in order to calculate $x^{(k+1)}$. The initial vector satisfies $x^{(1)}>0$ by hypothesis, and $x^{(k+1)}$ inherits strict positivity from $x^{(k)}$ if $\alpha^{(k)} < \bar{\alpha}^{(k)}$, where $\bar{\alpha}^{(k)}$ is defined soon after expression (2.22). The alternative case $\alpha^{(k)} = \bar{\alpha}^{(k)}$ is very unusual, because then the $-\sum_i \log\hat{x}_i$ term of the potential function (2.16) blows up, so the reduction (2.22) implies that $\hat{c}^{(k)T}(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)})$ is zero, although all points of $\hat{\mathcal{S}}^{(k)}$, in particular $\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)}$, are nonzero. It follows by analogy with inequality (2.15) that the new value of the potential function is $V(x^{(k+1)})=\lim_{\alpha\to\alpha_k}\hat{V}(\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)})=-\infty$, which will satisfy the termination condition (2.23) when k is increased by one, the vector $x^{(k+1)}/a_0^Tx^{(k+1)}$ being a solution of the linear programming problem. Therefore we have $x^{(k)}>0$ at the beginning of every iteration that has to revise the variables.

3. Convergence properties and the restrictive assumption

The good convergence properties of Karmarkar's algorithm are due to the fact that the reduction in the potential function on every iteration is bounded away from zero. Several papers include proofs of this statement and we will do so too, in order to emphasise the success of the transformation (2.6) in providing room to move in $\hat{x}$-space, and in order to present a lemma that is the basis of the technique that removes the "restrictive assumption" that was made in Section 2. The key to the main convergence theorem is the question: when are we sure that the step from $\hat{x}^{(k)}$ along the search direction $\hat{d}^{(k)}$ can make a substantial reduction in $\hat{V}$? The following three conditions are sufficient for a favourable answer: (a) that the initial directional derivative $\hat{d}^{(k)T}\nabla\hat{V}(\hat{x}^{(k)})/\|\hat{d}^{(k)}\|$ is negative and bounded away from zero, (b) that there is an upper bound on the curvature of $\hat{V}$ that ensures that the directional derivative remains negative for a substantial distance along the search direction, and (c) that the step-length is not restricted severely by the boundary of an inequality constraint. So far we have given most attention to condition (c), noting that $\hat{x}^{(k)}=e$ and that the inequality constraints are the simple bounds $\hat{x}\geq 0$. Thus, because the definition (2.19) takes account of the equality constraints, feasibility is preserved if we impose the bound

$$\|\alpha\,\hat{d}^{(k)}\|_2 \leq \tfrac{1}{2} \tag{3.1}$$

on the change of variables in $\hat{x}$-space. The reason for the choice of right hand side will become clear.

One comment of Section 2 is highly relevant to condition (b), namely that at $\hat{x}=\hat{x}^{(k)}=e$ the second derivative matrix of the term $\{-\sum_i \log \hat{x}_i \mid \hat{x}\in\mathcal{R}^n\}$ of the potential function (2.16) is just the unit matrix. Further, for general $\hat{x}$, the second derivative matrix of this function is the expression

$$\nabla^2\hat{V}(\hat{x}) = -\frac{n}{(\hat{c}^{(k)T}\hat{x})^2}\,\hat{c}^{(k)}\hat{c}^{(k)T} + \operatorname{diag}(1/\hat{x}_i^2), \tag{3.2}$$

the last term being the $n\times n$ diagonal matrix whose diagonal elements have the values $\{1/\hat{x}_i^2 \mid i=1,2,\ldots,n\}$. Thus we have the upper bound

$$\hat{d}^{(k)T}\nabla^2\hat{V}(\hat{x})\hat{d}^{(k)} \leq 4\,\|\hat{d}^{(k)}\|_2^2 \tag{3.3}$$

when $\hat{x}$ satisfies $\{\hat{x}_i\geq\frac{1}{2} \mid i=1,2,\ldots,n\}$. We make use of this relation at the points $\{\hat{x}=\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)} \mid \alpha\geq 0\}$, whose components are all at least $\frac{1}{2}$ when inequality (3.1) is obtained.

In order to show that condition (a) also holds, we deduce the following conclusion from the restrictive assumption (2.14), using the notation

$$P^{(k)} = I - \hat{A}^{(k)}(\hat{A}^{(k)T}\hat{A}^{(k)})^{-1}\hat{A}^{(k)T} \tag{3.4}$$

for the symmetric projection matrix that occurs in the definition (2.19) of the search direction.

Lemma 3 At least one component of $P^{(k)}\hat{c}^{(k)}$ is nonpositive, where $\hat{c}^{(k)}$ is the vector (2.17).

Proof Let x^* be a solution of the linear programming problem and, as in Section 2, let $X^{(k)}$ be the diagonal matrix whose diagonal elements are the positive numbers $\{x_i^{(k)} \mid i=1,2,\ldots,n\}$. The constraint $A^Tx^*=0$ and the definitions (2.7) and (3.4) imply $X^{(k)}P^{(k)}X^{(k)-1}x^*=x^*$. Thus we can write the final value of the objective function in the form

$$c^Tx^* = c^TX^{(k)}P^{(k)}X^{(k)-1}x^* = (P^{(k)}\hat{c}^{(k)})^T(X^{(k)-1}x^*). \tag{3.5}$$

Now this expression is zero by the restrictive assumption, and, remembering $x^* \geq 0$ and the last statement of Lemma 1, the vector $X^{(k)-1}x^*$ has no negative and at least one positive components. Therefore we would have a contradiction if all the components of $P^{(k)}\hat{c}^{(k)}$ were positive, so the lemma is true.

Equations (2.19) and (2.20) imply the value

$$\hat{d}^{(k)} = e - \frac{n}{\hat{c}^{(k)T}e} P^{(k)}\hat{c}^{(k)}, \tag{3.6}$$

and the denominator $\hat{c}^{(k)T}e = \hat{c}^{(k)T}X^{(k)-1}x^{(k)} = c^Tx^{(k)}$ is positive. Hence we deduce from Lemma 3 that not every component of $\hat{d}^{(k)}$ is less than one, which provides the bound

$$\|\hat{d}^{(k)}\|_2 \geq 1. \tag{3.7}$$

It follows from the projection technique that defines $\hat{d}^{(k)}$ that the initial directional derivative of the line search satisfies the inequality

$$\begin{aligned} \hat{d}^{(k)T}\nabla\hat{V}(\hat{x}^{(k)}) \,/\, \|\hat{d}^{(k)}\|_2 &= -[P^{(k)}\nabla\hat{V}(\hat{x}^{(k)})]^T\nabla\hat{V}(\hat{x}^{(k)}) \,/\, \|\hat{d}^{(k)}\|_2 \\ &= -\|P^{(k)}\nabla\hat{V}(\hat{x}^{(k)})\|_2^2 \,/\, \|\hat{d}^{(k)}\|_2 \\ &= -\|\hat{d}^{(k)}\|_2 \leq -1. \end{aligned} \tag{3.8}$$

Conditions (a), (b) and (c) that are mentioned in the opening paragraph of this section are expressions (3.8), (3.3) and (3.1) respectively. Therefore it is now straightforward to establish the main convergence property of Karmarkar's algorithm.

Theorem 4 The given assumptions imply that the reduction $V(x^{(k)}) - V(x^{(k+1)})$ is bounded away from zero on every iteration that calculates $x^{(k+1)}$ from $x^{(k)}$.

Proof Let $\{\phi(\alpha) \mid \alpha \geq 0\}$ and $\bar{\alpha}$ denote the values of the potential function $\{\hat{V}(\hat{x}^{(k)} + \alpha\,\hat{d}^{(k)}) \mid \alpha \geq 0\}$ and $(2\,\|\hat{d}^{(k)}\|_2)^{-1}$ respectively. From the definition of $\alpha^{(k)}$, the Taylor series with explicit remainder and the three conditions that have just been mentioned, we deduce the relation

$$\begin{aligned} \hat{V}(\hat{x}^{(k)} + \alpha^{(k)}\hat{d}^{(k)}) &\leq \min_{0\leq\alpha\leq\bar{\alpha}} \phi(\alpha) \\ &= \min_{0\leq\alpha\leq\bar{\alpha}} [\,\phi(0) + \alpha\,\phi'(0) + \int_0^\alpha (\alpha-\theta)\,\phi''(\theta)\,d\theta\,] \\ &= \min_{0\leq\alpha\leq\bar{\alpha}} [\,\hat{V}(\hat{x}^{(k)}) + \alpha\,\hat{d}^{(k)T}\nabla\hat{V}(\hat{x}^{(k)}) \\ &\qquad + \int_0^\alpha (\alpha-\theta)\,\hat{d}^{(k)T}\nabla^2\hat{V}(\hat{x}^{(k)} + \theta\,\hat{d}^{(k)})\,\hat{d}^{(k)}\,d\theta\,] \\ &\leq \min_{0\leq\alpha\leq\bar{\alpha}} [\,\hat{V}(\hat{x}^{(k)}) - \alpha\,\|\hat{d}^{(k)}\|_2 + 2\,\alpha^2\,\|\hat{d}^{(k)}\|_2^2\,] \end{aligned}$$

$$= \hat{V}(\hat{x}^{(k)}) - \tfrac{1}{8}, \tag{3.9}$$

the last line being derived from the value $\alpha = (4\,\|\hat{d}^{(k)}\|_2)^{-1} = \frac{1}{2}\bar{\alpha}$. The theorem follows from the fact that the reduction in V is the same as the reduction in $\hat{V}$. □

The well-known polynomial-time complexity property of Karmarkar's algorithm is a consequence of a corollary of this theorem, namely that the number of iterations that are needed to satisfy the termination condition (2.23) is bounded above by a constant multiple of n.

We now relax the restrictive assumption $c^T x^* = 0$ that is made in Section 2. If the optimal value of the objective function, $c^T x^* = \gamma^*$ say, is known in advance, but γ^* is nonzero, then it is sufficient to modify c before beginning the calculation that is described in Section 2. Specifically, because the definition (2.1) implies that x^* minimizes $c^T x$ subject to $x \in \mathcal{S}$ if and only if it minimizes $(c - \gamma^* a_0)^T x$ subject to $x \in \mathcal{S}$, we can replace c by $c - \gamma^* a_0$ in order that the optimal value of the new objective function is zero. Usually, however, γ^* is not available. In this case for each k the calculation of $x^{(k+1)}$ from $x^{(k)}$ depends on an estimate $\gamma^{(k)}$ of γ^*, where $\gamma^{(1)}$ is given and where the subsequent values are generated automatically so that the sequence $\{\gamma^{(k)} \mid k = 1, 2, 3, \ldots\}$ converges to γ^*. On the k-th iteration c is replaced by the vector

$$c^{(k)} = c - \gamma^{(k)} a_0, \tag{3.10}$$

which induces the new value

$$\hat{c}^{(k)} = X^{(k)} c^{(k)} \tag{3.11}$$

of expression (2.17), and corresponding changes are made to equations (2.16), (2.18) and (2.19), but the sets $\mathcal{S}$, $\bar{\mathcal{S}}_0$, $\hat{\mathcal{S}}^{(k)}$ and $\mathcal{S}_0$ and the transformations (2.6) and (2.9) are the same as before. Having made these modifications, the procedure of Section 2 generates $x^{(k+1)}$ from $x^{(k)}$, unless it is detected that $\gamma^{(k)}$ should be revised.

