SEARCH DIRECTIONS FOR INTERIOR LINEAR PROGRAMMING METHODS

by

Clovis C. Gonzaga

Memorandum No. UCB/ERL M87/44

22 June 1987

SEARCH DIRECTIONS FOR INTERIOR LINEAR PROGRAMMING METHODS

by

Clovis C. Gonzaga

Memorandum No. UCB/ERL M87/44

22 June 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SEARCH DIRECTIONS FOR INTERIOR LINEAR PROGRAMMING METHODS

by

Clovis C. Gonzaga

Memorandum No. UCB/ERL M87/44

22 June 1987

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Search Directions for Interior Linear Programming Methods

Clovis C. Gonzaga *
Dept. of Electrical Engineering and Computer Sciences
University of California
Berkeley, California 94720

May 15, 1987

Abstract

Since Karmarkar published his algorithm for Linear Programming, several different interior directions have been proposed and much effort was spent on the problem transformations needed to apply these new techniques. This paper examines several search directions in a common framework that does not need any problem transformation. These directions result to be combinations of two problem-dependent vectors, and can all be improved by a bidirectional search procedure.

We conclude that there are essentially two polynomial algorithms: Karmarkar's method and the algorithm that follows a central trajectory, and they differ only in a choice of parameters (respectively lower bound and penalty multiplier). We finally present a complete path-following algorithm with a new strategy for choosing multipliers.

*On leave from COPPE-Federal University of Rio de Janeiro, Cx. Postal 68511, 21941 Rio de Janeiro, RJ, Brasil
Research partly sponsored by CNPq - Brazilian National Council for Scientific and Technological Development, by National Science Foundation grant ECS-8121149, Office of Naval Research contract N00014-83-K-0602, AFOSR grant 83-0361 and State of California Microelectronics Innovation and Computer Research Opportunities Program and General Electric.

1 Introduction

In this paper we survey several internal search directions for linear programming that were proposed since Karmarkar published his algorithm [14]. The resulting algorithms are classified and compared, to be then imbedded in a general framework using a bidirectional search procedure on the original feasible set.

Some of the interesting conclusions to be reached are:

- There is no need for problem transformations. Projective algorithms have an underlying motivation based on the minimization of a zero-degree homogeneous "potential function" on a cone, but this cone does not have to be described.

- Karmarkar's direction is more general than was expected. In fact, the following directions are equivalent (in the sense that the resulting algorithms generate the same points): Karmarkar's direction, the Newton-Raphson direction for the multiplicative potential function, and any directions obtained by spherical trust region minimizations of first or second order approximations of the potential functions on the cone. Incidentally, this proves the complete equivalence between Karmarkar's method and the algorithm by Iri and Imai [13], and establishes quadratic convergence (when the optimal value is known) and polynomial convergence for both algorithms.

- A conic function in the tradition of Davidon [4] can be used instead of the potential functions, and exactly minimized in a spherical trust region, giving rise to good descent directions.

- All surveyed algorithms generate at each iteration a step that is a linear combination of $c_p = Pc$ and $e_p = Pe$, where P is the projection matrix onto $Null(A)$ and $e = [1\ 1 \dots 1]'$. All of them can be understood as feasible direction methods with scaling, with directions obtained by solving a spherical trust region minimization. Most methods can also be understood as Newton-Raphson methods (and then scaling is irrelevant).

- All algorithms can be improved by a natural extension of the trust region, resulting in a bi-directional search in the directions c_p and e_p. This search can be sophisticated by a goal programming approach to ensure polynomiality (when costs are minimized) or to ensure decreasing costs (when potential functions are minimized).

- Karmarkar's algorithm and the homotopy method in [12] are not equivalent in general: they differ in the choice of a parameter at each iteration (respectively lower bound to the optimal cost and the penalty multiplier). The penalty multiplier seems to be more controllable than the lower bounds, and should lead to a better algorithm.

Algorithms will be classified according to the following classes:

(i) Underlying problem formulation: projective or affine.
(ii) Criterion: logarithmic potential, multiplicative potential, conic function, barrier function, cost function.
(iii) Direction-finding procedure: projected gradient, trust region minimization, Newton-Raphson.
(iv) Special requirements: decreasing costs, decreasing criterion.

Although the classes above allow a large number of combinations, the actual number of different algorithms is small.

The problem: All algorithms will solve the following linear programming problem:

$$\begin{array}{rrcl} \text{minimize} & c'x & & \\ \text{subject to} & Ax & = & b \\ & x & \geq & 0 \end{array} \tag{1}$$

where $b, c, x \in \mathbb{R}^n$, A is an $m \times n$ matrix, $m < n$.

The algorithms will always work in the relative interior of the feasible set, and the following notation will be used:

$$\begin{array}{rcl} S & = & \{x \in \mathbb{R}^n | Ax = b, x > 0\} \\ Q & = & \{\lambda x | Ax = b, \lambda \in \mathbb{R}\} \\ C & = & \{\lambda x | x \in S, \lambda > 0\} \\ D & = & Null(A) \\ \mathbb{R}^n_+ & = & \{x \in \mathbb{R}^n | x > 0\} \end{array} \tag{2}$$

S is the relative interior of the feasible set, C the positive cone generated by S, Q is the subspace generated by C and D is the set of feasible directions from a point in S.

We usually assume that the feasible set is bounded (unbounded feasible sets will be treated separately). We also assume that an initial non-optimal feasible solution $\bar{x}$ is known.

Transformed problem: Since the dimension of Q is one unit higher than the dimension the feasible set, the problem can be restated as:

$$\begin{aligned} \textit{minimize} \quad c'x & \\ x &\in \bar{C} \\ a'x &= 1 \end{aligned} \tag{3}$$

where $a \in Q$ and $\bar{C}$ is the closure of cone C .

The feasible set for this problem is a cone restricted by one constraint. Projective algorithms are constructed by replacing the cost in 3 by a zero-degree homogeneous function. This is the characteristic format for projective techniques, and it is easy to obtain it by computing a and a description of C , as we shall do in section 2. The description of transformed problems was studied in many papers, including [2], [7], [11], [14], [19]. Reference [11] shows a transformation that preserves the dimension of the problem by describing the conic set as we are using here. All other references achieve this format by increasing the dimension. We will show that there is no need to describe the transformed problem, and all techniques can be applied directly to formulation (1).

All algorithms evolve by a sequence of iterations composed of (possibly) a scaling operation, the determination of one or two search directions and a minimization along these directions. In this paper we shall concentrate on the determination of the search directions from a given point $\bar{x}$, in one typical iteration of the algorithm. The notation is kept simple by assuming that (1) is the formulation *after* a master algorithm has scaled the problem (if scaling is used).

Scaling: we shall usually assume that $\bar{x} = e$, meaning that a scaling operation has been performed. It is important to remark that scaling only affects first-order methods, and is totally irrelevant to the behavior of Newton-Raphson methods. The reason to use scaling to explain such algorithms is that it simplifies the mathematical treatment, puting all techniques in a common framework. The actual implementations do not depend on changes of variables.

The paper begins by presenting some tools necessary to develop the algorithms, and then examines separately projective methods in section 3, affine methods in section 4, bidirectional search procedures in section 5 and finally a complete new path-following algorithm in section 6.

2 The tools

This section describes mathematical tools to be used in the actual algorithms. The first two subsections deal with formulation (3) and with conical projections, and have interest in themselves. The remaining results are mostly common knowledge, included to fix the language and notation.

