Computing Weighted Analytic Center for Linear Matrix Inequalities Using Infeasible Newton's Method

Shafiu Jibrin

Department of Mathematics and Statistics

Northern Arizona University

Flagstaff, AZ 86011-5717

Abstract

We study the problem of computing weighted analytic center for system of linear matrix inequality constraints. The problem can be solved using the Standard Newton's method. However, this approach requires that a starting point in the interior point of the feasible region be given or a Phase I problem be solved. We address the problem by using infeasible Newton's method applied to the KKT system of equations which can be started from any point. We implement the method using backtracking line search technique and also study the effect of large weights on the method. We use numerical experiments to compare infeasible Newton's method with the Standard Newton's method. The results show that infeasible Newton's method moves in the interior of the feasible regions often very quickly, starting from any point. We recommend it as a method for finding an interior point by setting each weight to be 1. It appears to work better than the Standard Newton's method in finding the weighted analytic center when none of weights is very large relative to the other weights. However, we find that infeasible Newton's method is more sensitive than the Standard Newton's method to large variation in the weights.

1 Introduction

We consider a system of linear matrix inequality constraints given below:

$$\text{subject to } A^{(j)}(x) := A_0^{(j)} + \sum_{i=1}^{n} x_i A_i^{(j)} \succeq 0, \quad j = 1, 2, \ldots, q, \tag{1.1}$$

where $x \in \mathbb{R}^n$ is a variable and each $A_i^{(j)}$ is a m_j x m_j symmetric matrix. Linear matrix inequality (LMI) constraints have been well-studied especially in the field of semidefinite programming ([1], [9]. LMI constraints have applications in a variety of fields including engineering, geometry and statistics. We assume that feasible determined by the constraints is bounded and has a nonempty interior. This means that the set $\{diag(A_1^{(1)}, \ldots, A_1^{(q)}), \ldots, diag(A_n^{(1)}, \ldots, A_n^{(q)})\}$ is linearly independent ([11]).

In this paper, we are concerned with computing weighted analytic center for LMIs using Infeasible Newton's method. A feasible starting point is not required to start the method. In the special case of linear constraints, weighted analytic center has been studied extensively in the past (for example, [3]). A weighted analytic center for LMIs which extends the definition given in [3] was given in [7] and [4]. The study weighted analytic center is of interest in its own right. Many algorithms for linear programming and semidefinite programming are based on weighted analytic centers ([8], [6], [9],[11]).

Weighted analytic center for linear matrix inequalities can be found using the Standard Newton's method by minimizing the barrier function. This approach has the drawback that a starting in the interior of the feasible region must be given. Also, Newton's method does not work well when some of the weights are relatively very large relative to the other weights. Infeasible Newton's method for analytic center for single LMI constraint is given in [10]. We present infeasible Newton's method for finding weighted analytic center that can be started from any point. The method is applied to the Karush-Kuhn-Tucker (KKT) system of equations for the weighted analytic center problem. We implement the method using backtracking line search technique and also study the effect of large weights on the method. We use numerical experiments to compare infeasible Newton's method with the Standard Newton's method.

We find that infeasible Newton's moves very quickly into the interior of the feasible regions for most of our test problems. It seems to be a suitable method for finding an interior point for the system by setting each weight to be 1. It works better than the Standard Newton's method if none of the weights is relatively very large with respect to the other weights. We also find that infeasible Newton's method is more sensitive to large variations in the weights than the Standard Newton's method. In the case of very large variation in the weights, we recommend using Infeasible Newton's method to get into the interior with each weight set to 1, and then switching to the Standard Newton's method for convergence to the weighted analytic center using the original weights and starting from the interior.

2 Weighted Analytic Center for Linear Matrix Inequalities

Let $\mathcal{R}$ denote the feasible region defined by the inequalities (1.1). Given $\omega > 0$, define the barrier function $\phi_\omega(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$ by:

$$\phi_\omega(x) = \begin{cases} \sum_{j=1}^{q} \omega_j \text{logdet}[(A^{(j)}(x))^{-1}] & \text{if } x \in int(\mathcal{R}) \\ \infty & \text{otherwise} \end{cases}$$

In this section, we describe the infeasible Newton's method for finding weighted analytic center. "The problem of computing the weighted analytic center in (2.4) is a more general form of the determinant maximization problem ([11]). Its dual is given by:

$$\textbf{maximize} \sum_{j=1}^{q} \omega_j \text{logdet}(\frac{1}{\omega_j} Z^{(j)}) + \sum_{j=1}^{q} \omega_j m_j - \sum_{j=1}^{q} A_0^{(j)} \bullet Z^{(j)} \tag{2.1}$$

$$\text{subject to} \sum_{j=1}^{q} A_i^{(j)} \bullet Z^{(j)} = 0, \quad (i = 1, \ldots, n) \tag{2.2}$$

$$Z^{(j)} \succ 0, \quad (j = 1, \ldots, q) \tag{2.3}$$

where the bullet $\bullet$ is the matrix dot-product. Theorem 3.1 gives optimality conditions for computing the weighted analytic center $x_{ac}(\omega)$:

Theorem 2.1 *([11]) Suppose both the primal problem (2.4) and the dual problem (3.1) - (3.3) are strictly feasible. The set of primal optimal solutions x and dual* *optimal solutions Z is the set of feasible solutions to the system*

$$\begin{aligned} A_0^{(j)} + \sum_{i=1}^{n} x_i A_i^{(j)} - Y^{(j)} &= 0, \ (j = 1, \ldots, q) \\ \sum_{j=1}^{q} A_i^{(j)} \bullet Z^{(j)} &= 0, \quad (i = 1, \ldots, n) \\ Z^{(j)} Y^{(j)} &= w_j I_{m_j}, \ (j = 1, \ldots, q) \\ Y^{(j)} &\succ 0, \ (j = 1, \ldots, q) \\ Z^{(j)} &\succ 0, \ (j = 1, \ldots, q) \end{aligned}$$

The **weighted analytic center** for the system (1.1) is defined by ([7], [4]):

$$x_{ac}(\omega) = argmin\{\phi_\omega(x) \mid x \in \mathbb{R}^n\} \tag{2.4}$$

This definition extends that given in [3] for linear constraints. When $\omega = [1, \ldots, 1]$, $x_{ac}(\omega)$ is called the **analytic center**. Weighted analytic center has been used in interior point methods for linear programs and semidefinite programs ([8], [6], [9], [11] [2]). The primal-dual central path in semidefinite programming converges to the analytic center of the optimal solution set [5].

The Standard Newton's method has the choice for finding weighted analytic center. The gradient and Hessian of the barrier function $\phi_\omega(x)$ are given by [4]:

For $i, j = 1, \ldots, n$

$$\begin{aligned} \nabla_i \phi_\omega(x) &= -\sum_{j=1}^{q} \omega_j (A^{(j)}(x))^{-1} \bullet A_i^{(j)} \\ H_{ij}(x) &= \sum_{k=1}^{q} \omega_k [(A^{(k)}(x))^{-1} A_i^{(k)}]^T \bullet [(A^{(k)}(x))^{-1} A_j^{(k)}] \end{aligned}$$

The Standard Newton's Method for Computing Weighted Analytic Center

Input: an interior point x, tolerance $TOL > 0$

Set $k = 1$

Repeat

1. Compute the Newton's direction $s = -[H(x)]^{-1}\nabla\phi_\omega(x)$
2. Compute $d = \sqrt{s^T H(x) s}$
3. Do line search to get stepsize h
4. Update $x := x + hs$
5. Update $k = k + 1$

Until $d \leq TOL$

Line search technique such as Backtracking line search technique can be used in Newton's method to find weighted analytic center ([10]).

3 Infeasible Newton's Method for Computing Weighted Analytic Center

In this section, we describe the infeasible Newton's method for finding weighted analytic center. "The problem of computing the weighted analytic center in (2.4) is a more general form of the determinant maximization problem ([11]). Its dual is given by:

$$\textbf{maximize} \sum_{j=1}^{q} \omega_j \log(\frac{1}{\omega_j}) + \sum_{j=1}^{q} \omega_j m_j - \sum_{j=1}^{q} A_0^{(j)} \bullet Z^{(j)} \tag{3.1}$$

$$\text{subject to} \sum_{j=1}^{q} A_i^{(j)} \bullet Z^{(j)} = 0, \quad (i = 1, \ldots, n) \tag{3.2}$$

$$Z^{(j)} \succ 0, \quad (j = 1, \ldots, q) \tag{3.3}$$

where the bullet $\bullet$ is the matrix dot-product. Theorem 3.1 gives optimality conditions for computing the weighted analytic center $x_{ac}(\omega)$:

Theorem 3.1 *([11]) Suppose both the primal problem (2.4) and the dual problem (3.1) - (3.3) are strictly feasible. The set of primal optimal solutions x and dual optimal solutions Z is the set of feasible solutions to the system*

$$\begin{aligned} A_0^{(j)} + \sum_{i=1}^{n} x_i A_i^{(j)} - Y^{(j)} &= 0, \ (j = 1, \ldots, q) \\ \sum_{j=1}^{q} A_i^{(j)} \bullet Z^{(j)} &= 0, \quad (i = 1, \ldots, n) \\ Z^{(j)} Y^{(j)} &= w_j I_{m_j}, \ (j = 1, \ldots, q) \\ Y^{(j)} &\succ 0, \ (j = 1, \ldots, q) \\ Z^{(j)} &\succ 0, \ (j = 1, \ldots, q) \end{aligned}$$