Of course $\gamma^{(k)}$ is unacceptable if a negative value of $\hat{c}^{(k)T}\hat{x}$ occurs during the line search from $\hat{x}^{(k)} = e$ along the direction $\hat{d}^{(k)}$, because then the potential function (2.16) is not properly defined. We see that equations (3.10) and (3.11) give the identity

$$\hat{c}^{(k)T}\hat{x} = (c - \gamma^{(k)} a_0)^T (X^{(k)}\hat{x}). \tag{3.12}$$

Moreover, $\hat{x} \in \hat{\mathcal{S}}^{(k)}$ implies $X^{(k)}\hat{x} \in \bar{\mathcal{S}}_0$ and then Lemma 1 shows that $a_0^T X^{(k)}\hat{x}$ is positive, $\hat{x}$ being nonzero because it is in $\hat{\mathcal{S}}^{(k)}$. Therefore we can avoid the negative value of $\hat{c}^{(k)T}\hat{x}$ by reducing $\gamma^{(k)}$. We recommend being generous when decreasing this parameter, and perhaps reductions should be made even when

$\hat{c}^{(k)T}\hat{x}$ remains positive throughout the line search, because there is a highly suitable way of making the sequence $\{\gamma^{(k)} \mid k = j, j+1, j+2, \ldots\}$ increase monotonically to γ^*, where j is any integer such that $\gamma^{(j)} \leq \gamma^*$ (Todd and Burrell, 1986). This technique depends on the following lemma. Further, the lemma sometimes provides a useful lower bound on the new value of $\gamma^{(k)}$ when $\gamma^{(k)}$ is decreased.

Lemma 5 Let $x^{(k)}$ be any vector in $\mathcal{S}_0$, let $\hat{A}^{(k)}$ and $P^{(k)}$ be the matrices (2.7) and (3.4), where as usual $X^{(k)} = \mathrm{diag}\,(x_i^{(k)})$, and let γ^* be the least value of the original objective function $\{c^T x \mid x \in \mathcal{S}\}$. Then, if all components of the vector $P^{(k)}\hat{c}^{(k)}$ are positive, the strict inequality $\gamma^{(k)} < \gamma^*$ is satisfied.

Proof Let x^* solve the linear programming problem and suppose that, as in the statement of the lemma, we have $P^{(k)}\hat{c}^{(k)} > 0$. Since x^* is nonzero and nonnegative and since $x^{(k)} \in \mathcal{S}_0$ implies $x^{(k)} > 0$, these conditions give the strict inequality $\hat{c}^{(k)T} P^{(k)} X^{(k)-1} x^* > 0$. Moreover, as in the proof of Lemma 3, the equation $X^{(k)} P^{(k)} X^{(k)-1} x^* = x^*$ holds. Therefore the condition $\hat{c}^{(k)T} X^{(k)-1} x^* > 0$ is satisfied, and, remembering expressions (3.10) and (3.11), we write it in the form $(c - \gamma^{(k)} a_0)^T x^* > 0$. The lemma now follows from the definition $\gamma^* = c^T x^*$ and the constraint $a_0^T x^* = 1$. □

The lemma suggests the useful procedure that will be described in the next paragraph. It requires a lower bound $\gamma^{(1)}$ on the optimal value of the objective function to be given instead of the restrictive assumption $c^T x^* = 0$. For each γ in the set $\{\gamma^{(k)} \mid k = 1, 2, 3, \ldots\}$, the potential function is now the expression

$$V(x;\gamma) = n\,\log[\,(c-\gamma a_0)^T x\,] - \sum_{i=1}^{n} \log x_i, \quad x \in \mathcal{S}_0, \quad \gamma \leq \gamma^*, \tag{3.13}$$

which is still homogeneous in x. We see that the case $\gamma = \gamma^*$ corresponds to the potential function that we had before, so Lemma 2 shows that $V(x;\gamma^*)$ is well-defined for all $x \in \mathcal{S}_0$, the scalar product $(c - \gamma^* a_0)^T x$ being positive. Otherwise, when $\gamma < \gamma^*$, we apply Lemma 1 to $x \in \mathcal{S}_0 \subset \bar{\mathcal{S}}_0$ in order to deduce the condition

$$(c-\gamma a_0)^T x = (c-\gamma^* a_0)^T x + (\gamma^* - \gamma)\, a_0^T x > (c - \gamma^* a_0)^T x. \tag{3.14}$$

Thus $V(x;\gamma)$ is still well-defined. Expression (3.14) also provides the inequality

$$V(x;\gamma) > V(x;\gamma^*), \quad x \in \mathcal{S}_0, \quad \gamma < \gamma^*, \tag{3.15}$$

which will be needed later. We preserve the definition $\hat{V}(\hat{x};\gamma) = V(x;\gamma)$ when $\hat{x} = T(x)$, in order that the reductions in $\hat{V}$ that are achieved by revising the $\hat{x}$-variables are still enjoyed by V. Therefore equation (2.16) remains valid, since the dependence of $\hat{c}^{(k)}$ on $\gamma^{(k)}$ is present in the relations (3.10)–(3.11).

The k-th iteration of the revised procedure is supplied with a lower bound $\gamma^{(k)}$ on γ^*, given by the user when $k=1$ and inherited from the previous iteration when $k>1$. It is also supplied with a vector of variables $x^{(k)}$ that as before is in $\mathcal{S}_0$ or there is the remote possibility that $x^{(k)}$ has some zero components because it is the solution of the linear programming problem, which can happen only if $\gamma^{(k)}=\gamma^*$. The iterations are terminated if $V(x^{(k)};\gamma^{(k)}) \leq -Ln$, this test being no weaker than condition (2.23) in view of inequality (3.15). When termination does not occur, all components of $x^{(k)}$ are positive, we let $c^{(k)}$ and $\hat{c}^{(k)}$ have the values (3.10) and (3.11), and we define the search direction $\hat{d}^{(k)}$ by formula (2.19). If every component of $\hat{d}^{(k)}$ is less than one, then it follows from equation (3.6) and Lemma 5 that we have $\gamma^{(k)}<\gamma^*$. We seize this opportunity of being able to increase $\gamma^{(k)}$ in a way that preserves $\gamma^{(k)} \leq \gamma^*$, letting the new value of $\gamma^{(k)}$ be the least larger number such that the vector $P^{(k)}\hat{c}^{(k)} = P^{(k)}X^{(k)}(c-\gamma^{(k)}a_0)$ has a zero component. We then return to the beginning of the iteration to retry the termination condition and to recalculate the search direction if necessary. Thus $\hat{d}^{(k)}$ satisfies condition (3.7) on every iteration that changes the variables. As before, $x^{(k+1)}$ is any positive multiple of $X^{(k)}(\hat{x}^{(k)}+\alpha^{(k)}\hat{d}^{(k)})$, where the step-length $\alpha^{(k)}$ is found by a line search that minimizes the potential function. Finally, $\gamma^{(k+1)}$ is set to the current value of $\gamma^{(k)}$ and the next iteration is begun.

An important property of this technique for increasing $\gamma^{(k)}$ is that it guarantees that Lemma 3 is valid on every iteration. Therefore the deduction in inequality (3.9) and the definition of $\hat{V}$ provide the condition

$$V(x^{(k+1)};\gamma^{(k)}) < V(x^{(k)};\gamma^{(k)}) - \tfrac{1}{8}, \tag{3.16}$$

where $\gamma^{(k)}$ has the value that is chosen by the k-th iteration. Increases in $\gamma^{(k)}$ also reduce the potential function, because we have noted already that $a_0^T x$ is positive in the definition (3.13). Therefore Theorem 4 applies to the revised algorithm, giving the same polynomial-time complexity property as before. Further, if the termination condition is omitted and if none of the points $\{x^{(k)} \mid k=1,2,3,\ldots\}$ is an exact solution of the linear programming problem, then the sequence $\{V(x^{(k)};\gamma^{(k)}) \mid k=1,2,3,\ldots\}$ must decrease monotonically to $-\infty$. It follows from inequality (3.15) that the sequence $\{V(x^{(k)};\gamma^*) \mid k=1,2,3,\ldots\}$ also decreases to $-\infty$. Therefore, by Lemma 2, all limit points of $\{x^{(k)}/a_0^T x^{(k)} \mid k=1,2,3,\ldots\}$ are solutions of the linear programming problem as before. Further, the increasing sequence $\{\gamma^{(k)} \mid k=1,2,3,\ldots\}$ tends to γ^*, because, if its limit were $\bar{\gamma}<\gamma^*$, our deductions since equation (3.13) and the definition (2.5) would imply the inequality

$$\begin{aligned}\lim_{k\to\infty} V(x^{(k)};\gamma^{(k)}) &= \lim_{k\to\infty} V(x^{(k)}/a_0^T x^{(k)};\gamma^{(k)}) \\ &\geq \lim_{k\to\infty} V(x^{(k)}/a_0^T x^{(k)};\bar{\gamma})\end{aligned}$$

$$\begin{aligned} &\geq \lim_{k\to\infty} n \log\left[\,(c-\bar{\gamma}a_0)^T x^{(k)}/a_0^T x^{(k)}\,\right] - n \log M \\ &= \lim_{k\to\infty} n \log\left[\,(c-\gamma^* a_0)^T x^{(k)}/a_0^T x^{(k)} + (\gamma^*-\bar{\gamma})\,\right] - n \log M \\ &= n \log(\gamma^*-\bar{\gamma}) - n \log M, \qquad (3.17) \end{aligned}$$

which contradicts the fact that the sequence $\{V(x^{(k)};\gamma^{(k)}) \mid k=1,2,3,\ldots\}$ diverges to $-\infty$.

Thus the main convergence properties of Karmarkar's algorithm are retained by the above procedure. In the remainder of the paper, however, we revert to the basic algorithm of Section 2 in order to simplify notation, except that techniques are given in the final paragraphs of Sections 4 and 5 that extend the work of these sections to the case when only a lower bound on the optimal value of the objective function is known initially.

4. Making the transformations implicit

The explicit transformations from x-space to $\hat{x}$-space and back again, that occur in every iteration of the algorithm of Section 2, can be avoided by some algebra. It was pointed out by Gill *et al* (1986) that the resultant algorithm is a "barrier function method", due to the fact that the potential function (2.13) usually becomes infinite if one or more components of $x > 0$ tend to zero. The algebra does provide a big simplification, partly because it removes the need to distinguish between x-space and $\hat{x}$-space. Therefore this alternative form of Karmarkar's algorithm is derived in this section, in two different ways that are shown to be equivalent. Thus we identify some interesting and useful properties of the new form of the old method.

We begin by considering the choice of step-length of the line search of the algorithm of Section 2, which is defined in the paragraph that includes inequalities (2.21)–(2.22). We recall that $\alpha^{(k)}$ is the value of α that minimizes $\{\hat{V}(\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)} \mid 0\leq\alpha\leq\bar{\alpha}^{(k)}\}$, and we also recall that we can define $\hat{V}$ by the equation

$$\hat{V}(\hat{x}) = V(X^{(k)}\hat{x}), \quad \hat{x}\in\hat{\mathcal{S}}^{(k)}, \qquad (4.1)$$

as in expression (2.16). Hence, remembering the relation $x^{(k)}=X^{(k)}e=X^{(k)}\hat{x}^{(k)}$ and introducing the search direction

$$d^{(k)} = X^{(k)}\hat{d}^{(k)} \qquad (4.2)$$

in x-space, the step-length minimizes the function

$$\begin{aligned} \hat{V}(\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)}) &= V(X^{(k)}[\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)}]\,) \\ &= V(x^{(k)}+\alpha\,d^{(k)}), \quad 0\leq\alpha\leq\bar{\alpha}^{(k)}. \qquad (4.3) \end{aligned}$$

Having calculated $\alpha^{(k)}$, the procedure of Section 2 sets $x^{(k+1)}$ to any positive multiple of $X^{(k)}(\hat{x}^{(k)}+\alpha\,\hat{d}^{(k)})$, which includes the usual choice

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}. \tag{4.4}$$

Further, the bound $\bar{\alpha}^{(k)}$ is defined to be the greatest number such that all components of $\hat{x}^{(k)}+\bar{\alpha}^{(k)}\hat{d}^{(k)}$ are nonnegative, which is also the greatest number such that the components of $X^{(k)}(\hat{x}^{(k)}+\bar{\alpha}^{(k)}\hat{d}^{(k)}) = x^{(k)}+\bar{\alpha}^{(k)}d^{(k)}$ have this property, since $X^{(k)}$ is diagonal and positive definite. Therefore each iteration of Section 2 that changes the variables can be implemented in the following way. The search direction (4.2) is calculated. Then $\alpha^{(k)}$ is found by minimizing the function (4.3), the bound $\alpha \le \bar{\alpha}^{(k)}$ being the condition $x^{(k)}+\alpha\, d^{(k)} \ge 0$. Finally, $x^{(k+1)}$ is defined by formula (4.4).

