Fast Descent Methods for LPs With No Matrix Inversions

Katta G. Murty
Department of IOE, University of Michigan,
Ann Arbor, MI 48109-2117, USA;
Sandia National Labs, Livermore, CA
Phone: 734-763-3513, Fax: 734-764-3451, e-mail: murty@umich.edu
www-personal.umich.edu/~ murty/

March 2012

Abstract

Existing software implementations for solving Linear Programming (LP) models are all based on full matrix inversion operations involving every constraint in the model in every step. This **linear algebra component** in these systems makes it difficult to solve dense models even with moderate size, and it is also the source of accumulating roundoff errors affecting the accuracy of the output.

We present new methods for LP that help reduce the need for this linear algebra component significantly, or even eliminate it altogether, and still get comparable or better results.

Key words: Linear Programming (LP), Interior point methods (IPMs) , ball centers of a polytope, solving LPs by descent methods without using matrix inversions.

In Memorium: We dedicate this paper to the memory of our dear friend Santosh Kabadi with whom we had many fruitful discussions on the methods discussed in this paper, who passed away in a tragic drowning accident in the sacred Ganges river recently.

1 Introduction

For modeling decision making applications, LP is the most commonly used mathematical model. Software systems for solving LP models are based on either the simplex method, or interior point

methods (IPMs, in particular the primal-dual IPM) developed during the second half of the 20th century (Dantzig and Thappa [1997], Kojima, Mizuno, Yoshishe [1989], Megiddo [1989], Mehrotra [1992], Monteiro and Adler [1989], Sonnevend, Stoer and Zhao [1989], and the books Saigal [1995], Wright [1997], and Ye [1997])) and are able to solve large scale sparse models (those involving thousands of constraints) within reasonable times by exploiting the sparcity of the models. As several real world applications lead to sparse models, these systems are very popular in practice.

But the simplex method, and these IPMs are based on matrix inversion operations involving every constraint in the model in every step. In large scale applications, these matrix inversion operations limit the ability of these algorithms to only those with very sparse coefficient matrices. Typically, the effectiveness of these algorithms fades as the density of the coefficient matrix increases.

Many applications lead to LP models that are not sparse, and need near-optimum solutions in real time. In many of these applications, the LP models are only of moderate size. This provided a motivation for us to develop fast algorithms for LP without using matrix inversion operations, or using them sparingly if at all.

One such application is the graphics application posed by Watson [2010] recently. It needed the solution of LPs in real time on GPUs ("Graphical Processing Units", the hot new topic in high performance computing). GPU is the "CPU" in a graphic coprocessor. GPUs are basically vector computers, designed to do a few simple operations very fast on large arrays of data. They were specifically designed for graphics processing, but have the potential to be adapted to other more general uses. They are inherently parallel, performing at least 32 threads of computation (instruction streams) in parallel.

The present trend in supercomputing is to have a massively parallel machine, but with each "node" being a general purpose CPU plus several GPUs (currently each node is just a commodity CPU).

GPUs pose some interesting constraints. They can hold only 16K (about 4000 numbers) of transient data (storage locations for data that changes in the algorithm), including everything like loop indices, temporary variables, etc. But they have a much larger static memory that is accessed for reading only. Therefore GPUs favor algorithms that frequently evaluate fixed

functions. Small scale linear algebra (e.g., a 63x63 matrix fills up the entire dynamic memory) is possible, but not heavy computations. However, they can solve large problems by moving data into and out of this GPU memory, but that is slow, and too much data movement negates the speed advantage of the GPU.

Commercial software packages such as MOSEK and CPLEX cannot be used in GPUs because these codes are too big to fit in the GPU memory, and cannot exploit the GPU architecture. Therefore a new code needs to be developed that explicitly exploits the GPU architecture—it has to be small and highly memory-efficient.

The advantage of GPUs is that they perform arithmetic operations super fast, and can execute 32 threads in parallel (similar to operating on vectors of length 32 in one instruction). On some applications they are more than 100 times faster than the fastest general purpose CPU (e.g., the latest Intel chip). On the other hand one CPU can drive several GPUs, and thus speedups more than 100 times per node of a large parallel machine are possible.

In this paper we present SM-5 (Sphere Method 5), a variant of the Sphere Methods for LP using no matrix inversions, and discuss computational results obtained using it. It provides an efficient algorithm for LPs needing only small memory, as required in applications being solved using GPUs. Of course it can also be used for solving LP models using regular computers in other applications.

SMs consider LPs in the form:

$$\begin{aligned} \min \quad & z = cx \\ \text{s.t.} \quad & Ax \geq b \end{aligned} \tag{1}$$

where A is an $m \times n$ data matrix; with a known interior feasible solution x^0 (i.e., satisfying $Ax^0 > b$). We assume that K, the set of feasible solutions of (1), is bounded. Strategies for modifying any given LP into this form are discussed in (Murty[2009-2, 3]).

1.1 Nomenclature

Here is the notation we will use in this paper.

- **Notation for rows and columns of A:** $A_{i.}, A_{.j}$ denote the i^{th} row and j^{th} column of A.

- **Feasible region and its interior:** K denotes the set of feasible solutions of (1), and $K^0 = \{x : Ax > b\}$ its interior.

- **Facetal hyperplanes:** $FH_i = \{x : A_{i.}x = b_i\}$, the i-th facetal hyperplance of K for $i =$ 1 to m.

- **Largest inscribed ball with a given point as center, its radius:** $B(x, \delta(x, K))$, $\delta(x, K)$ defined for $x \in K^0$, $\delta(x, K) = \text{minimum}\{\frac{A_{i.}x - b_i}{||A_{i.}||} : i = 1, ..., m\}$ is the radius of the largest ball that can be inscribed in K with x as its center. $B(x, \delta(x, K)) = \{y : ||y - x|| \leq \delta(x, K)\}$ is that largest inscribed ball in K with x as its center.

- **Touching constraint set at a given point:** $T(x, K)$ defined for $x \in K^0$, it is the set of all indices i satisfying: $\frac{A_{i.}x - b_i}{||A_{i.}||} = \text{Minimum}\{\frac{A_{p.}x - b_p}{||A_{p.}||} : p = 1 \text{ to } m\} = \delta(x, K)$. The facetal hyperplane $FH_i = \{x : A_{i.}x = b_i\}$ is a tangent plane to $B(x, \delta(x, K))$ for each $i \in T(x, K)$, that's why $T(x, K)$ is called the **index set of touching constraints in (1) defining** K at x.

- **Updated feasible region:** When x^r is the current interior feasible solution in the algorithm, the set K^{r+1} defined as,

$$K^{r+1} = \{x : Ax \geq b, \quad A_{m+1.}x \geq b_{m+1}\} \tag{2}$$

  where $A_{m+1.} = -c$, $b_{m+1} = A_{m+1.}x^r - \epsilon$, and ϵ is a small positive tolerance, is the set of feasible solutions of (1) updated by the current objective value in the algorithm. The current objective value is strictly monotonic decreasing in the algorithm, and hence this **updated set of feasible solutions** keeps getting smaller during the algorithm.

- **Coefficient matrix for updated feasible region:** A^r, b^r refer to the $(m + 1) \times n$ coefficient matrix in (2), and the RHS vector belonging to R^{m+1} in (2) respectively.

