Note on Implementing the New IPM for LP Without Matrix Inversions

Katta G. Murty
Department of Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109-2117, USA
Phone: 734-763-3513, fax: 734-764-3451
e-mail:murty@umich.edu
Webpage: http://www-personal.engin.umich.edu/˜ murty/

December 2006

Abstract

We discuss improvments in a descent algorithm discussed in [9, 10] for solving a linear program (LP) without matrix inversions, and discuss techniques for implementing it.

Key words: Linear programming (LP), interior point methods (IPMs), solving LPs without matrix inversions.

1 Introduction

In [9, 10] a new predictor-corrector type interior point method (IPM) for linear programming (LP) has been dicsussed. It has the advantage of being a descent algorithm able to solve LPs without using matrix inversion operations. In this paper we discuss some improvements in that algorithm, and some techniques useful in implementing it. First we provide a summary of that algorithm. It considers LP in the form

$$\begin{aligned} \text{minimize} \quad & z(x) = cx \\ \text{subject to} \quad & Ax \geq b \end{aligned} \tag{1}$$

where A is an $m \times n$ data matrix, with a known initial interior feasible solution x^0 (i.e., $Ax^0 > b$). The rows of A, denoted by $A_{i.}$ for $i = 1$ to m, have been normalized, so $||A_{i.}|| = 1$ ($||.||$ denotes the Euclidean norm) for all $i = 1$ to m; also $||c|| = 1$. We will use the following notation:

$K =$	Set of feasible solutions of (1). We assume it is bounded.
$K^0 =$	$\{x : Ax > b\}$ = interior of K.
$\delta(x) =$	$\text{Min}\{A_{i.}x - b_i : i = 1 \text{ to } m\}$, defined for $x \in K^0$, it is the radius of the largest ball inside K with x as its center.
$B(x, \delta(x)) =$	Defined for $x \in K^0$, it is the largest ball inside K with x as its center.
$T(x) =$	Defined for $x \in K^0$, it is the index set $\{i :$ index i ties for the minimum in the definition of $\delta(x)\}$. The hyperplane $\{x : A_{i.}x = b_i\}$ is a tangent plane to $B(x, \delta(x))$ for each $i \in T(x)$, therefore $T(x)$ is called the **index set of touching constraints in (1)** at $x \in K^0$. See Figure 1.
$t_{\min}, t_{\max} =$	Minimum, maximum values of $z(x)$ over K respectively.
$\delta[t] =$	It is the $\text{Maximum}\{\delta(x) : x \in \{x : cx = t\}\}$, i.e., the maximum radius of the ball that can be inscribed inside K with its center restricted to $\{x : cx = t\}$. Notice the difference between $\delta(x)$ defined over K^0; and this $\delta[t]$ defined over the interval $[t_{\min}, t_{\max}]$ of the real line.
t^*	= Value of $t \in [t_{\min}, t_{\max}]$ that maximizes $\delta[t]$.
$\Gamma_1 =$	$\{A_{i.}^T, -A_{i.}^T : i = 1 \text{ to } m\}$. This is set of directions normal to facetal hyperplanes of K.
$\Gamma_2 =$	$\{P_{.1}, ..., P_{.m}, -P_{.1}, ..., P_{.m}\}$, where $P_{.i} = (I - c^T c)A_{i.}^T$, the orthogonal projection $A_{i.}$ (the direction normal to the facet of K defined by the i-th constraint in (1)) on the hyperplane $\{x : cx = 0\}$, for $i = 1$ to m

profitable direction to move at $x \in K^0$: = A direction y satisfying the property that $\delta(x + \alpha y)$ strictly increases as α increases from 0. It has been shown in [9] that y is a profitable direction to move at $x \in K^0$ iff $A_{i.}y \geq 0$ for all $i \in T(x)$.

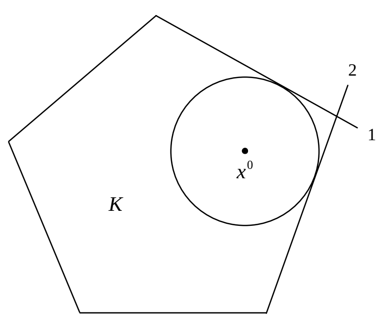

Figure 1: $x^0 \in K^0$, and the ball shown is the largest ball inside K with x^0 as center. Facetal hyperplanes of K corresponding to indices 1, 2 are tangent planes to this ball, so $T(x^0) = \{1, 2\}$.

