A Distributed Newton Method for Network Utility Maximization—Part II: Convergence

Ermin Wei, Student Member, IEEE, Asuman Ozdaglar, Member, IEEE, and Ali Jadbabaie, Senior Member, IEEE

Abstract—The existing distributed algorithms for network utility maximization (NUM) problems are mostly constructed using dual decomposition and first-order (gradient or subgradient) methods, which suffer from a slow rate of convergence. Part I of this paper proposed an alternative distributed Newton-type algorithm for solving NUM problems with self-concordant utility functions. For each primal iteration, this algorithm features distributed exact stepsize calculation with finite termination and decentralized computation of the dual variables using a finitely truncated iterative scheme obtained through novel matrix splitting techniques. This paper analyzes the convergence properties of a broader class of algorithms with potentially different stepsize computation schemes. In particular, we allow for errors in the stepsize computation. We show that if the error levels in the Newton direction (resulting from finite termination of dual iterations) and stepsize calculation are below a certain threshold, then the algorithm achieves local quadratic convergence rate in primal iterations to an error neighborhood of the optimal solution, where the size of the neighborhood can be explicitly characterized by the parameters of the algorithm and the error levels.

Index Terms—Distributed Newton method, distributed optimization, network utility maximization (NUM), superlinear rate of convergence.

I. INTRODUCTION

THERE has been much recent interest in developing distributed algorithms for solving convex optimization problems over networks. This is mainly motivated by resource allocation problems that arise in large-scale communication networks. This paper focuses on the rate allocation problem in wireline networks, which can be formulated as the network utility maximization (NUM) problem (see [1], [8], [18], [20], and [26]). NUM problems are characterized by a fixed set of sources with predetermined routes over a network topology. Each source in the network has a local utility, which is a function of the rate at which it transmits information over the network. The objective is to determine the source rates that maximize the sum of the utilities without violating link capacity constraints. The standard approach for solving NUM problems in a distributed way relies on using dual decomposition and first-order (subgradient) methods, which through a dual-price exchange mechanism enables each source to determine its transmission rate using only locally available information [17], [20], [23]. However, the drawback of these methods is their slow rate of convergence.

In this paper, we study the convergence properties of a distributed Newton-type method for solving NUM problems proposed in Part I of this paper. This method involves an iterative scheme to compute the dual variables based on matrix splitting and uses the same information-exchange mechanism as that of the first-order methods applied to the NUM problem. The stepsize rule is inversely proportional to the inexact Newton decrement (where the inexactness arises due to errors in the computation of the Newton direction) if this decrement is above a certain threshold and takes the form of a pure Newton step otherwise.

Since the method uses iterative schemes to compute the Newton direction, exact computation is not feasible. In this paper, we consider a truncated version of this scheme and present a convergence rate analysis of the constrained Newton method when the stepsize and the Newton direction are estimated with some error. We show that when these errors are sufficiently small, the value of the objective function converges superlinearly in terms of primal iterations to a neighborhood of the optimal objective function value, whose size is explicitly quantified as a function of the errors and bounds on them.

Our paper is most related to [5] and [14]. In [5], the authors have developed a distributed Newton-type method for the NUM problem using a belief propagation algorithm. Belief propagation algorithms, while performing well in practice, lack systematic convergence guarantees. Another recent paper [14] studied a Newton method for equality-constrained network optimization problems and presented a convergence analysis under Lipschitz assumptions. In this paper, we focus on an inequality-constrained problem, which is reformulated as an equality-constrained problem using barrier functions. Therefore, this problem does not satisfy Lipschitz assumptions. Instead, we assume that the utility functions are self-concordant and present a novel convergence analysis using properties of self-concordant functions.

Our analysis for the convergence of the algorithm also relates to work on convergence rate analysis of inexact Newton methods (see [11] and [16]). These works focus on providing conditions on the amount of error at each iteration relative to the norm of the gradient of the current iterate that ensures superlinear convergence to the exact optimal solution (essentially...
requiring the error to vanish in the limit). Even though these analyses can provide a superlinear rate of convergence, the vanishing error requirement can be too restrictive for practical implementations. Another novel feature of our analysis is the consideration of convergence to an approximate neighborhood of the optimal solution. In particular, we allow a fixed error level to be maintained at each step of the Newton direction computation and show that superlinear convergence is achieved by the primal iterates to an error neighborhood, whose size can be controlled by tuning the parameters of the algorithm. Hence, our work also contributes to the literature on error analysis for inexact Newton methods.