We complete this description by removing the dependence of $d^{(k)}$ on the transformations. Equations (4.2), (3.6) and (2.17) and further use of $x^{(k)} = X^{(k)}e$ imply the value

$$d^{(k)} = x^{(k)} - \frac{n}{c^T x^{(k)}}\, X^{(k)} P^{(k)} X^{(k)} c, \tag{4.5}$$

while the definitions (3.4) and (2.7) yield the matrix

$$P^{(k)} = I - X^{(k)} A (A^T X^{(k)2} A)^{-1} A^T X^{(k)}, \tag{4.6}$$

which give the search direction

$$d^{(k)} = x^{(k)} - \frac{n}{c^T x^{(k)}}\, [X^{(k)2} - X^{(k)2} A (A^T X^{(k)2} A)^{-1} A^T X^{(k)2}]\, c. \tag{4.7}$$

The calculation is now in a form that has no explicit dependence on $\hat{x}$-space.

A check on our algebra is that the method of Section 2 was designed to have the property that, if $x^{(k)}$ satisfies the equality constraints $A^T x^{(k)} = 0$, then $x^{(k+1)}$ also satisfies them. In view of formula (4.4), an equivalent statement is that we expect $A^T d^{(k)}$ to be zero whenever $A^T x^{(k)} = 0$, and we see that equation (4.7) passes this test. Another check is that, because we found using equation (2.20) that $e^T \hat{d}^{(k)} = 0$, we should now have the identity

$$e^T X^{(k)-1} d^{(k)} = 0. \tag{4.8}$$

Formula (4.7) satisfies this condition because of the relations $e^T X^{(k)-1} x^{(k)} = n$, $e^T X^{(k)} c = c^T x^{(k)}$ and $e^T X^{(k)} A = x^{(k)T} A = 0$. We gain further knowledge of $d^{(k)}$ by studying a different way of eliminating the $\hat{x}$ variables from the definition (4.2).

The starting point of this different approach is recalling that $\hat{d}^{(k)}$, the search direction in $\hat{x}$-space, is generated by applying to the negative gradient vector

(2.18) the orthogonal projection operator into the null space of $\hat{A}^{(k)T}$. This construction implies that $\hat{d}^{(k)}$ is the point in the subspace $\{d \mid \hat{A}^{(k)T}d=0\}$ that is closest to $-\nabla\hat{V}(e)$, where distance is measured in the Euclidean metric. Therefore the squared distance $\|\hat{d}^{(k)}+\nabla\hat{V}(e)\|_2^2$ is made as small as possible. In other words, $\hat{d}^{(k)}$ is the solution of the quadratic programming problem

$$\left.\begin{array}{ll}\text{minimize} & d^T\nabla\hat{V}(e)+\frac{1}{2}\,\|d\|_2^2, \quad d\in\mathcal{R}^n \\ \text{subject to} & \hat{A}^{(k)T}d=0\end{array}\right\}. \tag{4.9}$$

Therefore the solution of the analogous calculation

$$\left.\begin{array}{ll}\text{minimize} & (X^{(k)-1}d)^T\nabla\hat{V}(e)+\frac{1}{2}\,\|X^{(k)-1}d\|_2^2, \quad d\in\mathcal{R}^n \\ \text{subject to} & \hat{A}^{(k)T}(X^{(k)-1}d)=0\end{array}\right\} \tag{4.10}$$

must occur when $X^{(k)-1}d=\hat{d}^{(k)}$, which is when d is the search direction (4.2) in x-space. Since the derivative of the identity (4.1) with respect to $\hat{x}$ provides the relation

$$\nabla\hat{V}(\hat{x})=X^{(k)}\nabla V(X^{(k)}\hat{x}), \tag{4.11}$$

and since $\hat{A}^{(k)}$ is the matrix (2.7), it follows that the required search direction in x-space is the solution of the quadratic programming problem

$$\left.\begin{array}{ll}\text{minimize} & d^T\nabla V(x^{(k)})+\frac{1}{2}\,d^TX^{(k)-2}d, \quad d\in\mathcal{R}^n \\ \text{subject to} & A^Td=0\end{array}\right\}. \tag{4.12}$$

Again we have a definition of $d^{(k)}$ that does not depend explicitly on the transformations of Section 2.

In order to solve this problem analytically, we let $\mu^{(k)}$ be the vector of Lagrange multipliers at its solution. Therefore $d^{(k)}$ and $\mu^{(k)}$ satisfy the linear equations

$$\left.\begin{array}{c}\nabla V(x^{(k)})+X^{(k)-2}d^{(k)}=A\mu^{(k)} \\ A^Td^{(k)}=0\end{array}\right\}. \tag{4.13}$$

Multiplication of the first equation by $A^TX^{(k)2}$, in order that $d^{(k)}$ can be eliminated by the second equation, yields the vector

$$\mu^{(k)}=(A^TX^{(k)2}A)^{-1}A^TX^{(k)2}\nabla V(x^{(k)}). \tag{4.14}$$

By substituting this expression into the first equation, we find the search direction

$$d^{(k)}=-X^{(k)2}\,[\,I-A(A^TX^{(k)2}A)^{-1}A^TX^{(k)2}\,]\,\nabla V(x^{(k)}). \tag{4.15}$$

Therefore, since the potential function (2.13) has the gradient

$$\nabla V(x^{(k)}) = \frac{n}{c^T x^{(k)}}\, c - X^{(k)-2} x^{(k)}, \tag{4.16}$$

where the last term is a convenient form of the vector with components $\{1/x_i^{(k)} \mid i=1,2,\ldots,n\}$, and since $A^T x^{(k)}=0$, we see that formulae (4.7) and (4.15) are equivalent. More importantly, the quadratic programming problem (4.12) provides a valid definition of the search direction.

We extend this point of view by considering the second derivative matrix of the potential function, which has the value

$$\nabla^2 V(x^{(k)}) = -\frac{n}{(c^T x^{(k)})^2}\, c\, c^T + X^{(k)-2}. \tag{4.17}$$

We note that the diagonal matrix $X^{(k)-2}$ also occurs in expression (4.12). Indeed, the work so far of this section has established the following theorem, which provides a description of an iteration of Karmarkar's algorithm that seems to be much clearer than the one that is developed in Section 2.

Theorem 6 Given a point $x^{(k)} \in \mathcal{S}_0$, an iteration of the algorithm first tests the termination condition (2.23). If a further improvement to the vector of variables is required, then the gradient of the potential function (2.13) and the diagonal second derivative matrix of the $-\sum_i \log x_i$ term are calculated at $x=x^{(k)}$. The search direction $d^{(k)}$ is defined to be the solution of the quadratic programming problem (4.12). Finally, the new vector of variables is given the value (4.4), where the step-length $\alpha^{(k)}$ minimizes the univariate function $\{V(x^{(k)}+\alpha\, d^{(k)}) \mid \alpha \geq 0\}$, subject to the condition $x^{(k+1)} \geq 0$. □

The relation to the "log barrier method" for constrained optimization that was noticed by Gill *et al* (1986) is as follows. We let $\overline{V}^{(k)}$ be the approximation to the potential function that is formed by replacing the $n \log c^T x$ term of the definition (2.13) by the first order Taylor series expansion

$$n \log c^T x \approx n \log c^T x^{(k)} + \frac{n}{c^T x^{(k)}}\,(x - x^{(k)})^T c. \tag{4.18}$$

Thus the gradients $\nabla V(x^{(k)})$ and $\nabla \overline{V}^{(k)}(x^{(k)})$ are equal and the second derivative matrix of expression (4.12) is $\nabla^2 \overline{V}^{(k)}(x^{(k)})$. Therefore $d^{(k)}$ is the Newton–Raphson search direction at $x^{(k)}$ of the linearly constrained nonlinear programming problem

$$\text{minimize} \quad \{\overline{V}^{(k)}(x) \mid x \in \mathcal{R}^n\} \quad \text{subject to} \quad A^T x = 0. \tag{4.19}$$

Lemma 2, however, shows that actually the algorithm solves the problem

$$\text{minimize} \quad \{V(x) \mid x \in \mathcal{R}^n\} \quad \text{subject to} \quad A^T x = 0, \tag{4.20}$$

since the $-\sum_i \log x_i$ term of V keeps x nonnegative. Perhaps, therefore, we should be considering the Newton–Raphson search direction of this calculation at $x^{(k)}$, which is normally defined to be the solution to the problem (4.12) after replacing $X^{(k)-2} = \nabla^2 \overline{V}^{(k)}(x^{(k)})$ by the matrix $\nabla^2 V(x^{(k)})$. Unfortunately this approach is unsuitable, because now the quadratic programming calculation cannot have a unique solution due to the homogeneity of V. Indeed, we suppose that $d^{(k)}$ is a solution and we consider the set of vectors $\{d = d^{(k)} + \theta x^{(k)} \mid \theta \in \mathcal{R}\}$. For every θ the constraints $A^T d = 0$ are satisfied due to $A^T x^{(k)} = 0$, and the new quadratic objective function

$$Q(d) = d^T \nabla V(x^{(k)}) + \tfrac{1}{2}\, d^T \nabla^2 V(x^{(k)})\, d, \quad d \in \mathcal{R}^n, \tag{4.21}$$

takes the value

$$Q(d^{(k)} + \theta x^{(k)}) = Q(d^{(k)}) + \theta d^{(k)T} \nabla^2 V(x^{(k)})\, x^{(k)}, \quad \theta \in \mathcal{R}, \tag{4.22}$$

because, by homogeneity, the terms $x^{(k)T} \nabla V(x^{(k)})$ and $x^{(k)T} \nabla^2 V(x^{(k)})\, x^{(k)}$ are zero, which is shown explicitly in the derivatives (4.16) and (4.17). It follows that $d^{(k)}$ is well-defined only if the function (4.22) has a unique minimum at $\theta = 0$, which is impossible as the dependence on θ is linear. Therefore the use of the function (4.19) instead of the function (4.20) when calculating the search direction by a Newton–Raphson procedure provides some stabilization that is certainly needed if one employs a general numerical method for quadratic programming.

In fact the homogeneity makes the vector $x^{(k)}$ in the definition (4.7) redundant. A good way of interpreting this remark is to take the view that is presented in the paragraph that follows equation (2.10), namely that $\{V(x) \mid x \in \mathcal{S}_0\}$ is not a function of the point x but instead is a function of the half-line $\{\lambda x \mid \lambda > 0\}$. Thus the set of points $\{x^{(k)} + \alpha\, d^{(k)} \mid \alpha \geq 0\}$ can be regarded as a set of half-lines, and the line search must find the step-length $\alpha^{(k)}$ that minimizes the constant value of the potential function on the half-line $\{\lambda\, (x^{(k)} + \alpha^{(k)} d^{(k)}) \mid \lambda > 0\}$. Thus only the directions and not the magnitudes of the vectors $\{x^{(k)} + \alpha\, d^{(k)} \mid \alpha \geq 0\}$ are important. Further, the positive line search parameter makes all half-lines in the convex hull of the half-lines $\{\lambda\, x^{(k)} \mid \lambda > 0\}$ and $\{\lambda\, d^{(k)} \mid \lambda > 0\}$ accessible. Now, if we drop the $x^{(k)}$ term from the definition (4.7), the new search direction is $d^{(k)} - x^{(k)}$, and the new line search has access to all half-lines in the convex hull of $\{\lambda\, x^{(k)} \mid \lambda > 0\}$ and $\{\lambda\, (d^{(k)} - x^{(k)}) \mid \lambda > 0\}$. Therefore, by taking the average of the new extreme half-lines, we find that the old line search over the range $0 \leq \alpha \leq \infty$ is equivalent to the new line search over the range $0 \leq \alpha \leq 1$. Further, the old range $0 \leq \alpha \leq \bar{\alpha}^{(k)}$ that occurs in equation (4.3) becomes the new range $0 \leq \alpha \leq \tilde{\alpha}^{(k)}$, where $\tilde{\alpha}^{(k)}$ is defined by the condition that $x^{(k)} + \tilde{\alpha}^{(k)}(d^{(k)} - x^{(k)})$ is a multiple of $x^{(k)}+\bar{\alpha}^{(k)}d^{(k)}$, which gives the value $\tilde{\alpha}^{(k)}=\bar{\alpha}^{(k)}/(1+\bar{\alpha}^{(k)})$. Therefore the deletion of $x^{(k)}$ from expression (4.7) would preserve the choice (4.4), apart from an unimportant scaling factor.