Each iteration of the algorithm consists of only two steps, a centering step and a descent step. The centering step is a corrector step, it tries to move the current interior feasible solution into another one with higher value for $\delta(x)$ without sacrificing objective quality. The descent step is a predictor step, that results in a strict decrease in objective value.

2 The Centre of a Polytope

In [9], when the LP (2) given below has alternate optima, the definition of the center used in the algorithm is left imprecise. Here we complete the definition and make it ‘precise’.

The definition of the center of a polytope used in our algorithm is very different from that used in earlier IPMs [1 to 7; 12 to 15]. To distinguish, we therefore use the word **centre** (this is the common British spelling for the word “center”) for the center that we use.

A polytope of dimension 1 is a line segment, its centre is its unique midpoint. See Figure 2.

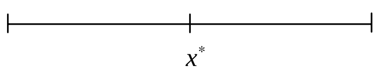

Figure 2: The centre of a 1-dimensional polytope (a line segment) is its midpoint x^*.

Now consider the polytope K of dimension n represented by (1). Its centre x^* is a point in K^0 which is the center of a lrgest radius ball inscribed inside K. Letting $\delta^* = \delta(x^*)$, (x^*, δ^*) is therefore an optimum solution of the LP

$$\begin{array}{ll} \text{Maximize} & \delta \\ \text{subject to} & \delta \quad - \quad A_{i.}x \leq -b_i, \quad = 1 \text{ to } m \end{array} \tag{2}$$

If the optimum solution of this LP is unique, it will be (x^*, δ^*), and x^* is the centre of K. See Figure 3.

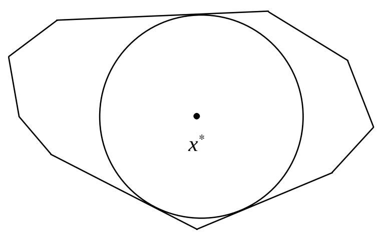

Figure 3: When the largest inscribed ball in K is unique, its center x^* is the centre of K.

If the optimum solution of (2) is not unique, all optimum solutions are of the form (x, δ^*) for $x \in S$, where S is the optimum face of (2) in the x-space. In this case the centre of K is defined recursively by dimension to be the centre of the lower dimensional polytope S. This definition guarantees that every polytope has a unique centre. See Figure 4 for an illustration.

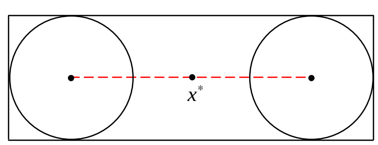

Figure 4: A 2-dimensional polytope K for which the largest inscribed ball is not unique. S, the set of centers of all such balls, the optimum face of (2) in the x-space, is the dashed line segment in this polytope. So here the centre of K is the centre of S, which is its mid-point x^*.

3 Centre for (1), On the Objective Plane $\{x : cx = t\}$ for Given t

Each iteration of our algorithm begins with the current point, which is the interior feasible solution obtained at the end of the previous iteration. Consider Iteration $r + 1$, suppose it begins with the current point x^r. Let $cx^r = t$ be the current objective value in (1). The centering step in this iteration tries to find the point $x \in K^0 \cap \{x : cx = t\}$ which maximizes $\delta(x)$, it is an optimum solution of the LP

$$
\begin{array}{lrcll}
\text{Maximize} & \delta & & & \\
\text{subject to} & \delta & - & A_{i.}x \leq -b_i, & i = 1 \text{ to } m \\
& cx & = & t &
\end{array} \tag{3}
$$

If the optimum solution x for (3) is unique, denote it by $x(t)$, it is called the **centre for (1) on the current objective plane** $\{x : cx = t\}$. In this case, the unique optimum solution of (3) is $(x(t), \delta[t] = \delta(x(t)))$. See Figure 5.

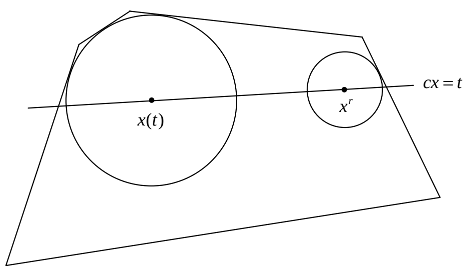

Figure 5: When the optimum solution for (3) is unique, the largest ball inside K with center on the current objective plane $\{x : cx = t\}$ is unique (like here, it is the large ball in the figure), its center is $x(t)$, the centre for (1) corresponding to the present objective value t.