- **Radius of largest inscribed sphere for updated feasible region at a given point as center, and its touching constraint set:** $\delta(x, K^{r+1})$, $T(x, K^{r+1})$ defined for interior

points x of K^{r+1} same way as $\delta(x, K)$, $T(x, K)$ for K defined above, using the system of constraints (2) characterizing K^{r+1}.

- **GPTC (gradient projection on touching constraint) directions:** Let c^i denote the orthogonal projection of c^T on $\{x : A_{i.}x = 0\}$, i.e., $c^i = (I - A_{i.}^T(A_{i.}))c^T$ for $i = 1$ to m. When the ball $B(x, \delta(x, K))$ is under consideration, the directions $-c^i$ for $i \in T(x, K)$ are called the GPTC directions at the current center x in K.

- **Ball center of K:** It is the center of a largest ball in K, it maximizes $\delta(x, K)$ over $x \in K$.

- **Ball center of K^{r+1}:** When x^r is the current interior feasible solution in the algorithm, this is the ball center of the updated set of feasible solutions defined by (2).

- **IFS:** We will use this abbreviation for "interior feasible solution", i.e., for K it will be an x satisfying $Ax > b$.

INSERT FIGURE 1 HERE

Figure 1: $x^0 \in K$ and the ball shown is $B(x^0, \delta(x^0))$, 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 $T(x^0) = \{1, 2\}$.

INSERT FIGURE 2 HERE

Figure 2: K here is the feasible region bounded by 5 facetal hyperplances numbered 1 to 5. The ball B is the largest inscribed ball in K and its center $\bar{x}$ is the ball center of K. Here the ball center and the largest inscribed ball in K are unique, but in general they may not be unique. SM will work with any approximation of a ball center computed by the centering cycle.

INSERT FIGURE 3 HERE

Figure 3: K is the feasible region in the original problem. When x^r is the current IFS, the updated feasible region K^{r+1} (the shaded region), a largest inscribed ball B in K^{r+1}, and its center $\bar{x}^r$, a ball center of K^{r+1}, are shown.

1.2 Initialization Routine

In this paper we will assume that $K^0 \neq \emptyset$. For solving (1), the sphere methods for LP need an initial IFS for K. If an IFS for (1) is not available, , we can use the following Phase I procedure to find an IFS of K. This procedure introduces an artificial variable x_{n+1} associated with the column vector e (a column vector of appropriate order in which all entries are 1) changing the constraints in (1) into

$$Ax + ex_{n+1} \geq b \tag{3}$$

Let $X = \binom{x}{x_{n+1}}$, $A_1 = (A \dot{:} e)$ of order $m \times (n+1)$.

Then the Phase I problem is (here ϵ is a small positive number)

$$\begin{aligned} \min \quad & x_{n+1} \\ \text{s.t.} \quad & A_1 X \geq b + \epsilon e \end{aligned} \tag{4}$$

Let $x^0_{n+1} = 1 + \max\{0, b_i : i = 1, \cdots, m\}$, and $X^0 = \binom{x^0=0}{x^0_{n+1}}$. Then X^0 satisfies $A_1 X^0 > b + \epsilon e$. Now we apply the sphere method to solve (4) beginning with X^0 as the initial IFS. Under our assumption, (4) has an optimum solution at which the minimum objective value in (4) is ≤ 0. In solving (4) by the sphere method, when we get an IFS of (4), $X' = \binom{x'}{x'_{n+1}}$ at which x'_{n+1} is close to 0, then x' will be an IFS of (1). Now we apply the sphere method on the original LP (1) beginning with x' as the initial IFS for it.

For another option to use when an initial IFS for (1) is not available, but we know that $K^0 \neq \emptyset$, see Descent Step D5.8 based on Blow-up-2 routine discussed later.

1.3 Outline of Sphere Methods to solve LPs

We will now describe the main strategy used by the SMs to solve (1). Each iteration of the method begins with the best IFS obtained at the end of the previous iteration; and consists of two cycles; a **centering cycle**, and a **descent cycle** consisting of several descent steps. Details of both these steps are discussed next.

SM-5 uses the framework used in Sphere Method 2 (SM-2). In SM-2 discussed in Murty and Oskoorouchi [2010], in iteration $r+1$ with x^r as the initial IFS, the current set of feasible

solutions considered is K^{r+1}, the set of feasible solutions updated by the current objective value cx^r (defined in (2)).

The **center**, the output of the centering cycle in this iteration in SM-5, is a ball center of K^{r+1}, computed approximately.

The main strategy used in SM-2 to compute the ball center of K^{r+1} starting with an initial IFS $\hat{x}$ in K^{r+1}, is to select a direction y called a **profitable direction to move at $\hat{x}$ for K^{r+1}**, i.e., one satisfying the property that $\delta(\hat{x} + \alpha y, K^{r+1})$ strictly increases as α increases from 0; and determines the optimum step length to maximize $\delta(\hat{x} + \alpha y, K^{r+1})$ over $\alpha \geq 0$.

INSERT FIGURE 4 HERE. Figure 4: Illustration of a profitable direction y at an IFS x^0 in K^{r+1}. As we move from x^0 in the direction y indicated by the arrow, $\delta(x^0 + \alpha y, K^{r+1})$ increases until we reach the point x^*.

A direction y has been shown to be a profitable direction at $\hat{x}$ for K^{r+1} iff $A_{i.}y > 0$ for all $i \in T(\hat{x}, K^{r+1})$ [8, 9, 14], so checking a given direction for profitability is easy.

Once a profitable direction y at the current point $\hat{x}$ for K^{r+1} has been determined, the optimum step length that maximizes $\delta(\hat{x} + \alpha y, K^{r+1})$ is $\bar{\alpha}$, where $(\bar{\delta}, \bar{\alpha})$ is the optimum solution in the following 2-variable LP:

$$\begin{aligned} \max \quad & \delta \\ \text{s.t.} \quad & \delta - \alpha A_{i.}y \leq A_{i.}\hat{x} - b_i \quad \forall i = 1, \cdots, m+1 \\ & \delta, \alpha \geq 0 \end{aligned} \tag{5}$$

and $\delta(\hat{x} + \bar{\alpha} y, K^{r+1}) = \bar{\delta}$, the optimum objective value in this 2-variable LP, and the output point of this move in the profitable direction y at $\hat{x}$ is $\hat{x} + \bar{\alpha} y$. We will discuss an efficient algorithm to solve this 2-variable LP in Section 3.

As an illustration, in Figure 4, as we move from x^0 in the profitable direction y indicated by the arrow, the optimum step length $\bar{\alpha}$ for the move takes us to the point x^*, where $\delta(x^0 + \alpha y)$ is maximum over $\alpha \geq 0$. If this step length is larger that $\bar{\alpha}$, it can be verified that $\delta(x^0 + \alpha y)$ actually decreases from $\delta(x^*)$.

2 The Centering Cycle in SM-5

The centering cycle in SM-5 computes the ball center of the current set of feasible solutions K^{r+1} approximately starting with x^r as the current IFS of K^{r+1}, using the method based on results from Xie, Snoeyink, Xu [2006] and Clarkson [2010], not using any matrix inversions. Here is the method.

