Convergence Analysis of a Long-Step Primal-Dual Infeasible Interior-Point LP Algorithm Based on Iterative Linear Solvers

Renato D.C. Monteiro* Jerome W. O'Neal †

October 31, 2003

Abstract

In this paper, we consider a modified version of a well-known long-step primal-dual infeasible IP algorithm for solving the linear program $\min\{c^T x : Ax = b,\, x \geq 0\}$, $A \in \mathbb{R}^{m \times n}$, where the search directions are computed by means of an iterative linear solver applied to a preconditioned normal system of equations. We show that the number of (inner) iterations of the iterative linear solver at each (outer) iteration of the algorithm is bounded by a polynomial in m, n and a certain condition number associated with A, while the number of outer iterations is bounded by $\mathcal{O}(n^2 \log \epsilon^{-1})$, where ϵ is a given relative accuracy level. As a special case, it follows that the total number of inner iterations is polynomial in m and n for the minimum cost network flow problem.

Keywords: Linear programming, interior-point methods, polynomial bound, network flow problems, condition number, preconditioning, iterative methods for linear equations, normal matrix.

AMS 2000 subject classification: 65F35, 90C05, 90C35, 90C51.

1 Introduction

Consider the standard-form linear programming (LP) problem

$$\min\{c^T x : Ax = b,\, x \geq 0\} \tag{1}$$

*School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0205. (email: `monteiro@isye.gatech.edu`). This author was supported in part by NSF Grants CCR-0203113 and INT-9910084 and ONR grant N00014-03-1-0401.

†School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0205 (email: `joneal@isye.gatech.edu`).

which we refer to as the *primal* problem, and its associated *dual* problem

$$\max\{b^T y : A^T y + s = c,\ s \geq 0\}, \tag{2}$$

where the data consists of $(A, b, c) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$ and the primal-dual variable consists of $(x, s, y) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m$.

This paper deals with interior-point (IP) algorithms for solving the pair of LP problems (1) and (2) whose search directions are computed by means of iterative linear solvers. We refer to such algorithms as *iterative IP methods*. An outer iteration of an iterative IP algorithm is similar to that of an exact IP method, except that the Newton search directions are computed approximately by means of an iterative linear solver. In this context, the iterations of the linear solver will be referred to as the *inner iterations* of the iterative IP method. We will consider an iterative version of the long-step primal-dual infeasible IP algorithm considered in [4, 16] and show that its total number of inner iterations is polynomially bounded by m, n and a certain condition number associated with A, while the number of outer iterations is bounded by $\mathcal{O}(n^2 \log \epsilon^{-1})$, where ϵ is a given relative accuracy level.

Given a current iterate (x, s, y) in a generic primal-dual IP algorithm, one usually solves for the Newton search direction $(\Delta x, \Delta s, \Delta y)$ using one of two well-known approaches. In the first approach, one solves an *augmented system* for $(\Delta x, \Delta y)$, which then immediately yields Δs. In the second one, one solves the *normal equation* system

$$AD^2 A^T \Delta y = p, \tag{3}$$

for some appropriate $p \in \mathbb{R}^m$, where $D^2 = S^{-1}X$. Our method is based on the second approach where we solve (3) using an iterative solver.

Most IP solvers using the normal equation usually solve (3) via a *direct solver* which computes a factorization of AD^2A^T to obtain Δy. The use of an iterative linear solver to solve (3) has two main potential advantages: (i) it can be significantly faster than a direct solver in some LP instances (see e.g. [10, 12]); and (ii) it can take complete advantage of the sparsity of the matrix A. However, iterative solvers possess two potential drawbacks when compared with direct solvers: (a) they solve (3) less accurately than direct methods; and (b) they may have slow convergence if the coefficient matrix AD^2A^T is ill-conditioned.

The effect of item (a) above is that the search direction $(\Delta x, \Delta s, \Delta y)$ can only satisfy the Newton system approximately, regardless of the choice of Δx and Δs. In our approach, we will choose these components so that the equations of the Newton system corresponding to primal and dual feasibility are satisfied exactly, while the equation corresponding to the centrality condition is violated. This way of choosing the search direction is crucial for us to establish that the number of outer iterations of our method is polynomially bounded (see Section 2.4).

Item (b) has been a significant problem for those wishing to use iterative methods to solve (3). It is well-known that in degenerate cases, the condition number of AD^2A^T "blows up" as we approach an optimal solution, even if our iterates (x, s, y) lie on the central path (see e.g. [6]). A cure to this problem is to use a preconditioner T so as to make the condition number of $TAD^2A^TT^T$ small. One such preconditioner was introduced by Resende and Veiga [12] in the context of the minimum cost network flow problem, and later generalized by Oliveira and Sørensen [10] for general LP problems. The proof that the above preconditioner makes the condition number of $TAD^2A^TT^T$ uniformly bounded regardless of the values of the diagonal elements of D was proved by Monteiro, O'Neal, and Tsuchiya [9]. In view of this nice property, we will use this preconditioner in our algorithm.

Global convergence analysis of algorithms using inexact search directions has been presented in several papers (see e.g. [3, 4, 5, 8]). Several authors have also used iterative linear solvers to compute an approximate Newton search direction (see e.g. [1, 5, 10, 11, 12]). In particular, Resende and Veiga [12] and Oliveira and Sørensen [10] used an iterative solver in conjunction with the above preconditioner to compute (3) inexactly for network-flow problems and general LP problems, respectively. Their computational results show that iterative IP methods can be extremely useful in practice. To our knowledge, though, no one to date has obtained strong theoretical arithmetic complexities for iterative IP methods.

Our paper is organized as follows. Section 1.1 describes the terminology and notation used in our paper. Section 2 gives the main results of our paper, and is divided into five parts. Section 2.1 describes an exact variant of an infeasible-interior-point algorithm on which the algorithm we study in this paper is based. Section 2.2 discusses the preconditioner T mentioned above and gives some background results. Section 2.3 states the convergence results about a generic iterative for solving (3). Section 2.4 gives our algorithm and main results, and Section 2.5 discusses the application of our algorithm to network flow problems. In Section 3, we prove the results stated in Section 2, and in Section 4, we give some concluding remarks.

1.1 Terminology and Notation

Throughout this paper, upper-case Roman letters denote matrices, lower-case Roman letters denote vectors, and lower-case Greek letters denote scalars. For a matrix A, $A \in \mathbb{R}^{m\times n}$ means that A is an $m \times n$ matrix with real entries; while for a vector x, $x \in \mathbb{R}^n$ means that x is an n-dimensional real vector. More specifically, $x \in \mathbb{R}^n_+$ means that $x \in \mathbb{R}^n$ and $x_i \geq 0$ for all i, while $x \in \mathbb{R}^n_{++}$ means that $x \in \mathbb{R}^n$ and $x_i > 0$ for all i. The notation $x \circ y$ denotes the Hadamard product of two vectors x and y, i.e. $(x \circ y)_i = x_i y_i$. Next, the vector $|v|$ is the vector whose ith component is $|v_i|$. Also, given a vector v, $\mathrm{Diag}(v)$ is a diagonal matrix whose diagonal elements are the elements of v, i.e. $(\mathrm{Diag}(v))_{ii} = v_i$ for all i.

Three matrices bear special mention: the matrices X, S, and D, all in $\mathbb{R}^{n\times n}$. These matrices are diagonal matrices having the elements of the vectors x, s, and d, respectively, along their diagonals (i.e. $X = \mathrm{Diag}(x)$, $S = \mathrm{Diag}(s)$, and $D = \mathrm{Diag}(d)$). The symbol 0 will be used to denote a scalar, vector, or matrix of all zeroes; its dimensions should be clear from the context. Also, the vector e is the vector of all 1's, whose dimension is implied by the context.