In general, even though the optimum δ in (3) is always unique, there may be alternate x which are optimal. So, let $S(t)$ denote the optimum face of (3) in the x-space. In this case, the centre for (1) on the objective plane $\{x : cx = t\}$ is defined to be the centre of the polytope $S(t)$ as defined in Section 2.

This definition guarantees that for each $t \in (t_{\min}, t_{\max})$, the centre for (1) on the objective plane $\{x : cx = t\}$ is unique.

4 The Concept of the Algorithm

Iteration 1 begins with the initial interior feasible solution x^0 that is available. We will discuss the steps in the genral iteration $r + 1$ coceptually.

Iteration $r + 1$**:** It begins with x^r, the current point, the interior feasible solution obtained at the end of the previous iteration. Let $cx^r = t$ be the current objective value in (1). Go to the centering step.

Centering step: Starting with the current point x^r, find the centre for (1) on the current objective plane $\{x : cx = t\}$, which we will denote by $\bar{x}^r$. So, $(\bar{x}^r, \delta(\bar{x}^r))$ is an optimum solution of the LP (3).

Descent step: Consider only two descent directions, $-c^T, \bar{x}_r - \bar{x}^{r-1}$ (where $\bar{x}^{r-1}$ is the centre obtained in the previous iteration). Take maximum step lengths from $\bar{x}^r$ in each of these directions to within a tolerance ϵ of the boundary of K, and take the best of these two resulting points as x^{r+1}, the initial interior feasible solution for the next iteration. Go to the next iteration.

Comments: What is the reason to get a ball with the maximum possible radius with center on the current objective plane in the centering step? From the center of a ball with radius δ, we can move a step length of at least δ in any direction. Maximizing δ helps to make longer steps towards optimality in each iteration.

The centering problem (3) is itself another LP of the same size as the original LP (1). But it is an LP with a very special structure. For example, it is a parametric right hand side LP with the parameter t, and it can be shown that the optimum objective value function in it has only $O(m)$ slope changes, as opposed to the worst case behavior of an exponential number of slope changes [8]. Also, to implement this algorithm for solving (1), an exact solution of (3) is not essential, and we show that the special structure of (3) can be exploited to get a good approximate solution for it fast.

5 Summary of Theoretical Results

Let $x(t)$ be the centre for (1) computed in the centering step when the current objective value is t. So, $\delta[t] = \delta(x(t))$.

Define $J(t) = T(x(t))$ called **essential touching constraint index set** at objective value t, it is $\cap\{T(x) : x$ in the optimum face for (3)$\}$; i.e., $J(t) = \{i : A_{i.}x = b_i + \delta[t]$ for every optimum solution (δ, x) of (3)$\}$.

Proofs for the Results 2 to 7 listed below can be seen from [9, 11]. For some of these results, we provide figures that suggest an intutive justification for the result.

1. Every polytope has a unique centre. For each $t \in (t_{\min}, t_{\max})$, $x(t)$ is unique.

2. $\delta[t]$ is piecewise linear concave. So it increases monotonically as t decreases from $t_{\max}$ to t^*; and decreases monotonically as t continues to decrease from t^* to $t_{\min}$. See Figures 6, 7.

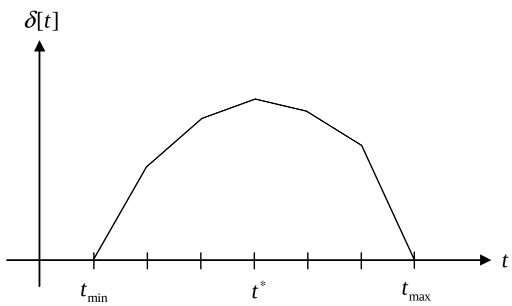

Figure 6: $\delta[t]$ is a PL concave function.