Some use will be made of this freedom in Section 6, but until then we retain the usual choice of search direction that we have derived in two different ways. The fact that this choice satisfies equation (4.8) provides a noteworthy property. Specifically, since we have $X^{(k)-1}e=X^{(k)-2}x^{(k)}$, condition (4.8) implies that the search direction is orthogonal to the last term of the gradient (4.16). Therefore there is no first order change to the penalty term $\{-\sum_i \log x_i \mid x\in\mathcal{S}_0\}$ of the potential function when the step-length of the line search is small. Because this term is strictly convex for all $x\in\mathcal{S}_0$, its second derivative matrix being $\text{diag}(1/x_i^2)$, it follows that every iteration of the algorithm increases the penalty term. Therefore Theorem 4 shows that not only $\{V(x)\mid x\in\mathcal{S}_0\}$ but also $\{\log c^Tx \mid x\in\mathcal{S}_0\}$ is reduced by a substantial amount on every iteration. Although this observation is interesting, it does not illuminate the progress of the variables towards the solution of the original linear programming problem, because we are not taking account of the normalization condition $a_0^Tx=1$ that is imposed by the inhomogeneous constraint. Further, it is straightforward to construct examples where the change to the variables gives $c^T(x^{(k+1)}/a_0^Tx^{(k+1)})>c^T(x^{(k)}/a_0^Tx^{(k)})$, although the reduction $c^Tx^{(k+1)}<c^Tx^{(k)}$ is guaranteed.

When the restrictive assumption is replaced by a lower bound $\gamma^{(k)}$ on the final value of the objective function, we recall from the last three paragraphs of Section 3 that we replace c by the vector (3.10) wherever it occurs. Thus formula (4.7) provides the search direction

$$d^{(k)}=x^{(k)}-\frac{n}{(c-\gamma^{(k)}a_0)^Tx^{(k)}}\,X^{(k)}P^{(k)}X^{(k)}(c-\gamma^{(k)}a_0). \tag{4.23}$$

We also recall that it is advantageous to increase $\gamma^{(k)}$ if all components of the vector

$$P^{(k)}\hat{c}^{(k)}=P^{(k)}X^{(k)}(c-\gamma^{(k)}a_0) \tag{4.24}$$

are positive, the new value being the least larger value of $\gamma^{(k)}$ that causes a component of this vector to become zero. After any increase the search direction (4.23) is recalculated. Thus at least one component of $d^{(k)}$ is no less than the corresponding component of $x^{(k)}$, which gives the bound

$$\|X^{(k)-1}d^{(k)}\|_2\geq 1, \tag{4.25}$$

corresponding to inequality (3.7). This construction provides the polynomial-time property as before. Indeed, all the material of this section is just a reformulation of techniques and results that are given or that could have been derived in Sections 2 and 3.

5. General inequality constraints

In many linear programming problems one has initially far fewer variables than constraints. For example, one may require the cubic polynomial fit to one hundred given measured values of a function of one variable, $\{(t_j, \bar{f}_j) \mid j = 1, 2, \ldots, 100\}$ say, that minimizes the maximum residual of the fit. In this case it is convenient to employ five variables, which are the four coefficients of the cubic polynomial and an upper bound on the maximum residual that is going to be made as small as possible. Thus the calculation is expressed in the linear programming form

$$\left.\begin{array}{ll} \text{minimize} & c^T x, \quad x \in \mathcal{R}^5 \\ \text{subject to} & x_1 + x_2\, t_j + x_3\, t_j^2 + x_4\, t_j^3 - x_5 \leq \bar{f}_j \\ & \leq x_1 + x_2\, t_j + x_3\, t_j^2 + x_4\, t_j^3 + x_5, \quad j = 1, 2, \ldots, 100 \end{array}\right\}, \qquad (5.1)$$

where c is the fifth coordinate vector, because we wish to minimize the value of x_5. We see that there are two hundred linear inequality constraints and no simple bounds. Therefore one might introduce some new variables in order to express the calculation in standard form, some suitable techniques being mentioned in the opening paragraph of Section 2. The standard form, however, has as many variables as inequalities, so in this case the number of variables would increase from five to at least two hundred. We are going to show, therefore, that Karmarkar's algorithm can handle general inequalities directly, avoiding the need to increase the number of variables, except that one extra variable is usually required to make all but one of the constraints homogeneous.

Our claim that the standard form of the problem (5.1) requires at least two hundred variables presumes that there is no preliminary calculation that depends on the details of the data, an extreme and unreasonable example of preliminary work being the solution of the problem itself. Assuming that the actual numerical values of the elements of the vectors and matrices that specify the problem are the concern of the main linear programming algorithm, it is not possible to say in advance that any of the inequality constraints are redundant. Therefore, as mentioned already, the number of variables of the standard form is at least the initial number of inequality constraints. Moreover, whenever the standard form demands a new variable, a new equality condition is introduced too, in order that the amount of freedom to change the variables remains the same. Hence it is usual for the equations $A^T x = 0$ in the calculation of Section 2 to have the property that it is easy to employ some of them to eliminate some of the new variables from the standard form, in order that these new variables do not occur explicitly.

Therefore we address the case when the m linearly independent constraints of the standard form $A^Tx=0$ have the structure

$$\left(\begin{array}{c|c} B^T & 0 \\ \hline -Z^T & I \end{array}\right)\left(\begin{array}{c} y \\ \hline z \end{array}\right)=0, \quad y\in\mathcal{R}^r, \quad z\in\mathcal{R}^{n-r}, \tag{5.2}$$

the components of y and z being the first r and last $n-r$ components of x respectively. Thus, because the total number of equations is still m, the dimensions of B^T and Z^T are $(m+r-n)\times r$ and $(n-r)\times r$ respectively. Further, if A does not give the structure (5.2) immediately, we postmultiply it by a nonsingular $m\times m$ matrix, U say, that provides the partitioning

$$AU=\left(\begin{array}{c|c} B & -Z \\ \hline 0 & I \end{array}\right) \tag{5.3}$$

in order that expression (5.2) is equivalent to $A^Tx=0$. Such transformations exist for any choice of the integer r from $[n{-}m,\ n{-}1]$, because of the assumption that the homogeneous equality constraints are linearly independent, but the $n-m$ variables that are expressed in terms of the remaining variables by the identity $z=Z^Ty$ cannot always be the last $n-m$ components of x. Therefore we assume that the variables are reordered if necessary.

For any $y\in\mathcal{R}^r$ we form the vector $x=x(y)\in\mathcal{R}^n$ by appending $z=Z^Ty$ to y. It follows that the problem of minimizing $\{V(x) \mid x\in\mathcal{S}_0\}$ is equivalent to minimizing the new potential function

$$W(y)=V(x(y))=n\,\log h^Ty-\sum_{i=1}^{r}\log y_i-\sum_{j=1}^{n-r}\log\left(\sum_{i=1}^{r}Z_{ij}y_i\right), \quad y\in\mathcal{Y}_0, \tag{5.4}$$

where $h\in\mathcal{R}^r$ is defined by the identity $h^Ty=c^Tx(y)$, which gives the components

$$h_i=c_i+\sum_{j=1}^{n-r}Z_{ij}c_{r+j}, \quad i=1,2,\ldots,r, \tag{5.5}$$

and where $\mathcal{Y}_0$ is the set $\{y \mid x(y)\in\mathcal{S}_0\}$. Thus, by analogy with expression (2.11), $\mathcal{Y}_0$ is the set of vectors that provide positive arguments of the logarithms of the potential function (5.4), and that satisfy the relevant homogeneous equality constraints. These constraints are now just the $m+r-n$ conditions $B^Ty=0$, which are empty if $r=n-m$, because $n-r$ of the original constraints have been used to eliminate the last $n-r$ variables of the standard form of Section 2.

We go back to the standard form, however, in order to consider the search direction (4.7) that is generated in x-space. We recall that $A^Tx^{(k)}=0$ implies $A^Td^{(k)}=0$. Therefore the points on the search direction in x-space can be expressed in the form $\{x^{(k)}+\alpha\,d^{(k)}=x(y^{(k)}+\alpha\,e^{(k)}) \mid \alpha\geq 0\}$, where $x(y)$ is defined at the beginning of the previous paragraph and where $y^{(k)}$ and $e^{(k)}$ are composed of the first r components of $x^{(k)}$ and $d^{(k)}$ respectively. Therefore, in view of the definitions (2.13) and (5.4), searching along $d^{(k)}$ from $x^{(k)}$ to minimize the potential function V is equivalent to searching along $e^{(k)}$ from $y^{(k)}$ to minimize the potential function W. Further, the second search gives the vector

$$y^{(k+1)} = y^{(k)} + \alpha^{(k)} e^{(k)} \qquad (5.6)$$

in $\mathcal{R}^r$, where $\alpha^{(k)}$ is the step-length of equation (4.4), and where $y^{(k+1)} \in \mathcal{R}^r$ inherits the first r components of $x^{(k+1)}$.

We now take the view that we are working in y-space, so h, B and Z are available, the objective function is $\{h^T y \mid y \in \mathcal{R}^r\}$, the constraints are $y \geq 0$, $Z^T y \geq 0$ and $B^T y = 0$, and the potential function is expression (5.4). We let $y^{(k)} \in \mathcal{Y}_0 \subset \mathcal{R}^r$ be the initial vector of variables for an iteration of Karmarkar's algorithm. If the usual termination condition, which now has the form

$$W(y^{(k)}) \leq -Ln, \qquad (5.7)$$

is not satisfied, then we are going to calculate $y^{(k+1)}$ in the way that is suggested in the previous paragraph. Therefore we construct the vectors and matrix

$$x^{(k)} = \left(\frac{y^{(k)}}{Z^T y^{(k)}}\right), \quad c = \left(\frac{h}{0}\right) \quad \text{and} \quad A = \left(\begin{array}{c|c} B & -Z \\ \hline 0 & I \end{array}\right), \qquad (5.8)$$

in order that formula (4.7) provides a suitable search direction in x-space. As before, the components of $e^{(k)} \in \mathcal{R}^r$ are the first r components of $d^{(k)}$. Then $y^{(k+1)}$ is chosen to be the vector (5.6), where, as mentioned already, the step-length $\alpha^{(k)}$ is the value of α that minimizes the univariate function $\{W(y^{(k)} + \alpha\, e^{(k)}) \mid \alpha \geq 0\}$, subject to the condition that the arguments of the logarithms of the potential function remain nonnegative, which should be satisfied automatically. We note that the constraints $A^T x = 0$ when A is defined by expression (5.8) are equivalent to the original constraints $A^T x = 0$, even if U is different from the identity matrix in expression (5.3). Further, when these constraints are satisfied, the definitions (5.5) and (5.8) make the new value of $c^T x$ the same as the original one, even if some of the last $n - r$ components of the original c are nonzero. Thus we preserve the property that each reduction in the potential function is bounded away from zero.