The optimality conditions in Theorem 3.1 can be written equivalently as:

$$\begin{bmatrix} A_0^{(1)} + \sum_{i=1}^n x_i A_i^{(1)} - Y^{(1)} \\ \vdots \\ A_0^{(q)} + \sum_{i=1}^n x_i A_i^{(q)} - Y^{(q)} \\ \sum_{j=1}^q A_1^{(j)} \bullet Z^{(j)} \\ \vdots \\ \sum_{j=1}^q A_n^{(j)} \bullet Z^{(j)} \\ Z^{(1)} Y^{(1)} - w_1 I_{m_1} \\ \vdots \\ Z^{(q)} Y^{(q)} - w_q I_{m_q} \end{bmatrix} = 0 \tag{3.4}$$

$$Z^{(j)} \succ 0, \ (j = 1, \ldots, q) \tag{3.5}$$

$$Y^{(j)} \succ 0, \ (j = 1, \ldots, q) \tag{3.6}$$

Now, as in [11], let

$$
\begin{aligned}
z^{(j)} &= vec\ Z^{(j)} \\
y^{(j)} &= vec\ Y^{(j)} \\
B^{(j)} &= \begin{bmatrix} (vec\ A_1^{(j)})^T \\ \vdots \\ (vec\ A_n^{(j)})^T \end{bmatrix} \\
R_p^{(j)} &= Y^{(j)} - A_0^{(j)} - mat\ (B^{(j)})^T x \\
R_d &= \begin{bmatrix} -\sum_{j=1}^q A_1^{(j)} \bullet Z^{(j)} \\ \vdots \\ -\sum_{j=1}^q A_n^{(j)} \bullet Z^{(j)} \end{bmatrix} \\
R_c^{(j)} &= \omega_j I_{m_j} - Z^{(j)} Y^{(j)} \\
r_p^{(j)} &= vec\ R_p^{(j)} = y^{(j)} - vec\ A_0^{(j)} - (B_j^{(j)})^T x \\
r_d &= vec\ R_d = -\sum_{j=1}^q B^{(j)} z^{(j)} \\
r_c^{(j)} &= vec\ R_c = vec(\omega_j I_{m_j}) - (I_{m_j} \otimes Z^{(j)}) y^{(j)} \\
r_c^{(j)} &= vec\ R_c = vec(\omega_j I_{m_j}) - (Y^{(j)} \otimes I_{m_j}) z^{(j)}
\end{aligned}
$$

$$
G(x, y^{(1)}, \ldots, y^{(q)}, z^{(1)}, \ldots, z^{(q)}) = \begin{bmatrix} -r_p^{(1)} \\ \vdots \\ -r_p^{(q)} \\ -r_d \\ -r_c^{(1)} \\ \vdots \\ -r_c^{(q)} \end{bmatrix}
$$

where *vec* is the map that stacks the columns of a matrix on top of each other into a single vector and *mat* is the inverse map. Also, let

$$
\begin{aligned}
r_p &= \begin{bmatrix} r_p^{(1)} \\ \vdots \\ r_p^{(q)} \end{bmatrix}, \; r_c = \begin{bmatrix} r_c^{(1)} \\ \vdots \\ r_c^{(q)} \end{bmatrix} \\
y &= \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(q)} \end{bmatrix}, \; z = \begin{bmatrix} z^{(1)} \\ \vdots \\ z^{(q)} \end{bmatrix} \\
A &= [B^{(1)}, \ldots, B^{(q)}] \\
E &= \mathrm{diag}(Y^{(1)} \otimes I_{m_1}, \ldots, Y^{(q)} \otimes I_{m_q}) \\
F &= \mathrm{diag}(I_{m_1} \otimes Z^{(1)}, \ldots, I_{m_q} \otimes Z^{(q)}) \\
I &= \mathrm{diag}(I_{m_1^2}, \ldots, I_{m_q^2})
\end{aligned}
$$

Then, the system of equations (3.4) becomes $G(x, y, z) = 0$. The Newton's directions for the system are found by solving the linear system:

$$
\begin{bmatrix} A^T & -I & 0 \\ 0 & 0 & A \\ 0 & F & E \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}
$$

Using block elimination, we get

$$M\Delta x = AE^{-1}(Fr_p + r_c) - r_d \tag{3.7}$$

$$\Delta z = E^{-1}(F(r_p - A^T\Delta x) + r_c) \tag{3.8}$$

$$\Delta y = F^{-1}(r_c - E\Delta z) \tag{3.9}$$

where

$$M = AE^{-1}FA^T$$

The matrix M Is positive definite. The following is an iteration of the infeasible Newton's method. In Step 2, the iterate $\Delta Z^{(j)}$ is not symmetric. We symmetrize $\Delta Z^{(j)}$ in Step 3.