2.1 The problem transformation

Karmarkar's algorithm in its original formulation [14] needs a special formulation that coincides with (3) with $a = e$, that is, the feasible set must lie on the unit simplex at each iteration (after scaling). A "projective transformation" is needed at each iteration to achieve this format, and this made the original algorithm difficult to understand. Todd [19] showed that Karmarkar's direction was actually the steepest descent direction for his potential function, and the simplex constraint became irrelevant: any problem in formulation (3) could be treated with straightforward scaling transformations.

A sequence of papers cited in section 1 improved the problem formulation, and in this paper we shall advance a step further by showing that no reformulation is needed in the practical construction of the algorithms. The actual description of the set Q will never be needed. In some cases (to compute lower bounds to optimal solutions) the description of projections onto Q will be needed, but this will be obtained indirectly.

Consider the problem (1) and the sets defined in (2). Let P be the projection matrix onto D, $P = I - A'(AA')^{-1}A$, and assume that $e \in S$.

Lemma 2.1 *S can be expressed as*

$$S = C \cap \{x \in \mathbb{R}^n | a'x = 1\} = \{x > 0 | x \in Q, a'x = 1\}$$

where $a \in Q$ is the unique vector computed by

$$a = \frac{e - Pe}{\|e - Pe\|^2}. \tag{4}$$

Besides this, for any $y \in C$ $a'y > 0$.

Proof: Q is the subspace generated by S and the origin. Then $e \in Q$ by hypothesis, $Null(A) \subset Q$ by construction and consequently $Pe \in Q$, $e - Pe \in Q$, $a \in Q$.

Consider any $y \in C$. Then $y = \lambda x$ for some $x \in S, \lambda > 0$.
But $x = e + (x - e)$, and then

$$a'y = \lambda[a'e + a'(x - e)] \ .$$

Note that $a'(x - e) = 0$, since $x - e \in Null(A)$ and $e - Pe \perp Null(A)$, and it follows that

$$a'y = \lambda a'e$$

Developing this expression

$$a'e = \frac{(e - Pe)'e}{\|e - Pe\|^2} = \frac{(e - Pe)'(e - Pe)}{\|e - Pe\|^2} = 1$$

since $(e - Pe)'Pe = 0$. Finally, $a'y = \lambda$, and consequently $a'y = 1$ if and only if $y = x \in S$, completing the proof.

The lemma above guarantees that (1) can be recast into format (3), and provides a unique value for $a \in Q$.

Consider now an arbitrary vector $d \in \mathbb{R}^n$ and define respectively d_p, d_Q and d_a as the projections of d onto D , Q and $Null(a')$. Assume that $d_p = Pd$ is known, and note that d_a is easily computed. Next lemma shows how to obtain d_Q .

Lemma 2.2 *The projection of a vector* $d \in \mathbb{R}^n$ *onto* Q *is given by*

$$d_Q = \frac{d'a}{\|a\|^2}a + d_p = \frac{d'(e - e_p)}{\|e - e_p\|^2}(e - e_p) + d_p \tag{5}$$

Proof: From lemma 2.1, $a \in Q$, and then $Null(a') \perp Q$. It follows that d_p is the orthogonal projection of d_Q onto $Null(a')$, $d_p = d_{Q_a}$.

On the other hand, $d - d_Q \perp a$, since $a \in Q$, and consequently $d_a - d_{Q_a} = d - d_p$.
It follows that $d_Q - d_p = d - d_q$, or $d_Q = d_p + d - d_a$. We can now compute

$$d - d_a = \frac{d'a}{\|a\|^2}a \ ,$$

and finally obtain

$$d_Q = d_p + \frac{d'a}{\|a\|^2}a \ .$$

The expression in terms of e follows trivially from (4), completing the proof.

Projections onto Q will be needed at only one point: the computation of lower bounds in Karmarkar's algorithm, which depend on c_Q . This computation will then be inexpensive since c_p and e_p are used anyway by all algorithms.

2.2 Conical projections

The projective algorithms to be studied in section 3 are based on solving a problem with a different criterion, but equivalent to (1). The new criterion is a zero-degree homogeneous function $f : \mathbb{R}^n_+ \mapsto \mathbb{R}$ (that is, for any $x > 0, \lambda > 0$, $f(\lambda x) = f(x)$), and each iteration tries to improve the available solution to the problem

$$minimize\{f(x) \mid x \in S\}$$

The properties of zero-degree homogeneous functions become apparent if we use formulation (3)

$$\begin{array}{rcl} minimize & f(x) & \\ x & \in & C \\ a'x & = & 1 \end{array} \qquad (6)$$

We shall study the special properties of descent directions for a problem in formulation (6). The essential fact is that since $f(\cdot)$ is constant on rays and C is a cone, the constraint $a'x = 1$ can be dropped and we can work in a very simple set: the subspace Q with positivity constraints.

Definition 2.3 *Given $x \in C$, the* conical projection *of x onto S is the intersection of S and the ray through x , computed by*

$$K(x) = \frac{x}{a'x}$$

$K(x)$ is well defined for any $x \in C$, since then $a'x > 0$ by lemma 2.1. It is immediate to verify that $a'K(x) = 1$ and $f(K(x)) = f(x)$.

We now study the result of a line search from a point $x \in S$ along a direction in the cone, $h \in Q$. If $a'h \neq 1$ then the points $x + \alpha h$ are infeasible for the original problem, but the conical projections of these points are feasible and have the same objective values. These conical projections follow a direction in D , called the conical projection of h from x .

Definition 2.4 *Given* $x \in S$ *and* $h \in Q$ *, the* conical projection of h from x onto S *is given by*

$$K_x(h) = h - a'h\,x$$

Lemma 2.5 *Let* $x \in S$*,* $h \in Q$ *and* $\alpha \in I\!R$ *be given, and let* $\bar{h} = K_x(h)$*. If* $x + \alpha h \in C$ *then there exists* $\beta \in I\!R$ *such that* $x + \beta\bar{h} = K(x + \alpha h)$*.*

Proof: Let us compute $K(x + \alpha h) - x$.

$$K(x + \alpha h) - x \;=\; \frac{x + \alpha h}{a'(x + \alpha h)} - x$$

Since $a'x = 1$, clearing denominators gives

$$K(x + \alpha h) - x \;=\; \frac{1}{1 + \alpha a'h}(\alpha h - \alpha a'hx) \;=\; \frac{\alpha}{1 + \alpha a'h}\bar{h}\ .$$

Setting $\beta = \alpha/(1 + \alpha a'h)$ completes the proof.

Lemma 2.6 *Consider a direction* $h \in C$ *from* $x \in S$*, and set* $\bar{h} = K_x(h)$*. Then*

$$\inf_{\beta \geq 0}\{f(x + \beta\bar{h}) \mid x + \beta\bar{h} > 0\} \;\leq\; \inf_{\alpha \geq 0}\{f(x + \alpha h) \mid x + \alpha h > 0\}\,. \quad (7)$$

Proof: immediate consequence of lemma 2.5.

Definition 2.7 *Two directions* $h^1, h^2 \in C$ *are* equivalent from $x \in S$ *if their conical projections are collinear.*

At this point we must make some comments on equivalence. C is always an unbounded set, and the line $\{x + \alpha h \mid \alpha \geq 0\}$ may be unbounded in C. Then the 'inf' in the right-hand side of (7) cannot be replaced by 'min' unless we know that a minimum exists. This is not relevant, since the directions that really matter are in D (the conical projections), and we do not need a minimum on the right-hand side. The 'inf' in the left-hand side is more important, and we need conditions to guarantee a the existence of a minimum. The most usual is to assume that S is bounded, but weaker assumptions were made in [2] and [11]. These references describe algorithms in which the cost decreases in all iterations, and will be commented ahead.