If a matrix $W \in \mathbb{R}^{m\times m}$ is symmetric ($W = W^T$) and positive definite (has all positive eigenvalues), we write $W \succ 0$. The condition number of W, denoted $\kappa(W)$, is the ratio between its maximum eigenvalue divided by its minimum eigenvalue. We will denote sets by upper-case script Roman letters (e.g. $\mathcal{B}$, $\mathcal{N}$). For a set $\mathcal{B}$, we denote the cardinality of the set by $|\mathcal{B}|$. Given a matrix $A \in \mathbb{R}^{m\times n}$ and a set $\mathcal{B} \subseteq \{1,\ldots,n\}$, the matrix $A_\mathcal{B}$ is the submatrix consisting of the columns $\{A_i : i \in \mathcal{B}\}$. Similarly, given a vector $v \in \mathbb{R}^n$ and a set $\mathcal{B} \subseteq \{1,\ldots,n\}$, the vector $v_\mathcal{B}$ is the subvector consisting of the elements $\{v_i : i \in \mathcal{B}\}$.

We will use several different norms throughout the paper. For a vector $z \in \mathbb{R}^n$, $\|z\| = \sqrt{z^T z}$ is the Euclidian norm, and $\|z\|_\infty = \max_{i=1,\ldots,n} |z_i|$ is the "infinity norm". For a matrix $V \in \mathbb{R}^{m\times n}$, $\|V\|$ denotes the operator norm associated with the Euclidian norm: $\|V\| = \max_{z:\|z\|=1} \|Vz\|$. Finally, $\|V\|_F$ denotes the Frobenius norm: $\|V\|_F = (\sum_{i=1}^m \sum_{j=1}^n V_{ij}^2)^{1/2}$.

2 Main Results

The main results of the paper are stated in this section, which is divided into five parts. Section 2.1 gives some background and motivation for the method proposed in the paper. Section 2.2 describes the maximum weight basis preconditioner. Section 2.3 considers a generic iterative linear solver and derives an upper bound on the number of iterations for it to obtain a reasonably accurate solution of (3). Section 2.4 describes how the overall search direction is obtained and states the main algorithm. Finally, Section 2.5 describes a tighter complexity for the algorithm when applied to network flow problems.

2.1 Preliminaries and Motivation

In this subsection, we discuss a well-known infeasible primal-dual long-step IP algorithm (see for example [4] and [16]) which will serve as the basis for the iterative IP method proposed in this paper. We also state the complexity results which have been obtained for this algorithm. For the sake of concreteness, we have chosen to work with one specific primal-dual IP method. We note, however, that our analysis applies to other long-step variants as well as to short-step IP methods.

As stated in the introduction, we will be working with the pair of LPs (1) and (2). Letting $\mathcal{S}$ denote the set of primal-dual optimal solutions $(x, s, y) \in \mathbb{R}^{2n} \times \mathbb{R}^m$ of (1) and (2), it is well-known that $\mathcal{S}$ consists of the triples $(x, s, y) \in \mathbb{R}^{2n} \times \mathbb{R}^m$ satisfying

$$Ax = b, \quad x \geq 0 \tag{4}$$
$$A^T y + s = c, \quad s \geq 0 \tag{5}$$
$$x \circ s = 0. \tag{6}$$

Throughout the paper, we will make the following assumptions:

Assumption 1 *A has full row rank.*

Assumption 2 *The set $\mathcal{S}$ is nonempty.*

For a point $(x, s, y) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$, let us define

$$\begin{aligned} \mu = \mu(x,s) &:= x^T s/n, \\ r_p = r_p(x) &:= Ax - b, \\ r_d = r_d(s,y) &:= A^T y + s - c, \\ r = r(x,s,y) &:= (r_p, r_d). \end{aligned}$$

Moreover, given $\gamma \in (0,1)$ and an initial point $(x^0, s^0, y^0) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$, we define the following neighborhood of the central path:

$$\mathcal{N}(\gamma) := \left\{ (x,s,y) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m : x \circ s \geq (1-\gamma)\mu e, \ \frac{\|r\|}{\|r^0\|} \leq \frac{\mu}{\mu_0} \right\}, \tag{7}$$

where $r^0 := r(x^0, s^0, y^0)$ and $\mu_0 := \mu(x^0, s^0)$. Here, we use the convention that $\nu/0$ is equal to 0 if $\nu = 0$ and ∞ if ν is positive.

The infeasible primal-dual algorithm which will serve as the basis for our iterative IP method is as follows:

Algorithm IIP

1. **Start:** Let $\epsilon > 0$, $\gamma \in (0,1)$, $(x^0, s^0, y^0) \in \mathcal{N}(\gamma)$ and $0 < \underline{\sigma} < \overline{\sigma} < 1$ be given. Set $k = 0$.
2. **While** $\mu_k := \mu(x^k, s^k) > \epsilon$ **do**
   (a) Let $(x,s,y) := (x^k, s^k, y^k)$ and $w := (x,s,y)$; choose $\sigma \in [\underline{\sigma}, \overline{\sigma}]$
   (b) Let $\Delta w := (\Delta x, \Delta s, \Delta y)$ denote the solution of the linear system
   $$\begin{aligned} x \circ \Delta s + s \circ \Delta x &= -x \circ s + \sigma \mu e, &(8)\\ A \Delta x &= -r_p, &(9)\\ A^T \Delta y + \Delta s &= -r_d. &(10) \end{aligned}$$
   (c) Let
   $$\tilde{\alpha} = \operatorname{argmax} \left\{ \alpha \in [0,1] : w + \alpha' \Delta w \in \mathcal{N}(\gamma), \ \forall \alpha' \in [0, \alpha] \right\}.$$
   (d) Let $\bar{\alpha} = \operatorname{argmin}\{(x + \alpha \Delta x)^T (s + \alpha \Delta s) : \alpha \in [0, \tilde{\alpha}]\}$.
   (e) Let $(x^{k+1}, s^{k+1}, y^{k+1}) = w + \bar{\alpha}\, \Delta w$, and set $k \leftarrow k+1$.

   end (while)

The main complexity result for Algorithm IIP (see for example [4] and [16]) is as follows:

Theorem 2.1 *Assume that the constants γ, $\underline{\sigma}$ and $\overline{\sigma}$ are such that*

$$\max\left\{\gamma^{-1}, (1-\gamma)^{-1}, \underline{\sigma}^{-1}, (1-\overline{\sigma})^{-1}\right\} = \mathcal{O}(1),$$

and that the initial point $(x^0, s^0, y^0) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ satisfies $(x^0, s^0) \geq (\bar{x}, \bar{s})$ for some $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$. Then, Algorithm IIP finds an iterate $(x^k, s^k, y^k) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ satisfying $\mu_k \leq \epsilon\mu_0$ and $\|r^k\| \leq \epsilon\|r^0\|$ within $\mathcal{O}\left(n^2 \log(1/\epsilon)\right)$ iterations.

One way of computing the solution $(\Delta x, \Delta s, \Delta y)$ of (8)-(10) is to first solve for Δy using the following equation, known as the *normal equation*:

$$AD^2A^T\Delta y \;=\; -r_p - \sigma\mu AS^{-1}e + Ax - AD^2r_d, \tag{11}$$

where $D^2 = S^{-1}X$, and then compute Δs and Δx using the following formulae:

$$\Delta s \;=\; -r_d - A^T\Delta y, \tag{12}$$

$$\Delta x \;=\; -x + \sigma\mu S^{-1}e - D^2\Delta s. \tag{13}$$

Theorem 2.1 assumes that we can solve (8)-(10), and hence (11), exactly. Normally, the exact solution of (11) is obtained via a Cholesky factorization of AD^2A^T. Instead of this, we would like to use an iterative solver to obtain an approximate solution of (11). However, as mentioned in the introduction, the condition number of AD^2A^T may "blow up" as we approach an optimal solution, making the use of iterative methods for solving (11) undesirable. One cure is to use a preconditioner T such that the condition number $\kappa(TAD^2A^TT^T)$ remains bounded and hopefully small. One such preconditioner will be described in the next subsection, and will play an important role in our main algorithm described in Section 2.4.

2.2 Preconditioner

In this subsection, we will describe the preconditioner T that we will use to solve (11), and we will state the main results for this preconditioner, as given in [9].