An Iteration of Infeasible Newton's Method for Computing Weighted Analytic Center

Step 1: Compute Newton's direction $(\Delta x,\ \Delta y,\ \Delta z)$ using equations (3.7)-(3.9)

This gives $(\Delta x,\ \Delta y^{(1)}, \ldots, \Delta y^{(q)},\ \Delta z^{(1)}, \ldots, \Delta z^{(q)})$

Step 2: For each j, determine:

$$\Delta Y^{(j)} = mat\ \Delta y^{(j)}$$

$$\Delta Z^{(j)} = mat\ \Delta z^{(j)}$$

Step 3: Symmetrize $\Delta Z^{(j)}$

Replace $\Delta Z^{(j)}$ by $\frac{1}{2}(\Delta Z^{(j)} + (\Delta Z^{(j)})^T)\ \ (j = 1, \ldots, q)$

Step 4: Do line search to get stepsize h

Step 5: Update the iterates

$$\begin{aligned} x &\leftarrow x + h\Delta x \\ Y^{(j)} &\leftarrow Y^{(j)} + h\Delta Y^{(j)} \quad (j = 1, \dots, q) \\ Z^{(j)} &\leftarrow Z^{(j)} + h\Delta Z^{(j)} \quad (j = 1, \dots, q) \end{aligned}$$

Any point $x \in \mathbb{R}^n$ can be picked as a starting point. Then, for $j = 1, \dots, q$ choose

$$\begin{aligned} Y^{(j)} &= \begin{cases} A^{(j)}(x) & \text{if } A^{(j)}(x) \succ 0 \\ I_{m_j} & \text{otherwise} \end{cases} \\ Z^{(j)} &= \omega_j (Y^{(j)})^{-1} \end{aligned}$$

The above iteration is repeated until $\| r(x, y, z) \| < TOL$, where $r = (r_p, r_d, r_c)$ is the residual and TOL is a given tolerance. One can use backtracking line search ([10]) or other techniques to get the stepsize h.

4 Numerical Experiments

In this section, we give numerical experiments to compare infeasible Newton's method with the Standard Newton's method. We also investigate the effects of large weights on the two methods.

Table 1 describes the 35 random test problems used for our numerical experiments. The second column of Table 1 gives the dimension n of the ambient space and the third column is the number q of constraints. The dimensions m_j of the matrices are given in the fourth column. For each random problem, n, q and m_j are given and the LMI $A_0^{(j)} + \sum_{i=1}^{n} x_i A_i^{(j)} \succeq 0$ was generated randomly as follows: $A_0^{(j)}$ is an $m_j \times m_j$ diagonal matrix with each diagonal entry chosen from $U(0,1)$. Each $A_i^{(j)}$ $(1 \leq i \leq n)$ is a random $m_j \times m_j$ symmetric and sparse matrix with approximately $0.8 * m_j^2$ nonzero entries generated using the Matlab command $sprandsym(m_j, 0.8)$. Each problem has a nonempty interior.

LMI Test Problem	n	q	m
1	2	2	[2,1]
2	3	4	[3,4,1,2]
3	2	2	[2,2]
4	5	3	[4,1,3]
5	4	3	[5,4,3]
6	4	5	[4,3,1,1,4]
7	3	3	[4,2,3]
8	3	4	[4,2,2,5]
9	5	3	[4,1,1]
10	3	5	[5,3,5,1,4]
11	2	7	[2,5,3,5,2,5,1]
12	5	6	[5,1,3,4,1,4]]
13	14	5	[5,1,3,4,2]
14	20	5	[5,2,5,1,5]
15	3	8	[5,4,1,5,3,5,1,3]
16	9	7	[1,4,2,4,4,2,2]
17	6	5	[4,4,2,1,4]
18	10	2	[3,5]
19	15	9	[2,5,3,1,2,3,3,1,2]
20	8	2	[4,5]
21	19	7	[5,2,2,2,5,5,5]
22	9	10	[3,4,1,1,3,5,5,4,5,2]
23	3	4	[2,3,2,5]
24	8	2	[5,1]
25	2	8	[5,2,1,1,1,5,3,3]
26	13	8	[4,1,4,2,3,1,2,1]
27	24	10	[5,4,5,1,4,2,3,5,5,2]
28	5	6	[4,1,4,2,1,3]
29	16	3	[2,2,3]
30	2	2	[4,5]
31	2	4	[1,5,5,5]
32	4	4	[5,1,4,5]
33	4	4	[1,2,3,5]
34	17	9	[1,5,2,1,2,5,1,4,3]
35	2	8	[5,2,1,1,1,5,3,3]

Table 1: Test Problems

Our codes were written in Matlab and ran on Dell OPTIPLEX 880 computer.