Another remark is in place regarding equivalence: h^1 equivalent to h^2 does not mean that the line search along both directions give the same result: it means that the same result is obtained if we use their conical projections. These details are actually irrelevant in general, since most algorithms guarantee the existence of minimizers on both sides of (7). Special cases appear only for unbounded feasible sets, and will be treated separately in this paper.

We now present the two most important lemmas on conical projections.

Lemma 2.8 *Two directions $h^1, h^2 \in C$ are equivalent from a point $x \in C$ if and only if there exist real numbers α, β , $\alpha > 0$ such that $h^2 = \alpha h^1 + \beta x$.*

Proof:
($\Leftarrow$) Assume that $h^2 = \alpha h^1 + \beta x$, $\alpha > 0$. Using definition 2.4,

$$\begin{aligned} K_x(\alpha h^1 + \beta x) &= \alpha h^1 + \beta x - a'(\alpha h^1 + \beta x)x \\ &= \alpha(h^1 - a'h^1 x) \qquad \text{since } a'x = 1 \\ &= \alpha K_x(h^1) \end{aligned}$$

and the conical projections are collinear.

($\Rightarrow$) Assume that $K_x(h^2) = \alpha K_x(h^1)$, $\alpha > 0$. Then, using definition 2.4,

$$h^2 - a'h^2 x = \alpha h^1 - \alpha a'h^1 x \qquad \text{or}$$

$$h^2 = \alpha h^1 + (a'h^2 - \alpha a'h^1)x \ ,$$

completing the proof.

Consider now an arbitrary vector $d \in \mathbb{R}^n$ and its projections d_p , d_Q and d_a respectively onto D , Q and $Null(a')$.

Lemma 2.9 *Given a vector $d \in \mathbb{R}^n$, consider the direction d_Q and a point $x \in C$. Then*

$$K_x(d_Q) = d_p - a'dx_p \ . \tag{8}$$

Proof: Using the definition,

$$K_x(d_Q) = d_Q - a'd_Q x \ .$$

But for $a \in Q$, $a'd_Q = a'd$, and

$$K_x(d_Q) = d_Q - a'dx \ .$$

Now project both sides orthogonally onto $D = Null(A)$. The left hand side is already in D ; since $D = Q \cap Null(a')$ and $a \in Q$, it is easy to see that $(d_Q)_p = d_p$, and the projection coincides with (8), completing the proof.

This result means that $K_x(d_Q)$ is a linear combination of the projections of the vectors d and x onto D . Conically projected directions can then be computed directly from orthogonal projections, without explicit knowledge of the subspace Q : only the vector a must be known, but this is easy by lemma 2.1. This result will be used to construct projective algorithms without transforming the original linear programming problem. In particular, if a problem is scaled, then $x = e$ and the conically projected direction depends only on the projected vectors c_p and e_p , since by (4) a depends only on e_p and e .

2.3 Trust regions and search directions

Consider a general non-linear programming problem

$$\text{minimize}\{f(x) \mid x \in C \subset \mathbb{R}^n\}.$$

Most non-linear programming algorithms evolve by constructing a sequence of points (x^k) in C , such that x^{k+1} is obtained by a line search along a feasible direction from x^k .

The procedures to obtain a search direction h can usually be interpreted as a trust region minimization of some approximation of $f(\cdot)$ about x^k . More specifically, consider a function $g(\cdot)$ such that $g(\Delta x) \approx f(x + \Delta x)$ and a set T that contains the origin in its interior and such that the approximation is good in T (that is the origin of the name 'trust region'). Then a promising direction to decrease the values of $f(\cdot)$ is computed by

$$h = \text{argmin}\{g(\Delta x) \mid \Delta x \in T\}$$

The most typical trust region is a sphere, and the most typical approximations are linear, quadratic and exact ($g(\cdot) = f(\cdot)$). A linear approximation in a spherical region generates the steepest descent direction; a second order approximation is used in the most usual "trust region methods", and the Newton-Raphson direction corresponds to the limiting case of $T = \mathbb{R}^n$.

Ellipsoidal trust regions can also be used, but the trust region minimization becomes more difficult. In all linear programming algorithms considered in this paper spherical trust regions are used after scaling: this is equivalent to an ellipsoidal trust region minimization without scaling.

One should note that there is no good reason (besides easy computation) to adhere to spherical trust regions: if the region can be increased to better approximate the shape of the feasible set and still keep a low computational cost, this should be done. This is the reasoning that will lead us to bi-directional search procedures instead of simple line searches.

3 Projective algorithms

Projective algorithms consist of a "master algorithm" that at each iteration starts with a feasible point, (possibly) scales the problem about this point and calls an "internal algorithm". This internal algorithm constructs a zero-degree homogeneous function $f(\cdot)$ and an internal problem in the format

$$\begin{array}{rrcl} \text{minimize} & f(x) & & \\ \text{subject to} & x & \in & C \\ & a'x & = & 1 \end{array} \tag{9}$$

using the sets defined in (2). An initial point $\bar{x}$ is available, and if scaling was used $\bar{x} = e$.

This structure is described in [11], and the scope of each iteration is to improve the value of the objective by finding a point x such that $f(x)$ is significantly smaller than $f(\bar{x})$. The essence of conical projection algorithms is in the fact that (9) can be rewritten as

$$\begin{array}{rrcl} \text{minimize} & f(x) & & \\ \text{subject to} & x & \in & C \end{array} \tag{10}$$

This is the same as (9), without the constraint $a'x = 1$. The feasible set is now very simple (the subspace Q with positivity constraints), and to each point in C we can associate its conical projection with the same value for the objective function. To each descent direction from $\bar{x}$ we can associate its conical projection from $\bar{x}$. Conical projection methods will then improve the value of the objective function for (10) and then translate this result to (9) by conical projection.

The reason why conical projection methods should be more efficient than methods based on a spherical trust region on S is that a sphere in the cone

C is mapped by conical projection onto a larger ellipsoid in S . This ellipsoid can be roughly visualized as the shadow caused by the sphere when S is illuminated by a light source at the origin.

We shall study three different objective functions: the multiplicative potential function, its logarithm – the logarithmic potential function and the "natural" conic function. For each of these functions, a descent direction will be obtained by a spherical trust region minimization in C . The final feasible direction will be the conical projection $K_x(h)$.

The main results: most of the section will be devoted to show that all possible spherical trust region minimizations involving first or second order approximations of both potential functions result in equivalent directions, all reproducing Karmarkar's direction. The conical projection of these directions is given by a simple expression to be derived in lemma 3.1, and is a combination of the directions c_p and e_p .

The natural conic function is also studied, and an exact minimization in a spherical trust region also results in a combination of those two directions in D . Similar conclusions will be obtained in the next section for affine methods, suggesting bi-directional search procedures to be explored in the last section.

3.1 Potential functions – Karmarkar's algorithm

The criterion used in Karmarkar's algorithm is the potential function

$$f(x) = n \log(c'x - v) - \sum_{i=1}^{n} \log x_i \tag{11}$$

where v is a lower bound for the value $\hat{v}$ of an optimal solution for (1). Note that for $x \in S$, $c'x - v > 0$ and $f(\cdot)$ is well defined. Since $f(\cdot)$ is in general not zero-degree homogeneous, it can be rewritten as

$$f(x) = n \log \bar{c}'x - \sum_{i=1}^{n} \log x_i \tag{12}$$

where $\bar{c} = c - va$. Since $a'x = 1$ in S , both expressions are equivalent in S and $f(\cdot)$ is now zero-degree homogeneous.