The rest of this paper is organized as follows. Section II defines the problem formulation and related transformations. Section III describes the exact constrained primal-dual Newton method for this problem. Section IV outlines the distributed inexact Newton-type algorithm developed in Part I of the paper. Section V contains the rate of convergence analysis for our algorithm. Section VI contains our concluding remarks. Due to space constraints, we omit some of the proofs here and include them in the longer version of our paper [30].

Basic Notation and Notions: A vector is viewed as a column vector, unless clearly stated otherwise. We write $\mathbb{R}_+$ to denote the set of non-negative real numbers, that is, $\mathbb{R}_+ = [0, \infty)$. We use subscripts to denote the components of a vector and superscripts to index a sequence (i.e., $x_i$ is the $i$th component of vector $x$ and $x^k$ is the $k$th element of a sequence). When $x_i \geq 0$ for all components $i$ of a vector $x$, we write $x > 0$.

For a matrix $A$, we write $A_{ij}$ to denote the matrix entry in the $i$th row and $j$th column. We write $I(n)$ to denote the identity matrix of dimension $n \times n$. We use $x'$ and $A'$ to denote the transpose of a vector $x$ and a matrix $A$, respectively. For a real-valued function $f : X \to \mathbb{R}$, where $X$ is a subset of $\mathbb{R}^n$, the gradient vector and the Hessian matrix of $f$ at $x$ in $X$ are denoted by $\nabla f(x)$ and $\nabla^2 f(x)$, respectively. We use the vector $e$ to denote the vector of all ones.

A real-valued convex function $g : X \to \mathbb{R}$, where $X$ is a subset of $\mathbb{R}$, is self-concordant if it is three times continuously differentiable and $g'''(x) \leq 2g''(x)^{3/2}$ for all $x$ in its domain.1 For real-valued functions in $\mathbb{R}^n$, a convex function $g : X \to \mathbb{R}$, where $X$ is a subset of $\mathbb{R}^n$, is self-concordant if it is self-concordant along every direction in its domain, that is, if the function $g(t) = g(x + tv)$ is self-concordant in $t$ for all $x$ and $v$. Operations that preserve self-concordant property include summing, scaling by a factor $\alpha \geq 1$, and composition with affine transformation (see [7, Ch. 9] for more details).

II. NETWORK UTILITY MAXIMIZATION PROBLEM

For a fixed network, we use $E = \{1, \ldots, L\}$ to denote the set of (directed) links of finite nonzero capacity, given by $c = |c_{l} |_{l \in E}$. and use $S = \{1, \ldots, S\}$ to indicate the set of sources, each of which transmits information along a predetermined route.2 For each link $l$, we write $S(l)$ to denote the set of sources using it. For each source $i$, we use $L(i)$ to denote the set of links it uses. Let the non-negative source rate vector be denoted by $s = |s_i |_{i \in S}$. Let matrix $R$ be the routing matrix of dimension $L \times S$, given by

$$
R_{ij} = \begin{cases} 
1 & \text{if link } i \text{ is on the route of source } j, \\
0 & \text{otherwise.}
\end{cases}
$$

(1)

For each $i$, we use $U_i : \mathbb{R}_+ \to \mathbb{R}$ to denote the utility function of source $i$. The NUM problem involves choosing the source rates to maximize a global system function given by the sum of all utility functions and can be formulated as

$$
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{S} U_i(s_i) \\
\text{subject to} & \quad Rs \leq c, \quad s \geq 0.
\end{align*}
$$

(2)

We adopt the following assumptions on the utility function.

Assumption 1: The utility functions $U_i : \mathbb{R}_+ \to \mathbb{R}$ are continuous, strictly concave, monotonically nondecreasing on $\mathbb{R}_+$, and twice continuously differentiable on the set of positive real numbers. The function $-\sum_{i=1}^{S} U_i$ is self-concordant on $(0, \infty)$. The self-concordance assumption is satisfied by standard utility functions considered in the literature, for instance, logarithmic (i.e., weighted proportionally fair, see [21] and [26]) utility functions, and is adopted here to allow a self-concordant analysis in establishing local quadratic convergence. We use $h(x)$ to denote the (negative of the) objective function of problem (2), that is, $h(x) = -\sum_{i=1}^{S} U_i(x_i)$, and $h^*$ to denote the (negative of the) optimal value of this problem.3 Since $h(x)$ is continuous and the feasible set of problem (2) is compact, it follows that problem (2) has an optimal solution and, therefore, $h^*$ is finite. Moreover, the interior of the feasible set is nonempty, that is, a feasible solution $x$ exists with $x_i = c/S + 1$ for all $i \in S$ with $\epsilon > 0$.