Our proposed approach consists of solving the preconditioned system of linear equations:

$$Wz = q, \tag{14}$$

where

$$W \;:=\; TAD^2A^TT^T, \tag{15}$$

$$q \;:=\; -Tr_p - \sigma\mu TAS^{-1}e + TAx - TAD^2r_d, \tag{16}$$

and T is the preconditioner matrix (which we refer to as the *maximum weight basis* preconditioner) determined by the following algorithm:

Maximum Weight Basis Algorithm

Start: Given $A \in \mathbb{R}^{m\times n}$ and $d \in \mathbb{R}^n_{++}$,

1. Order the elements of d so that $d_1 \geq \ldots \geq d_n$; order the columns of A accordingly.

2. Let $\mathcal{B} = \emptyset$, $l = 1$.

3. **While** $|\mathcal{B}| < m$ **do**

   (a) If A_l is linearly independent of $\{A_i : i \in \mathcal{B}\}$, set $\mathcal{B} \leftarrow \mathcal{B} \cup \{l\}$.

   (b) $l \leftarrow l + 1$.

4. Return to the original ordering of A and d; determine the set $\mathcal{B}$ according to this ordering and set $\mathcal{N} := \{1, \ldots, n\} \backslash \mathcal{B}$.

5. Set $B := A_{\mathcal{B}}$, $N := A_{\mathcal{N}}$, $D_{\mathcal{B}} := \mathrm{Diag}(d_{\mathcal{B}})$ and $D_{\mathcal{N}} := \mathrm{Diag}(d_{\mathcal{N}})$.

6. Let $T := D_{\mathcal{B}}^{-1} B^{-1}$.

end

This preconditioner was originally proposed by Resende and Veiga in [12] in the context of network flow problems. In this case, A is a node-arc incidence matrix of a connected directed graph (with one row deleted to ensure that A has full row rank), and the elements of d are weights on the edges of the graph. Using this algorithm, we see that the set $\mathcal{B}$ created by the algorithm above defines a maximum spanning tree on the digraph. Oliveira and Sørensen [10] later proposed the use of this preconditioner for general matrices A.

For the purpose of stating the next result, we now introduce some notation. Let us define

$$\varphi_A := \max\{\|B^{-1}A\|_F : B \text{ is a basis of } A\}. \tag{17}$$

It is easy to show that $\varphi_A \leq \sqrt{m}\,\bar{\chi}_A$, where $\bar{\chi}_A$ is a well-known condition number (see [14]) defined as

$$\bar{\chi}_A := \sup\{\|A^T(ADA^T)^{-1}AD\| : D \in \mathrm{Diag}(\mathbb{R}^n_{++})\}.$$

Indeed, this follows from the fact that $\|C\|_F \leq \sqrt{m}\|C\|$ for any matrix $C \in \mathbb{R}^{m \times n}$ with $m \leq n$ and that an equivalent characterization of $\bar{\chi}_A$ is

$$\bar{\chi}_A := \max\{\|B^{-1}A\| : B \text{ is a basis of } A\}, \tag{18}$$

as shown in [13] and [14].

Recently, Monteiro, O'Neal and Tsuchiya showed the following result in [9].

Proposition 2.2 *Let a full row rank matrix $A \in \mathbb{R}^{m \times n}$ and $d \in \mathbb{R}^n_{++}$ be given. Let $T = T(A, d)$ be the preconditioner determined according to the Maximum Weight Basis Algorithm, and define $W := TAD^2A^TT^T$. Then, $\|TAD\| \leq \varphi_A$ and $\kappa(W) \leq \varphi_A^2$.*

Note that the bound φ_A^2 on $\kappa(W)$ is independent of the diagonal matrix D and depends only on A. In the next subsection, we derive bounds on the number of iterations needed by an iterative solver to solve (14) to a desired accuracy level.

2.3 Iteration Complexity for the Iterative Solver

In this subsection, we will develop a bound on the number of iterations that an iterative linear solver needs to perform to obtain a suitable approximate solution to (14). Instead of focusing on one specific solver, we will assume that we have a generic iterative linear solver with a prescribed rate of convergence. More specifically, we will assume that the generic iterative linear solver when applied to (14) generates a sequence of iterates $\{z^j\}$ satisfying the following condition:

$$\|q - Wz^j\| \leq c(\kappa) \left[1 - \frac{1}{f(\kappa)}\right]^j \|q - Wz^0\|, \quad \forall\, j = 0, 1, 2, \ldots, \tag{19}$$

where c and f are positive functions of $\kappa \equiv \kappa(W)$. For our purposes, we will also assume that the initial iterate $z^0 = 0$, so that $q - Wz^0 = q$.

Examples of solvers which satisfy (19) include the steepest descent (SD) and conjugate gradient (CG) methods, with the following values for $c(\kappa)$ and $f(\kappa)$:

Solver	$c(\kappa)$	$f(\kappa)$
SD	$\sqrt{\kappa}$	$(\kappa + 1)/2$
CG	$2\sqrt{\kappa}$	$(\sqrt{\kappa} + 1)/2$

Table 2.3

The justification for the table above follows from Section 7.6 and Exercise 10 of Section 8.8 of [7].

Before we give the main convergence result, we state the following lemma, which we will prove in Section 3.1:

Lemma 2.3 *Assume that $T = T(A, d)$ and the initial point (x^0, s^0, y^0) is such that $s^0 \geq |c - A^T y^0|$ and $(x^0, s^0) \geq (\bar{x}, \bar{s})$ for some $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$. Suppose also that $(x, s, y) \in \mathcal{N}(\gamma)$ and that $r = \eta r^0$ for some $\eta \in [0, 1]$. Then, the vector q defined in (16) satisfies $\|q\| \leq \Psi\sqrt{\mu}$, where*

$$\Psi := \frac{9n\varphi_A}{\sqrt{1-\gamma}} + \sqrt{n}\varphi_A + \sigma\varphi_A\sqrt{\frac{n}{1-\gamma}}. \tag{20}$$

■

The following result gives an upper bound on the number of iterations that the generic iterative linear solver needs to perform to obtain an iterate z^j satisfying $\|q - Wz^j\| \leq \rho\sqrt{\mu}$, for some constant $\rho > 0$.

Theorem 2.4 *Suppose that the conditions of Lemma 2.3 are met and $(1 - \gamma)^{-1} = \mathcal{O}(1)$. Then, a generic iterative solver with a convergence rate given by (19) generates an iterate z^j satisfying $\|q - Wz^j\| \leq \rho\sqrt{\mu}$ in*

$$\mathcal{O}\left(f(\kappa)\log\left(\frac{c(\kappa)n\varphi_A}{\rho}\right)\right) \tag{21}$$

iterations, where $\kappa := \kappa(W)$.

Proof: Let j be any index satisfying

$$j \geq f(\kappa) \log \left(\frac{c(\kappa)\Psi}{\rho} \right). \tag{22}$$

Using the fact that $\log(1+x) \leq x$ for all $x > -1$ and the above inequality, we conclude that

$$\begin{aligned} \log(\rho\sqrt{\mu}) &\geq \log(c(\kappa)\Psi\sqrt{\mu}) - \frac{j}{f(\kappa)} \\ &\geq \log(c(\kappa)\Psi\sqrt{\mu}) + j\log\left(1 - \frac{1}{f(\kappa)}\right). \end{aligned}$$

This together with the assumption that $z^0 = 0$, relation (19) and Lemma 2.3 imply that

$$\rho\sqrt{\mu} \ \geq \ c(\kappa)\left[1 - \frac{1}{f(\kappa)}\right]^j \Psi\sqrt{\mu} \ \geq \ c(\kappa)\left[1 - \frac{1}{f(\kappa)}\right]^j \|q\| \ \geq \ \|q - Wz^j\|.$$

Since $\Psi = \mathcal{O}(n\varphi_A)$ in view of Lemma 2.3, it follows that the right hand side of (22) is majorized by (21), from which the result follows. ∎

We will refer to an inner iterate z^j satisfying $\|q - Wz^j\| \leq \rho\sqrt{\mu}$ to as ρ-approximate solution of (14). In our interior-point algorithm in the next subsection, we will choose the constant ρ as $\rho = \gamma\sigma/(4\sqrt{n})$. As a consequence of Proposition 2.2, we obtain the following corollary.