Let $X = diag(x_1, \ldots, x_n)$. The derivatives for the potential function are given by:

$$\begin{array}{rclrcl} \nabla f(e) & = & \frac{n}{\bar{c}'e}\bar{c} - e & \nabla f(x) & = & \frac{n}{\bar{c}'x}\bar{c} - [x_i^{-1}] \\ \nabla^2 f(e) & = & -\frac{n}{(\bar{c}'e)^2}\bar{c}\bar{c}' + I & \nabla^2 f(x) & = & -\frac{n}{(\bar{c}'x)^2}\bar{c}\bar{c}' + X^{-2} \end{array} \tag{13}$$

References [10] and [19] show that Karmarkar's direction is the steepest descent direction for $f(\cdot)$ on Q from $x = e$, given by $-\nabla f_Q(e)$, or equivalently

$$h = -\bar{c}_Q + \frac{\bar{c}'e}{n}e \tag{14}$$

This direction is in Q. The actual direction in D is obtained by computing the conical projection of h.

Lemma 3.1 *Karmarkar's direction for problem (1) from the point e is given by*

$$h^0 = -c_p + \frac{1}{\|e - e_p\|^2}(c'(e - e_p) - v)e_p \tag{15}$$

Proof: Using lemma 2.8, it is immediate to see that the steepest descent direction h (14) is equivalent to $-\bar{c}_Q$. Computing the conical projection for this direction using (8),

$$h^0 = K_e(-\bar{c}_Q) = -\bar{c}_p + a'\bar{c}e_p$$

Substituting now $\bar{c} = c - va$ and noting that $a_p = 0$,

$$h^0 = -c_p + a'ce_p - \|a\|^2 ve_p$$

Finally, substitute the expression for a from (4) to get

$$h^0 = -c_p + c'\frac{e - e_p}{\|e - e_p\|^2}e_p - \frac{1}{\|e - e_p\|^2}ve_p$$

Simplifying this expression leads to (15) and completes the proof.

Expression (15) gives Karmarkar's direction for the linear programming problem in the general format (1), with no use of problem transformations.

The lemma to follow shows that this and all other directions obtained by spherical trust region minimizations of approximations of $f(\cdot)$ are equivalent.

Lemma 3.2 *The following directions from* e *are equivalent for the logarithmic potential function:*
(i)$-\nabla f_Q(e)$ *: Karmarkar's direction*
(ii)$-\bar{c}_Q$
(iii)argmin$\{\Delta x'\nabla f(e)+\frac{1}{2}\Delta x'\nabla^2 f(e)\Delta x \mid \Delta x \in Q, \|\Delta x\| \leq \alpha\}$ *for any* $\alpha > 0$*: trust region minimization of the quadratic approximation for any spherical trust region.*

Proof: If e is an optimal solution for (9) then the result is trivial. Assume then that $\nabla f_Q(e) \neq 0$.

(i) The equivalence between $-c_Q$ and h is an immediate consequence of lemma 2.8.

(ii) Consider the second order approximation for $f(\cdot)$:

$$\begin{aligned} g(\Delta x) &= \Delta x'\nabla f(e) + \frac{1}{2}\Delta x'\nabla^2 f(e)\Delta x \qquad (16) \\ \nabla g(\Delta x) &= \nabla f(e) + \nabla^2 f(e)\Delta x \end{aligned}$$

Let $h = \text{argmin}\{g(\Delta x \mid \Delta x \in Q, \|\Delta x\| \leq \alpha)\}$, $\alpha > 0$. Then the following facts are true:

— h is a descent direction for $g(\cdot)$: this is well known for a quadratic function, since $g(h) - g(-h) = 2h'\nabla f(e)$. Since $g(h) < g(-h)$, necessarily $h'\nabla f(e) < 0$

— By an immediate application of Kuhn-Tucker theorem on the subspace Q ,

$$\nabla g_Q(h) + \lambda h = 0 \qquad \text{for some } \lambda \geq 0 \qquad (17)$$

Developing the expression (16) with help of (13),

$$\nabla g(h) = \frac{n}{\bar{c}'e}\bar{c} - e - \frac{n}{(\bar{c}'e)^2}\bar{c}'h\bar{c} + h$$

Projecting onto Q and using (17),

$$\mu\bar{c}_Q - e + h + \lambda h = 0 \qquad \text{with } \mu = \frac{n}{\bar{c}'e} - \frac{n}{(\bar{c}'e)^2}\bar{c}'h$$

or

$$(1+\lambda)h = -\mu\bar{c}_Q + e$$

Now note that $\mu \neq 0$. This is true because if $\mu = 0$ then $h = e$ and this is not a descent direction, since $\nabla f(e)'e = 0$.

It follows from lemma 2.8 that h is equivalent either to $\bar{c}_q$ or to $-\bar{c}_q$, depending on the sign of μ. But h is a descent direction, and thus can only be equivalent to $-\bar{c}_Q$, since by (i) this last direction is equivalent to the steepest descent direction. This completes the proof.

The multiplicative potential function: Karmarkar's potential function is the logarithm of the multiplicative potential defined by

$$p(x) = \frac{(\bar{c}'x)^n}{\prod_{i=1}^n x_i} . \tag{18}$$

Problem (9) is obviously equivalent for both potential functions as objectives, since the logarithm function is monotonically increasing, but the same is not trivial with respect to the trust region minimizations. While $f(\cdot)$ is not convex (and actually strictly concave along the direction $-\bar{c}_Q$) and thus not adapted to the Newton-Raphson method, $p(\cdot)$ is convex. This was proved by Iri and Imai [13] for a different formulation of the linear programming problem, but the equivalence to our format is immediate.

The lemma to follow will show that unfortunately nothing can be gained by choosing the multiplicative instead of the logarithmic potential, or even by using Newton-Raphson algorithm. First let us write the expressions for the derivatives of $p(\cdot)$ at e.

$$\nabla p(e) = p(e)\nabla f(e) \quad , \quad \nabla^2 p(e) = p(e)(\nabla^2 f(e) + \nabla f(e)\nabla f(e)') \tag{19}$$

Lemma 3.3 *The following directions from e are equivalent for the multiplicative potential function:*
(i)$-\nabla f_Q(e)$ *: Karmarkar's direction*
(ii)$-\nabla p_Q(e)$
(iii)$\text{argmin}\{\Delta x'\nabla p(e) + \frac{1}{2}\Delta x'\nabla^2 p(e)\Delta x \mid \Delta x \in Q, \|\Delta x\| \leq \alpha\}$ *for any* $\alpha > 0$*: trust region minimization of the quadratic approximation for any spherical trust region.*
(iv)A Newton-Raphson direction, if any exists.
(v) A Newton-Raphson direction restricted to D .

Proof: The equivalence between (i) and (ii) is trivial, since the directions are collinear. We must prove an equivalence for (iii).

Following the same procedure as in the proof of lemma 2.1, construct

$$g(\Delta x) = \Delta x'\nabla p(e) + \frac{1}{2}\Delta x'\nabla^2 p(e)\Delta x$$

Taking the gradient and substituting (19)

$$\frac{1}{p(e)}\nabla g(\Delta x) = \nabla f(e) + \nabla^2 f(e)\Delta x + \nabla f(e)'\Delta x \nabla f(e)$$

If h solves (iii), then again using Kuhn-Tucker theorem and (13),

$$\frac{n}{\bar{c}'e}\bar{c}_Q - e + h - \frac{n}{(\bar{c}'e)^2}\bar{c}'h\bar{c}_Q + \nabla f(e)'h\frac{n}{\bar{c}'e}\bar{c}_Q + \lambda h = 0 \qquad (20)$$

Again an expression with the format $(1+\lambda)h = -\beta\bar{c}_Q + e$ is obtained, and h is equivalent to $-\bar{c}_Q$ by lemma 2.8.