We reformulate the problem into one with only equality constraints by introducing non-negative slack variables $|y_l|_{l \in L}$, such that

$$
\sum_{j=1}^{S} R_{ij}s_j + y_l = c_l \quad \text{for } l = 1, 2 \ldots, L,
$$

(3)

and using logarithmic barrier functions for the non-negativity constraints (which can be done since the feasible set of (2) has a nonempty interior).4 The new decision vector is $x = (|s_i|_{i \in S}, |y_l|_{l \in L})$ and problem (2) can be rewritten as

$$
\begin{align*}
\text{minimize} & \quad -\sum_{i=1}^{S} U_i(x_i) - \mu \sum_{i=1}^{S} \log(x_i) \\
\text{subject to} & \quad Ax = c.
\end{align*}
$$

(4)

We assume that each source flow traverses at least one link and each link is used by at least one source.

We consider the negative of the objective function value to work with a minimization problem.

We adopt the convention that $\log(x) = -\infty$ for $x \leq 0$. 

1Self-concordant functions are defined through the following more general definition: a real-valued three times continuously differentiable convex function $g : X \to \mathbb{R}$, where $X$ is a subset of $\mathbb{R}$, is self-concordant, if there exists a constant $\alpha > 0$, such that $\|g''(x)\| \leq 2\alpha g'''(x)^{3/2}$ for all $x$ in its domain [15], [25]. Here, we focus on the case $\alpha = 1$ for notational simplification in the analysis.

2We assume that each source flow traverses at least one link and each link is used by at least one source.

3We consider the negative of the objective function value to work with a minimization problem.

4We adopt the convention that $\log(x) = -\infty$ for $x \leq 0$. 

5Self-concordant functions are defined through the following more general definition: a real-valued three times continuously differentiable convex function $g : X \to \mathbb{R}$, where $X$ is a subset of $\mathbb{R}$, is self-concordant, if there exists a constant $\alpha > 0$, such that $\|g''(x)\| \leq 2\alpha g'''(x)^{3/2}$ for all $x$ in its domain [15], [25]. Here, we focus on the case $\alpha = 1$ for notational simplification in the analysis.
where $A$ is the $L \times (S + L)$-dimensional matrix given by

$$A = R^T f(I),$$  \tag{5}$$

and $\mu$ is a non-negative barrier function coefficient. We use $f(x)$ to denote the objective function of problem (4), i.e.,

$$f(x) = \sum_{i=1}^{S} U_i(x_i) - \mu \sum_{i=1}^{S+L} \log(x_i);$$  \tag{6}$$

and $f^*$ to denote the optimal value of this problem, which is finite for positive $\mu$.\footnote{This problem has a feasible solution, hence $f^*$ is upper bounded. Each of the variables $x_i$ is upper bounded by $\bar{c}$, where $\bar{c} = \max\{c_i\}$; hence, by monotonicity of utility and logarithm functions, the optimal objective function value is lower bounded. Note that in the optimal solution of problem (4), $x_i \neq 0$ for all $i$, due to the logarithmic barrier functions.}

By Assumption 1, the function $f(x)$ is separable, strictly convex, and has a positive definite diagonal Hessian matrix on the positive orthant. The function $f(x)$ is also self-concordant for $\mu \geq 1$, since both summing and scaling by a factor $\mu \geq 1$ preserve self-concordance property.

We write the optimal solution of problem (4) for a fixed barrier function coefficient $\mu$ as $x(\mu)$. One can show that as the barrier function coefficient $\mu$ approaches 0, the optimal solution of problem (4) approaches that of problem (2), when the constraint set in (2) has a nonempty interior and is convex\cite{3,12}. Hence, by continuity from Assumption 1, $h(x(\mu))$ approaches $h^*$. Therefore, in the rest of this paper, unless clearly stated otherwise, we study iterative distributed methods for solving problem (4) for a given $\mu$. In order to preserve the self-concordance property of the function $f$, which will be used in our convergence analysis, we first develop a Newton-type algorithm for $\mu \geq 1$. In Section V-C, we show that problem (4) for any $\mu > 0$ can be tackled by solving two instances of problem (4) with different coefficients $\mu \geq 1$, leading to a solution $x(\mu)$ that satisfies $((h(x(\mu)) - h^*)/h^*) \leq \alpha$ for any positive scalar $\alpha$.