Both infeasible Newton's method and the Standard Newton's method were implemented using a tolerance of $TOL = 10^{-4}$ and up to a maximum of 500 iterations. The starting point is random such that each of its components is chosen from a normal distribution with mean 0 and variance 10^6. We use the backtracking line search technique in the two methods. Table 2 compares Infeasible Newton's method with the Standard Newton's method for different sets of weights. In each of Problems 1-15, one weight was set at 10^{12}, which is very high value relative to the others. For Problems 16-35, none of the weights was relatively very large. In Table 2, the third and the fourth column give the number of iterations and time required to find the first point in the interior of the feasible region (respectively) by the Infeasible Newton's method. The fourth and the fifth columns give the number of iterations and time required to find the weighted analytic center (respectively) starting from the first interior point found. The sixth and the seventh columns in Table 2 give the number of iterations and time required to find the weighted analytic center (respectively) by the Standard Newton's method. The Standard Newton's method is started from the same interior point as in Infeasible Newton's method. Infeasible Newton's method and the Standard Newton's method are compared in Table 3 with weights $\omega = [1, 1, \ldots, 1]$.

Figure 1 shows the iterates taken by Infeasible Newton's method to converge in Problem 3 with $\omega = [10^{12}, 1000]$. It is clear from the figure that Infeasible Newton's method slowed down considerably before converging to the weighted analytic center. On the other hand, as seen in Figure 2, the method converged quickly with $\omega = [1, 1]$. Figure 3 shows how the norm of the gradient varies with the number of iterations for the two values of the weights. In Table 2, the entry $\star$ means that Infeasible Newton's method has failed to converge after the maximum number of 500 iterations

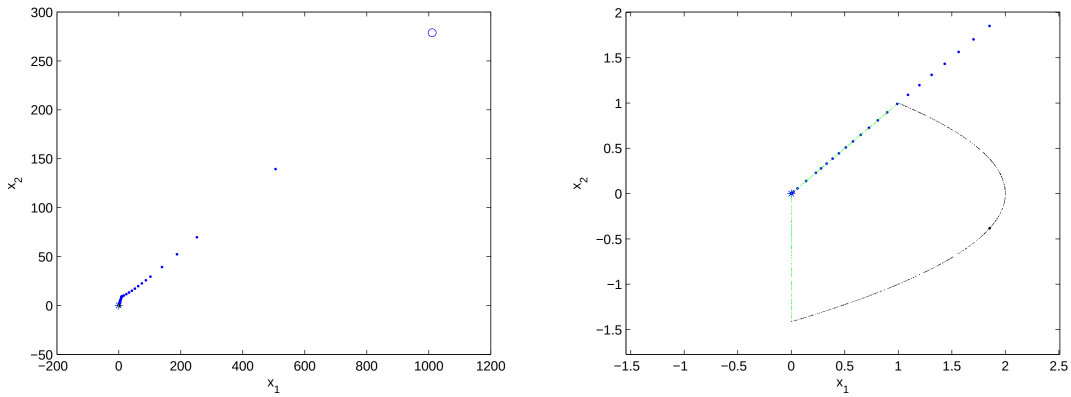


Figure 1: Iterates (51+46=97) taken by Infeasible Newton's method to converge in Problem 3 with $\omega = [10^{12}, 1000]$. The graph on the right is the graph on the left zoomed and showing the last iterate $\star$ in the interior, but near the boundary of the feasible region.