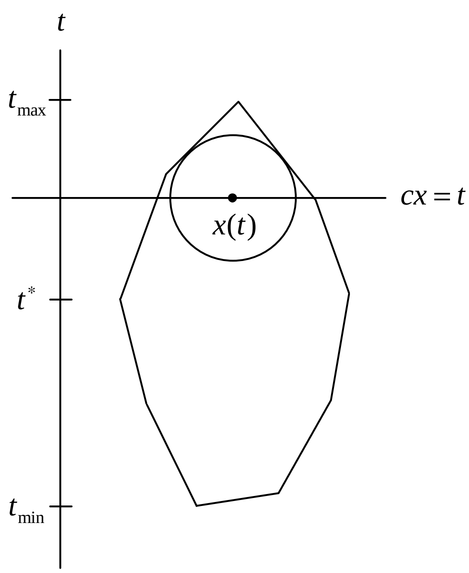

Figure 7: In this figure the current objective value t decreases as you move from top to bottom. For one value of t, the objective plane $\{x : cx = t\}$, $x(t)$, and the largest ball inside K with $x(t)$ as center are shown.

3. If $J(t)$ remains the same for all $t_2 \leq t \leq t_1$, then both $\delta[t]$ and $x(t)$ are linear in this interval. See Figure 8.

4. If t_1 is a value of t where the slope of $\delta[t]$ changes, then $J(t)$ changes as t decreases through t_1.

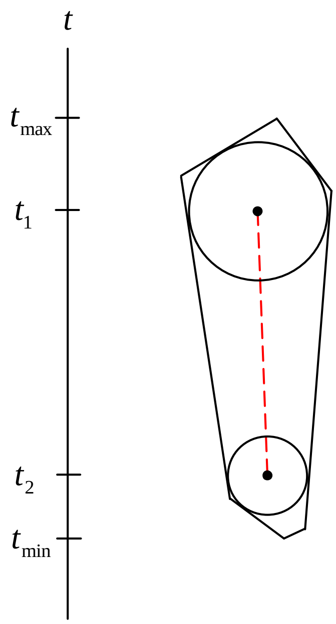

Figure 8: In this figure, the essential touching constraint index set, $J(t)$ remains the same for $t_1 \geq t \geq t_2$. In this objective value interval, the centre for (1), $x(t)$, traces the dashed line segment; also $\delta[t]$ is linear.

5. For $t^* \leq t \leq t_{\max}$, $x(t)$ is an optimum solution for the perturbed LP: maximize cx, subject to $Ax \geq b + e\delta[t]$, where e is a column vector of all 1s in R^m. For $t_{\min} \leq t \leq t^*$, $x(t)$ is an optimum solution for the perturbed LP: minimize cx, subject to $Ax \geq b + e\delta[t]$. See Figures 9 (a), (b).

6. If $t_1 > t_2$ are two consecutive values of t where the slope of $\delta[t]$ changes, and if the algorithm is implemented by finding the exact centre in every iteration, then it needs no more than 3 iterations to descend from objective value t_1 to t_2.

7. When the algorithm is carried out by finding the exact centre in every iteration, once a constraint, say the i-th, leaves the touching constraint index set $J(t)$ as the current objective value t is decreasing in the algorithm; it cannot reenter the set $J(t)$ as t decreases further.

Therefore, the touching constraint index set $J(t)$ changes at most $2m$ times during the algorithm. Consequently, the number of slope changes in $\delta[t]$, the optimum objective value function in the parametric RHS LP (3) is at most $2m$; and the algorithm terminates after at most $6m$ iterations.

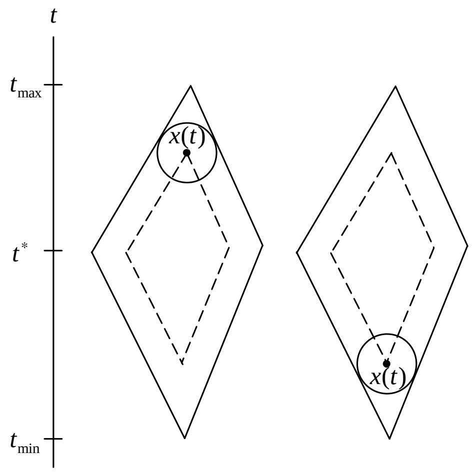

Figure 9: (a) (left), (b) (right) show the same polytope K in solid lines; and the current objective value t decreases as you move from top to bottom. (a) illustrates the case when $t > t^*$, and (b) the situation $t < t^*$. In each, the sphere is the largest inscribed sphere in K with $x(t)$ as center; and the polytope in dashed lines is the perturbed polytope $\{x : Ax \geq e\delta[t]\}$. In (a), $x(t)$ maximizes (and in (b) it minimizes) cx on the perturbed polytope.