It is important to the main result of this section to eliminate from this use of formula (4.7) the explicit dependence on $x^{(k)}$, c, $X^{(k)}$ and A. Indeed, we are going to express the search direction $e^{(k)}$ in terms of h, B, Z and $y^{(k)}$. In theory this can be done by substituting the definitions (5.8) into equation (4.7) and identifying the first r components of the resultant $d^{(k)}$, but the algebra is awkward, and it is difficult to extract a simple specification of $e^{(k)}$. Instead, therefore, we take advantage of the fact that we know already that $d^{(k)}$ is the solution of the quadratic programming problem (4.12). Fortunately, this derivation of $e^{(k)}$ is easy and highly instructive.

We separate the variables of the problem (4.12) and the matrix $X^{(k)}$ by the partitions

$$d=\left(\begin{array}{c} e \\ \hline f \end{array}\right) \quad \text{and} \quad X^{(k)}=\left(\begin{array}{c|c} Y^{(k)} & 0 \\ \hline 0 & \Omega^{(k)} \end{array}\right), \tag{5.9}$$

where e and f are in $\mathcal{R}^r$ and $\mathcal{R}^{n-r}$, and where the dimensions of $Y^{(k)}$ and $\Omega^{(k)}$ are $r \times r$ and $(n-r)\times(n-r)$ respectively, so e is no longer the vector of ones. Of course $Y^{(k)}$ and $\Omega^{(k)}$ are diagonal matrices whose diagonal elements are the components of the vector $x^{(k)}$ of expression (5.8). We also let $Z(\cdot,j)$ denote the j-th column of Z, which gives the values

$$\Omega_{jj}^{(k)} = Z(\cdot,j)^T y^{(k)}, \quad j=1,2,\ldots,n-r. \tag{5.10}$$

The definitions (5.8) and (5.9) show that the constraints $A^T d=0$ on the variables of the problem (4.12) are the equalities

$$B^T e = 0 \quad \text{and} \quad f = Z^T e. \tag{5.11}$$

We use the second condition to eliminate f from the quadratic objective function of expression (4.12), which provides a strictly convex quadratic function of the variables e. The required search direction $e^{(k)}$ is the vector that minimizes this function subject to $B^T e=0$.

It follows from equations (5.9), (4.16), (5.8), (5.10) and (5.11) that the linear term of this objective function has the value

$$\begin{aligned} d^T \nabla V(x^{(k)}) &= \frac{n}{h^T y^{(k)}}\, e^T h - e^T Y^{(k)-2} y^{(k)} - f^T \Omega^{(k)-2} Z^T y^{(k)} \\ &= \frac{n}{h^T y^{(k)}}\, e^T h - e^T Y^{(k)-2} y^{(k)} - \sum_{j=1}^{n-r} \frac{f_j}{Z(\cdot,j)^T y^{(k)}} \\ &= \frac{n}{h^T y^{(k)}}\, e^T h - e^T Y^{(k)-2} y^{(k)} - \sum_{j=1}^{n-r} \frac{e^T Z(\cdot,j)}{Z(\cdot,j)^T y^{(k)}}. \end{aligned} \tag{5.12}$$

This is a very nice result, because, by differentiating the potential function (5.4), we see that expression (5.12) is the scalar product $e^T \nabla W(y^{(k)})$. Further, the quadratic part of the objective function (4.12) is the term

$$\begin{aligned} \tfrac{1}{2}\, d^T X^{(k)-2} d &= \tfrac{1}{2}\, e^T Y^{(k)-2} e + \tfrac{1}{2}\, f^T \Omega^{(k)-2} f \\ &= \tfrac{1}{2}\, e^T \left(Y^{(k)-2} + \sum_{j=1}^{n-r} \frac{Z(\cdot,j)\, Z(\cdot,j)^T}{[Z(\cdot,j)^T y^{(k)}]^2} \right) e, \end{aligned} \tag{5.13}$$

which gives the good news that the large brackets contain the second derivative matrix of the last two terms of W. We summarise these discoveries by stating that the comment in Section 4 on the calculation (4.19) extends to the current problem. Specifically, $e^{(k)}$ is the Newton–Raphson search direction at $y^{(k)}$ of the problem

$$\text{minimize} \quad \{\overline{W}^{(k)}(y) \mid y \in \mathcal{R}^r\} \quad \text{subject to} \quad B^T y = 0, \tag{5.14}$$

where $\overline{W}^{(k)}$ is the function

$$\begin{aligned} \overline{W}^{(k)}(y) &= n \log h^T y^{(k)} + \frac{n}{h^T y^{(k)}} (y - y^{(k)})^T h - \sum_{i=1}^{r} \log y_i \\ &\qquad - \sum_{j=1}^{n-r} \log Z(\cdot,j)^T y, \quad y \in \mathcal{R}^r. \end{aligned} \tag{5.15}$$

A strong benefit of this technique is that it allows departures from the standard form of Section 2, that provide much useful freedom in the specification of linear programming problems that can be treated directly by Karmarkar's algorithm. It is now convenient not to distinguish between simple bounds and general linear inequality constraints. Therefore we let the original calculation be the problem

$$\left.\begin{array}{ll} \text{minimize} & h^T y, \quad y \in \mathcal{R}^r \\ \text{subject to} & Z^T y \geq 0, \quad B^T y = 0 \quad \text{and} \quad b_0^T y = 1 \end{array}\right\}, \tag{5.16}$$

where h and b_0 are given vectors in $\mathcal{R}^r$, and Z and B are given $r \times n$ and $r \times s$ matrices respectively. This is the standard form when $r = n$ and Z is the identity matrix, but, when Z is rectangular and $r < n$, the standard form requires some extra variables, which can be eliminated in the way that has been described. Alternatively, the potential function (5.4) anticipates the elimination, and, including any bounds with the general inequalities, it is the expression

$$W(y) = n \log h^T y - \sum_{j=1}^{n} \log Z(\cdot,j)^T y, \quad y \in \mathcal{Y}_0, \tag{5.17}$$

where $\mathcal{Y}_0$ is the set

$$\mathcal{Y}_0 = \{ y \mid Z^T y > 0, \ B^T y = 0 \}. \tag{5.18}$$

Thus the problem can be solved in the following way, that avoids the construction of the standard form and occasional large increases in the number of variables.

We require a feasible starting point $y^{(1)} \in \mathcal{Y}_0$ to be given that is not optimal, and we require the set of feasible points to be bounded. Further, we continue to make the restrictive assumption that the final value of the objective function is zero, but this condition will be replaced by a lower bound on h^Ty^* later, where y^* denotes a solution of the calculation. The k-th iteration in y-space is as follows for the sequence of positive integers k. A nonzero starting point $y^{(k)}$ is available that satisfies $Z^Ty^{(k)} \geq 0$ and $B^Ty^{(k)} = 0$. The calculation ends if the termination condition (5.7) is satisfied. Otherwise $y^{(k)}$ is known to be in the set (5.18) and the vector of variables has to be revised. Therefore we let $e^{(k)}$ be the solution of the quadratic programming problem

$$\left.\begin{array}{ll} \text{minimize} & e^T\nabla W(y^{(k)}) + \tfrac{1}{2}\, e^T \left(\sum_{j=1}^{n} \dfrac{Z(\cdot,j)\, Z(\cdot,j)^T}{[Z(\cdot,j)^Ty^{(k)}]^2} \right) e, \quad e \in \mathcal{R}^r \\ \text{subject to} & B^Te = 0 \end{array}\right\}, \qquad (5.19)$$

the gradient having the value

$$\nabla W(y^{(k)}) = \frac{n}{h^Ty^{(k)}}\, h - \sum_{j=1}^{n} \frac{Z(\cdot,j)}{Z(\cdot,j)^Ty^{(k)}}. \qquad (5.20)$$

The new vector of variables is $y^{(k+1)} = y^{(k)} + \alpha^{(k)} e^{(k)}$, where, as in the paragraph that includes equation (5.8), the step-length $\alpha^{(k)}$ is found by a line search that minimizes $\{W(y^{(k)} + \alpha\, e^{(k)}) \mid \alpha \geq 0\}$. Then the next iteration is begun.

However, we have not yet proved that the above algorithm is Karmarkar's algorithm for all the problems (5.16) that satisfy the conditions that have just been stated, because the starting point of the analysis of this section, namely that the equality constraints in x-space can yield the equations (5.2), is not sufficiently general. Indeed, because the elimination that has been described reduces the number of variables, our work so far is restricted to the case when the dimensions of the $r \times n$ matrix Z satisfy $r \leq n$. There is no need for this restriction on the problem (5.16), however, because the boundedness of the feasible region allows more variables than inequalities when there are some equality constraints. Therefore we present a general result that fills this gap.

Theorem 7 Let the problem (5.16) satisfy the conditions that have been stated, and let the sequence of points $\{y^{(k)} \mid k = 1, 2, 3, \ldots\}$ be generated by the given algorithm that employs the quadratic programming calculation (5.19). Then there exists a linear programming problem in standard form, having n variables and the following property. Defining the sequence $\{x^{(k)} \mid k = 1, 2, 3, \ldots\}$ by applying Karmarkar's algorithm to the standard form, these points satisfy the identities $\{x^{(k)} = Z^Ty^{(k)} \mid k = 1, 2, 3, \ldots\}$. Further, if $x \in \mathcal{R}^n$ and $y \in \mathcal{R}^r$ are any vectors such that $x = Z^Ty$ and $B^Ty = 0$, then the potential function of the standard form has the value $V(x) = W(y)$, where W is the potential function (5.17).

Proof Let $\bar{\mathcal{Y}} \subset \mathcal{R}^r$ be the linear space $\{y \mid B^T y = 0\}$. Thus the set of vectors $\{x = Z^T y \mid y \in \bar{\mathcal{Y}}\}$ is a linear subspace of $\mathcal{R}^n$, and we call it $\bar{\mathcal{S}}$. It is important that the conditions of the theorem imply that the correspondence between $\bar{\mathcal{Y}}$ and $\bar{\mathcal{S}}$ is one-to-one.

To prove this assertion, we suppose that $y^{(a)}$ and $y^{(b)}$ are different elements of $\bar{\mathcal{Y}}$ such that $Z^T y^{(a)} = Z^T y^{(b)}$. Hence the nonzero vector $y^{(c)} = y^{(b)} - y^{(a)}$ satisfies $Z^T y^{(c)} = 0$ and $B^T y^{(c)} = 0$. The condition $b_0^T y^{(c)} \neq 0$ must hold, because otherwise the feasible set of expression (5.16) would not be bounded. Further, we let $y^{(d)}$ be the starting vector $y^{(1)}$ if $y^{(c)}$ and $y^{(1)}$ are linearly independent, but otherwise we let $y^{(d)}$ be a solution y^* of the linear programming problem (5.16). It follows that the vector

$$y^{(e)} = y^{(d)} - y^{(c)} / b_0^T y^{(c)} \tag{5.21}$$

is nonzero and satisfies $Z^T y^{(e)} \geq 0$, $B^T y^{(e)} = 0$ and $b_0^T y^{(e)} = 0$. Therefore the addition of any positive multiple of $y^{(e)}$ to any feasible vector preserves feasibility. In this case, however, the feasible region would not be bounded, so we have a contradiction that gives the required result.