Corollary 2.5 *Suppose that the conditions of Lemma 2.3 are met and that* $\max\{\underline{\sigma}^{-1}, \gamma^{-1}, (1-\gamma)^{-1}\} = \mathcal{O}(1)$. *If* $\rho = \gamma\sigma/(4\sqrt{n})$, *then the SD and CG methods generate a* ρ*-approximate solution in* $\mathcal{O}(\varphi_A^2 \log(n\varphi_A))$ *and* $\mathcal{O}(\varphi_A \log(n\varphi_A))$ *iterations, respectively.*

Proof: This result follows immediately from the assumptions, Theorem 2.4, Table 2.3 and Proposition 2.2. ∎

Note that the inner-iteration complexity bounds derived in Corollary 2.5 are not polynomial in general, since they depend on φ_A. However, these bounds will be polynomial if φ_A is polynomial. We will discuss this impact for general matrices A at the end of Section 2.4. In Section 2.5, we will consider a specific case when φ_A is polynomial, namely when A is the node-arc incidence matrix of a directed graph.

2.4 The Iterative IP Algorithm

In this subsection, we describe our main algorithm. It is essentially the IIP algorithm, except that the search direction $(\Delta x, \Delta s, \Delta y)$ is computed approximately with the use of iterative methods applied to (14).

Notice that under exact computations, the primal and dual residuals $r^k = (r_p^k, r_d^k)$ corresponding to the k-th iterate of Algorithm IIP always lie on the line segment between 0 and r^0 because of (9) and (10). It is well known that this property plays an important role in the convergence analysis of infeasible interior point methods. We will now show that it is still possible to ensure that r^k lies on the segment between 0 and r^0, even when Δy is an approximate solution of (11). Indeed, consider an approximate solution Δy which satisfies the following equation:

$$AD^2A^T\Delta y = -r_p - \sigma\mu AS^{-1}e + Ax - AD^2r_d + f \tag{23}$$

where f is some error vector. Next, we solve for Δs using (12), so that $(\Delta s, \Delta y)$ satisfies (10).

The usual approach for choosing Δx is to use (13). However, this does not ensure that (9) is satisfied. To ensure that (9) is satisfied, we use the following equation:

$$\Delta x = -x + \sigma\mu S^{-1}e - D^2\Delta s - S^{-1}v \tag{24}$$

where v is a perturbation vector satisfying

$$AS^{-1}v = f. \tag{25}$$

Condition (25) is necessary and sufficient for Δx given by (24) to satisfy (9), since

$$\begin{aligned} A\Delta x &= A\left(-x + \sigma\mu S^{-1}e - D^2\Delta s - S^{-1}v\right) \\ &= -Ax + \sigma\mu AS^{-1}e - AD^2\Delta s - AS^{-1}v \\ &= -Ax + \sigma\mu AS^{-1}e + AD^2r_d + AD^2A^T\Delta y - AS^{-1}v \\ &= -r_p + f - AS^{-1}v, \end{aligned} \tag{26}$$

due to (24), (12) and (23). Note from (24) that (8) is satisfied exactly if and only if $v = 0$, which in turn satisfies the necessary and sufficient condition $AS^{-1}v = f$ if and only if $f = 0$, i.e. when Δy is an exact solution of (11).

There are numerous choices for v. An obvious choice for v is to choose the least squares solution, i.e., to choose the optimal solution to $\min\{\|v\| : AS^{-1}v = f\}$. However, this is the type of computation we wish to avoid when using an iterative solver. A more effective choice for v is to choose a basis $\tilde{B}$ of A and let

$$v = (v_{\tilde{\mathcal{B}}}, v_{\tilde{\mathcal{N}}}) = (S_{\tilde{\mathcal{B}}}\tilde{B}^{-1}f\,,\,0), \tag{27}$$

where $(\tilde{\mathcal{B}}, \tilde{\mathcal{N}})$ is the index partition corresponding to the basis $\tilde{B}$.

It turns out that an obvious choice for $\tilde{B}$ in our approach is to let $\tilde{B}$ be equal to the maximum weight basis B corresponding to (A, d). More specifically, recall that in our algorithm, we compute Δy approximately using the preconditioned system $Wz = q$. Let $\widetilde{\Delta y}$ denote the final iterate z^j in the approximate solution of $Wz = q$, and let $\tilde{f} = W\widetilde{\Delta y} - q$. Letting

$\Delta y = T^T \widetilde{\Delta y}$, it is easy to see that Δy is an approximate solution satisfying (23) with error $f = T^{-1}\tilde{f} = BD_{\mathcal{B}}\tilde{f}$. Using this expression for f and letting $\tilde{B} = B$ in (27), we obtain

$$v = (S_{\tilde{\mathcal{B}}}\tilde{B}^{-1}f\,,\,0) = (S_{\mathcal{B}}B^{-1}BD_{\mathcal{B}}\tilde{f}\,,\,0) = ((X_{\mathcal{B}}S_{\mathcal{B}})^{1/2}\tilde{f}\,,\,0). \tag{28}$$

Note that by (24), we have $X\Delta s + S\Delta x = -XSe + \sigma\mu e - v$, i.e. (8) is satisfied only approximately. To ensure convergence of our method, it turns out that it is important to keep $\|\tilde{f}\|$, and hence $\|v\|$, small. In our algorithm below, we require that

$$\|\tilde{f}\| \leq \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu} \tag{29}$$

so as to ensure that the number of outer iterations of our method is still polynomially bounded.

We now present our main algorithm:

Algorithm IIP-IS:

1. **Start:** Let $\epsilon > 0$, $\gamma \in (0,1)$, $(x^0, s^0, y^0) \in \mathcal{N}(\gamma)$ and $0 < \underline{\sigma} < \overline{\sigma} < 4/5$ be given. Set $k = 0$.

2. **While** $\mu_k := \mu(x^k, s^k) > \epsilon$ **do**

   (a) Let $(x, s, y) := (x^k, s^k, y^k)$, and choose $\sigma \in [\underline{\sigma}, \overline{\sigma}]$.

   (b) Set $d = S^{-1/2}X^{1/2}e$, $r_p = Ax - b$, $r_d = A^Ty + s - c$, and $r = (r_p, r_d)$.

   (c) Build the preconditioner $T = T(A, d)$ using the Maximum Weight Basis Algorithm.

   (d) Find an approximate solution $\widetilde{\Delta y}$ of (14) such that $\tilde{f} = W\widetilde{\Delta y} - q$ satisfies (29).

   (e) Let v be computed according to (28). Set $\Delta y = T^T\widetilde{\Delta y}$, and compute Δs and Δx by (12) and (24), respectively.

   (f) Compute $\tilde{\alpha} := \operatorname{argmax}\{\alpha \in [0,1] : w + \alpha'\Delta w \in \mathcal{N}(\gamma),\ \forall \alpha' \in [0, \alpha]\}$, where $w := (x, s, y)$ and $\Delta w := (\Delta x, \Delta s, \Delta y)$.

   (g) Compute $\bar{\alpha} := \operatorname{argmin}\{(x + \alpha\Delta x)^T(s + \alpha\Delta s) : \alpha \in [0, \tilde{\alpha}]\}$.

   (h) Let $(x^{k+1}, s^{k+1}, y^{k+1}) = w + \bar{\alpha}\Delta w$, and set $k \leftarrow k + 1$.

   end (while)

Using this algorithm, we obtain nearly the exact same polynomial convergence result as Theorem 2.1. The results for Algorithm IIP-IS are summarized in the following theorem, which we will prove in Section 3.2.