Translate the origin to the current IFS x^r, i.e., define the new vector of variables $y = x - x^r$. In terms of y, $K^{r+1} = \{y : A^r y \geq b^r - A^r x^r\}$. In the y-space x^r becomes the origin 0. Since it is an interior point of K^{r+1}, we have $b^r - A^r x^r < 0$. Define $A'^r_{i.} = A^r_{i.}/(b^r_i - A^r_{i.} x^r)$ for all $i = 1$ to $m+1$. Then in the y-space $K^{r+1} = \{y : A'^r y \leq e\}$ where e is a column vector of all 1s; and A'^r is the matrix with $A'^r_{i.}$ as the row vectors for $i = 1$ to $m+1$.

We know that A'^r is of order $(m+1) \times n$. Let $S = \{P_i = (A'^r_{i.})^T : i = 1 \text{ to } m+1\}$.

Let Q denote the center of the minimum Enclosing Sphere (min ES) for the set S of points, i.e., the smallest radius sphere containing all the points in S. An approximation for this can be found by the following scheme.

Scheme for computing Q: Q = center of min ES containing all the points in $S = \{P_1, ..., P_{m+1}\}$.

Strating with Q_0 = average of points in S, obtain the sequence Q_t, $t = 1, 2, ...$ as described below.

Having obtained Q_t, let P^{t+1} be the farthest point (by Euclideam distance) from Q_t in the set S. Then define $Q_{t+1} = (1-a)Q_t + aP^{t+1}$ where $a = 2/(t+3)$.

When the sequence Q_t converges take the final point as an approximation for Q.

INSERT FIGURE 5 HERE

Figure 5: The set S of points and the approximated MES for S and its center Q.

Centering step in SM-5 continued: Once Q is obtained, it corresponds to $Q + x^r$ in the original x-space. Now maximize δ using our 2-variable LP algorithm in K^{r+1} by finding the optimum step length to move on the straight line joining $Q + x^r$ to x^r and let the best point be called x^{r1}.

With x^{r1} as the new current IFS, repeat this centering step.

Continue repeating this centering step a few times until change in δ per step becomes small. Take the final point as the output of the centering step called the **center** in this iteration. Go to the descent steps with it.

2.1 The Various Descent Steps Used in SMs

Considering the genral iteration $r+1$, suppose the center obtained in the centering step in this iteration is $\bar{x}^r$. From this center, the descent cycle in this iteration carries out various descent steps. In this section we describe all the descent steps used in various SMs, and a new one, the most productive ones among these need to be determined in computational tests, to carry out in the descent cycle in SM-5 in each iteration.

In a general descent step from an IFS x^* in descent direction d (i.e., d satisfying $cd < 0$), we move from x^* in this direction, the maximum distance possible while still remaining at a distance ϵ from the boundary. This gives the step length to be γ, where

$$\gamma = \text{Minimum}\{(-A^r_{i.}x^* + b^r_i + \epsilon)/(A^r_{i.}d) : \quad \text{over all } i \text{ satisfying } A^r_{i.}d < 0\} \tag{6}$$

and the output of this descent step is $x^* + \gamma d$.

Here are the various descent steps used in SM-1 in the descent cycle when the center is $\bar{x}^r$.

D1.1: Descent step from $\bar{x}^r$ in the direction $d^1 = -c^T$.

D1.2: Let $G = \{(A_{i.})^T : i \in T(\bar{x}^r, K) \text{ such that } c(A_{i.})^T < 0\} \cup \{(-A_{i.})^T : i \in T(\bar{x}^r, K) \text{ such that } c(A_{i.})^T > 0\}$. Take a descent step from the center $\bar{x}^r$ in the direction which is the average of all the directions in G.

D2: Descent step from $\bar{x}^r$ in the direction $d^2 = \bar{x}^r - \bar{x}^{r-1}$, direction of the path of centers being generated, here $\bar{x}^{r-1}$ is the center obtained in the previous iteration.

D3: Descent step from $\bar{x}^r$ in each of the directions $-c^i$ for $i \in T(\bar{x}^r, K)$.

D4: Descent step from $\bar{x}^r$ in the average of the directions in D3.

D5.1 For each $i \in T(\bar{x}^r, K)$, let x^{ir} be the orthogonal projection of $\bar{x}^r$ on FH_i. It is $\bar{x}^r + (A_{i.})^T(b_i - A_{i.}\bar{x}^r)$.

Let $\hat{x}^{ir} = (1-\epsilon)x^{ir} + \epsilon\bar{x}^r$, the point on the line segment joining x^{ir} and $\bar{x}^r$ at a distance of ϵ from x^{ir}. $\hat{x}^{ir}$ is called the NTP (**near touching point**) of $B(\bar{x}^r, \delta(\bar{x}^r))$ with its tangent plane FH_i.

For each $i \in T(\bar{x}^r, K)$, take a descent step from the NTP $\hat{x}^{ir}$ in the direction $-c^i$.

INSERT FIGURE 6 HERE Figure 6: The largest inscribed ball with the current center $\bar{x}^r$ of K^{r+1} is shown. Facetal planes 1,2,3 are the touching planes to this ball with the touching points x^{1r}, x^{2r}, x^{3r} shown. The dashed lines indicate descent steps in D5.1. The dashed-dotted line from $\bar{x}^r$ is the descent step from $\bar{x}^r$ in the direction $-c^T$.

Now we describe additional descent steps D5.2, D5.3 used in this general iteration $r+1$ in SM-2 (Murty and Oskoorouchi [2010, 2011]).

D5.2, Descent Step 5.2: Let $\tilde{x}^{r1}$ denote the best point (by objective value) obtained in descent steps D1 to D5.1 in this iteration. This $\tilde{x}^{r1}$ is the initial IFS for Descent Step 5.2 (D5.2).

For each $i \in T(\tilde{x}^{r1}, K)$, from $\tilde{x}^{r1}$ take a descent step in the GPTC direction $-c^i$. Also, from $\tilde{x}^{r1}$ take a descent step in the direction which is the average of $-c^i$ for $i \in T(\tilde{x}^{r1}, K)$. Let $\tilde{x}^{r2}$ denote the best point obtained in all these descent steps, by objective value. If $c\tilde{x}^{r1} - c\tilde{x}^{r2}$ is:

$>$ the selected tolerance ϵ for objective value reduction, with $\tilde{x}^{r2}$ as the initial IFS repeat this D5.2; and continue the same way.

$\leq \epsilon$, take $\tilde{x}^{r2}$ as the output of this D5.2, with this point go to D5.3.

D5.3, Descent Step 5.3: We come to this step from the output point of D5.2, let us denote it by x^s. Clearly $\delta(x^s) \leq \epsilon$ from the manner it is obtained.

For each $i \in T(x^s, K^{r+1})$, define $x^{is} = x^s + (A_{i.})^T(b_i - A_{i.}x^s)$, the orthogonal projection of x^s on facetal hyperplane FH_i . Define $\bar{x} = [\sum_{i \in T(x^s, K^{r+1})} x^{is}]/|T(x^s, K^{r+1})|$. Typically, a move from x^s in the direction $x^s - \bar{x}$ goes through the central portion of K^{r+1}, so a step in this

direction at this stage can be expected to lead to good improvement in objective value. We have 2 cases to consider.

Case 1: If $c(x^s - \bar{x}) < 0$ carry out a descent step at x^s in the descent direction $(x^s - \bar{x})$, and make the output of this descent step the new current point (new x^s) and repeat this step with it, as long as the improvement in objective value is greater than the selected tolerance.