We now choose two matrices Z^+ and A. The isomorphism that has just been proved and linearity imply that we can let Z^+ be an $n \times r$ matrix that satisfies the equation

$$Z^{+T} Z^T y = y, \quad y \in \bar{\mathcal{Y}}. \tag{5.22}$$

Further, we let A be any matrix with linearly independent columns such that $\bar{\mathcal{S}}$ is the space $\{x \mid A^T x = 0\}$. Further, we complete the definition of the linear programming problem in standard form, where we are using the notation of Section 2, by setting $c = Z^+ h$ and by letting a_0 be any vector, satisfying the normalization condition $a_0^T Z^T y^{(1)} = 1$, such that the feasible region (2.1) is bounded. Thus $\bar{\mathcal{S}}_0$ in Lemma 1 is $\{x = Z^T y \mid y \in \bar{\mathcal{Y}}_0\}$, where $\bar{\mathcal{Y}}_0$ is the set

$$\bar{\mathcal{Y}}_0 = \{\, y \mid Z^T y \geq 0, \ y \in \bar{\mathcal{Y}} \,\}. \tag{5.23}$$

The two calculations that we are comparing confine the sequences $\{x^{(k)} \mid k = 1,2,3,\ldots\}$ and $\{y^{(k)} \mid k = 1,2,3,\ldots\}$ to $\bar{\mathcal{S}}_0$ and $\bar{\mathcal{Y}}_0$ respectively. Further, the isomorphism gives a sequence $\{\bar{y}^{(k)} \mid k = 1,2,3,\ldots\}$ in $\bar{\mathcal{Y}}_0$ such that $\{x^{(k)} = Z^T \bar{y}^{(k)} \mid k = 1,2,3,\ldots\}$. We have to prove that $y^{(k)} = \bar{y}^{(k)}$ can be achieved for all k.

Of course the relation between the potential functions (2.13) and (5.17) depends on the choice $c = Z^+ h$. If $B^T y = 0$ and $x = Z^T y$, then the identity

$$c^T x = h^T Z^{+T} Z^T y = h^T y \tag{5.24}$$

is a consequence of equation (5.22), the condition $B^T y = 0$ being equivalent to $y \in \bar{\mathcal{Y}}$. It follows from the definitions of V and W that the last statement of the theorem is true.

We begin the calculation of the standard form of Karmarkar's algorithm at the feasible point $x^{(1)} = Z^T y^{(1)}$, which is in $\mathcal{S}_0$ because we are given $y^{(1)} \in \mathcal{Y}_0$. Employing the usual notation, we let $x^{(2)} = x^{(1)} + \alpha^{(1)} d^{(1)}$, and, because $x^{(2)}$ is in $\bar{\mathcal{S}}$, we recall from the isomorphism that $x^{(2)} = Z^T \bar{y}^{(2)}$ and $\bar{y}^{(2)} \in \bar{\mathcal{Y}}$ define $\bar{y}^{(2)}$. Further, because $d^{(1)}$ is in $\bar{\mathcal{S}}$, we can define $\bar{e}^{(1)}$ by the conditions $d^{(1)} = Z^T \bar{e}^{(1)}$ and $\bar{e}^{(1)} \in \bar{\mathcal{Y}}$, which provides the relation $\bar{y}^{(2)} = y^{(1)} + \alpha^{(1)} \bar{e}^{(1)}$. It follows from the equivalence of the potential functions V and W and the exact line searches that the calculation in y-space gives $y^{(2)} = \bar{y}^{(2)}$ if and only if the search direction $e^{(1)}$ is a positive multiple of $\bar{e}^{(1)}$. In fact we are going to establish $e^{(1)} = \bar{e}^{(1)}$, and then a straightforward inductive argument completes the proof of the theorem. Therefore it remains to show that $x^{(k)} = Z^T y^{(k)}$ implies $e^{(k)} = \bar{e}^{(k)}$.

The search direction $d^{(k)}$ minimizes the objective function of the quadratic programming problem (4.12) subject to $d^{(k)} \in \bar{\mathcal{S}}$. Since, for every $d \in \bar{\mathcal{S}}$, there is a unique $\bar{e} \in \bar{\mathcal{Y}}$ such that $d = Z^T \bar{e}$, and since we have $Z^T \bar{e} \in \bar{\mathcal{S}}$ for every $\bar{e} \in \bar{\mathcal{Y}}$, it follows that $d^{(k)}$ is the vector $Z^T \bar{e}^{(k)}$, where $\bar{e}^{(k)}$ is the value of $\bar{e}$ that solves the calculation

$$\left.\begin{array}{ll} \text{minimize} & \bar{e}^T Z \nabla V(x^{(k)}) + \tfrac{1}{2} \bar{e}^T Z X^{(k)-2} Z^T \bar{e}, \quad \bar{e} \in \mathcal{R}^r \\ \text{subject to} & B^T \bar{e} = 0 \end{array}\right\}. \qquad (5.25)$$

We compare this calculation with the quadratic programming problem (5.19). The analogy of expression (5.12) is that equations (4.16) and (5.20) give the identity

$$\begin{aligned} \bar{e}^T Z \nabla V(x^{(k)}) &= \frac{n}{c^T x^{(k)}} \bar{e}^T Z c - \bar{e}^T \sum_{j=1}^{n} \frac{Z(\cdot,j)}{x_j^{(k)}} \\ &= \frac{n}{h^T y^{(k)}} \bar{e}^T h - \sum_{j=1}^{n} \frac{\bar{e}^T Z(\cdot,j)}{Z(\cdot,j)^T y^{(k)}} \\ &= \bar{e}^T \nabla W(y^{(k)}), \quad \bar{e} \in \bar{\mathcal{Y}}, \end{aligned} \qquad (5.26)$$

where the middle line depends on $c = Z^+ h$ and on the inductive hypothesis $x^{(k)} = Z^T y^{(k)}$. Further, the analogy of expression (5.13) is the identity

$$\tfrac{1}{2} \bar{e}^T Z X^{(k)-2} Z^T \bar{e} = \tfrac{1}{2} \bar{e}^T \left(\sum_{j=1}^{n} \frac{Z(\cdot,j)\, Z(\cdot,j)^T}{[Z(\cdot,j)^T y^{(k)}]^2} \right) \bar{e}. \qquad (5.27)$$

Therefore the calculations (5.19) and (5.25) are the same. Hence they provide $e^{(k)} = \bar{e}^{(k)}$ as required, which completes the proof of the theorem. □

An immediate corollary is that every iteration of the calculation in y-space reduces its potential function by an amount that is bounded away from zero.

Therefore the polynomial-time property of Karmarkar's algorithm is enjoyed by our version that treats the linear programming problem (5.16) directly.

We now address the removal of the restrictive assumption, obtaining guidance from the standard form that is constructed in the proof of Theorem 7. Since equation (5.24) shows that the objective functions of the two calculations of the theorem agree, it is suitable to pretend for a moment that we are working in x-space, and to increase $\gamma^{(k)}$ by the technique that is described at the end of Section 4, after giving further attention to the choice of a_0. Specifically, we require the relation

$$(c-\gamma^{(k)}a_0)^T x = (h-\gamma^{(k)}b_0)^T y \tag{5.28}$$

to hold when $x = Z^T y$ and $B^T y = 0$, so we set $a_0 = Z^+ b_0$. Therefore we can follow the rule, stated in the last paragraph of Section 4, that any increases in $\gamma^{(k)}$ are to provide the property that at least one component of $d^{(k)}$ is no less than the corresponding component of $x^{(k)}$. Our analysis shows that in y-space this means that at least one component of $Z^T(e^{(k)}-y^{(k)})$ is to be nonnegative, but we have to look more closely at the calculation (5.19) in order to express this condition in a convenient form.

Let H be the second derivative matrix of the problem (5.19). Then, by analogy with the derivation of equation (4.15) from expression (4.12), the search direction is the vector

$$e^{(k)} = -H^{-1}\,[\,I - B(B^T H^{-1} B)^{-1} B^T H^{-1}\,]\,\nabla W(y^{(k)}). \tag{5.29}$$

Moreover the identity

$$\sum_{j=1}^{n} \frac{Z(\cdot,j)}{Z(\cdot,j)^T y^{(k)}} = \left(\sum_{j=1}^{n} \frac{Z(\cdot,j)\,Z(\cdot,j)^T}{[Z(\cdot,j)^T y^{(k)}]^2} \right) y^{(k)} = H y^{(k)} \tag{5.30}$$

implies that the last term of the gradient (5.20) is simply $-Hy^{(k)}$. It follows, remembering the constraint $B^T y^{(k)} = 0$, that the contribution from this term to the search direction (5.29) is just $y^{(k)}$. Therefore we have the equation

$$e^{(k)}-y^{(k)} = -\frac{n}{(h-\gamma^{(k)}b_0)^T y^{(k)}}\, H^{-1}\,[\,I - B(B^T H^{-1} B)^{-1} B^T H^{-1}\,]\,(h-\gamma^{(k)}b_0), \tag{5.31}$$

after replacing h by $h-\gamma^{(k)}b_0$ in the gradient (5.20) in order to take account of the current lower bound, which corresponds to the definition (3.10). Now as usual the lower bound makes the denominator $(h-\gamma^{(k)}b_0)^T y^{(k)}$ positive. Hence our rule for changing $\gamma^{(k)}$ is that we require at least one component of the vector

$$Z^T H^{-1}\,[\,I - B(B^T H^{-1} B)^{-1} B^T H^{-1}\,]\,(h-\gamma^{(k)}b_0) \tag{5.32}$$

to be nonpositive. If necessary we make the smallest increase in $\gamma^{(k)}$ that provides this condition, which is analogous to the change that depends on equation (4.24).

6. An example that has an infinite number of constraints

The work of Section 5 shows that Karmarkar's algorithm can be applied directly to linear programming problems that have few variables and a large number of constraints. We are going to study a simple example of such a calculation, which is derived from the nonlinear problem

$$\left.\begin{array}{ll} \text{minimize} & x_2, \quad x \in \mathcal{R}^2 \\ \text{subject to} & x_1^2 + x_2^2 \le 1 \end{array}\right\}. \tag{6.1}$$

We approximate the nonlinear constraint by an envelope of tangents to the unit circle, each tangential condition having the form

$$\cos\theta\, x_1 + \sin\theta\, x_2 \le 1 \tag{6.2}$$

for some angle θ. We see that x is feasible if and only if inequality (6.2) holds for all $\theta \in [0,\, 2\pi]$. For interest, we let θ have an infinite number of values, distributed uniformly over $[0,\, 2\pi]$, which allows the sum that occurs in the potential function to be replaced by an integral. Moreover, we require all constraints to be homogeneous except for a single equation. Thus the number of variables increases from two to three, and we use the notation $y \in \mathcal{R}^3$ for the new variables because we will go back to the original variables $x \in \mathcal{R}^2$ later. Therefore we express the calculation as the semi-infinite programming problem

$$\left.\begin{array}{lcc} \text{minimize} & y_2 + y_3, \quad y \in \mathcal{R}^3 & \\ \text{subject to} & \cos\theta\, y_1 + \sin\theta\, y_2 + y_3 \ge 0, & 0 \le \theta \le 2\pi \\ \quad\text{and} & y_3 = 1 & \end{array}\right\}, \tag{6.3}$$

the term y_3 being included in the objective function in order that the "restrictive assumption" is satisfied. Hence, choosing a normalization that provides homogeneity in y, the potential function (5.17) is the expression

$$W(y) = 2\pi \log(y_2 + y_3) - \int_0^{2\pi} \log(y_1 \cos\theta + y_2 \sin\theta + y_3)\, d\theta, \quad y \in \mathcal{R}^3. \tag{6.4}$$

Because there are no homogeneous equality constraints in this example, the algorithm can be regarded as a procedure for unconstrained minimization that is applied to $\{W(y) \mid y \in \mathcal{R}^3\}$. We expect the integral term of expression (6.4) to keep the homogeneous inequality constraints satisfied. The fact that we have only three variables shows the advantage of the version of the algorithm that is developed in Section 5.