Theorem 2.6 *Assume that the constants γ, $\underline{\sigma}$ and $\overline{\sigma}$ are such that*

$$\max\left\{\gamma^{-1}\,,\,(1-\gamma)^{-1}\,,\,\underline{\sigma}^{-1}\,,\,\left(1-\frac{5}{4}\overline{\sigma}\right)^{-1}\right\} = \mathcal{O}(1), \tag{30}$$

and that the initial point $(x^0, s^0, y^0) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ satisfies $(x^0, s^0) \geq (\bar{x}, \bar{s})$ for some $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$. Then, Algorithm IIP-IS generates an iterate $(x^k, s^k, y^k) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ satisfying $\mu_k \leq \epsilon\mu_0$ and $\|r^k\| \leq \epsilon\|r^0\|$ within $\mathcal{O}\left(n^2 \log(1/\epsilon)\right)$ iterations.

Note that while the number of "outer" iterations is polynomial for our algorithm, the overall complexity is not, due to the total number of "inner" iterations. To see this, consider a given iteration of our algorithm with the CG solver. It is easy to see that steps (a), (b), and (e) through (h) can be carried out in $\mathcal{O}(mn)$ flops. Let us now examine the other two steps (c) and (d). The sorting part of the Maximum Weight Basis algorithm can be done in $\mathcal{O}(n \log n)$ with a quick sorting algorithm. Computing T and its corresponding LU factorization can be done in $\mathcal{O}(m^2 n)$ flops. Hence, the entire step (c) takes $\mathcal{O}(n \max\{m^2, \log n\})$ flops. Now, notice that each step of the CG solver requires $\mathcal{O}(mn)$ flops. Since $\mathcal{O}(\varphi_A \log(n\varphi_A))$ iterations of the CG solver are required, we conclude that step (d) takes $\mathcal{O}(mn\varphi_A \log(n\varphi_A))$ flops. Thus, the number of flops per iteration of Algorithm IIP-IS is $\mathcal{O}(n \max\{m\varphi_A \log(n\varphi_A), m^2\})$.

In the next subsection, we will consider a specific case where φ_A is polynomial, namely when A is the node-arc incidence matrix of a directed graph.

2.5 Application to Network Flows

Consider a standard network flow problem of the form (1). In this case, A is the node-arc incidence matrix of a simple connected directed graph $\mathcal{G}$, with one row deleted to ensure that A has full row rank. We will show that for this particular problem, Algorithm IIP-IS is indeed a polynomial-time algorithm. The key result comes from the following observation: since A is a node-arc incidence matrix, it is totally unimodular so that every element of $B^{-1}A$ is 1, 0, or -1. Thus, $\varphi_A \leq \sqrt{mn}$, as can be seen from definition (17) of φ_A.

Additional savings in the complexity of our algorithm can be obtained by using the special structure of the matrix A. Indeed, consider a single iteration of Algorithm IIP-IS using the CG solver. Steps (a), (b), and (e) through (h) now take $\mathcal{O}(n)$ flops since A has only $2n$ nonzero entries. Since a maximum weight basis is now a maximum spanning tree on $\mathcal{G}$, it can be found with $\mathcal{O}(n \log n)$ flops using either Prim's or Kruskal's algorithm (see e.g. [2]). The basis matrix B, under a suitable ordering, will be upper triangular, so we do not need to form B^{-1} explicitly; thus step (c) requires $\mathcal{O}(n \log n)$ flops. Next, Corollary 2.5 and the fact that $\varphi_A \leq \sqrt{mn}$ imply that the CG method takes $\mathcal{O}(\sqrt{mn} \log n)$ iterations to find a suitably accurate solution of (14). As each step of the CG method takes $\mathcal{O}(n)$ flops, step (d) requires $\mathcal{O}(m^{1/2} n^{3/2} \log n)$ flops. Thus, a single outer iteration of Algorithm IIP-IS applied to a minimum-cost network flow problem requires $\mathcal{O}(m^{1/2} n^{3/2} \log n)$ flops.

3 Technical Results

In this section, we give detailed proofs for the results given in Sections 2.3 and 2.4. Section 3.1 will be devoted to the proofs of the results in Section 2.3, while Section 3.2 will give the proofs for the results in Section 2.4.

3.1 Results for the Iterative Solver

Consider a generic iterative solver which solves (14), and suppose that this solver satisfies (19) at each iteration. We will seek to get $\|q - Wz^j\| \leq \rho\sqrt{\mu}$ for some term $\rho > 0$. We begin with some technical lemmas:

Lemma 3.1 *Let* (x^0, s^0, y^0) *and* (x, s, y) *be points such that* $r(x, s, y) = \eta r(x^0, s^0, y^0)$ *for some* $\eta \in \mathbb{R}$*, and let* $(\bar{x}, \bar{s}, \bar{y})$ *be a point such that* $r(\bar{x}, \bar{s}, \bar{y}) = 0$*. Then,*

$$\begin{aligned} 0 = \eta^2 {x^0}^T s^0 + (1-\eta)^2 \bar{x}^T \bar{s} + x^T s + \eta(1-\eta)({x^0}^T \bar{s} + \bar{x}^T s^0) \\ -\eta({x^0}^T s + x^T s^0) - (1-\eta)(\bar{x}^T s + x^T \bar{s}). \end{aligned} \tag{31}$$

Proof: Using the definition of r, it is easy to see that

$$\begin{aligned} A(x - \eta x^0 - (1-\eta)\bar{x}) &= 0 \\ (s - \eta s^0 - (1-\eta)\bar{s}) + A^T(y - \eta y^0 - (1-\eta)\bar{y}) &= 0 \end{aligned}$$

Multiplying the second relation by $[x - \eta x^0 - (1-\eta)\bar{x}]^T$ on the left and using the first relation, we get

$$[x - \eta x^0 - (1-\eta)\bar{x}]^T [s - \eta s^0 - (1-\eta)\bar{s}] = 0. \tag{32}$$

Expanding this equality, we obtain (31). ∎

Lemma 3.2 *Let* $(x^0, s^0, y^0) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ *be a point such that* $(x^0, s^0) \geq (\bar{x}, \bar{s})$ *for some* $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$*. Then, for any point* $(x, s, y) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ *such that* $r = \eta r^0$ *for some* $\eta \in [0, 1]$ *and* $\eta \leq x^T s / {x^0}^T s^0$*, we have that* $\eta({x^0}^T s + {s^0}^T x) \leq 3n\mu$*.*

Proof: By assumption, there exists $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$ such that $\bar{x} \leq x^0$ and $\bar{s} \leq s^0$. Since $r = \eta r^0$ and $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$, the points (x, s, y), (x^0, s^0, y^0), and $(\bar{x}, \bar{s}, \bar{y})$ satisfy the assumption of the previous lemma. Hence, by equation (31), along with the facts that $\eta \leq x^T s / {x^0}^T s^0$, $\bar{x}^T \bar{s} = 0$, $(x, s) \geq 0$, $(\bar{x}, \bar{s}) \geq 0$, $(x^0, s^0) \geq 0$, $\eta \in [0, 1]$, $\bar{x} \leq x^0$, and $\bar{s} \leq s^0$, we conclude that

$$\begin{aligned} \eta({x^0}^T s + {s^0}^T x) &\leq \eta^2 {x^0}^T s^0 + x^T s + \eta(1-\eta)({x^0}^T \bar{s} + {s^0}^T \bar{x}) \\ &\leq \eta^2 {x^0}^T s^0 + x^T s + 2\eta(1-\eta){x^0}^T s^0 \\ &\leq 2\eta {x^0}^T s^0 + x^T s \leq 3x^T s. \end{aligned}$$

∎

Next, we turn to the proof of Lemma 2.3:

Proof of Lemma 2.3: By (16) and the triangle inequality for norms, we have

$$\|q\| \leq \|Tr_p\| + \|\sigma\mu TAS^{-1}e\| + \|TAx\| + \|TAD^2 r_d\|. \tag{33}$$

We will now bound each of the terms in the right hand side of (33). Let $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$ and (x^0, s^0, y^0) satisfy the assumptions of Lemma 2.3, so that $x^0 \geq \bar{x}$, $s^0 \geq \bar{s}$ and $s^0 \geq |c - A^T y^0|$. Using these inequalities, the assumption that $(x, s, y) \in \mathcal{N}(\gamma)$ and Lemma 3.2, we obtain

$$\eta\|S(\bar{x} - x^0)\| \;\leq\; \eta\|Sx^0\| \;\leq\; \eta s^T x^0 \;\leq\; 3n\mu \tag{34}$$

$$\eta\|X(s^0 + A^T y^0 - c)\| \;\leq\; 2\eta\|Xs^0\| \;\leq\; 2\eta x^T s^0 \;\leq\; 6n\mu. \tag{35}$$