$$III. EXACT NEWTON METHOD$$

For each fixed $\mu$, problem (4) is feasible and has a convex objective function, affine constraints, and a finite optimal value $f^*$. Therefore, we can use a strong duality theorem to show that, for problem (4), there is no duality gap and there exists a dual optimal solution (see [4]). Moreover, since matrix $A$ has full-row rank, we can use a (feasible start) equality constrained Newton method to solve problem (4) (see [7, Ch. 10], which serves as a starting point in the development of a distributed algorithm. In our iterative method, we use $x_k$ to denote the primal vector at the $k$th iteration.

$$A. FEASIBLE INITIALIZATION$$

We initialize the algorithm with some feasible and strictly positive vector $x^0$. One example is given in Part I of this paper.

$$B. ITERATIVE UPDATE RULE$$

We denote $H_k = \nabla^2 f(x_k)$ for notational convenience. Given an initial feasible vector $x^0$, the algorithm generates the iterates by $x_{k+1} = x_k + \Delta x_k$, where $\Delta x_k$ is a positive stepsize, and $\Delta x_k$ is the (primal) Newton direction given as

$$\Delta x_k = -H_k^{-1} (\nabla f(x_k) + A^t w_k),$$  \tag{7}$$

and

$$\begin{align*}
AH_k^{-1} A^t w_k &= -AH_k^{-1} \nabla f(x_k),
\end{align*}$$  \tag{8}$$

where $w_k = [w_k^t]_{i \in E}$ is the dual vector and the $w_k^i$ are the dual variables for the link capacity constraints at primal iteration $k$, where the matrix $H_k$ is a diagonal matrix with entries

$$\langle H_k \rangle_{ii} = \left\{ \begin{array}{ll}
\frac{-\bar{c} U_i(x_k^i)}{\bar{c}^2} + \frac{\mu}{\bar{c}^2} & 1 \leq i \leq S,
\frac{\mu}{\bar{c}^2} & S + 1 < i < S + L.
\end{array} \right.$$  \tag{9}$$

IV. DISTRIBUTED INEXACT NEWTON METHOD

Our distributed Newton algorithm is developed based on (7) and (8). In Section IV-A, we introduce some preliminaries on matrix splitting techniques. In Section IV-B, we use ideas from matrix splitting to compute the dual vector $w^k$ at each $k$ [cf. (8)] using an iterative scheme, which serves as the theoretical foundation for the distributed implementation. In Section IV-C, we outline the distributed primal update of our inexact Newton method.

$$A. PRELIMINARIES ON MATRIX SPLITTING$$

Matrix splitting can be used to solve a system of linear equations given by $G y = a$, where $G$ is an $n \times n$ matrix and $a$ is an $n$-dimensional vector. Suppose that the matrix $G$ can be expressed as the sum of an invertible matrix $M$ and a matrix $N$, i.e.,

$$G = M + N.$$  \tag{10}$$

Let $y_0$ be an arbitrary $n$-dimensional vector. A sequence $\{y^k\}$ can be generated by the following iteration:

$$y^{k+1} = -M^{-1} Ny^k + M^{-1} a.$$  \tag{11}$$

It can be seen that the sequence $\{y^k\}$ converges as $k \to \infty$ if and only if the spectral radius of the matrix $M^{-1} N$ is strictly bounded above by 1. When the sequence $\{y^k\}$ converges, its limit $y^*$ solves the original linear system, that is, $G y^* = a$ (see [2] and [10] for more details). Hence, the key to solving the linear equation via matrix splitting is the bound on the spectral radius of the matrix $M^{-1} N$. Such a bound can be obtained by using the following result (see [10, Theor. 2.5.3]).

**Theorem IV.1**: Let $G$ be a real symmetric matrix. Let $M$ and $N$ be matrices such that $G = M + N$ and assume that $M$ is invertible and both matrices $M + N$ and $M - N$ are positive definite. Then, the spectral radius of $M^{-1} N$, denoted by $\rho(M^{-1} N)$, satisfies $\rho(M^{-1} N) < 1$.