(iv) A Newton-Raphson direction is a solution of (20) with $\lambda = 0$. The equivalence to (iii) is then trivial.
If h is a Newton-Raphson direction, then we can easily see that $h + \alpha e$ is also a Newton-Raphson direction, for any real α, since $\nabla p(e)'e = 0$.
In particular, for $h + \alpha e \in S$, the resulting direction is the conical projection of h onto S, and must be a Newton-Raphson direction restricted to D. This argument shows (v) and completes the proof.

Iri and Imai [13] studied the multiplicative potential function for a problem stated with inequality constraints. Their conclusions are easily transported to our formulation by the introduction of slack variables, and we conclude that $p(\cdot)$ is strictly convex on S whenever the origin is not an optimal solution to (1). The direction is then given by the unique Newton-Raphson direction restricted to D. Using the lemma above, the resulting direction is equivalent to Karmarkar's direction, and the algorithms are equivalent. Following Iri and Imai, we conclude that if the optimal value for (1) is known and $v = \hat{v}$, then Karmarkar's algorithm is quadratically convergent.

Directions of decreasing cost: Consider the transformed problem (9) and the Karmarkar direction from e, $h^0 = K_e(h)$. It is well known that h^0 can be a direction of increasing cost, and modifications of this direction to avoid this behavior were proposed by Padberg [17], Anstreicher [2] and in [11]. Following this last reference, whenever h^0 is a direction of decreasing cost, a modified direction

$$\hat{h} = h - za \qquad , z > 0 \tag{21}$$

can be found such that $c'K_e(\hat{h}) \leq 0$. Reference [11] shows and explicit expression to achieve equality, and a straightforward algorithm to achieve inequality while preserving the same polynomial bound as Karmarkar's algorithm. The result is very simple: (21) corresponds to an increase in the value of the parameter v . The lower bound must be increased in the expression for h^0 until this direction becomes a direction of decreasing cost (and v may become larger than $\hat{v}$).

These modified directions are important to deal with unbounded feasible sets. If we guarantee that the cost decreases strictly along the search directions, then convergence in values can be guaranteed for unbounded feasible sets. If we guarantee non-increasing costs, then Anstreicher [2] shows that only the optimal set must be bounded to guarantee Karmarkar's polynomial bound.

Steepest descent without scaling: Whenever the problem is rescaled a new projection matrix must be computed. It may be convenient to do steepest descent iterations without rescaling to get the most of the last projection matrix. In this case the steepest descent is not equivalent to second order descent directions. A complete algorithm using such iterations is described in [11].

The steepest descent direction from a point x will be proportional to $-\nabla f(x)$, expanded in (13)

$$h = -\bar{c}_Q + \frac{c'x - v}{n} + x_Q^- \qquad , x^- = [x_i^{-1}] \tag{22}$$

To obtain the conical projection of h , we can use (8), again noting that $a_p = 0$. The simplified expression results in

$$h^0 = K_x(h) = -c_p + a'[c - va - wx_p^-]x_p + wx_p^- \tag{23}$$

where

$$w = \frac{c'x - v}{n} \qquad \text{and } x_p^- = (x^-)_p$$

The computation of (23) needs only the computation of one projection $x_p^- = Px^-$ with the available matrix P . The vector x_p is computed from $x_p = e_p + x - e$ since $x - e \in D$

3.2 The conic function

We now study the zero-degree homogeneous function defined on C by

$$f(x) = \frac{c'x}{a'x} \tag{24}$$

Functions of this form were first used by Davidon [4], in a different context. $f(\cdot)$ is well defined, since $a'x > 0$ for any $x \in C$ by lemma 2.1. It is a zero-degree homogeneous function and for any $x \in C$, $f(x) = c'K(x)$.

An interesting feature of this function is that a spherical trust region minimization can be easily solved exactly. Consider then the trust region minimization for initial point e :

$$\text{minimize}\{f(e+h) \mid h \in B\} \tag{25}$$

where

$$B = \{h \in I\!R^n \mid h \in Q, \|h\| \leq 1\}$$

This problem is equivalent to

$$\text{minimize}\{c'x \mid x \in K(B)\}$$

where $K(B)$ is the set of conical projections of points in B . The set $K(B)$ is the intersection of a cone and a hyperplane, and therefore an ellipsoïd. We conclude that problem (25) has a unique solution that must be on the boundary of B , corresponding to a point on the boundary of the ellipsoid.

The ellipsoid can be much larger than the sphere $B \cap S$ that would be the trust region in the affine gradient projection method (see section 4.1 ahead), and therefore the resulting direction can be more efficient than the projected gradient direction.

The analytical solution of the trust region minimization is summarized in the following lemma:

Lemma 3.4 *The trust region minimization problem (25) has a unique solution h whose conical projection is given by*

$$h^1 = -c_p + \frac{1}{\|e - e_p\|^2}\left(\frac{\|c_p\|}{\beta} - c_p'e\right)e_p \tag{26}$$

where $\beta \in (0,1)$ is the positive root of the second order equation

$$(k^2 + \|e - e_p\|^2)\beta^2 - 2k\beta + 1 - \|e - e_p\|^2 = 0 \qquad , k = \frac{c_p'e}{\|c_p\|}$$

Proof: From the reasoning above, the problem has a unique solution h , $\|h\| = 1$. The point $e+h$ is in the boundary of the ball, and consequently the ray $\{\lambda(e+h) \mid \lambda > 0\}$ is tangent to the ball (otherwise h would not be unique).
It follows that $h \perp e+h$, or

$$h'e \;=\; -h'h \;=\; -1 \tag{27}$$

Now consider the set

$$P \;=\; \{e+d \mid A(e+d) = A(e+h)\} \;=\; \{e+d \mid Ad = Ah\}$$

Now, h must solve

$$\text{minimize}\{f(e+d) \mid e+d \in P, \|d\| \leq 1\}$$

This set is a "slice" of the ball B parallel to S. In particular, for $e+d \in P$, $a'(e+d) = a'(e+h)$, and the denominator of $f(e+d) = c'(e+d)/(a'(e+d))$ is constant in Q . It follows that h must solve

$$\text{minimize}\{c'(e+d) \mid Ad = Ah, \|d\| \leq 1\}$$

The solution for this problem (minimization of a linear function in a ball) must satisfy

$$h_p = -\beta\frac{c_p}{\|c_p\|} \qquad \text{, for some } \beta > 0 \tag{28}$$

But $h_p = (h_Q)_p = h_a$, since $h \in Q$, and by definition of projection,

$$h = h_a + \alpha a \qquad , \|h\|^2 = \|h_a\|^2 + \alpha^2\|a\|^2$$

for some real α. Substituting (28) into these two equalities, and noting that $\|h\| = 1$,

$$h = \alpha a - \beta\frac{c_p}{\|c_p\|} \qquad , \tag{29}$$

$$\beta^2 + \alpha^2\|a\|^2 = 1 \qquad . \tag{30}$$

Merging now (27) and (29),

$$h'e \;=\; \alpha a'e - \beta\frac{c_p'e}{\|c_p\|} \;=\; -1 \qquad .$$

Defining $k = c_p'e/\|c_p\|$, and noting that $a'e = 1$,

$$\alpha = k\beta - 1 \tag{31}$$

Computing now α^2 from (31) and substituting into (30), we arrive to the expression

$$(\|a\|^2 k^2)\beta^2 - 2k\|a\|^2\beta + \|a\|^2 - 1 = 0 \qquad ,$$

and the final result is obtained by setting $\|a\| = \|e - e_p\|^{-1}$ from (4).
Since we know that the solution is unique and that $\beta > 0$ from (28), β must be the positive solution to the equation, and the proof is complete.