Thus, using the relations $r = \eta r^0$, $b = A\bar{x}$, (34) and (35), the fact that $(x, s, y) \in \mathcal{N}(\gamma)$ and Proposition 2.2, we obtain

$$\begin{aligned} \|Tr_p\| &= \eta\,\|Tr_p^0\| = \eta\,\|T(b - Ax^0)\| = \eta\,\|TA(\bar{x} - x^0)\| = \eta\,\|(TAD)(XS)^{-1/2}S(\bar{x} - x^0)\| \\ &\leq \eta\,\|TAD\|\,\|(XS)^{-1/2}\|\,\|S(\bar{x} - x^0)\| \leq \varphi_A \frac{1}{\sqrt{(1-\gamma)\mu}}\,3n\mu = \frac{3n\varphi_A}{\sqrt{1-\gamma}}\sqrt{\mu} \end{aligned}$$

and

$$\begin{aligned} \|TAD^2 r_d\| &\leq \|TAD\|\,\|Dr_d\| = \eta\|TAD\|\,\|Dr_d^0\| = \eta\|TAD\|\,\|D(s^0 + A^T y^0 - c)\| \\ &\leq \eta\|TAD\|\,\|(XS)^{-1/2}\|\,\|X(s^0 + A^T y^0 - c)\| \\ &\leq \varphi_A \frac{1}{\sqrt{(1-\gamma)\mu}}\,6n\mu = \frac{6n\varphi_A}{\sqrt{1-\gamma}}\sqrt{\mu}. \end{aligned}$$

Similarly, we have

$$\|\sigma\mu TAS^{-1}e\| \;\leq\; \sigma\mu\|TAD\|\,\|(XS)^{-1/2}\|\,\|e\| \leq \sigma\mu\varphi_A \frac{1}{\sqrt{(1-\gamma)\mu}}\sqrt{n} = \sigma\varphi_A\sqrt{\frac{n}{1-\gamma}}\sqrt{\mu}$$

and

$$\|TAx\| = \|TAD(XS)^{1/2}e\| \leq \|TAD\|\,\|(XS)^{1/2}e\| \leq \varphi_A\sqrt{n\mu},$$

where in the last inequality we used the fact that $\|(XS)^{1/2}e\| = \sqrt{n\mu}$. The result now follows by combining the four bounds obtained above with (33). ∎

3.2 Convergence Results for Algorithm IIP-IS

In this subsection, we will provide the proof of Theorem 2.6.

For the sake of future reference, we note that $(\Delta x, \Delta s, \Delta y)$ satisfies

$$\begin{aligned} A\Delta x &= -r_p && (36)\\ A^T\Delta y + \Delta s &= -r_d && (37)\\ x \circ \Delta s + s \circ \Delta x &= -x \circ s + \sigma\mu e - v && (38)\end{aligned}$$

by equations (25), (26), (12), and (24), respectively. Throughout this section, we use the following notation:

$$\begin{aligned} (x(\alpha), s(\alpha), y(\alpha)) &:= (x, s, y) + \alpha(\Delta x, \Delta s, \Delta y),\\ \mu(\alpha) &:= x(\alpha)^T s(\alpha)/n,\\ r(\alpha) &:= r(x(\alpha), s(\alpha), y(\alpha)) = (Ax(\alpha) - b, A^T y(\alpha) + s(\alpha) - c).\end{aligned}$$

Lemma 3.3 *Assume that $(\Delta x, \Delta s, \Delta y)$ satisfies (36)-(38) for some $\sigma \in \mathbb{R}$, $v \in \mathbb{R}^n$ and $(x, s, y) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$. Then, for every $\alpha \in \mathbb{R}$, we have:*

(a) $x(\alpha) \circ s(\alpha) = (1-\alpha)x \circ s + \alpha\sigma\mu e - \alpha v + \alpha^2 \Delta x \circ \Delta s$;

(b) $\mu(\alpha) = [1 - \alpha(1-\sigma)]\mu - \alpha v^T e/n + \alpha^2 \Delta x^T \Delta s/n$;

(c) $r(\alpha) = (1-\alpha)r$.

Proof: Using (38), we obtain

$$\begin{aligned} x(\alpha) \circ s(\alpha) &= (x + \alpha\Delta x) \circ (s + \alpha\Delta s)\\ &= x \circ s + \alpha(x \circ \Delta s + s \circ \Delta x) + \alpha^2 \Delta x \circ \Delta s\\ &= x \circ s + \alpha(-x \circ s + \sigma\mu e - v) + \alpha^2 \Delta x \circ \Delta s\\ &= (1-\alpha)x \circ s + \alpha\sigma\mu e - \alpha v + \alpha^2 \Delta x \circ \Delta s,\end{aligned}$$

thereby showing that a) holds. Left multiplying the above equality by e^T and dividing the resulting expression by n, we easily conclude that b) holds. Statement c) can be easily verified by means of (36) and (37). ∎

Lemma 3.4 *Assume that $(\Delta x, \Delta s, \Delta y)$ satisfies (36)-(38) for some $\sigma > 0$, $(x, s, y) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ and $v \in \mathbb{R}^n$ satisfying $v^T e/n \le \sigma\mu/2$. Then, for every scalar α satisfying*

$$0 \le \alpha \le \min\left\{1\,,\ \frac{\sigma\mu}{2\,\|\Delta x \circ \Delta s\|_\infty}\right\}, \tag{39}$$

we have

$$\frac{\|r(\alpha)\|}{\|r\|} \le \frac{\mu(\alpha)}{\mu}. \tag{40}$$

Proof: Using Lemma 3.3(b) and the assumption that $v^T e/n \leq \sigma\mu/2$, we conclude for every α satisfying (39) that

$$\begin{aligned} \mu(\alpha) &= [1-\alpha(1-\sigma)]\mu - \alpha v^T e/n + \alpha^2 \Delta x^T \Delta s/n \\ &\geq [1-\alpha(1-\sigma)]\mu - \frac{1}{2}\,\alpha\sigma\mu + \alpha^2 \Delta x^T \Delta s/n \\ &\geq (1-\alpha)\mu + \frac{1}{2}\,\alpha\sigma\mu - \alpha^2 \|\Delta x \circ \Delta s\|_\infty \\ &\geq (1-\alpha)\mu. \end{aligned}$$

The result now follows from the last relation and Lemma 3.3(c). ∎

Lemma 3.5 *Assume that $(\Delta x, \Delta s, \Delta y)$ satisfies (36)-(38) for some $\sigma > 0$, $(x, s, y) \in \mathcal{N}(\gamma)$ with $\gamma \in [0, 1]$, and $v \in \mathbb{R}^n$ satisfying $\|v\|_\infty \leq \gamma\sigma\mu/4$. Then, $(x(\alpha), s(\alpha), y(\alpha)) \in \mathcal{N}(\gamma)$ for every scalar α satisfying*

$$0 \leq \alpha \leq \min\left\{1\,,\ \frac{\gamma\sigma\mu}{4\,\|\Delta x \circ \Delta s\|_\infty}\right\}. \tag{41}$$

Proof: Since the assumption that $\gamma \in [0, 1]$ and $\|v\|_\infty \leq \gamma\sigma\mu/4$ imply that $v^T e/n \leq \sigma\mu/2$, it follows from Lemma 3.4 that (40) holds for every α satisfying (39), and hence (41). Thus, for every α satisfying (41), we have

$$\frac{\|r(\alpha)\|}{\|r^0\|} = \frac{\|r(\alpha)\|}{\|r\|}\,\frac{\|r\|}{\|r^0\|} \leq \frac{\mu(\alpha)}{\mu}\,\frac{\mu}{\mu_0} = \frac{\mu(\alpha)}{\mu_0}. \tag{42}$$