By the above theorem, if $G$ is a real, symmetric, positive definite matrix, and $M$ is a nonsingular matrix, then one sufficient condition for the iteration (11) to converge is that the matrix $M - N$ is positive definite. This can be guaranteed using the Gershgorin Circle Theorem, which we introduce next (see [28] for more details).

**Theorem IV.2 (Gershgorin Circle Theorem)**: Let $G$ be an $n \times n$ matrix, and define $r_i(G) = \sum_{j \neq i} |G_{ij}|$. Then, each eigenvalue of $G$ lies in one of the Gershgorin sets $\{\Gamma_i\}$, with
defined as disks in the complex plane, that is, \( \Gamma_i = \{ z \in \mathbb{C} \; | \; z - G_{ii} \leq r_i \} \).

One corollary of the above theorem is that if a matrix \( G \) is strictly diagonally dominant, that is, 
\[ G_{ii} > \sum_{j \neq i} |G_{ij}|, \] and for all \( s \), then the real parts of all the eigenvalues lie in the positive half of the real line and, thus, the matrix is positive definite. Hence, a sufficient condition for the matrix \( M - N \) to be positive definite is that \( M - N \) is strictly diagonally dominant with strictly positive diagonal entries.

### B. Distributed Computation of the Dual Vector

We use the matrix splitting scheme introduced in the preceding section to compute the dual vector \( u^k \) in (8) in a distributed manner for each primal iteration \( k \). For notational convenience, we may suppress the explicit dependence of \( u^k \) on \( k \).

Let \( D_k \) be a diagonal matrix, with diagonal entries
\[ (D_k)_{ii} = (AH_k^{-1}A')_{ii} \] (12)
and matrix \( B_k \) be given by
\[ B_k = AH_k^{-1}A' - D_k. \] (13)

Let matrix \( \bar{B}_k \) be a diagonal matrix, with diagonal entries
\[ (\bar{B}_k)_{ii} = \sum_{j=1}^{L} (B_k)_{ij}. \] (14)

By splitting the matrix \( AH_k^{-1}A' \) as the sum of \( D_k + \bar{B}_k \) and \( B_k - \bar{B}_k \), we obtain the following result.

**Theorem IV.3:** For a given \( k > 0 \), let \( D_k, B_k, \bar{B}_k \) be the matrices defined in (12) – (14). Let \( u(0) \) be an arbitrary initial vector and consider the sequence \( \{w(t)\} \) generated by the iteration
\[ w(t + 1) = (D_k + \bar{B}_k)^{-1}(\bar{B}_k - B_k)w(t) + (D_k + \bar{B}_k)^{-1}\left(-AH_k^{-1}\nabla f(x^k)\right) \] (15)
for all \( t \geq 0 \). Then, the spectral radius of the matrix \( (D_k + \bar{B}_k)^{-1}(\bar{B}_k - B_k) \) is strictly bounded above by 1 and the sequence \( \{w(t)\} \) converges as \( t \to \infty \), and its limit is the solution to (8).

**a) Proof:** We split the matrix \( AH_k^{-1}A' \) as
\[ (AH_k^{-1}A') = (D_k + \bar{B}_k) + (B_k - \bar{B}_k) \] (16)
and use the iterative scheme presented in (10) and (11) to solve (8). For all \( k \), the real matrix \( H_k \) and its inverse \( H_k^{-1} \) are positive definite and diagonal. The matrix \( A \) has full-row rank and is element-wise non-negative. Therefore, the product \( AH_k^{-1}A' \) is real, symmetric, element-wise non-negative, and positive definite. We let
\[ Q_k = (D_k + \bar{B}_k) - (B_k - \bar{B}_k) = D_k + 2\bar{B}_k - B_k \] (17)
denote the difference matrix. By definition of \( \bar{B}_k \) [cf. (14)], the matrix \( 2\bar{B}_k - H_k \) is diagonally dominant, with non-negative diagonal entries. Moreover, due to strict positivity of the second derivatives of the logarithmic barrier functions, we have \((D_k)_{ii} > 0 \) for all \( i \). Therefore, the matrix \( Q_k \) is strictly diagonally dominant. By Theorem IV.2, such matrices are positive definite. Therefore, by Theorem IV.1, the spectral radius of the matrix \( (D_k + \bar{B}_k)^{-1}(B_k - \bar{B}_k) \) is strictly bounded above by 1. Hence, the splitting scheme (16) guarantees the sequence \( \{w(t)\} \) generated by iteration (15) to converge to the solution of (8).