3.3 Conclusions for projective methods

Summing up the results in this section, we end with very simple results. Projective methods are built without changing the original problem. All directions based on potential functions are equivalent, and produce the direction (15), repeated here:

$$h^0 = -c_p + \frac{c'(e-e_p) - v}{\|e-e_p\|^2} e_p$$

The lower bound: In this expression v is a lower bound to an optimal solution of (1). To find a lower bound following the method in Todd and Burrell [19] and in [11], we must find v such that the vector $\bar{c}_Q = c_Q - va$ has a non-positive component.
Using lemma 2.2,

$$\bar{c}_Q = c_p + (c'(e-e_p) - v)\frac{e-e_p}{\|e-e_p\|^2} \tag{32}$$

If a value of v is available such that $\bar{c}_Q$ has a nonpositive component, then v is a lower bound for $\hat{v}$. Otherwise, a lower bound v is found by a simple ratio test to solve maximize$\{v | \bar{c}_Q \geq 0\}$:

$$v = \min_{i=1,\ldots,n}\left\{\frac{\|e-e_p\|^2 c_{p_i}}{(e-e_p)_i} + c'(e-e_p) \mid (e-e_p)_i > 0\right\} \tag{33}$$

Guaranteed descent for the cost function is not assured by the direction h^0, but it can be obtained by increasing the value of v until $c'h^0$ becomes negative, and polynomial complexity can still be obtained.

The other direction obtained by a projective method is the "natural" conical projection direction h^1 in (26). This does not depend on lower bounds for an optimal solution, but does not guarantee polynomial complexity.

We saw then that there are only two different directions coming from primal feasible direction projective methods. Both have the format

$$-c_p + \beta e_p \qquad ,$$

a linear combination of the directions $-c_p$ and e_p . The extension to bi-directional search procedures is immediate, and will be studied in section 5, after examining affine methods.

4 Affine methods

Affine methods are based on directions obtained by spherical trust region minimizations on $D = Null(A)$. Again we assume that the problem has been rescaled to bring the current point to e .

4.1 Projected gradients

This method was immediately considered after the publication of Karmarkar's algorithm, and uses the direction

$$h^2 = -c_p \tag{34}$$

Studies of this direction were made by Vanderbei, Meketon and Freeman [21], Cavalier and Soyster [3] and a computer code was built by Adler, Resende and Veiga [1]. The line search along $e + \lambda h^2$ is a simple ratio test, and the next point is chosen by a heuristic rule to avoid approaching too much the boundary of $\mathbb{R}^n_+$.

4.2 Barrier methods

Barrier function methods work with the penalized function

$$x \in S \mapsto f_\epsilon(x) = c'x - \epsilon \sum_{i=1}^{n} \log x_i \qquad , \tag{35}$$

where ϵ is the penalty parameter. They were first introduced by Frisch [6], and developed by Fiacco and McCormick [5]. The similarity between Karmarkar's potential function and a barrier function method was immediately noticed, and the use of a gradient method for this function was proposed by Gill and al. [8].

The derivatives of the penalized function are given by

$$\nabla f_\epsilon(e) = c - \epsilon e \qquad , \nabla^2 f_\epsilon(e) = \epsilon I \tag{36}$$

and the steepest descent direction from e is given by

$$h^3 = -\frac{1}{\epsilon} c_p + e_p \tag{37}$$

Since the Hessian matrix is a multiple of the identity, $f_\epsilon(\cdot)$ is strictly convex and the steepest descent h^3 coincides with the Newton-Raphson step, as well as with the result of any spherical trust region minimization of the second order approximation.

There is actually no need for scaling the problem, since the Newton-Raphson direction can be computed directly for any point in S , but this computation is precisely equivalent to scaling and projecting [11].

Barrier methods need at each iteration a value for the parameter ϵ . The master algorithm generates a sequence of values ϵ_k , and it is known that the minimizers of the corresponding sequence of penalized problems converge to an optimal solution of (1) under our assumptions. It is not easy to design a strategy to reduce the values of the penalty parameter: large reductions tend to reproduce the projected gradient method, while small reductions produce path-following algorithms as we shall see below. The method is closely related to Karmarkar's algorithm, and the values of ϵ are related to the values of the lower bounds v used by that method (we shall also comment on this below).

4.3 Homotopy algorithms

The minimizers of the penalized function (35) for ϵ varying between $+\infty$ and 0 describe a smooth trajectory that has been known for a long time. It was described by Fiacco and McCormick [5] for non-linear programming problems, and recently by Megiddo [15] for linear programming. An algorithm to solve (1) by following this trajectory was proposed by the author in [9], resulting in a complexity of $O(n^3L)$ arithmetical operations with L bits of precision. The algorithm proceeds by changing ϵ by small steps $\epsilon_{k+1} = (1-\sigma)\epsilon_k$, $\sigma \leq 0.05$ in each iteration, with improvements obtained by one Newton-Raphson search per iteration. The algorithm is essentially the classical barrier function method, and the originality is in the strategy used to update the penalty parameter.

Other homotopy equations can be considered, and again we will show that several variations result to be equivalent. Consider the following functions, each dependent on a parameter:

$$x \in \mathbb{R}^n_+ \mapsto n \log(c'x - v) - \sum_{i=1}^{n} \log x_i \tag{38}$$

$$x \in \mathbb{R}^n_+, c'x < J \mapsto \frac{n}{J - c'x} - \sum_{i=1}^{n} \log x_i \tag{39}$$

$$x \in \mathbb{R}^n_+, c'x < K \mapsto -n \log(K - c'x) - \sum_{i=1}^{n} \log x_i \tag{40}$$

(38) corresponds to Karmarkar's criterion, and the other two correspond to the problem of finding centers for the regions in which they are defined. For (40), the problem of centers is

$$\text{maximize}\{\log(K - c'x) + \sum_{i=1}^{n} \log x_i \mid x \in S, c'x < K\}$$

This problem was studied by Renegar [18], who by the first time designed an algorithm to solve (1) in $O(n^{0.5}L)$ iterations. The same problem was studied by Vaidya [20], who proved a bound of $O(n^3L)$ operations for a path-following algorithm.

Each of the functions above contains a parameter, and varying these parameters generates solution paths for the respective minimization problems. Next lemma shows that these paths coincide.

Lemma 4.1 *The solution paths associated to the functions (35), (38), (39) and (40) coincide. If x minimizes the penalized function (35) for a parameter ϵ, then x minimizes the other functions for parameters given respectively by*

$$v = c'x - n\epsilon \quad , \quad J = c'x + \sqrt{n\epsilon} \quad , \quad K = c'x + n\epsilon \tag{41}$$

Proof: The minimizers for all functions correspond to points in which the projected gradients vanish. All we need to do is to compare the projected gradients for all functions at a point x. They are, for the functions in the order presented above:

$$c_p - \epsilon x_p^-$$

$$\frac{n}{c'x - v} c_p - x_p^-$$

$$\frac{n}{(J - c'x)^2} c_p - x_p^-$$

$$\frac{n}{K - c'x} c_p - x_p^-$$

It is immediate to check that the substitution of the parameter values in (41) reduce all four expressions above to identical formulas, and consequently with the same solution, completing the proof.