6 How to Solve the Centering Step?

First consider (2), the problem of computing the centre of the polytope K itself beginning with an initial interior feasible solution, say x^r. [9, 10] proposes to solve this approximately using a line search algorithm. Beginning with the initial $x^{r,0} = x^r$, it generates a sequence of points $x^{r,k}$, $k = 1, 2, ...$ along which the radius of the ball δ is strictly increasing.

At the current point $x^{r,k}$, a direction y is called a **profitable direction**, if $\delta(x^{r,k} + \alpha y)$ strictly increases as α increases from 0. [9] has the following result, which makes it easy to check whether any given direction y is profitable at the current point.

Result: A given direction $y \in R^n$ is a profitable direction at the current interior feasible solution $x^{r,k}$ iff $A_{i.}y \geq 0$ for all $i \in T(x^{r,k})$. Also, $x^{r,k}$ is an optimum solution for (2) iff there is no profitable direction at it, i.e., iff the system: $A_{i.}y \geq 0$ for all $i \in T(x^{r,k})$ has no nonzero solution y.

Since the goal in this centering step is to increase the minimum distance of x from each facetal hyperplane of K, the procedure uses only the directions normal to the facetal hyperplanes of K for the line searches, i.e., directions in

Γ_1. The procedure continues as long as profitable directions for line search are found in Γ_1, and terminates with the final point as an approximate centre of K, which is also denoted by $\bar{x}^r$.

Once a profitable direction y at the current point in the sequence being generated by this procedure, $x^{r,k}$, has been found, the optimum step length α in this direction that maximizes $\delta(x^{r,k}+\alpha y)$ over $\alpha \geq 0$ is $\bar{\alpha}$, where $(\bar{\delta}, \bar{\alpha})$ is the optimum solution of the 2-variable LP

$$\begin{array}{rrcll} \text{Maximize} & \delta & & & \\ \text{subject to} & \delta - \alpha A_{i.}y & \leq & A_{i.}x^{r,k} - b_i & i = 1, \ldots, m \\ & \delta, \alpha \geq 0 & & & \end{array} \qquad (4)$$

and $\bar{\delta}$ is the optimum objective value $\delta(x^{r,k} + \bar{\alpha} y)$. So, the line search for the maximum value of δ in the direction y involves solving this 2-variable LP, which can be carried out efficiently (e.g., by the simplex algorithm) as discussed in [9, 10].

To solve (3), finding the centre for (1) on the current objective plane $\{x : cx = t = cx^r\}$, it uses the same procedure with profitable directions selected from the set Γ_2. See Figure 10.

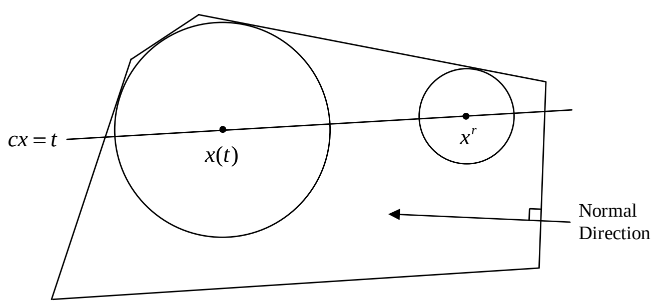

Figure 10: Moving from the current point x^r, in the direction which is the orthogonal projection of the normal to the facet of K on the right, on the objective plane $\{x : cx = t\}$, leads to $x(t)$.

We are investigating additional directions for line search to include in the sets Γ_1, Γ_2, to accelerate the convergence of this procedure, and to improve the quality of the approximation to the optimum centre.

7 How to Get An Initial Interior Feasible Solution for (1)?

The standard Phase I procedure adds a nonnegative artificial variable x_0 and modifies (1) into

$$\begin{aligned} \text{minimize} \quad & z(x) = cx + Mx_0 & & \\ \text{subject to} \quad & Ax + ex_0 & \geq & \; b \\ & x_0 & \geq & \; 0 \end{aligned} \tag{5}$$

where e is the column vector of all 1s in R^m, and M is a large positive penalty parameter. ($x^0 = 0$, x_0^0) where x_0^0 is a sufficiently large positive number, gives an initial interior feasiblew solution to (5). (5) is in the same form as (1), we solve it instead.