This provides an iterative scheme to compute the dual vector \( u^k \) at each primal iteration \( k \) using an iterative scheme. We will refer to the iterative scheme defined in (15) as the **dual iteration**. There are many ways to split the matrix \( AH_k^{-1}A' \). The particular one in (16) is chosen here due to two desirable features. First, it guarantees that the difference matrix \( Q_k \) [cf. (17)] is strictly diagonally dominant and, hence, ensures convergence of the sequence \( \{w(t)\} \). Second, with this splitting scheme, the matrix \( D_k + \bar{B}_k \) is diagonal, which eliminates the need for global information when calculating its inverse. As shown in Part I of this paper, we can rewrite iteration (15) as in a distributed way to compute the dual variables \( u^k \). By using (7), an inexact primal Newton direction, denoted by \( \Delta x^k \), can then be obtained, where the inexactness arises due to the finite truncation of the dual iteration.

### C. Distributed Computation of the Primal Update

We can now formally present our distributed inexact Newton algorithm. Starting from an initial feasible vector \( x^0 \) (e.g., the initialization in Part I of this paper), the distributed Newton algorithm generates the primal vectors \( x^k \) as follows:
\[ x^{k+1} = x^k + d^k \Delta x^k \] (18)
where \( d^k \) is a positive stepsize, and \( \Delta x^k \) is the inexact Newton direction at the \( k \)th iteration.

The stepsize used in the distributed algorithm is based on an inexact Newton decrement, which we introduce next. We refer to the exact solution of the system of (7) as the **exact Newton direction**, denoted by \( \Delta x^k \). The inexact Newton direction \( \Delta \hat{x}^k \) computed by our algorithm is a feasible estimate of \( \Delta x^k \). At a given primal vector \( x^k \), we define the **exact Newton decrement** \( \lambda(x^k) \) as
\[ \lambda(x^k) = \sqrt{(\Delta x^k)^T \nabla^2 f(x^k) \Delta x^k}. \] (19)

Similarly, the **inexact Newton decrement** \( \tilde{\lambda}(x^k) \) is given by
\[ \tilde{\lambda}(x^k) = \sqrt{(\Delta \hat{x}^k)^T \nabla^2 f(x^k) \Delta \hat{x}^k}. \] (20)

Note that \( \lambda(x^k) \) and \( \tilde{\lambda}(x^k) \) are non-negative and well defined because the matrix \( \nabla^2 f(x^k) \) is positive definite.

We assume that \( \tilde{\lambda}(x^k) \) is obtained through some distributed computation procedure and denote \( \theta^k \) as its approximate value. One possible procedure with finite termination yielding \( \theta^k = \tilde{\lambda}(x^k) \) is described in Part I of this paper. However, other estimates \( \theta^k \) can be used, which can potentially be obtained by exploiting the diagonal structure of the Hessian matrix, writing the inexact Newton decrement as
\[ \tilde{\lambda}(x^k) = \sqrt{\sum_{i \in \mathcal{E}} (\Delta \hat{x}^k)^2 (H_k)_{ii} = \sqrt{(I + S)\tilde{\gamma}}}, \]
where \( \tilde{g} = (1/S + L) \sum_{i \in S} \sum_{j \in C} (\Delta \tilde{x})^2_{ij} (H_k)_{ii} \) and using iterative consensus-type algorithms.

Given the scalar \( \theta^k \), an approximation to the inexact Newton decrement \( \lambda(x^k) \), at each iteration \( k \), we choose the stepsize \( d^k \) as follows: Let \( V \) be some positive scalar with \( 0 < V < 0.267 \). Based on [25], we have

\[
d^k = \begin{cases} \frac{V}{1 + V}, & \text{if } \theta^k > V \text{ for all previous } k \\ 1, & \text{otherwise} \end{cases}
\]

(21)

where \( (V+1)/(2V+1) < b \leq 1 \). The upper bound on \( V \) will be used in the analysis of the quadratic convergence phase of our algorithm [cf. Assumption 4]. This bound will also ensure the strict positivity of the generated primal vectors [cf. Theo. IV.5]. The lower bound on \( b \) will be used to guarantee a lower bounded improvement in the damped convergent phase. The stepsize rule in part I of this paper uses \( \theta^k = \hat{\lambda}(x^k) \) and \( b = 1 \) as a special case of this broader class of stepsize rules.