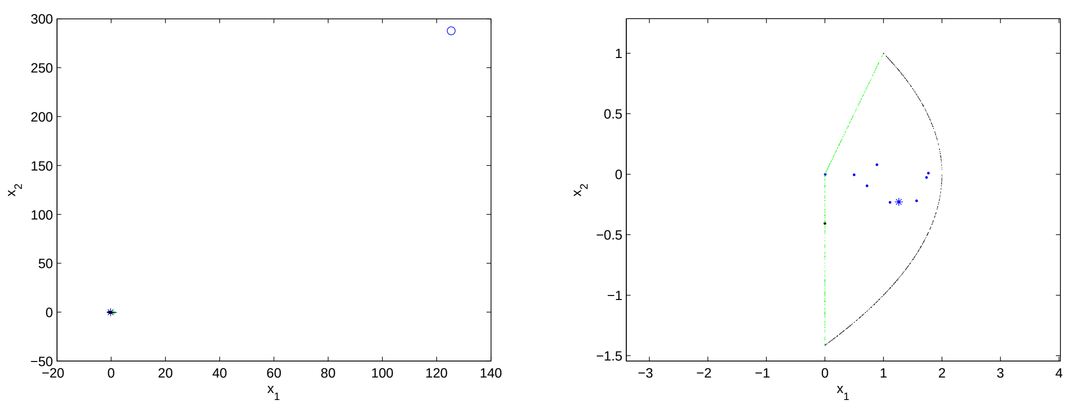


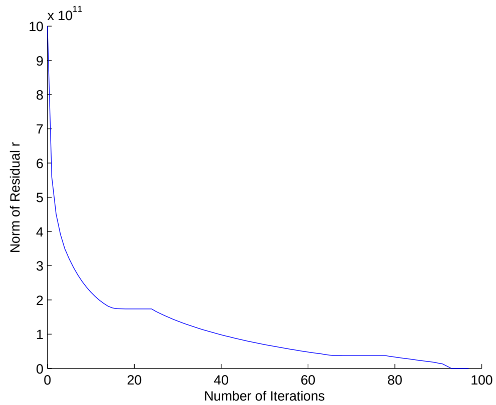
Figure 2: Iterates (1+9=10) taken by Infeasible Newton's method to converge in Problem 3 with $\omega = [1, 1]$. The graph on the right is the graph on the left zoomed and showing the last iterate $\star$ well inside the interior of the feasible region.

Problem	Weights ω	Inf Newton 1st Feas Point Iter	Inf Newton 1st Feas Point Time(sec)	Inf Newton Iter	Inf Newton Time (sec)	St Newton Iter	St Newton Time (sec)
1	$[10^{12},10]$	2	0.0185	4	0.0096	2	0.0056
2	$[10^{12},100,100,1]$	3	17.2562	*	*	3	0.0161
3	$[10^{12},1000]$	51	0.3771	46	0.1690	41	0.1042
4	$[10^{12},10,1]$	20	10.8898	*	*	9	0.0687
5	$[10^{12},1,10]$	2	8.8548	*	*	4	0.0215
6	$[1,10^{12},1,10,100]$	*	*	*	*	*	*
7	$[100,10,10^{12}]$	129	9.5582	*	*	33	0.1810
8	$[1,1000,10^{12},10]$	*	*	*	*	*	*
9	$[10^{12},1000,1000]$	1	8.5943	*	*	4	0.0278
10	$[10^{12},1000,100,1000,100]$	6	25.3801	*	*	2	0.0158
11	$[10^{12},10,100,1000,1000,10,100]$	*	*	*	*	*	*
12	$[10^{12},1,100,10,1,10]$	3	17.5497	*	*	5	0.0701
13	$[1000,10^{12},1000,100,100]$	*	*	*	*	*	*
14	$[1,1000,10^{12},1000,1]$	*	*	*	*	*	*
15	$[1,1,1,100,10^{12},100,100,10]$	*	*	*	*	*	*
16	[10,1,10,1000,1000,10,1]	24	0.2885	3	0.0234	5	0.2108
17	[100,100,10,1,1000]	3	0.0508	4	0.0252	3	0.0534
18	[1,1000]	2	0.0286	6	0.0182	4	0.0579
19	[100,10,1000,1,1000,100,100,1000,10]	10	0.2117	7	0.0749	8	0.9943
20	[1000,1000]	1	0.0180	4	0.0125	7	0.0664
21	[1,1,10,1,100,10,1]	21	0.3301	3	0.0342	3	0.4761
22	[1,1000,100,10,1000,1,100,100,100,1]	7	0.2006	3	0.0491	3	0.2011
23	[100,10,1,1]	4	0.0609	5	0.0231	6	0.0297
24	[1000,1]	12	0.0564	2	0.0052	3	0.0348
25	[1,1,10,1,1000,1,1,10]	8	0.1360	3	0.0280	3	0.0224
26	[100,1,100,1,1,1,1,1]	20	0.2815	2	0.0208	3	0.3256
27	[1,1,1,10,1,10,1,100,1,10]	8	0.3212	4	0.0945	6	2.1028
28	[10,10,100,1,1,1]	7	0.0986	3	0.0246	4	0.0718
29	[1,1,10]	1	0.0092	*	*	*	*
30	[10,1]	2	0.0268	4	0.0149	3	0.0121
31	[1,10,1,1]	1	0.0444	4	0.0326	4	0.0270
32	[100,1,10,1]	8	0.0994	4	0.0278	4	0.0444
33	[100,100,1,10]	6	0.0612	3	0.0168	3	0.0318
34	[1,100,1,1,1,10,100,100,10]	7	0.1870	3	0.0426	5	0.9478
35	[1,10,100,1,1]	31	0.3678	2	0.0175	2	0.3845