8 Implementation of the Algorithm to Solve (1) With an Initial Point $x_0 \in K^0$

We will discuss the general iteration $r + 1$. Let x^r be current interior feasible solution obtained at the end of the previous iteration.

We will discuss how the centre is approximated by the line search procedure in this iteration beginning with the initial point $x^{r,0} = x^r$. In a general stage of this procedure let $x^{r,k}$ be the current point in the sequence of points obtained in this procedure. Now we look for a line search direction in 2 stages.

Stage 1: Look for a direction to increase the ball radius and decrease the objective value simultaneously. So, look for $y \in \Gamma_1$ satisfying $cy < 0$ and $A_{i.}y \geq 0$ for all $i \in T(x^{r,k})$. If such y is found, carry out line search by finding the optimum step length as discussed in Section 6. If there is no such y go to Stage 2.

Stage 2: Look for profitable direction $y \in \Gamma_2$ and carry out line search as discussed in Section 6. If no such y is found, terminate the procedure with $x^{r,k}$ as an approximate centre, which is also denoted by $\bar{x}^r$. Now carry out the descent step at $\bar{x}^r$ as discussed in Section 4 and continue.

9 Another Way to Implement the Algorithm to Solve (1)

For any given value, t, of the objective function cx, the set of feasible solutions of (1) with this objective value is $K \cap \{x : cx = t\}$, represented by

$$
\begin{aligned}
Ax &\geq b \\
cx &= t
\end{aligned} \tag{6}
$$

$c = (c_1, ..., c_n)$ is the vector of cost coefficients in (1). Take any nonzero entry in c, say c_n. Then in (6), we can use the equality constraint to express the variable x_n as $(t - c_1x_1 - ... - c_{n-1}x_{n-1})/c_n$ in terms of the objective value t and the other variables. Substituting this expression for x_n in all the inequality constraints in (6), we get a representation of $K \cap \{x : cx = t\}$ in terms of the remaining variables $X = (x_1, ..., x_{n-1})^T$ in the form

$$
DX \geq d + td^* \tag{7}
$$

say. We will denote the set of feasible solutions of (6) for given t by $\mathcal{K}(t)$ in the X-space. Each point $X \in \mathcal{K}(t)$ corresponds to a unique point x in $K \cap \{x : cx = t\}$ through the expression given above for x_n.

Let $X(t)$ denote the centre for the polytope $\mathcal{K}(t)$ as defined in Section 2. $X(t)$ may correspond to a different point in $K \cap \{x : cx = t\}$ than $x(t)$ = the centre for (1) on the objective plane $\{x : cx = t\}$ defined in Section 3. We can apply the algorithm to solve (1) discussed in Section 4 using the point in $K \cap \{x : cx = t\}$ corresponding to $X(t)$ as the centre corresponding to objective value t, instead of $x(t)$.

In the implementation of the algorithm to solve (1) discussed in Section 8, in Iteration $r+1$ in which the initial interior feasible solution is x^r, the approximate centre $\bar{x}^r$ for (1) is computed in the centering step by the Stage 1, 2 procedure discussed there, and then carry out the descent step at $\bar{x}^r$ as discussed in Section 4 and continue.

In the new implementation, in this iteration we define $t = cx^r$, and compute the approximate centre of the polytope $\mathcal{K}(t)$ defined by (7), as discussed in Section 6, and take the point corresponding to it in $K \cap \{x : cx = t\}$ as the result $\bar{x}^r$ of the centering step in this iteration, carry out the descent step at $\bar{x}^r$ as discussed in Section 4 and continue.

10 How to Implement the Algorithm for Solving a General LP ?

The most popular IPM for software implementations is the primal-dual IPM [1 to 7, 12 to 15], because: **(i)** it gives optimum solutions to both the primal and the dual when both have feasible solutions; and **(ii)** it provides a lower bound that serves as an indicator to check how far left to go to reach the optimum. Also, in Sections 8, 9, we did not discuss a good termination condition for the algorithm. The lower bound in the primal-dual format provides an automatic practical termination condition.