There are three sources of inexactness in this algorithm: finite precision achieved in the computation of the dual vector due to truncation of the matrix splitting scheme, two-stage computation of an approximate primal direction to maintain feasibility, and inexact stepsize value obtained from a finitely truncated consensus algorithm. The following assumptions quantify the bounds on the resulting error levels.

**Assumption 2:** Let \( \{x^k\} \) denote the sequence of primal vectors generated by the distributed inexact Newton method. Let \( \Delta x^k \) and \( \Delta \hat{x}^k \) denote the exact and inexact Newton directions at \( x^k \), and \( \gamma^k \) denote the error in the Newton direction computation, i.e.,

\[
\Delta x^k = \Delta \hat{x}^k + \gamma^k.
\]

(22)

For all \( k \), \( \gamma^k \) satisfies

\[
\left( \gamma^k \right)^T \nabla^2 f(x^k) \gamma^k \leq p^2 (\Delta \hat{x}^k)^T \nabla^2 f(x^k) \Delta \hat{x}^k + \epsilon
\]

(23)

for some positive scalars \( p < 1 \) and \( \epsilon \).

This assumption imposes a bound on the weighted norm of the Newton direction error \( \gamma^k \) as a function of the weighted norm of \( \Delta \hat{x}^k \) and a constant \( \epsilon \). Note that without the constant \( \epsilon \), we would require this error to vanish when \( x^k \) is close to the optimal solution, that is, when \( \Delta \hat{x}^k \) is small, which is impractical for implementation purposes. Since the errors arise due to finite truncation of the dual iteration (15), the primal Newton direction can be computed with arbitrary precision. Therefore, given any \( p \) and \( \epsilon \), the dual computation can terminate after a certain number of iterations such that the resulting error \( \gamma^k \) satisfies this Assumption.

Recent papers [29] (Part I of this paper) and [31] presented two different distributed methods to determine when to terminate the dual computation procedure such that the aforementioned error tolerance level is satisfied. The method in Part I of this paper has two stages: in the first stage, a predetermined number of dual iterations is implemented; in the second stage, the error bound is checked after each dual iteration. The method in [31] computes an upper bound on the number of dual iterations required to satisfy Assumption 2 at each primal iteration. Simulation results suggest that the method proposed in [31] yields a loose upper bound, while it does not require distributed error checking at each dual iteration and, hence, involves less communication and computation overhead in terms of error checking.

We bound the error in the inexact Newton decrement calculation as follows.

**Assumption 3:** Let \( \tau^k \) denote the error in the Newton decrement calculation, i.e.,

\[
\tau^k = \hat{\lambda}(x^k) - \theta^k.
\]

(24)

For all \( k \), \( \tau^k \) satisfies

\[
\tau^k \leq \frac{1}{b - 1} (1 + V).
\]

(25)

This assumption will be used in establishing the strict positivity of the generated primal vectors \( x^k \). When the method presented in Part I of this paper is used to compute \( \theta^k \), then we have \( \tau^k = 0 \) and \( b = 1 \) for all \( k \) and the preceding assumption is satisfied clearly. Throughout the rest of this paper, we assume the conditions in Assumptions 1–3 hold.

In Part I of this paper, we have shown that the stepsize choice with \( \theta^k = \hat{\lambda}(x^k) \) and \( b = 1 \) can guarantee strict positivity of the primal vector \( x^k \) generated by our algorithm, which is important since it ensures that the Hessian \( H^k \) and, therefore, the (inexact) Newton direction is well defined at each iteration. We next show that the stepsize choice in (21) will also guarantee strict positivity. We first establish a bound on the error in the stepsize under Assumption 3.

**Theorem IV.5:** Given a strictly positive feasible primal vector \( x^0 \), let \( \{x^k\} \) be the sequence generated by the inexact distributed Newton method (18). Assume that the stepsize \( d^k \) is selected according to (21), and the constant \( b \) satisfies \((V + 1)/(2V + 1) < b \leq 1 \). Then, the primal vector \( x^k \) is strictly positive for all \( k \).