Case 2: If $c(x^s - \bar{x}) \geq 0$, let y be the orthogonal projection of $(x^s - \bar{x})$ on the hyperplane $\{x : cx = 0\}$, $y = (I - c^T c)(x^s - \bar{x})$.

Solve the 2-variable LP: max δ subject to $\delta - \alpha A_{i.} y \leq A_{i.} x^s - b_i$ for all i, and $\delta, \alpha \geq 0$. Let $\bar{\delta}, \bar{\alpha}$ be the optimum solution of this 2-variable LP. The point $x^s + \bar{\alpha} y$ has objective value $= cx^s$ because $cy = 0$, from this point take all descent steps D1 up to D5.2. Call the final output point of these descent steps as the new current point (new x^s), and with it repeat this D5.3 until the improvement in objective value becomes less than the selected tolerance.

Next we will describe a descent steps D5.4, D5.5 discussed in (Murty and Oskoorouchi [2011]) for use in every iteration of SM-2 , SM-3, SM-4.

D5.4, Descent Step 5.4: This descent step is carried out in the descent cycle after all the descent steps D1 to D5.3 have been carried out in this cycle. . Let $\bar{K}$ denote the current updated set of feasible solutions, here it is the current K^{r+1}.

Let $x^1, ..., x^s$ be all the points obtained at the end of all the descent steps carried out in the latest D5.1 above in this iteration; and suppose x^s is the best among all these by objective value. Let $H = \{x : cx = cx^s\}$, the objective plane through x^s, called **the current objective plane**. Let ϵ_1 be a small positive number, e.g. $\epsilon_1 = 0.1$ or smaller. Here $s =$ the number of touching constraints at the center using which this D5.1 was carried out.

For each $t \in \{1, ..., s-1\}$, let $\tilde{x}^t$ be the orthogonal projection of $x^s + \epsilon_1(x^t - x^s)$ on H. For all t such that $\tilde{x}^t \in K$, leave $\tilde{x}^t$ as it is.

For any $t \in \{1, ..., s-1\}$ such that $\tilde{x}^t \notin \bar{K}$, do the following: For each $i = 1$ to $m+1$ such that $A_{i.}\tilde{x}^t < b_i$ we know that $A_{i.}\tilde{x}^t - b_i < 0$ and $A_{i.}x^s - b_i > 0$ (because x^s is an interior point of $\bar{K}$), and so $(A_i x^s - b_i) - (A_{i.}\tilde{x}^t - b_i) = A_{i.}x^s - A_i\tilde{x}^t > 0$. Hence for such i, the smallest value of β that would make $A_{i.}(\beta x^s + (1-\beta)\tilde{x}^t) = \beta(A_i x^s - A_{i.}\tilde{x}^t) + A_{i.}\tilde{x}^t \geq b_i$ is $\beta = (b_i - A_{i.}\tilde{x}^t)/(A_{i.}x^s - A_{i.}\tilde{x}^t)$

which is > 0 and < 1.

So if we define $\theta =$ Maximum$\{(-(A_{i.}\tilde{x}^t - b_i)/(A_{i.}x^s - A_{i.}\tilde{x}^t))$: over all constraints i defining $\bar{K}$ and satisfying $A_{i.}\tilde{x}^t - b_i < 0\}$; then $\theta x^s + (1-\theta)\tilde{x}^t$ on the line segment joining x^s and $\tilde{x}^t$ is in $\bar{K}$.

Now replace $\tilde{x}^t$ by $(\theta)x^s + (1-\theta)\tilde{x}^t$. It can be verified that after these changes all $\tilde{x}^t \in \bar{K}$ for all $t = 1$ to $s-1$.

Now define the direction y as the average of $\{(\tilde{x}^t - x_s)/||\tilde{x}^t - x^s|| : \ \ t = 1 \text{ to } s-1\}$. All the $\tilde{x}^t$ for $t = 1$ to $s-1$, are spread out in different directions all around $\bar{K} \cap H$. So the half-line from x_s in the direction y will be in the central portion of $\bar{K} \cap H$, and hence the point which maximizes $\delta(x^s + \alpha y)$ over $\alpha \geq 0$ for the current set of feasible solutions $\bar{K}$ may be a reasonable approximation to the ball center of $\bar{K}$ on H.

Solve the 2-variable LP (of the form (5)) to find the point $x^s + \alpha y$, $\alpha \geq 0$ which maximizes $\delta =$ the radius of the largest ball inscribed inside $\bar{K}$ with $x^s + \alpha y$ as center, for the current set of feasible solutions $\bar{K}$. Let $\bar{\bar{x}}^2$ be the resulting point.

Let $S(\bar{\bar{x}}^2) = \{(A_{i.})^T : i \in T(\bar{\bar{x}}^2, K)$ such that $c(A_{i.})^T < 0\} \cup \{(-A_{i.})^T : i \in T(\bar{\bar{x}}^2, K)$ such that $c(A_{i.})^T > 0\}$, and let y be the average of all the directions in $S(\bar{\bar{x}}^2)$. Redefine $\bar{K} = \{x : Ax \geq b$ and $A_{m+1.}x \geq b_{m+1}\}$ where $A_{m+1.} = -c$ as defined earlier, and $b_{m+1} = -c\bar{\bar{x}}^2 - \epsilon$ (here ϵ is a small positive number), as the current set of feasible solutions. Solve the 2-variable LPs to maximize the radius of the largest ball inscribed inside the current set of feasible solutions with its center on each of the half-lines $\{\bar{\bar{x}}^2 + \alpha(-c^T) : \alpha \geq 0\}$ and $\{\bar{\bar{x}}^2 + \beta y : \beta \geq 0\}$; and let $\bar{\bar{x}}^3$ be the point among the outputs which corresponds to the maximum radius of the inscribed ball.

With $\bar{\bar{x}}^3$ as the center carry out the descent cycle with all descent steps D1 to D5.3. Using the set of output points from the latest application of D5.1, carry out D5.4 again. Continue this way repeating D5.4 as long as good reductions in objective value are obtained

If the reduction in objective value in two successive applications of D5.4 is less than the selected tolerance, the best point among the outputs of all the descent steps carried out in this iteration is the output of this iteration. With that point the method goes to D5.5.

D5.5: Let $x^1, ..., x^s$ be all the points obtained at the end of all the descent steps carried out in the latest D5.1 above in this iteration; and suppose x^s is the best among all these by objective

value. For $t = 1$ to $s-1$, define $x^t(\alpha) = x^s + \alpha(x^t - x^s)$.

For $t = 1$ to $s-1$, carry out the following step.

Step: Take $\alpha = 2^{-p}$, start with $p = 1$. Take a descent step from $x^t(\alpha)$ in the direction $-c^T$.

If the output point corresponds to an objective value $< cx^s$, call this point $\tilde{x}^t$, then go to the next value of t. If the output point corresponds to an objective value $> cx^s$, keep the value of t the same but increment p by 1 and repeat the above step.

Let $\tilde{x}$ denote the best by objective value among the $\tilde{x}^t$. Take $\tilde{x}$ as the initial IFS for carrying out D5.2 followed by D5.3.

D5.6: This step may not be suitable if the number of constraints m in (1) is large.

For each $i = 1$ to $m+1$ do the following. Define $c^{m+1} = c$.