Now, it is easy to see that for every $u \in \mathbb{R}^n$ and $\tau \in [0, n]$, there holds $\|u - \tau(u^T e/n)e\|_\infty \leq (1+\tau)\|u\|_\infty$. Using this inequality twice, the fact that $(x, s, y) \in \mathcal{N}(\gamma)$ and statements (a) and (b) of Lemma 3.3, we conclude for every α satisfying (41) that

$$\begin{aligned} & x(\alpha) \circ s(\alpha) - (1-\gamma)\,\mu(\alpha) e \\ &= (1-\alpha)\,[x \circ s - (1-\gamma)\mu e] + \alpha\gamma\sigma\mu e - \alpha\left[v - (1-\gamma)\left(\frac{v^T e}{n}\right) e\right] \\ &\quad + \alpha^2\left[\Delta x \circ \Delta s - (1-\gamma)\left(\frac{\Delta x^T \Delta s}{n}\right) e\right] \\ &\geq \alpha\left[\gamma\sigma\mu - \left\|v - (1-\gamma)\frac{v^T e}{n}\,e\right\|_\infty - \alpha\left\|\Delta x \Delta s - (1-\gamma)\frac{\Delta x^T \Delta s}{n}\,e\right\|_\infty\right] e \\ &\geq \alpha\,(\gamma\sigma\mu - 2\|v\|_\infty - 2\alpha\|\Delta x \circ \Delta s\|_\infty)\, e \ \geq\ \alpha\left(\gamma\sigma\mu - \frac{1}{2}\gamma\sigma\mu - \frac{1}{2}\gamma\sigma\mu\right) e \ \geq\ 0. \end{aligned}$$

We have thus shown that $(x(\alpha), s(\alpha), y(\alpha)) \in \mathcal{N}(\gamma)$ for every α satisfying (41). ∎

Next, we consider the minimum step length allowed under our algorithm:

Lemma 3.6 *In every iteration of Algorithm IIP-IS, the step length* $\bar{\alpha}$ *satisfies*

$$\bar{\alpha} \geq \min\left\{1, \frac{\min\{\gamma\sigma, 1-\frac{5}{4}\sigma\}\mu}{4\left\|\Delta x \circ \Delta s\right\|_\infty}\right\} \tag{43}$$

and

$$\mu(\bar{\alpha}) \leq \left[1 - \left(1 - \frac{5}{4}\,\sigma\right)\frac{\bar{\alpha}}{2}\right]\mu. \tag{44}$$

Proof: Using (28) and (29), we conclude that

$$\|v\|_\infty = \|(X_{\mathcal{B}}S_{\mathcal{B}})^{1/2}\tilde{f}\|_\infty \leq \|X_{\mathcal{B}}S_{\mathcal{B}}\|^{1/2}\|\tilde{f}\|_\infty \leq \sqrt{n\mu}\,\frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu} = \frac{1}{4}\gamma\sigma\mu. \tag{45}$$

Hence, by Lemma 3.5, the quantity $\tilde{\alpha}$ computed in step (g) of Algorithm IIP-IS satisfies

$$\tilde{\alpha} \geq \min\left\{1, \frac{\gamma\sigma\mu}{4\left\|\Delta x \circ \Delta s\right\|_\infty}\right\}. \tag{46}$$

Moreover, by (45), it follows that the coefficient of α in the expression for $\mu(\alpha)$ in Lemma 3.3(b) satisfies

$$-(1-\sigma)\mu - \frac{v^Te}{n} \leq -(1-\sigma)\mu + \|v\|_\infty \leq -(1-\sigma)\mu + \frac{1}{4}\,\gamma\sigma\mu = -\left(1-\frac{5}{4}\,\sigma\right)\mu < 0, \tag{47}$$

since $\sigma \in (0, \frac{4}{5})$. Hence, if $\Delta x^T\Delta s \leq 0$, it is easy to see that $\bar{\alpha} = \tilde{\alpha}$, and hence that (43) holds in view of (46). Moreover, by Lemma 3.3(b) and (47), we have

$$\mu(\bar{\alpha}) \leq [1 - \bar{\alpha}(1-\sigma)]\mu - \bar{\alpha}\frac{v^Te}{n} \leq \left[1 - \left(1-\frac{5}{4}\,\sigma\right)\bar{\alpha}\right]\mu \leq \left[1 - \left(1-\frac{5}{4}\,\sigma\right)\frac{\bar{\alpha}}{2}\right]\mu,$$

showing that (44) also holds. We now consider the case where $\Delta x^T\Delta s > 0$. In this case, we have $\bar{\alpha} = \min\{\alpha_{\min}\,, \tilde{\alpha}\}$, where $\alpha_{\min}$ is the unconstrained minimum of $\mu(\alpha)$. It is easy to see that

$$\alpha_{\min} \;=\; \frac{n\mu(1-\sigma) + v^Te}{2\Delta x^T\Delta s} \;\geq\; \frac{n\mu(1-\sigma) - \frac{1}{4}\sigma n\mu}{2\Delta x^T\Delta s} \;\geq\; \frac{\mu(1-\frac{5}{4}\sigma)}{2\left\|\Delta x \circ \Delta s\right\|_\infty}.$$

The last two observations together with (46) imply that (43) holds in this case too. Moreover, since the function $\mu(\alpha)$ is convex, it must lie below the function $\phi(\alpha)$ over the interval $[0, \alpha_{\min}]$, where $\phi(\alpha)$ is the affine function interpolating $\mu(\alpha)$ at $\alpha = 0$ and $\alpha = \alpha_{\min}$. Hence,

$$\mu(\bar{\alpha}) \leq \phi(\bar{\alpha}) = [1 - (1-\sigma)\frac{\bar{\alpha}}{2}]\mu - \bar{\alpha}\frac{v^Te}{2n} \leq \left[1 - \left(1-\frac{5}{4}\,\sigma\right)\frac{\bar{\alpha}}{2}\right]\mu, \tag{48}$$

where the second inequality follows from (47). We have thus shown that $\bar{\alpha}$ satisfies (44). ∎

Our next task will be to show that the stepsize $\bar{\alpha}$ remains bounded away from zero. In view of (43), it sufficient to show that the quantity $\|\Delta x \circ \Delta s\|_\infty$ remains bounded. The next lemma addresses this issue.

Lemma 3.7 *Let $(x^0, s^0, y^0) \in \mathbb{R}^{2n}_{++} \times \mathbb{R}^m$ be such that $(x^0, s^0) \geq (\bar{x}, \bar{s})$ for some $(\bar{x}, \bar{s}, \bar{y}) \in \mathcal{S}$, and let $(x, s, y) \in \mathcal{N}(\gamma)$ be such that $r = \eta r^0$ for some $\eta \in [0, 1]$. Then, the search direction $(\Delta x, \Delta s, \Delta y)$ generated by Algorithm IIP-IS satisfies*

$$\max(\|D^{-1}\Delta x\|, \|D\Delta s\|) \leq \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)^{1/2} \sqrt{n\mu} + \frac{6n}{\sqrt{1-\gamma}}\sqrt{\mu} + \frac{\gamma\sigma}{4\sqrt{n}}\sqrt{\mu}.$$

Proof: Relations (36) and (37) and the assumption $r = \eta r^0$ imply that

$$\begin{aligned} A(\Delta x + \eta(x^0 - \bar{x})) &= 0 \\ A^T(\Delta y + \eta(y^0 - \bar{y})) + (\Delta s + \eta(s^0 - \bar{s})) &= 0, \end{aligned}$$

from which it follows that $(\Delta x + \eta(x^0 - \bar{x}))^T(\Delta s + \eta(s^0 - \bar{s})) = 0$. Multiplying (38) on the left by $(XS)^{-1/2}$, we obtain $D^{-1}\Delta x + D\Delta s = H(\sigma) - (XS)^{-1/2}v$, where $H(\sigma) := -(XS)^{1/2}e + \sigma\mu(XS)^{-1/2}e$. Equivalently, we have that

$$\begin{aligned} & D^{-1}(\Delta x + \eta(x^0 - \bar{x})) + D(\Delta s + \eta(s^0 - \bar{x})) \\ = \;& H(\sigma) + \eta(D(s^0 - \bar{s}) + D^{-1}(x^0 - \bar{x})) - (XS)^{-1/2}v. \end{aligned}$$