This lemma shows that again, there are few possible variations on algorithms to follow these trajectories, since they coincide and the projected gradients are similar. The barrier function is the simplest of these functions, and is strictly convex. Karmarkar's function is not convex, and can impose difficult numerical problems in the line-search procedures; the methods of centers are limited by the need to keep the present iterate feasible ($K > c'x^k$ for Renegar's function), thus preventing great reductions in cost per iteration. The barrier function approach is the the most promising, since bold variations of the parameter can be used without serious numerical problems.

A complete algorithm to follow the center trajectory can be stated now, but we shall first look at bidirectional search procedures, and a further step in the equivalence of algorithms.

5 Bi-directional search procedures

Summing up the results of the previous sections, all studied algorithms are composed by iterations with the following structure:

Consider (with some abuse of notation) Karmarkar's potential function $f_v(\cdot)$ and the barrier function $f_\epsilon(\cdot)$, with parameters respectively denoted by v and ϵ .

Start at point e.

Choose an objective function ($f_v(\cdot)$, $f_\epsilon(\cdot)$, $c'x$).

Choose a parameter value (v, ϵ) if one is used.

Choose a direction of the form $h = -c_p + \alpha e_p$.

Do a line search along h to minimize the chosen objective along the chosen direction.

The line search procedure examines points $e - \lambda c_p + \lambda\alpha e_p$, with fixed α, reflecting the fact that fixed spherical trust regions were used in all the algorithms. But there is no reason (besides simplicity) to use spherical regions. A still simple but expanded trust region corresponds in each case to the bi-directional search on the bi-dimensional region defined by

$$B = \{e - \alpha c_p + \beta e_p \mid \alpha \geq 0, \beta \in \mathbb{R}, e - \alpha c_p + \beta e_p \in \mathbb{R}^n_+\} \qquad (42)$$

The search region is now the same for all methods surveyed by us. The only difference is in what to minimize. All methods are improved, since the region in which the minimization is done always increases.

Some insight in the role of the parameters is gained by the following lemma:

Lemma 5.1 *Let $x = e - \alpha c_p + \beta e_p$ be a point in B. Then x minimizes $f_v(\cdot)$ in B if and only if x minimizes $f_\epsilon(\cdot)$ in B with*

$$\epsilon = \frac{c'x - v}{n} \qquad (43)$$

Proof: The proof is essentially the same as that for lemma 4.1: choosing the value of ϵ given by (43), the gradients of both functions at x become collinear, and the optimality conditions on B coincide.

The lemma above asserts that given the parameter for one of the methods there exists a value for the parameter in the other method such that the search gives the same results for both methods. But it does not say how to relate the parameters without knowing the solution of the bi-directional search. We end up with two different methods, but the only difference is in the choice of the parameter.

There is a natural way of choosing a lower bound at each iteration of Karmarkar's method, as we commented before (33), but there is no guarantee neither that this is a good lower bound nor that even for $v = \hat{v}$ the resulting improvement is good.

There is also a natural way of choosing the parameter ϵ in a path–following algorithm, and we shall show it in our last section.

The bi-directional search procedures: minimizing one of the three criteria in B is an optimization problem in two dimensions. If the criterion is cost, then it is a linear programming problem, which can easily be solved by heuristic techniques based on geometric reasoning. For the other criteria, it is easy to compute derivatives along the variables α and β, and use a Newton or Quasi-Newton method.

A complete bi-directional search for the barrier function will be presented in the next section.

6 The complete path following algorithm

We are ready to present a complete path following algorithm using the barrier function and a bi-directional search solved by a quasi-Newton algorithm. The master algorithm is the same as in [12]. Each iteration scales the problem, chooses a value for ϵ and performs a bi-directional search to reduce $f_\epsilon(\cdot)$. The key point is the choice of ϵ, to be discussed in detail below. We assume that an initial feasible point x^0 is given.

Algorithm 6.1 *Path-following : given* $x^0 \in S$, $r \in (0,1)$, $\delta > 0$.

$k := 0$

Repeat

Scale the original problem about x^k obtaining problem (1) with initial point $D^{-1}x^k = e$, with $D = diag(x_1^k, \ldots, x_n^k)$.

Compute the projection matrix P onto the feasible set and set

$$c_p := Pc \quad , \quad e_p := Pe$$

Compute ϵ by algorithm 6.2 below.

Use a bi-directional search algorithm (see below) to find a new point $y \in S$ such that $f_\epsilon(y) < f_\epsilon(e)$

Return to the original space with $x^{k+1} = Dy$

$k := k + 1$

Until $\epsilon < \delta$

The choice of ϵ :

Reference [12] shows that good results may be expected if one tries to follow as closely as possible the central trajectory

$$\{x_\epsilon \in S \mid x_\epsilon > 0,\ x_\epsilon \text{ minimizes } f_\epsilon(\cdot) \text{ on } S\}$$

The choice of the penalty parameter will be based on an estimate of the distance from e to this trajectory.

Consider the scaled problem and the feasible point e. At this point the penalized function for a given $\epsilon > 0$ has

$$\nabla f_\epsilon(e) = c - \epsilon e \qquad , \qquad \nabla^2 f_\epsilon(e) = \epsilon I$$

The projected Newton Raphson step h_ϵ from e is the given by

$$P\nabla f_\epsilon(e) + \epsilon I h_\epsilon = 0 \qquad \text{, or}$$

$$h_\epsilon = -\frac{c_p}{\epsilon} + e_p \tag{44}$$

Given a value $\epsilon > 0$, the Newton–Raphson direction h_ϵ corresponds to the minimization of the quadratic approximation of $f_\epsilon(\cdot)$, and consequently $e + h_\epsilon$ estimates the point x_ϵ, and $\|h_\epsilon\|$ estimates $\|e - x_\epsilon\|$. The distance from e to the central trajectory is then estimated by

$$d = \inf\{\|h_\epsilon\| \mid \epsilon > 0\} \tag{45}$$

If $c_p' e_p \leq 0$ then for any $\epsilon > 0$, $\|-\frac{c_p}{\epsilon} + e_p\| > \|e_p\|$, and $d = \|e_p\|$ for $\epsilon \to \infty$. In this case the only possible conclusion is that e is too far from the central trajectory for the estimate to be reliable. Otherwise, we shall be concerned with

$$\bar{\epsilon} = \operatorname{argmin}\{\|-\frac{c_p}{\epsilon} + e_p\| \mid \epsilon > 0\}$$

The solution to this problem is easily obtaining by differentiating the expression for $\|h_\epsilon\|^2$ and equating the derivative to zero:

$$2(-\frac{c_p}{\epsilon} + e_p)' c_p \frac{1}{\epsilon^2} = 0 \qquad \text{, or}$$

$$\bar{\epsilon} = \frac{\|c_p\|^2}{c_p' e_p} \qquad \text{, for } c_p' e_p > 0 \tag{46}$$

The algorithm below will examine these two possibilities.

Algorithm 6.2 *Computation of ϵ : given a reduction factor $r \in (0,1)$ (typically 0.1).*

If $c_p' e_p \leq 0$ then set $\epsilon := \|c_p\| / \|e_p\|$

Else set $\epsilon := r\bar{\epsilon}$, with $\bar{\epsilon}$ computed by (46)

In the algorithm above, if e is far from the central trajectory then ϵ is chosen to weight equally cost reduction and proximity to the "center" of the feasible set (the point that maximizes $\sum_{i=1}^{n} \log x_i$).