As earlier let x^{ri} = orthogonal projection of x^r on $F_i = \{x : A_{i.}x = b_i\}$, which is $x^r + (A_{i.}^T(b_i - A_{i.}x^r))$. Two cases to consider.

Case 1: If $x^{ri} \in K^{r+1}$, let $\hat{x}^{ri} = \epsilon x^r + (1-\epsilon)x^{ri}$ as before (it is the NTP). From $\hat{x}^{ri}$ take a descent step in the direction $-c^i$.

Case 2: If $x^{ri} \notin K^{r+1}$, let α_i be the largest value of α such that $x^r + \alpha(x^{ri} - x^r) \in K^{r+1}$. In this case define $\hat{x}^{ri} = x^r + (\alpha_i - \epsilon)(x^{ri} - x^r)$. Again take the descent step from $\hat{x}^{ri}$ in the direction $-c^i$.

The best point among the output points obtained in these descent steps is the output point in this descent step.

D5.7: The Kabadi Descent Step 1: This and the following D5.8 were developed based on our discussions with Santosh Kabadi.

Let K^{r+1} be the current set of feasible solutions, and $\bar{x}^r$ its center obtained at the end of the centering cycle.We will use $\bar{x}^r$ as the **initial IFS for initiating this step**.

Blow-up-1 for K^{r+1}: If $\delta(\bar{x}^r, K^{r+1})$ is small, we can **blow up** K^{r+1} to make it fat for carrying out this step. This involves replacing the RHS constants vector b^r in (2) by κb^r where κ, known as the **blow up factor**, is a positive number > 1 to be chosen appropriately to blow

up $\delta(\bar{x}^r, K^{r+1})$ into a reasonable size number. This has the effect of multiplying $\bar{x}^r$, $\delta(\bar{x}^r, K^{r+1})$ etc. by κ. But we will continue to denote $K^{r+1}, \bar{x}^r, \delta(\bar{x}^r, K^{r+1})$, etc. by these same symbols. At the end you need to remember to divide the results by κ to bring them to refer to the original problem (1) again.

Let $B = B(\bar{x}^r, \delta(\bar{x}^r, K^{r+1}))$, $T = T(\bar{x}^r, K^{r+1})$ denote the current insphere of K^{r+1}, the index set of touching constraints respectively. As defined earlier, let $x^{ir} = \bar{x}^r + (A_{i.})^T(b_i - A_{i.}\bar{x}^r)$ be the orthogonal projection of $\bar{x}^r$ on FH_i for $i \in T$.

For each $i \in T$ extend the line joining x^{ir} to $\bar{x}^r$ until it intersects the boundary of K^{r+1} again on the other side of $\bar{x}^r$, and let that point of intersection be $\bar{x}^{ir}$. Finding $\bar{x}^{ir}$ requires only one minimum ratio computation at x^{ir} in the direction $\bar{x}^r - x^{ir}$ to find the maximum step length, γ^{ir}, all the way to the boundary of K^{r+1}. Similar to (6), we know that $\gamma^{ir} =$ minimum$\{(-A^r_{i.}x^{ir} + b^r_i)/(A^r_{i.}(\bar{x}^r - x^{ir})) :$ over all $i = 1$ to $m+1$ satisfying $A^r_i(\bar{x}^r - x^{ir}) < 0\}$. Then $\bar{x}^{ir} = x^{ir} + \gamma^{ir}(\bar{x}^r - x^{ir})$.

Let L_i denote the line segment joining x^{ir} and $\bar{x}^{ir}$. A general point on L_i is $x^i(\alpha) = x^{ir} + \alpha(\bar{x}^{ir} - x^{ir})$, $0 \leq \alpha \leq 1$. In D5.7, we will take descent steps in a descent direction, q^i say, from points along L_i, for each $i \in T$.

There are three possible choices of descent directions that can be used for carrying out these descent steps. They are $-c^T$, $-(c^i)$, $-(c^i + c^{\bar{i}})/2$, where $\bar{i}$ is the p satisfying $A_{p.}\bar{x}^{ir} = b_p$ if it is unique. If there are several p satisfying this equation, then $\bar{i}$ is a p among these that corresponds to the best output point in the step D5.1 if it was carried out in this iteration, or the one which minimizes $-c(c^p)$ (remember c^i is defined as a column vector for each i). We will use the descent direction among the 3 choices that gives the best results in computational tests, as the q^i to implement.

Let q^i denote the descent direction selected, then $cq^i < 0$. A general point on L_i is $x^i(\alpha) = x^{ir} + \alpha(\bar{x}^{ir} - x^{ir})$, $\quad 0 \leq \alpha \leq 1$.

Let $d^i(\alpha)$ denote the maximum step length that can be taken from $x^i(\alpha)$ in the direction q^i for $0 \leq \alpha \leq 1$. Computing $d^i(\alpha)$ for each α takes one minimum ratio computation, i.e., $d^i(\alpha)$ is the value of γ in the formula in (6) obtained by setting the point where this descent step is taken from $x^* = x^i(\alpha)$, and the direction in which this descent step is taken $d = q^i$. So, $d^i(\alpha) =$

Minimum$\{(-A^r_{p.}x^i(\alpha)+b^r_p+\epsilon)/A^r_{p.}q^i$: over all p satisfying $A^r_{p.}q^i<0\}$. Clearly $d^i(\alpha)$ is a concave function over $0\leq\alpha\leq 1$.

So, the objective value $f^i(\alpha)$ of the point obtained at the end of the descent step from $x^i(\alpha)$ is: $f^i(\alpha)=cx^i(\alpha)+cq^id^i(\alpha)$; which is a convex function in $0\leq\alpha\leq 1$, since $cq^i<0$ as q^i is a descent direction.

The minimum value of $f^i(\alpha)$ over $0\leq\alpha\leq 1$ can be found efficiently using a line search subroutine in NLP (nonlinear programming) codes (based on some efficient method like the **bisection section search method**).

After completing this step for each $i\in T$, we take the best among the output points obtained corresponding to all $i\in T$, and that point is the output point of this D5.7. With that point as the new IFS D5.7 can be repeated again. Continue the same way as long as there is good reduction after each repitition.

If blow up of solution set is carried out, divide the output point and the RHS constants vector by the blowup factor to bring them to refer to the original problem (1).

D5.8: The Kabadi Descent Step 2: Here we describe another **blow-up routine, Blow-up-2** to use before applying D5.8. Let Δ be a positive blow-up parameter , selected appropriately. In fact this Blow-up-2 can be used right at the beginning of the method when an initial IFS to (1) is not available, if we know that $K^0\neq\emptyset$; with D5.8 as the only deescent step carried out.

Blow-up-2: Assuming that each constriant in the updated system (1) is normalized to make sure that $||A_{i.}||=1$ for all i, replace each b_i in (1) by $b_i-\Delta=b^e_i$. This leads to the **expanded system** which we will denote by (1e),

$$A_{i.}x\geq b^e_i, \qquad \text{for } i=1 \text{ to } m \qquad (1e)$$

(**Note:** If constraint normalization is not carried out as discussed above, then we must take "$b^e_i=b_i-\Delta||A_{i.}||$ for each $i=1$ to m.) Also notice that the Blow-up-2 is applied only to constraints 1 to m in the original system, and not the updating constraint $m+1$ in (2) defining the updated set of feasible solutions K^{r+1}_e for the expanded system defined below in

(6) corresponding to K^{r+1} for the original system.