We will show how to convert our algorithm into a primal-dual algorithm for LPs in general form. Consider an LP in general form in which there may be equality constraints on the variables, inequality constraints, and bounds on individual variables. By combining the bounds on individual variables with the inequality constraints the problem is in the form

$$\begin{array}{rrcl} \text{Minimize} & f\xi & & \\ \text{subject to} & F\xi & = & h \\ & G\xi & \geq & g \end{array} \tag{8}$$

where F is a matrix of order $p \times q$, say. Let π, μ be dual vectors corresponding to the constraints in the two lines in (8). Solving (8), and its dual involves finding a feasible solution to the following system

$$\begin{array}{rcl} F\xi & = & h \\ \pi F + \mu G & = & f \\ (G\xi, \mu, -f\xi + \pi h + \mu g) & \geq & (g, 0, 0) \end{array} \tag{9}$$

.

Solving (9) is the same as solving the LP

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{p}(F_{i.}\xi - h_i) + \sum_{j=1}^{q}(\pi F_{.j} + \mu G_{.j} - f_j) \\ \text{subject to} & (F\xi, G\xi, \pi F + \mu G, \mu, -f\xi + \pi h + \mu g) \;\; \geq \;\; (h, g, f, 0, 0) \end{array} \tag{10}$$

(10) is in same form as (1). Also, since we are applying the algorithm without matrix inversions, having all these additional constraints over those in the original LP (8) in the model does not make it numerically difficult to handle. If both (8) and its dual have feasible solutions, at the optimum, the objective value in (10) will be 0, so this provides a convenient lower bound to judge how far is left to go.

Acknowledgements: I thank Huang Chia-Hui for the nice figures.

11 References

[1] G. B. Dantzig and M. N. Thapa, *Linear Programming, 1. Introduction, 2. Theory and Extensions.* Springer-Verlag New York, 1997.

[2] O. Güler, C. Roos, T. Terlaky, and J. -P. Vial, "A survey of the implications of the behavior of the central path for the duality theory of linear programming", *Management Science*, 41 (1922-1934) 1995.

[3] M. Kojima, S. Mizuno, and A. Yoshise, "A primal-dual interior point algorithm for linear programming", *Progress in Mathematical Programming: Interior Point and Related Methods*, N. Meggiddo, ed., Springer-Verlag, New York, ch. 2 (29-47) 1989.

[4] N. Meggiddo, "Pathways to the optimal set in linear programming", *Progress in Mathematical Programming: Interior Point and Related Methods*, N. Meggiddo, ed., Springer-Verlag, New York, ch. 8 (131-158) 1989.

[5] S. Mehrotra, "On the implementation of a primal-dual interior point method", *SIAM Journal on Optimization*, 2 (575-601) 1992.

[6] S. Mizuno, M. Todd, and Y. Ye, "On adaptive step primal-dual interior point algorithms for linear programming", *Mathematics of Operations Research* 18 (964-981) 1993.

[7] R. D. C. Monteiro and I. Adler, "Interior path-following primal-dual algorithms, Part I: Linear programming", *Mathematical Programming* 44 (27-41) 1989.

[8] K. G. Murty, "Computational Complexity of Parametric Linear Programming", *Mathematical Programming*, 19, no. 2(213 - 219)Sept. 1980.

[9] K. G. Murty, "A new practically efficient interior point method for LP", *Algorithmic Operations Research*, 1 (3-19) 2006; paper can be seen at the website: http://journals.hil.unb.ca/index.php/AOR/index.

[10] K. G. Murty, "Linear equations, Inequalities, Linear Programs (LP), and a New Efficient Algorithm" Pages 1-36 in *Tutorials in OR*, INFORMS, 2006.

[11] K. G. Murty, "A New Practically Efficient Interior Point Method for Quadratic Programming", Working Paper, 2006.

[12] R. Saigal. *Linear Programming A Modern Integrated Analysis.* Kluwer Academic Publishers, Boston, MA, 1995.

[13] G. Sonnevend, J. Stoer, and G. Zhao, "On the complexity of following the central path of linear programming by linear extrapolation", *Mathematics of Operations Research* 62 (19-31) 1989.

[14] S. J. Wright. *Primal-Dual Interior-Point Methods.* SIAM, Philadelphia, PA, 1997.

[15] Y. Ye. *Interior Point Algorithms, Theory and Analysis*, Wiley-Interscience, New York, 1997.