Otherwise, $\bar{\epsilon}$ is found corresponding to a point in the central trajectory near e, and the parameter is reduced for the next iteration. Small reductions of the penalty parameter produce well behaved sequences that follow the trajectory and guarantee polynomial complexity; bold reductions of ϵ may lead to fast convergence. A typical value of the reduction is $r = 0.1$, and it can be made adaptive: a deeper reduction can be used in an iteration if the former search produced a point very near the central trajectory.

An interesting remark about the computation of $\bar{\epsilon}$ is obtained by substituting (46) into (44) to get

$$h_{\bar{\epsilon}} = -\frac{c_p' e_p}{\|c_p\|^2} c_p + e_p$$

and then $c_p' h_{\bar{\epsilon}} = 0$. The Newton-Raphson direction associated to $\bar{\epsilon}$ is orthogonal to the cost, and corresponds to moving from e to a point on the central trajectory with the same cost as e.

The bi-directional search

The search problem corresponds to a minimization of the penalized criterion $f_\epsilon(\cdot)$ in the bidimensional region B defined in (42). Let us state this as a problem in two dimensions.

For each pair $(\alpha, \beta) \in I\!R^2$, define $p(\alpha, \beta) = e - \alpha c_p + \beta e_p$, and define

$$g_\epsilon(\alpha, \beta) = f_\epsilon(p(\alpha, \beta)) \qquad \text{for } (\alpha, \beta) \text{ such that } p(\alpha, \beta) > 0 \qquad (47)$$

Then the bi-directional search problem is

$$\text{minimize}\{g_\epsilon(\alpha, \beta) \mid \alpha > 0, p(\alpha, \beta) > 0\}$$

The derivatives of $g_\epsilon(\cdot)$ are :

$$\frac{\partial g_\epsilon(\alpha, \beta)}{\partial \alpha} = -\|c_p\|^2 + \epsilon \sum_{i=1}^{n} \frac{c_{p_i}}{p_i(\alpha, \beta)}$$

$$\frac{\partial g_\epsilon(\alpha,\beta)}{\partial\beta} = c_p' e_p - \epsilon\sum_{i=1}^{n}\frac{e_{p_i}}{p_i(\alpha,\beta)}$$

It is also easy to compute the hessian matrix:

$$H(\alpha,\beta) \;=\; \epsilon\sum_{i=1}^{n}\frac{1}{[p_i(\alpha,\beta)]^2}\begin{bmatrix} c_{p_i}^2 & -c_{p_i}e_{p_i} \\ -c_{p_i}e_{p_i} & e_{p_i}^2 \end{bmatrix}$$

Now any non-linear programming algorithm can be used to find the unique solution to the two-dimensional problem. The objective function is strictly convex, and Newton-Raphson algorithm can be used. We chose BFGS algorithm, with good preliminary results in about three iterations.

We shall not describe the algorithm, since it is well known. Each iteration chooses a direction $d = -H^{-1}\nabla g_\epsilon(\alpha,\beta)$ in the two-dimensional space, where H is either the hessian matrix or an approximation to it that is updated at each iteration. A line search in S along the direction $h = -d_1 c_p + d_2 e_p$ produces the new iterate. The line search can be efficiently implemented using an algorithm by Murray and Wright [16].

7 Conclusions

At this time only small problems have been solved by us using the path-following algorithm. These preliminary results show a very good behavior, but the affirmation that it is the best among the algorithms cited in this paper is still premature. It was definitely superior to all others, including the bi-directional versions of Karmarkar's algorithm for the small problems tested up to now, including the problem in Iri and Imai [13] with 29 variables (after introducing slacks) and 17 restrictions and some problems of similar size generated randomly.

The theoretical advantage of the method in relation to the others is in the sound interpretation of the parameter ϵ from the homotopy approach. This parameter is much more controllable than the lower bounds used in Karmarkar's algorithm, since there is no way of predicting how well the available methods to compute them will approach the optimal value of the problem.

References

[1] I. Adler, M. Resende, and G. Veiga. *An Implementation of Karmarkar's Algorithm for Linear Programming.* Report ORC86-8, Operations Research Center, University of California, Berkeley, May 1986.

[2] K. Anstreicher. *A Monotonic Projective Algorithm for Fractional Linear Programming.* Manuscript, Yale School of Organization and Management, New Haven, CT, November 1985.

[3] T. Cavalier and A. Soyster. *Some Computational Experience and a Modification of the Karmarkar Algorithm.* Working Paper 85-105, Dept. of Industrial and Management System Eng., Pennsylvania State Univ, February 85.

[4] W. C. Davidon. Conic approximations and collinear scalings for optimizers. *SIAM Journal on Numerical Analysis*, 17:268–281, 1980.

[5] A. Fiacco and G. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley and Sons, New York, 1955.

[6] K. R. Frisch. *The Logarithmic Potential Method of Convex Programming.* Memorandum, University Institute of Economics, Oslo, Norway, May 1955.

[7] D. Gay. *A Variant of Karmarkar's Linear Programming Algorithm for Problems in Standard Form.* Technical Report Numerical Analysis Manuscript 85-10, AT&T Bell Laboratories, Murray Hill, NJ, October 1985.

[8] P. Gill, W. Murray, M. Saunders, J. Tomlin, and M. Wright. *On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method.* Report SOL 85-11, Systems Optimization Laboratory, Dept. of Operations Research, Stanford University, Stanford, 1985.

[9] Clovis Gonzaga. *An Algorithm for Solving Linear Programming Problems in $O(n^3L)$ operations.* Memorandum UCB/ERL M87/10, Electronics Research Laboratory, University of California, Berkeley, CA, March 1987.

[10] Clovis Gonzaga. *A Conical Projection Algorithm for Linear Programming.* Memorandum UCB/ERL M85/61, Electronics Research Laboratory, University of California, Berkeley, CA, July 1985.

[11] Clovis Gonzaga. *Conical Projection Algorithms for Linear Programming.* Memorandum UCB/ERL M87/11, Electronics Research Laboratory, University of California, Berkeley, CA, March 1987.

[12] Clovis Gonzaga. *Search Directions for Interior Linear Programming Methods.* In preparation, University of California, Berkeley, CA, 1987.

[13] M. Iri and H. Imai. A multiplicative penalty function method for linear programming - another "new and fast" algorithm. In *Proc. of the 6th. Mathematical Programming Symposium*, Tokio, Japan, November 1985.

[14] N. Karmarkar. A new polynomial time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984.

[15] N. Megiddo. *Pathways to the Optimal Set in Linear Programming.* Research Report , IBM Almaden Research Center, S. Jose, California, 1986.

[16] W. Murray and M. Wright. *Efficient Linear Search Algorithms for the Logarithmic Barrier Function.* Report SOL 76-18, Dept. of Operations Research, Stanford University, Stanford, California, 1976.

[17] M. Padberg. *Solution of a Nonlinear Programming Problem Arising in the Projective Method for Linear Programming.* Manuscript, New York University, New York, NY, March 1985.

[18] James Renegar. *A Polynomial-time Algorithm Based on Newton's Method for Linear Programming.* Report MSRI 07 118 – 86, Mathematical Sciences Research Institute, Berkeley, California, June 1986.

[19] M. Todd and B. Burrell. *An extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables.* Technical Report 648, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, January 1985.

[20] Pravin M. Vaidya. *An Algorithm for Linear Programming which Requires $O(((m+n)n^2+(m+n)^{1.5}n)L)$ Arithmetic Operations.* preprint, AT&T Bell Laboratories, Murray Hill, NJ, 1987.

[21] R. J. Vanderbei, M. J. Meketon, and B. A. Freedman. A modification of Karmarkar's linear programming algorithm. *Algorithmica*, 1, 1987. to appear.