To keep the discussion simple, we will discuss this step under the assumption that each constraint in (1) is normalized at the beginning of this D5.8 step to make sure that $||A_{i.}|| = 1$ for all $i = 1$ to m, and that $||c|| = 1$. Let K_e denote the set of feasible solutions of (1e), K_e includes the original K as a subset, as each constraint is extended outwards, that is why we will call K_e the **extended set of solutions**, and verify that each facetal hyperplane of K_e is parallel to the corresponding facetal hyperplane of K. Also the following results hold.

Result 1: For each x in the interior of K, $\delta(x, K_e) = \delta(x, K) + \Delta$.

Result 2: A ball center of K_e is also a ball center of K and vice versa.

How does this Blow-up-2 help avoid doing Phase I ?: If an initial IFS of (1) is not available, let x^0 be any point not in K but close to K, which may be infeasible to the original system of constraints (1), or if feasible, satisfies at least one of the constraints in (1) as an equation. So, Minimum$\{A_{i.}x^0 - b_i : i = 1, ..., m\}$ is ≤ 0. Taking $\Delta = 1 + |$ Minimum$\{A_{i.}x^0 - b_i : i = 1, ..., m\}|$ to expand K to K_e will provide x^0 as the initial IFS of K_e to start the algorithm without any Phase I.

When x^r in the interior of K_e is the initial IFS for Step $r + 1$, define the updated set of solutions to consider in this step to be:

$$K_e^{r+1} = \{x : A_{i.}x \geq b_i - \Delta, \quad i = 1 \text{ to } m, \text{ and } A_{m+1.}x \geq b_{m+1} \qquad (6)$$

where $A_{m+1.} = -c$ and $b_{m+1} = -cx^r - \epsilon\}$. K_e^{r+1} is called the **updated extended set of solutions** in this iteration.

Apply the centering routine (using LSFN discussed in [13], and the centering step discussed in Section 2) beginning with the initial IFS x^r of K_e^{r+1} to find the ball center of K_e^{r+1}; and let $\bar{x}^r$ denote the output of this centering step. Now define:

$B_e = B(\bar{x}^r, K_e^{r+1})$, the largest insphere of K_e^{r+1} with $\bar{x}^r$ as center

$T_e = T(\bar{x}^r, K_e^{r+1})$, the touching set of constraint indices for the updated extended set of solutions K_e^{r+1} at $\bar{x}^r$.

Let x_e^{ir} for $i \in T_e$ be the touching points of B_e with its touching facetal hyperplanes of K_e .

Join x_e^{ir} to $\bar{x}^r$ and extend this line until it intersects the boundary K_e^{r+1} at the other end at a point denoted by $\bar{x}_e^{ir}$, and denote this line segment by L_i^e. To find $\bar{x}_e^{ir}$, compute γ_e = Minimum$\{(-A_{i.}^r x_e^{ir} + b^{er})/A_{i.}^r(\bar{x}^r - x_e^{ir})$: over all i satisfying $A_{i.}^r(\bar{x}^r - x_e^{ir}) < 0\}$. Then $\bar{x}_e^{ir} = x_e^{ir} + \gamma_e(\bar{x}^r - x_e^{ir})$; and $L_e^i = \{x : x = x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir}) : 0 \leq \alpha \leq \gamma_e\}$.

In general, L_i^e can be divided into 3 parts, the two end parts are in K_e^{r+1} but not in the original K^{r+1}, while the middle part is in K^{r+1}. These various parts can be identified as follows:

In the system of constraints $A^r(x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir})) - b_i^{er} \geq 0$ all quantities except α are given. When expanded and simplified, these constraints just become lower and upper boundes for α, let these be $\alpha_1 \leq \alpha \leq \alpha_2$. Then the various parts of L_i^e are $L_{i1}^e, L_{i2}^e, L_{i3}^e$ consisting of all $x = x_i^{ir} + \alpha(\bar{x}^r - x_e^{ir})$ for α in the intervals $0 \leq \alpha \leq \alpha_1$, $\alpha_1 \leq \alpha \leq \alpha_2$, $\alpha_2 \leq \alpha \leq \gamma_e$ respectively.

However, when $\bar{x}_e^{ir}$ satisfies the $(m+1)$th constraint in the definition of K_e^{r+1} as an equation, there is no L_{i3}^e; in this case L_i^e consists of only two parts L_{i1}^e, L_{i2}^e.

For each $i \in T$, find the set of all p such that $A_{p.}\bar{x}_e^{ir} = b_p^{er}$, and among all these p call the one which minimizes $-c(c^p)$ as $\bar{i}$. Then $\bar{x}_e^{ir} \in F_{\bar{i}}^e$, the facet of K_e corresponding to the constraint $\bar{i}$. For each $i \in T$, $(i, \bar{i})$ form a pair of indices corresponding to the facets of K_e on which the end points of L_i^e lie.

Options for descent directions y for descent steps in D5.8: As in D5.7 these are $-c^t, -(c^i), -(c^i + c^{\bar{i}})/2)$. Compare performance in computational tests and select the best.

Once a y is chosen, the step length $\gamma_i(\alpha)$ for descent step from $x^i(\alpha) = x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir})$ on L_i^e to a point in the interior of K^{r+1} while still remaining at a distance of ϵ from the boundary; and the objective value $f^i(\alpha)$ at the output point is discussed below:

(i) When $\alpha_1 \leq \alpha \leq \alpha_2$: In this case $\gamma_i(\alpha) = \text{Minimum}\{(-A_{i.}^r x(\alpha) + b_i^r + \epsilon)/(A_{i.}y) : \text{over all } i = 1 \text{ to } m \text{ satisfying } A_{i.}^r y < 0\}$, and $f^i(\alpha) = cx(\alpha) + cy\gamma_i(\alpha)$.

Discussion relating to descent steps from $x(\alpha)$ **for** $0 \leq \alpha \leq \alpha_1$ **and** $\alpha_2 \leq \alpha \leq \gamma_i^e$

In these cases, a descent step from $x(\alpha)$ in the direction y may not intersect the original set K^{r+1} at all. So first we need to determine the subset of α in these intervals satisfying the property that $x(\alpha) + \lambda y$ lies in the original K^{r+1} for some $\lambda \geq 0$. Here is the procedure for that. For $x(\alpha)+\lambda y$ to be in K^{r+1} it should satisfy $A_{i.}(x(\alpha)+\lambda y)-b_i \geq 0$ for all $i = 1$ to m; i.e.

$\lambda \geq (b_i - A_{i.}x(\alpha))/(A_{i.}y)$: for all i satisfying $A_{i.}y > 0$
$\lambda \leq (b_i - A_{i.}x(\alpha))/(A_{i.}y)$: for all i satisfying $A_{i.}y < 0$

Now Define

$$\lambda_1(\alpha) = \text{Maximum}\{0, (b_i - A_{i.}x(\alpha))/(A_{i.}y : \text{over all } i \text{ satisfying } A_{i.}y > 0\}$$
$$\lambda_2(\alpha) = \text{Minimum}\{(b_i - A_{i.}x(\alpha))/(A_{i.}y : \text{over all } i \text{ satisfying } A_{i.}y < 0\} \qquad (7)$$

then we need $\lambda_1(\alpha) \leq \lambda \leq \lambda_2(\alpha)$ for $(x(\alpha) + \lambda y)$ to be in K^{r+1}.