There is an analytic expression for the above integral. Indeed, the equation that is numbered 865.44 in Dwight (1961) is the formula

$$\int_0^{\pi} \log(\,a \pm b\cos t\,)\,dt = \pi \,\log\left(\tfrac{1}{2}\,a + \tfrac{1}{2}\sqrt{a^2-b^2}\right). \tag{6.5}$$

Since the range of θ is $[0,\,2\pi]$, we may replace the term $y_1\cos\theta + y_2\sin\theta$ by $(y_1^2+y_2^2)^{1/2}\cos\theta$ in equation (6.4). It follows that the potential function has the value

$$W(y) = 2\pi\,\log(y_2+y_3) - 2\pi\,\log\left(\tfrac{1}{2}\,y_3 + \tfrac{1}{2}\sqrt{y_3^2-y_1^2-y_2^2}\right), \quad y \in \mathcal{R}^3. \tag{6.6}$$

Now, for every $y^{(k)}$ that occurs during the calculation of the algorithm, we are going to take advantage of the homogeneity by picking the normalization $y_3^{(k)}=1$. Therefore every iteration is going to reduce the function

$$F(x) = \log(1+x_2) - \log\left(1 + \sqrt{1-x_1^2-x_2^2}\right), \quad x \in \mathcal{R}^2, \tag{6.7}$$

which is derived by setting $y_1 = x_1$, $y_2 = x_2$ and $y_3 = 1$ in the expression $\{W(y)/(2\pi) - \log 2 \mid y \in \mathcal{R}^3\}$.

We see that we no longer have the usual property that the potential function tends to infinity when y (or x) approaches the boundary of the feasible region, because this feature has been removed by the analytic integration and our scaling of the potential function. However, the gradient

$$\nabla W(y) = 2\pi \begin{pmatrix} y_1/[\sigma(y_3+\sigma)] \\ 1/(y_2+y_3) + y_2/[\sigma(y_3+\sigma)] \\ 1/(y_2+y_3) - 1/\sigma \end{pmatrix}, \quad y \in \mathcal{R}^3, \tag{6.8}$$

where σ is the term

$$\sigma = \sqrt{y_3^2-y_1^2-y_2^2}, \tag{6.9}$$

becomes unbounded if y tries to violate the constraints. Thus, if $y^{(k)}$ is strictly inside the feasible region and $e^{(k)}$ is any search direction that goes to the boundary of this region, then the line search function $\{W(y^{(k)}+\alpha\, e^{(k)}) \mid \alpha \geq 0\}$ has a minimum at a feasible point.

Another important consequence of the infinite number of constraints is that we do not know if the algorithm will converge successfully. Indeed, the guaranteed reduction of Theorem 4 depends on the potential function being normalized so that the logarithm of the objective function ($c^T x$ or $h^T y$) is multiplied by the number of inequality constraints. Therefore, before applying the analysis of Section 3, we should rescale the function (6.4) so that the factor 2π becomes infinite. Thus the guaranteed reduction in our potential function is a positive constant multiplied by zero, so the theorem is of no relevance to the present example.

Nevertheless, it is still straightforward to apply the algorithm that is given in Section 5. We begin at any interior feasible point $y^{(1)}$. For each positive integer k, the search direction $e^{(k)}$ is derived from a quadratic programming problem that is analogous to the calculation (5.19). In the present case there are no constraints, so $e^{(k)}$ is obtained by equating the gradient of the quadratic objective function to zero. The matrix of this linear system in Section 5 is the second derivative matrix of the last part of the potential function (5.17), so now we must employ the analogous part of the function (6.6), which is the expression

$$-2\pi \log\left(\tfrac{1}{2}y_3 + \tfrac{1}{2}\sqrt{y_3^2 - y_1^2 - y_2^2}\right), \quad y \in \mathcal{R}^3. \qquad (6.10)$$

It has the second derivative matrix

$$H(y) = 2\pi \begin{pmatrix} \dfrac{\sigma^2(y_3+\sigma) + y_1^2\,(y_3+2\sigma)}{\sigma^3\,(y_3+\sigma)^2} & \dfrac{y_1\,y_2\,(y_3+2\sigma)}{\sigma^3\,(y_3+\sigma)^2} & \dfrac{-y_1}{\sigma^3} \\[2ex] \dfrac{y_1\,y_2\,(y_3+2\sigma)}{\sigma^3\,(y_3+\sigma)^2} & \dfrac{\sigma^2\,(y_3+\sigma) + y_2^2\,(y_3+2\sigma)}{\sigma^3\,(y_3+\sigma)^2} & \dfrac{-y_2}{\sigma^3} \\[2ex] \dfrac{-y_1}{\sigma^3} & \dfrac{-y_2}{\sigma^3} & \dfrac{y_3}{\sigma^3} \end{pmatrix}, \qquad (6.11)$$

where σ is still the term (6.9). Thus $e^{(k)}$ is defined by the system

$$H(y^{(k)})\,e^{(k)} = -\nabla W(y^{(k)}), \qquad (6.12)$$

which has the solution

$$e^{(k)} = -\sigma \begin{pmatrix} y_1/(y_2+y_3) \\ 1+\sigma/(y_2+y_3) \\ 1-\sigma/(y_2+y_3) \end{pmatrix}. \qquad (6.13)$$

One can verify this claim by substituting the values (6.8), (6.11) and (6.13) into equation (6.12), remembering the definition of σ. A partial check is provided by a remark in the penultimate paragraph of Section 4, namely that the line search makes no first order change to the barrier term of the potential function, which in the present case is expression (6.10). It follows from the derivation of the gradient (6.8) that $e^{(k)}$ should be orthogonal to the vector $(\, y_1 \;\; y_2 \;\; -y_3{-}\sigma)$. This condition is satisfied.

We are going to return to a two-dimensional form of our calculation in order to provide some numerical results and a picture. Therefore, for each $y \in \mathcal{R}^3$ we let $x \in \mathcal{R}^2$ have the components y_1/y_3 and y_2/y_3. As mentioned already, we normalize $y^{(k)}$ so that $y_3^{(k)} = 1$, so it would be convenient if the last component of the search direction in y-space were zero. Fortunately, the half-line argument in the complete paragraph that follows equation (4.22) shows that some additions of multiples of $y^{(k)}$ to $e^{(k)}$ make no difference to the calculation in y-space because of homogeneity. Specifically, since $y_3 = 1$, we construct the vector

$$-\sigma \begin{pmatrix} \dfrac{y_1}{y_2+1} \\ 1+\dfrac{\sigma}{y_2+1} \\ 1-\dfrac{\sigma}{y_2+1} \end{pmatrix} + \sigma\left(1-\frac{\sigma}{y_2+1}\right)\begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix} = \frac{\sigma}{y_2+1}\begin{pmatrix} y_1\,(y_2-\sigma) \\ (y_2+1)\,(y_2-1-\sigma) \\ 0 \end{pmatrix}, \qquad (6.14)$$

which suggests the search direction

$$d^{(k)} = \begin{pmatrix} x_1^{(k)}\left(x_2^{(k)} - \sqrt{1 - x_1^{(k)\,2} - x_2^{(k)\,2}}\right) \\ \left(x_2^{(k)}+1\right)\left(x_2^{(k)} - 1 - \sqrt{1 - x_1^{(k)\,2} - x_2^{(k)\,2}}\right) \end{pmatrix} \qquad (6.15)$$

in x-space, the positive multiplying factor $\sigma/(y_2{+}1)$ being unimportant. Since $d^{(k)}$ must go to the boundary of the unit circle from any interior point $x^{(k)} \in \mathcal{R}^2$, the addition in equation (6.14) of a multiple of $y^{(k)}$ to $e^{(k)}$ does not exclude any relevant points of the line search, which can happen sometimes because the feasible region in y-space is a cone.

Therefore our two-dimensional calculation is as follows. We let the initial vector of variables $x^{(1)}$ be any point that is strictly inside the unit circle. Each iteration sets $x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}$, where $d^{(k)}$ is the direction (6.15) and where $\alpha^{(k)}$ is calculated to minimize the new value of the function (6.7). Because the usual termination condition may be unattainable, we continue the iterations until the inequality

$$\| \, x^{(k+1)} - x^{(k)} \|_\infty \le 10^{-6} \qquad (6.16)$$

is satisfied.

Numerical results using two starting points are reported in Table 1. The first starting point has the components (0.1, −0.5), which is a fair estimate of the solution at (0, −1), but we see that the sequence $\{x^{(k)} \mid k = 1, 2, 3, \ldots\}$

First calculation		k	Second calculation	
$x_1^{(k)}$	$x_2^{(k)}$		$x_1^{(k)}$	$x_2^{(k)}$
0.1000000	−0.5000000	1	0.5990000	0.8000000
0.0424728	−0.9990972	2	0.7052093	0.7021610
0.0415769	−0.9991353	3	0.8954367	0.4010370
0.0415583	−0.9991361	4	0.9773643	−0.0878220
0.0415579	−0.9991361	5	0.8981845	−0.4253666
		6	0.8166685	−0.5747441
		7	0.7713221	−0.6360218
		8	0.7497547	−0.6616377
		9	0.7400236	−0.6725662
		10	0.7357213	−0.6772816
		11	0.7338350	−0.6793272
		12	0.7330110	−0.6802167
		13	0.7326515	−0.6806040
		14	0.7324948	−0.6807727
		15	0.7324265	−0.6808461
		16	0.7323967	−0.6808782
		17	0.7323837	−0.6808921
		18	0.7323781	−0.6808982
		19	0.7323756	−0.6809008
		20	0.7323746	−0.6809020
		21	0.7323741	−0.6809025

Table 1: The semi-infinite programming example

seems to be converging to a wrong point on the unit circle. The second calculation is pathological, because $x^{(1)}$ is very close to the edge of the feasible region. In this case the Karmarkar search direction is nearly parallel to the nearest constraint boundary. Consequently, the table shows that the sequence $\{x^{(k)} \mid k = 1, 2, 3, \ldots\}$ stays close to the boundary of the unit circle. Again the numerical results indicate that quite rapid convergence is occurring to a spurious point. Another calculation is shown in Figure 2. Here the starting vector is (0.8, 0.5), the points $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, $x^{(4)}$ and the limit of the sequence $\{x^{(k)} \mid k = 1, 2, 3, \ldots\}$ being plotted. Further, the dotted curves are level lines $\{x \mid F(x) = \text{constant}\}$ of the potential function.

These results seem to contradict the current opinion that Karmarkar's algorithm is suitable for very large linear programming calculations. Therefore

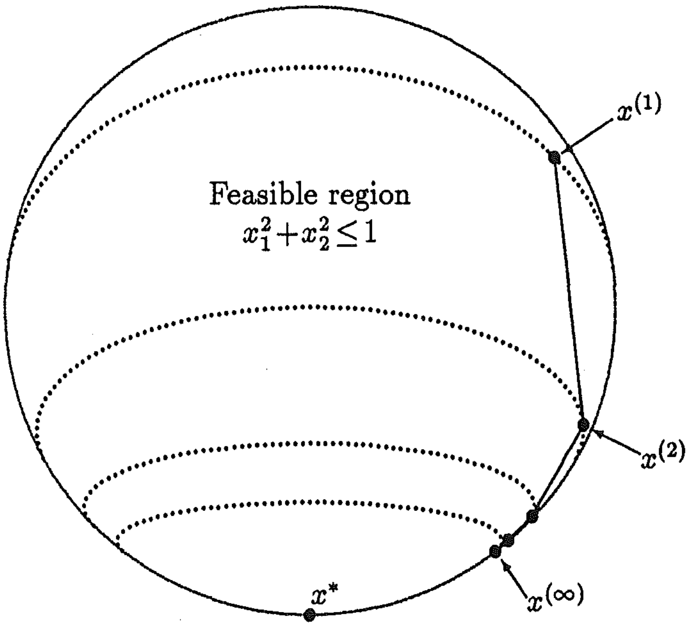

Figure 2: An example of premature convergence

analytic expressions were derived for the changes to the variables of a typical iteration, in order to prove theoretically that, when all arithmetic is exact, the algorithm fails to solve the problem that is studied in this section. Some of this analysis is given below, but most of the details are omitted because there are several pages of algebra in the complete proof.