Table 2: Infeasible Newton's vs. Standard Newton's Methods Using Different Weights ω

Problem	Inf Newton 1st Feas Point Iter	Inf Newton 1st Feas Point Time (sec)	Inf Newton Iter	Inf Newton Time (sec)	St Newton Iter	St Newton Time (sec)
1	1	0.0115	3	0.0068	3	0.0059
2	2	0.0327	4	0.0188	3	0.0177
3	1	0.0318	9	0.0271	10	0.0180
4	2	0.0230	3	0.0114	3	0.0249
5	1	0.0250	4	0.0180	4	0.0239
6	1	0.0348	4	0.0248	4	0.0355
7	1	0.0271	5	0.0206	7	0.0247
8	1	0.0284	4	0.0202	3	0.0170
9	1	0.0215	4	0.0145	3	0.0224
10	1	0.0641	6	0.0532	8	0.0520
11	1	0.0451	3	0.0311	3	0.0207
12	2	0.0460	3	0.0237	3	0.0461
13	2	0.0555	4	0.0268	6	0.3949
14	13	0.1481	4	0.0302	6	0.8056
15	1	0.0805	5	0.0629	3	0.0330
16	2	0.0599	4	0.0339	7	0.2742
17	2	0.0370	3	0.0185	3	0.0508
18	2	0.0184	3	0.0088	4	0.0569
19	3	0.0933	4	0.0390	3	0.4423
20	1	0.0196	4	0.0134	7	0.0665
21	12	0.2080	4	0.0463	5	0.7110
22	2	0.1079	4	0.0659	3	0.2002
23	1	0.0276	4	0.0196	3	0.0164
24	4	0.0281	4	0.0123	6	0.0616
25	2	0.0553	3	0.0286	3	0.0217
26	13	0.1980	3	0.0303	3	0.3295
27	3	0.1791	2	0.0902	2	1.6525
28	3	0.0612	3	0.0262	3	0.0567
29	1	0.0102	*	*	*	*
30	2	0.0275	4	0.0157	3	0.0140
31	1	0.0359	3	0.0231	3	0.0210
32	2	0.0431	3	0.0212	2	0.0264
33	4	0.0573	4	0.0240	6	0.0543
34	2	0.0992	4	0.0550	4	0.7863
35	13	0.1795	4	0.0367	6	0.8997

Table 3: Infeasible Newton's vs. Standard Newton's Methods Using Weights $\omega = [1, 1, \dots, 1]$



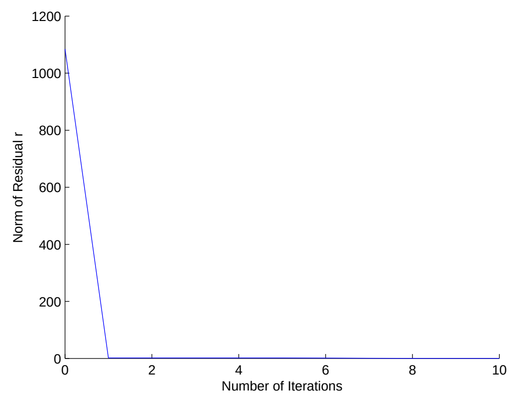


Figure 3: The graph on the left gives the number of iterations vs. norm of the residual in Problem 3 with $\omega = [10^{12}, 1000]$ using Infeasible Newton's method. The graph on the right gives the number of iterations vs. norm of the residual of the method with $\omega = [1, 1]$.

in Problems 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 29. However, it managed to find an interior point of (1.1) in Problems 2, 4, 5, 7, 9,10, 12 and 29. We see from the table that both methods might not work well when some of the weights are relatively larger than the other weights. The results also show that the Standard Newton's method performs better than Infeasible Newton's method in this case. In Infeasible Newton's method, the Jacobian of the residual function becomes increasingly ill-conditioned near the boundary of the systems (3.5) and (3.6) due to matrices E and F as the variation among the weights increases. Observe that Infeasible Newton's method failed to converge in 8 problems, even though it found a point in the interior of the feasible region (1.1). Standard Newton's method converged (within the 500 iterations limit) for each of the 8 problems except in Problem 29. Standard Newton's method converged after 536 iterations in Problem 29.

In Table 3, the entry $\star$ in Problem 29 indicates that Infeasible Newton's method has failed to converge to the analytic center after the maximum number of 500 iterations. However, it managed to find an interior point of (1.1) after 1 iteration.