So if $\lambda_2(\alpha) < \lambda_1(\alpha)$ then there is no $\lambda \geq 0$ such that $(x(\alpha) + \lambda y) \in K^{r+1}$. Remembering that $x(\alpha) = x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir})$, the condition $\lambda_2(\alpha) \geq \lambda_1(\alpha)$ is:

$$\text{Minimum}\{(b_i - A_{i.}(x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir})/(A_{i.}y : \text{over all } i \text{ satisfying } A_{i.}y < 0\} \qquad (8)$$
$$- \text{Maximum}\{0, (b_i - A_{i.}(x_e^{ir} + \alpha(\bar{x}^r - x_e^{ir}))/(A_{i.}y : \text{over all } i \text{ satisfying } A_{i.}y > 0\} \geq 0$$

In the left hand expression above in (8), everything is known except α; so the above leads

to bounds on α. Find the smallest value of α, say $\bar{\alpha}_1$, satisfying the above condition. Similarly determine the largest value of α, say $\bar{\alpha}_2$ satisfying the above condition.

(ii) When $0 \leq \alpha \leq \alpha_1$: Since our goal is to get an output point in the original K^{r+1} , if $\bar{\alpha}_1 > \alpha_1$, we do not perform descent steps in this case. On the other hand if $\bar{\alpha}_1 < \alpha_1$, we carry out descent steps in the direction y in the subinterval $\bar{\alpha}_1 \leq \alpha \leq \alpha_1$. The maximum step length will be $\lambda_2(\alpha) - \epsilon$ where $\lambda_2(\alpha)$ is defined in (7). The output point is $x(\alpha) + (\lambda_2(\alpha) - \epsilon)y$, and the objective value at this output point for α in this interval is $f(\alpha) = cx(\alpha) + (\lambda_2(\alpha) - \epsilon)cy$.

(iii) When $\alpha_2 \leq \alpha \leq \gamma_e$: Since our goal is to get an output point in the original K^{r+1} , if $\bar{\alpha}_2 < \alpha_2$, we do not perform descent steps in this case. On the other hand if $\bar{\alpha}_2 > \alpha_2$, we carry out descent steps in the direction y in the subinterval $\alpha_2 \leq \alpha \leq \bar{\alpha}_2$. The maximum step length will be $\lambda_2(\alpha) - \epsilon$ where $\lambda_2(\alpha)$ is defined in (7). The output point is $x(\alpha) + (\lambda_2(\alpha) - \epsilon)y$, and the objective value at this output point for α in this interval is $f(\alpha) = cx(\alpha) + (\lambda_2(\alpha) - \epsilon)cy$.

Bisection Searches Carry out bisection searches to minimize $f(\alpha)$ in each of the 3 intervals $\bar{\alpha}_1 \leq \alpha \leq \alpha_1$. $\alpha_1 \leq \alpha \leq \alpha_2$, $\alpha_2 \leq \alpha \leq \bar{\alpha}_2$. Take the output point corresponding to the best objective value as the initial IFS for the next iteration for repeating this work.

3 Method Used for Solving 2-variable LPs of the Form (5)

In SMs, we solve 2-variable LPs in variables (δ, α) of the form (5) in various stages. All these problems arise in finding the optimum step length (value of α that maximizes δ) from an IFS $\bar{x}$ of the current set of feasible solutions, with $\delta(\bar{x}) = \bar{\delta}$, in a profitable direction y. So in all such instances we have an initial feasible solution $(\delta, \alpha) = (\bar{\delta}, 0)$ for the instance of (5) being solved. We use the following method to solve this instance.

Let Γ denote the set of feasible solutions of the instance of (5) in the 2-dimensional space of (δ, α) with α plotted on the horizontal axis, and δ plotted on the vertical axis. The method performs a series of iterations. The first iteration begins with $(\bar{\delta}, 0)$ on the boundary of Γ. Each iteration begins with a feasible solution on the boundary of Γ, performs a (horizontal move + a vertical move) twice, and finally a diagonal move. We will now discuss a general iteration

beginning with the intial solution (δ_0, α_0).

Set Γ of feasible solution (δ, α) of the 2-variable LP (5) when y is not a profitable direction in (5). It can be verified that the maximum possible value of δ in Γ with α fixed at a nonnegative value $\hat{\alpha}$, decreases as $\hat{\alpha}$ increases from 0. So in this case the optimum value of α in the 2-variable LP (5) is zero.

Set of feasible solution (δ, α) of 2-variable LP (5) when y is a profitable direction. It can be verified that with α fixed at a value $\hat{\alpha}$ say, the maximum possible value of δ in Γ increases as $\hat{\alpha}$ increases from 0.

The first horizontal move: Keeping $\delta = \delta_0$, find $\alpha_1 =$ the value of α at the mid-point of the line segment $\{(\delta_0, \alpha) \in \Gamma\}$. Given $\delta = \delta_0$, we know from the constraints in (5) that $\theta_1(\delta_0) \leq \alpha \leq \theta_2(\delta_0)$ where

$\theta_1(\delta_0) =$ Maximum$\{0, (A_{i.}\bar{x} - b_i - \delta_0)/(-A_{i.}y)$: over i such that $A_{i.}y > 0\}$
$\theta_2(\delta_0) =$ Minimum$\{(A_{i.}\bar{x} - b_i - \delta_0)/(-A_{i.}y)$: over i such that $A_{i.}y < 0\}$.

It can be verified that $\theta_1(\delta_0) = 0$ whether the direction y chosen for this line search step is profitable or not. Also, if y is not profitable, then $\theta_2(\delta_0) = 0$, since for y not profitable $A_{i.}y < 0$ for at least one $i \in T(\bar{x}, K^{r+1})$. So, if a y not profitable is chosen by mistake, we will terminate at this first horizontal move with the conclusion that the point $(\delta_0, \alpha_0 = 0)$ is an optimum solution of this 2-variable LP.

So assuming that $\theta_2(\delta_0) > 0$, we know that the α_1 mentioned above is $(\theta_1(\delta_0) + \theta_2(\delta_0))/2$. We will call the corresponding point (δ_0, α_1) in Γ as the *Center of* Γ on $\delta = \delta_0$.

The vertical move: In this move α is held constant at present value α_1, and the maximum value of δ subject to the constraint that $(\delta, \alpha_1) \in \Gamma$ is computed. This is equal to:

$\delta_1 = \gamma(\alpha_1) =$ Minimum$\{A_{i.}\bar{x} - b_i + \alpha_1 A_{i.}y : i = 1$ to $m\}$.

and the point in Γ achieving this value of δ is (δ_1, α_1).

The 2nd (horizontal + vertical) moves: In the 2nd horizontal move, keeping δ fixed at δ_1, find lower and upper bounds $\theta_1(\delta_1)$, $\theta_2(\delta_1)$ for α such that $(\delta_1, \alpha) \in \Gamma$ using the formulas given above. If $\theta_1(\delta_1) = \theta_2(\delta_1)$, then $(\delta = \delta_1, \alpha = \theta_1(\delta_1))$ is an optimum solution of this

2-variable LP, terminate.