We suppose that at the starting point $x^{(k)}$ of a typical iteration we have $x_1^{(k)} > 0$ and $x_2^{(k)} < 0$, which does not lose generality because x_2 becomes negative after a finite number of iterations and the sign of x_1 is unimportant, except that in the special case when $x_1 = 0$ the search direction leads directly to the solution at $(0, -1)$. Further, we recall that $x^{(k)}$ is strictly inside the unit circle. It is therefore convenient to use the notation

$$x^{(k)} = \begin{pmatrix} \sin\psi \, \cos\omega \\ -\cos\psi \end{pmatrix} \tag{6.17}$$

for the starting point of the iteration, where $0 < \psi < \pi/2$ and $0 < \omega < \pi/2$. Thus, after dividing by the positive factor $\sin\psi$, formula (6.15) provides the search direction

$$d^{(k)} = \begin{pmatrix} -\cos\psi \, \cos\omega - \sin\psi \, \cos\omega \, \sin\omega \\ -\sin\psi - \sin\omega + \cos\psi \, \sin\omega \end{pmatrix}. \tag{6.18}$$

Because the components of $d^{(k)}$ are negative and the step-length $\alpha^{(k)}$ is going to be positive, the iteration preserves $x_2<0$. Further, x_1 will remain positive, because, if α is such that the first component of $x^{(k)}+\alpha\, d^{(k)}$ is zero, then this point is outside the unit circle. It follows that, after removing the termination condition, the algorithm generates a well-defined infinite sequence $\{x^{(k)} \mid k=1,2,3,\ldots\}$, where each vector of variables has the form (6.17) with $0<\psi<\pi/2$ and $0<\omega<\pi/2$.

The required step-length $\alpha^{(k)}$ is the unique positive value of α that satisfies the equation

$$d^{(k)T}\nabla F(x^{(k)}+\alpha\, d^{(k)}) = 0, \quad \alpha>0, \tag{6.19}$$

(Todd and Burrell, 1986), where F is the potential function (6.7). Using the definition of F, we write equation (6.19) in the form

$$\begin{aligned} d_2^{(k)} \left(1 - \|x^{(k)} + \alpha\, d^{(k)}\|_2^2 + \sqrt{1 - \|x^{(k)} + \alpha\, d^{(k)}\|_2^2}\right) \\ + (1 + x_2^{(k)} + \alpha\, d_2^{(k)})\, d^{(k)T}(x^{(k)} + \alpha\, d^{(k)}) = 0. \end{aligned} \tag{6.20}$$

Fortunately, the terms in α^2 that are not under the square root sign cancel each other, so $\alpha^{(k)}$ is a root of a quadratic polynomial. The two roots have opposite signs, and, by substituting expressions (6.17) and (6.18) into this polynomial and carrying out some laborious algebra, one can show that the positive root has the value

$$\alpha^{(k)} = (1 - \cos\psi + \sin\psi)\, \sin\omega \,/\, \Delta, \tag{6.21}$$

where the denominator is the number

$$\Delta = 1 + (\cos\psi + \sin\psi)\, \sin\omega + (1 - \cos\psi + \sin\psi)\, \sin^2\omega. \tag{6.22}$$

Thus the components of $x^{(k+1)}$ are given by the formula

$$\left.\begin{aligned} x_1^{(k+1)} &= \cos\omega\, [\,\sin\psi + (1-\cos\psi)\, \sin\omega\,] \,/\, \Delta \\ x_2^{(k+1)} &= -1 + (1-\cos\psi)\, (1-\sin\omega) \,/\, \Delta \end{aligned}\right\}. \tag{6.23}$$

In order to verify that $\alpha^{(k)}$ is correct, it is easier to check condition (6.19) instead of wrestling with the quadratic equation.

We now consider the reduction in the potential function (6.7) that is achieved by the iteration. A relatively easy calculation shows that formula (6.23) provides the identity

$$F(x^{(k+1)}) = F(x^{(k)}) - \log\left(\frac{1+\sin\omega}{1-\sin\omega}\right). \tag{6.24}$$

Further, it follows from the relation

$$\sin^2\omega = (\,1 - x_1^{(k)\,2} - x_2^{(k)\,2}\,) \,/\, (\,1 - x_2^{(k)\,2}\,) \tag{6.25}$$

that the value of ω on the $(k+1)$-th iteration satisfies the inequality

$$\begin{aligned}\sin^2\omega^{(k+1)} &= \frac{(1-\cos\psi)\,(1-\sin\omega)\,\sin^2\omega}{1+\cos\psi+(1+2\sin\psi+\cos\psi)\sin\omega+2\,(1-\cos\psi+\sin\psi)\sin^2\omega} \\ &\leq \frac{1-\cos\psi}{1+\cos\psi}\sin^2\omega, \qquad (6.26)\end{aligned}$$

where, as usual, the superscript is an iteration number. Now we recall that $\cos\psi = -x_2^{(k)}$, that $x_2^{(k)}$ is negative and that every iteration reduces this variable. Therefore expression (6.26) gives the relation

$$\sin^2\omega^{(k+1)} \leq \left(\frac{1-\cos\psi^{(\ell)}}{1+\cos\psi^{(\ell)}}\right)^{k+1-\ell} \sin^2\omega^{(\ell)}, \quad k \geq \ell, \qquad (6.27)$$

where ℓ is any fixed integer such that $\cos\psi^{(\ell)}$ is positive. It follows that the sum $\sum_k \sin\omega^{(k)}$ is absolutely convergent, so we have the bound

$$\sum_{k=\ell}^{\infty} \log\left(\frac{1+\sin\omega^{(k)}}{1-\sin\omega^{(k)}}\right) < \infty. \qquad (6.28)$$

Hence we deduce from equation (6.24) that the monotonically decreasing sequence $\{F(x^{(k)}) \mid k=1,2,3,\ldots\}$ is bounded below. Therefore the calculated vectors of variables $\{x^{(k)} \mid k=1,2,3,\ldots\}$ do not tend to the optimal point $(0,\,-1)$, which completes the proof that Karmarkar's algorithm can fail to provide convergence to the solution of a semi-infinite programming problem.

7. Discussion

I have greatly enjoyed studying Karmarkar's algorithm because of its elegant theoretical properties. The crucial ingredient seems to be that, due to the homogeneity of the potential function, there are no ill effects from the nonlinearity of the transformation (2.6). We have noted that the price of homogeneity is the need to assume that the optimal value of the objective function is zero, and it is brilliant that one can afford this price in practical calculations by working with a lower bound on $c^T x^*$ that is improved automatically in a way that preserves the polynomial time complexity. These ideas alone persuade me that the algorithm breaks new ground in the development of numerical methods for optimization calculations, and I feel sure that the ideas are of fundamental importance to future research, regardless of whether or not the actual performance of the algorithm is sometimes vastly superior to the simplex method as claimed. I know little about the comparisons that have been made by numerical experiments, my degree of interest in these activities being relatively small. On the other hand, I am very willing to be excited by techniques that improve efficiency, including the use of approximations and the implementations of matrix calculations.

The example of Section 6 shows a need for extensions to the basic algorithm, unless one takes the view that the method under investigation is not Karmarkar's algorithm. It is so easy to apply the results of Section 5 to semi-infinite programming problems with an infinite number of constraints, that the question of identity should surely be aimed at the procedure in Section 5 that tackles the general linear programming problem (5.16) directly, instead of requiring a preliminary calculation that expresses the problem in a standard form. I submit that Theorem 7 establishes that our procedure is equivalent to the original algorithm. Indeed, the components of Z^Ty in expression (5.16) have to be variables in the standard calculation because there the only inequality constraints are simple bounds. Thus the theorem proves that our calculation does correspond to a standard one that has the minimal number of variables.

The failure of the method in Section 6 is due partly to the fact that, when $x^{(k)}$ is very close to the boundary of the unit circle, then the search direction $d^{(k)}$ is nearly orthogonal to $x^{(k)}$. Therefore the preservation of feasibility demands a small step-length because the feasible region has a curved boundary. It might be possible to devise an *ad hoc* remedy that forces the line search to move away from the perimeter of the unit circle initially, but it can be shown that $d^{(k)}$ already satisfies this condition in the region that is sandwiched between the ellipse $\{x \in \mathcal{R}^2 \mid x_1^2+2x_2^2+2x_2=0\}$ and the unit circle, so one would need to strengthen this condition artificially. An alternative remedy might be to modify the potential function (6.4) in a way that recovers the usual property that this function is unbounded at the edge of the feasible region, but, when the number of constraints is infinite, this goal is in conflict with homogeneity.

The main reason why I had not studied Karmarkar's algorithm earlier is that I believe that linear constraints in optimization calculations should normally be treated explicitly, instead of being lumped together in a penalty function method. I still hold this view, and it has been strengthened by the results of this paper. On the other hand, I agree that it is important to avoid the inefficiencies that are liable to occur in the simplex method when many vertices of the feasible region are close to each other. Therefore I developed some Fortran software recently (Powell, 1989) for linearly constrained optimization calculations that generates search directions that are not allowed to move towards the boundaries of any constraints with "small" residuals, the meaning of "small" being dependent on a tolerance parameter that is adjusted automatically. This software is highly suitable for calculations that have far more constraints than variables, except that it takes no account of sparsity. Good techniques for managing large sparse matrices are essential for the efficient solution of most problems that are regarded as substantial by the linear programming community. Further, when such techniques are available, there may be no need for the elimination of variables that is performed in Section 5. The details of the elimination show, however, that the use of quadratic programming calculations for defining search directions provides a very convenient description of Karmarkar's algorithm for solving general linear programming problems.

We have viewed the most active field of research in optimization through a window that is shamefully narrow because I prefer to develop some ideas independently instead of reading the relevant literature comprehensively. The given description of Karmarkar's algorithm, however, should provide insight into its fundamental properties, because we have avoided several unnecessary details that are present in many published papers. In particular, I learnt from Gonzaga (1988) the simplification of ignoring the single inhomogeneous constraint until the end of the calculation when the restrictive assumption is satisfied. Further, Gill *et al* (1986) introduced me to the nonlinear programming point of view that initiates the findings of Section 5. I believe that the main conclusion of Section 6 is new, however, and it would be interesting to take this work further, investigating the deterioration in efficiency that surely occurs if more and more linear constraints are employed to approximate the feasible region $\{x \in \mathcal{R}^2 \mid x_1^2+x_2^2 \leq 1\}$. Indeed, there seem to be many opportunities for further research on Karmarkar's algorithm.

Acknowledgements I am very grateful to the Institute of Mathematics and its Applications for the opportunity to present this paper at its Silver Jubilee Conference. Most of the work was done after the conference, however, at the University of Canterbury, Christchurch, New Zealand, where I enjoyed excellent opportunities for research and some financial assistance as an Erskine Fellow. During this time I was helped greatly by the advice and encouragement of Ian Coope.

References

H.B. Dwight (1961), *Tables of Integrals and other Mathematical Data*, The Macmillan Company (Toronto).

P.E. Gill, W. Murray, M.A. Saunders, J.A. Tomlin and M.H. Wright (1986), "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method", *Mathematical Programming*, Vol. 36, pp. 183–209.

C.C. Gonzaga (1988), "A simple presentation of Karmarkar's algorithm", preprint, COPPE — Federal University of Rio de Janeiro.

N. Karmarkar (1984), "A new polynomial-time algorithm for linear programming", *Combinatorica*, Vol. 4, pp. 373–393.

M.J.D. Powell (1989), "TOLMIN: a Fortran package for linearly constrained optimization calculations", Report DAMTP 1989/NA2, University of Cambridge.

D.F. Shanno (1988), "Computing Karmarkar projections quickly", *Mathematical Programming*, Vol. 41, pp. 61–71.

G. Strang (1987), "Karmarkar's algorithm and its place in applied mathematics", *The Mathematical Intelligencer*, No. 2, Vol. 9, pp. 4–10.

M.J. Todd and B.P. Burrell (1986), "An extension of Karmarkar's algorithm for linear programming using dual variables", *Algorithmica*, Vol. 1, pp. 409–424.

J.A. Tomlin (1987), "An experimental approach to Karmarkar's projective method for linear programming", *Mathematical Programming Studies*, No. 31, pp. 175–191.