Problem 29 shows both Infeasible Newton's method and Standard Newton's method may fail to converge (within 500 iterations) even if the weights are all equal and an interior point is found. We see from Table 3 that Infeasible Newton's method took less number of iterations in 12 out of 35 problems while the Standard Newton's method took less number of iterations in 9 out of 35 problems. Infeasible Newton's method took less time in 23 out of 35 problems and the Standard Newton's method took less time in 11 out of 35 problems. It is interesting to note from the table that Infeasible Newton's method found an interior point of (1.1) within 1-3 iterations on most of the test problems.

The results from Table 2 and Table 3 suggest that when none of the weights is relatively very large, Infeasible Newton's method is a better method than the Standard Newton's method to find the weighted analytic center. When one weight is relatively very large, one could use Infeasible Newton's method with $\omega = [1, 1, \ldots, 1]$ to find an interior point, and then switch to Standard Newton's method using the original weights and starting from the interior point found.

5 Conclusion

We presented Infeasible Newton's method for computing weighted analytic center for system of linear matrix inequalities and compared it with the Standard Newton's method.

We found that Infeasible Newton's mehod finds a point in the interior point fairly quickly, starting from any point, especially when none of the weights is relatively very large. When none of the weights is relatively very large, Infeasible Newton's method seems to work better than the Standard Newton's method to find the weighted analytic center. However, Infeasible Newton's method does not work well when some of the weights are very large relative to others, but it still often finds a point in the interior, starting from any point. We find that infeasible Newton's method is suitable for finding an interior point for the system by setting each weight to be 1. We recommend that when some weights are relatively very large, one should use Infeasible Newton's method to find a point in the interior with $\omega = [1, 1, \ldots, 1]$ to find an interior point, and then switch to Standard Newton's method using the original weights and starting from the interior point found.

To improve the efficiency of Infeasible Newton's method for finding weighted analytic center, it would be useful to exploit the block-diagonal structure of the matrices A, E and F in future implementation. We would like to investigate how the method could handle weights where some are relatively much larger than the other weights. Furthermore, it would be of interest to study the application of Infeasible Newton's method for weighted analytic center to the SDP-CUT algorithm for semidefinite programming, presented in [6]. Implementing Infeasible Newton's method for weighted analytic center using different search directions is currently under investigation. The problem of weighted analytic center is a determinant maximization problem and can be solved with the algorithms for those problems. It would be of interest to compare the performance of those algorithms with the method presented in this paper.

6 Acknowledgment

I would like to thank Jubail University College in Saudi Arabia for providing a home for my 2014-2015 sabbatical leave. This research work was done during this leave.

References

[1] F. Alizadeh, "Interior Point Methods in Semidefinite Programming with Applications to Combinatorial Optimization", *SIAM Journal on Optimization*, Vol. 5, No. 1, 1995, pp. 13-51.

[2] F. Alizadeh, J. A. Haeberly and M. Overton, "Primal-Dual Methods for Semidefinite Programming: Convergence Rates, Stability and Numerical Results", *SIAM Journal on Optimization*, Vol. 8, 1998, no. 3, pp. 746-768.

[3] D. S. Atkinson and P. M. Vaidya, "A scaling technique for finding the weighted analytic center of a polytope," *Math. Prog.*, vol. 57, 1992, pp. 163–192

[4] S. Jibrin and J. W. Swift, "The Boundary of the Weighted Analytic Center for Linear Matrix inequalities." *Journal of Inequalities in Pure and Applied Mathematics*, Vol. 5, Issue 1, Article 14, 2004.

[5] Z.-Q. Luo, J. F.Sturm, and S. Zhang, "Superlinear convergence of a symmetric primal-dual path following algorithm for semide

nite programming", *SIAM Journal on Optimization*, vol. 8, 1998, pp. 59-81.

[6] J. Machacek and S. Jibrin, "An Interior Point Method for Solving Semidefinite Programs Using Cutting Planes and Weighted Analytic Centers", *Journal of Applied Mathematics*, Vol. 2012, Article ID 946893, 21 pages

[7] I. S. Pressman and S. Jibrin, "A Weighted Analytic Center for Linear Matrix Inequalities", *Journal of Inequalities in Pure and Applied Mathematics*, Vol. 2, Issue 3, Article 29, 2002.

[8] J. Renegar, "A polynomial-time algorithm, based on Newton's method, for linear programming", *Math. Programming*, Vol. 40, 1988, pp. 59–93.

[9] L. Vandenberghe and S. Boyd, "Semidefinite Programming", *SIAM Review*, Vol. 38, 1996, pp. 49-95.

[10] L. Vandenberghe and S. Boyd, "Convex Optimization", Cambridge University Press, New York 2004.

[11] L. Vandenberghe, S.-P. Boyd and S. Wu, "Determinant Maximization with Linear Matrix Inequality Constraints", *SIAM Journal on Matrix Analysis*, Vol. 19, no. 2, 1998, pp. 499-533.