On the other hand, if $\theta_1(\delta_1) < \theta_2(\delta_1)$, let $\alpha_2 = (\theta_1(\delta_1) + \theta_2(\delta_1))/2$, it is the center of Γ on $\delta = \delta_1$. Now carry out the 2nd vertical move keeping α fixed at α_2 to find the maximum value δ_2 of δ in Γ on $\delta = \delta_2$ attained at $(\delta_2, \alpha_2) \in \Gamma$.

Illustration of the horizontal, vertical and diagonal moves. The dotted lines are the horizontal moves, the large dashed lines are the vertical moves, and the medium dashed line is the diagonal moves.

The diagonal move: This move involves finding the maximum value of δ for points along the line joining the two centers of Γ obtained in the two horizontal moves in this iteration. The two centers are $(\delta_0, \alpha_1), (\delta_1, \alpha_2)$, where $\delta_1 > \delta_0$. Let L denote the line joining these two centers. From the coordianates of these two centers we know that L is defined by the equation

$$\delta \quad = \quad \delta_0 + s(\alpha - \alpha_1)$$

where $s = (\delta_1 - \delta_0)/(\alpha_2 - \alpha_1)$. Let

β_1 = minimum value of α in $L \cap \Gamma$ is = maximum$\{(0, A_{i.}\bar{x} - b_i - \delta_0 + s\alpha_1)/(s - A_{i.}y)$: over i such that $s - A_{i.}y < 0\}$.

β_2 = maximum value of α in $L \cap \Gamma$ is = minimum$\{(A_{i.}\bar{x} - b_i - \delta_0 + s\alpha_1)/(s - A_{i.}y)$: over i such that $s - A_{i.}y > 0\}$.

So, the maximum value of δ on $L \cap \Gamma$ is δ_3, where

$\delta_3 = \delta_0 + (\beta_2 - \alpha_1)s$ attained at the point (δ_3, β_2) if $\alpha_2 > \alpha_1$, or

$\delta_3 = \delta_0 + (\beta_1 - \alpha_1)s$ attained at the point (δ_3, β_1) if $\alpha_2 < \alpha_1$.

Let δ_4 = maximum$\{\delta_2, \delta_3\}$; and denote the associated value of α for it given above by α_4, i.e., $\alpha_4 = \alpha_2$ if $\delta_4 = \delta_2$; and if $\delta_4 = \delta_3$ then $\alpha_4 = \beta_2$ or β_1 depending on whether $\alpha_2 > \alpha_1$ or $\alpha_2 < \alpha_1$.

Then (δ_4, α_4) is the output of this iteration. With this point go to the next iteration.

Terminate the method with the output in an iteration when the improvement in the value of δ becomes small.

4 Computational results

5 References

1. K. L. Clarkson, "Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm", *Journal ACM Transactions on Algorithms (TALG)*, Volume 6 , Issue 4, August 2010 ACM New York, NY, USA

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

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

4. N. Megiddo, 1989, "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).

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

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

7. K. G. Murty, 1980, "Computational Complexity of Parametric Linear Programming", Mathematical Programming, 19 (213-219).

8. K. G. Murty, 1983, *Linear Programming*, Wiley, NY.

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

10. K. G. Murty, 2006-2, "Linear equations, Inequalities, Linear Programs (LP), and a New Efficient Algorithm" Pages 1-36 in *Tutorials in OR*, INFORMS.

11. K. G. Murty, 2009-1, "Ball Centers of Special Polytopes", IOE Dept., University of Michi-

gan, Ann Arbor, MI-48109-2217.

12: K. G. Murty, 2009-2, "New Sphere Methods for LP", *Tutorials in OR*, INFORMS.

13. K. G. Murty, 2009-3, *Optimization for Decision Making: Linear and Quadratic Models*, Springer, NY.

14. K. G. Murty, and S. N. Kabadi, 2008, "Additional Descent Steps in the Sphere Method", Dept. IOE, University of Michigan, Ann Arbor.

15. K. G. Murty, and M. R. Oskoorouchi, 2008, "Note on implementing the new sphere method for LP using matrix inversions sparingly", *Optimization Letters*, 3, 1, 137-160.

16. K. G. Murty, and M. R. Oskoorouchi, 2010, " Sphere Methods for LP", *Algorithmic Operations Research*, 5, 21-33.

17. K. G. Murty, and M. R. Oskoorouchi, 2011, "Fast descent methods for LPs with minimal or no matrix inversions", Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran-31261, Saudi Arabia.

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

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

20. L. T. Watson, 2010, *Personal Communication.* Computer Science, Virginia Tech.

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

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

23 Yulai Xie, Jack Snoeyink, Jinhui Xu,, 2006, " Efficient Algorithm for Approximating Maximum Inscribed Sphere in High Dimensional polytope", Proc. 22nd Annual ACM Symposium on Computational Geometry (SoCG06), pp. 21-29, June 5-7, Sedona, Arizona, USA.

NOTES ON SM-17.PDF

PAGE 6: **Reason for ϵ-term in the RHS vector in (4)**

This is to make sure when we get $X' = (x', x'_{n+1})$ at which x'_{n+1} is close to 0, then x' is an IFS for (1).

PAGE 8: BOTTOM AND PAGE 9 TOP: **Suggestions in coding the Centering Step:** Let current point at any stage of this centering cycle be $x^{r,k}$. The centering cycle continues if the Q obtained in this step satisfies the following **consistency condition**, which is

$A_{i.}Q \neq 0$ but has the same sign (+ve, or −ve) for all $i \in T(x^{r,k}, K^{r+1})$.

Take $y = Q$ or $-Q$ such that $A_{i.}y > 0$ for all $i \in T(x^{r,k}, K^{r+1})$; hence y is a profitable direction to move at $x^{r,k}$. Solve 2-variable LP of form (5) to select step length $\bar{\alpha} > 0$ to maximize $\delta(x^{r,k} + \alpha y)$ over $\alpha \geq 0$; and define $x^{r,k+1} = x^{r,k} + \bar{\alpha}y$; and repeat this centering step with this as the new current point.

Stop centering cycle when the Q obtained in a centering step does not satisfy consistency condition, or when improvement in the value of δ becomes small in some step. At that stage, take the current point in the sequence $x^{r,k}$ generated as the output of this centering cycle, i.e., the center.

Actually Mike implemented this centering cycle and tested it too during my stay in Sandia. You should be able to get this from him, check it, and use it for your implementation.

PAGE 19: INSERTING FIGURE 8 HERE illustrating D5.8 will be good. When you are ready I will scan it and send you.

PAGE 22 ABOVE SECTION 4: INSERTING A FIGURE 9 HERE TO ILLUSTRATE THIS 2-VAR. ALGO WILL BE GOOD. AGAIN can scan and send when you are ready.

ON DESCENT STEPS : MY suggestion is to go to D5.8 directly, and if needed use also D5.7. Both these steps together help improve the "aspect ratio" of K, which Reference 23 mentions as

being important for centering to work well. Seshadri at Sandia was also mentioning this "aspect ratio". I feel that these two steps may be adequate to get good performance.

A good "line search subroutine" from NLP codes is needed for good performance through D5.7, D5.8. Hope you will be able to dowbload a good one to use.

CLARIFICATIONS: If I can clarify anything, please let me know. Also if you can give some tentative results, I can try to make changes appropriately.

I believe this will work, but remember "proof of the pudding is in the eating". We cannot be sure until tested.