Using the fact that the two terms on the left hand side of the above identity are orthogonal, along with the fact that $\|(XS)^{-1/2}v\| = \|\tilde{f}\|$ by (28), we obtain

$$\begin{aligned} & \max\left(\|D^{-1}(\Delta x + \eta(x^0 - \bar{x}))\|, \|D(\Delta s + \eta(s^0 - \bar{s})\|\right) \\ & \quad \leq \|H(\sigma) + \eta(D(s^0 - \bar{s}) + D^{-1}(x^0 - \bar{x})) - (XS)^{-1/2}v\| \\ & \quad \leq \|H(\sigma)\| + \eta\left(\|D(s^0 - \bar{s})\| + \|D^{-1}(x^0 - \bar{x})\|\right) + \|\tilde{f}\|. \end{aligned}$$

This, together with the triangle inequality and the definition of D, imply that

$$\begin{aligned} \max(\|D^{-1}\Delta x\|, \|D\Delta s\|) &\leq \|H(\sigma)\| + 2\eta\left(\|D(s^0 - \bar{s})\| + \|D^{-1}(x^0 - \bar{x})\|\right) + \|\tilde{f}\| \\ &\leq \|H(\sigma)\| + 2\eta\|(XS)^{-1/2}\|\left(\|X(s^0 - \bar{s})\| + \|S(x^0 - \bar{x})\|\right) + \|\tilde{f}\| \\ &\leq \|H(\sigma)\| + \frac{2\eta}{\sqrt{(1-\gamma)\mu}}\left(\|X(s^0 - \bar{s})\| + \|S(x^0 - \bar{x})\|\right) + \|\tilde{f}\| \end{aligned} \tag{49}$$

It is well-known that

$$\|H(\sigma)\| \;\leq\; \left(1 - 2\sigma + \frac{\sigma^2}{1-\gamma}\right)^{1/2} \sqrt{n\mu}. \tag{50}$$

Moreover, using the fact that $\bar{s} \leq s^0$ and $\bar{x} \leq x^0$ along with Lemma 3.2, we obtain

$$\eta\|X(s^0 - \bar{s})\| + \|S(x^0 - \bar{x})\| \;\leq\; \eta({s^0}^T x + {x^0}^T s) \;\leq\; 3n\mu. \tag{51}$$

The result now follows by incorporating inequalities (50), (51) and (29) into (49). ▪

We are now ready to prove Theorem 2.6.

Proof of Theorem 2.6: Let $(\Delta x^k, \Delta s^k, \Delta y^k)$ denote the search direction, and let $r^k = r(x^k, s^k, y^k)$ and $\mu_k = \mu(x^k, s^k)$, at the k-th iteration of Algorithm IIP-IS. Clearly, $(x^k, s^k, y^k) \in \mathcal{N}(\gamma)$, and using Lemma 3.3, it is easy to see that $r^k = \eta r^0$ for some $\eta \in (0, 1)$. Hence, using Lemma 3.7, assumption (30) and the inequality

$$\|\Delta x^k \circ \Delta s^k\|_\infty \leq \|\Delta x^k \circ \Delta s^k\| \leq \|(D^k)^{-1}\Delta x^k\| \, \|D^k \Delta s^k\|,$$

we easily see that $\|\Delta x^k \circ \Delta s^k\|_\infty = \mathcal{O}(n^2)\mu_k$. Using this conclusion together with assumption (30) and Lemma 3.6, we see that, for some universal constant $\beta > 0$, we have

$$\mu_{k+1} \leq \left(1 - \frac{\beta}{n^2}\right)\mu_k, \quad \forall k \geq 0.$$

The conclusion of the theorem now follows by using the above inequality, the fact that $\|r^k\|/\|r^0\| \leq \mu_k/\mu_0$ for all $k \geq 0$, and some standard arguments (see, for example, Theorem 3.2 of [15]). ▪

4 Concluding Remarks

We have shown in this paper that the outer iteration complexity of Algorithm IIP-IS is the same as that of its direct counterpart, namely Algorithm IIP. We strongly believe that, using approaches similar to ours, it is possible to develop iterative versions of other primal-dual IP methods whose outer iteration complexities match those of their direct counterparts.

As we showed in Section 2.4, an inexact Δy leads to an error in the Newton equation (9) corresponding to primal feasibility. We have addressed this problem by introducing the vector v in equation (24) defining Δx. This correction term v can be used, not only in the context of iterative methods, but also in connection with direct methods, as was pointed out in Section 2.4. It would be interesting to see how the addition of this correction term v in the context of direct methods could help handle LP problems which are extremely hard to solve due to the ill-conditioning of AD^2A^T. Clearly, a certain overhead exists in computing v, but it might be worthwhile in such a case.

In order to satisfy the polynomial convergence of our methods, we have imposed stringent conditions on $\tilde{f}$. In a practical situation, it may be more appropriate to monitor v directly. Indeed, as long as v satisfies the requirements in Lemmas 3.4 and 3.5, the outer iteration convergence analysis used in this paper remains valid. Moreover, weaker requirements than those imposed on v in the two lemmas above might be more advantageous from the practical point of view. This is certainly a topic that deserves further investigation.

References

[1] V. Baryamureeba, T. Steihaug, and Y. Zhang. Properties of a class of preconditioners for weighted least squares problems. Technical Report 16, Department of Computational and Applied Mathematics, Rice University, 1999.

[2] W.J. Cook, W.H. Cunningham, W.R. Pulleybank, and A. Schrijver. *Combinatorial Optimization*. Wiley, 1997.

[3] R.W. Freund, F. Jarre, and S. Mizuno. Convergence of a class of inexact interior-point algorithms for linear programs. *Mathematics of Operations Research*, 24(1):50–71, 1999.

[4] M. Kojima, N. Megiddo, and S. Mizuno. A primal-dual infeasible-interior-point algorithm for linear programming. *Mathematical Programming*, 61(3):263–280, 1993.

[5] J. Korzak. Convergence analysis of inexact infeasible-interior-point algorithms for solving linear programming problems. *SIAM Journal on Optimization*, 11(1):133–148, 2000.

[6] V.V. Kovacevic-Vujcic and M.D. Asic. Stabilization of interior-point methods for linear programming. *Computational Optimization and Applications*, 14:331–346, 1999.

[7] D.G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, 1984.

[8] S. Mizuno and F. Jarre. Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation. *Mathematical Programming*, 84:357–373, 1999.

[9] R.D.C. Monteiro, J.W. O'Neal, and T. Tsuchiya. Uniform boundedness of a preconditioned normal matrix used in interior point methods. Technical report, Georgia Institute of Technology, 2003. Submitted to *SIAM Journal on Optimization*.

[10] A.R.L. Oliveira and D.C. Sørensen. Computational experience with a preconditioner for interior point methods for linear programming. Technical Report 28, Department of Computational and Applied Mathematics, Rice University, 1997.

[11] L.F. Portugal, M.G.C. Resende, G. Veiga, and J.J. Judice. A truncated primal-infeasible dual-feasible network interior point method. *Networks*, 35:91–108, 2000.

[12] M.G.C. Resende and G. Veiga. An implementation of the dual affine scaling algorithm for minimum cost flow on bipartite uncapacitated networks. *SIAM Journal on Optimization*, 3:516–537, 1993.

[13] M.J. Todd, L. Tunçel, and Y. Ye. Probabilistic analysis of two complexity measures for linear programming problems. *Mathematical Programming A*, 90:59–69, 2001.

[14] S.A. Vavasis and Y. Ye. A primal-dual interior point method whose running time depends only on the constraint matrix. *Mathematical Programming A*, 74:79–120, 1996.

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

[16] Y. Zhang. On the convergence of a class of infeasible interior-point methods for the horizontal linear complementarity problem. *SIAM Journal on Optimization*, 4(1):208–227, 1994.