**Proof:** We will prove this claim by induction. The base case of \( x^0 > 0 \) holds by the assumption of the theorem. Since the \( U_i \) are strictly concave [cf. Assumption 1], for any \( x^k \), we have \(-((\partial^2 U_i/\partial x^k_i)^T x^k_i) \geq 0 \). Given the form of the Hessian matrix [cf. (9)], this implies \( (H_k)_{ii} \geq \mu (x^k_i)^2 \) for all \( i \), and therefore \( \hat{\lambda}(x^k) - (\sum_{i = 1}^{S + L} (\Delta \hat{x}^k_i)^2 (H_k)_{ii})^{1/2} \geq (\sum_{i = 1}^{S + L} \mu (\Delta \hat{x}^k_i/x^k_i)^2)^{1/2} \geq \max_i |\Delta \hat{x}^k_i/x^k_i| \), where the last inequality follows from the non-negativity of the terms \( \mu (\Delta \hat{x}^k_i/x^k_i)^2 \). By taking the reciprocal on both sides, the above relation implies

\[
\frac{1}{\hat{\lambda}(x^k)} \leq \frac{1}{\max_i |\Delta \hat{x}^k_i/x^k_i|} \leq \frac{1}{\sqrt{\mu}} \min_i \frac{x^k_i}{\Delta \hat{x}^k_i} \leq \min_i \frac{x^k_i}{|\Delta \hat{x}^k_i|}
\]

(26)
where the last inequality follows from the fact that $\mu \geq 1$.

We show the inductive step by considering two cases.

Case 1) $\theta^k \geq V$

Since $0 < ((V + 1)/(2V + 1)) < b \leq 1$, we can apply Lemma IV-4 and obtain that the stepsize $d^k$ satisfies

$$d^k \leq 1 / (1 + \lambda(x^k)) < 1 / \lambda(x^k).$$

Using (26), this implies $d^k < \min \{1, x^k / \Delta x^k\}$. Hence, if $x^k > 0$, then $x^{k+1} - x^k + d^k \Delta x^k > 0$.

Case 2) $\theta^k < V$

By Assumption 3, we have $\lambda(x^k) < V + ((1/b) - 1)(1 + V)$. Using the fact that $b > ((V + 1)/(2V + 1))$, we obtain $\lambda(x^k) < V + ((1/b) - 1)(1 + V) < V + ((2V + 1)/(V + 1)) - 1)(1 + V) = 2V \leq 1$, where the last inequality follows from the fact that $V < 0.267$. Hence, we have $d^k - 1 < (1/\lambda(x^k)) \leq \min \{x^k / \Delta x^k\}$, where the last inequality follows from (26). Once again, if $x^k > 0$, then $x^{k+1} - x^k + d^k \Delta x^k > 0$.

In both cases, we have $x^{k+1} = x^k + d^k \Delta x^k > 0$, which completes the induction proof.

Hence, the algorithm with a more general stepsize rule is also well defined. In the rest of this paper, we will assume that the constant $b$ used in the definition of the stepsize satisfies $((V + 1)/(2V + 1)) < b \leq 1$.

V. CONVERGENCE ANALYSIS

We next present our convergence analysis for both primal and dual iterations of the algorithm presented before. We first establish convergence for dual iterations.

A. Convergence in Dual Iterations

We study the convergence rate of iteration (15) in terms of a dual (routing) graph, which we introduce next.

Definition 1: Consider a network $G = \{L, S\}$, represented by a set $L = \{1, \ldots, I\}$ of (directed) links, and a set $S = \{1, \ldots, S\}$ of sources. The links form a strongly connected graph, and each source sends information along a predetermined route. The weighted dual (routing) graph $\hat{G} = \{N, L\}$, where $N$ is the set of nodes, and $\hat{L}$ is the set of (directed) links defined by:

a) $N = L$;

b) a link is present between node $I_i$ to $I_j$ in $\hat{G}$ if and only if there is some common flow between $I_i$ and $I_j$ in $G$;

c) the weight $\hat{W}_{ij}$ on the link from node $I_i$ to $I_j$ is given by $\hat{W}_{ij} = (D_k + \hat{B}_k)^{-1}(B_k + \hat{B}_k)^{-1}(AH_k^1 A')_{ij}$.

Two sample network—dual graph pairs are presented in Fig. 1(a) and (b) and 2(a) and (b), respectively. Note that the unweighted indegree and outdegree of a node are the same in the dual graph; however, the weights are different depending on the direction of the links. The splitting scheme in the dual iteration involves the matrix $(D_k + \hat{B}_k)^{-1}(\hat{B}_k - B_k)$, which is the weighted Laplacian matrix of the dual graph.6 The weighted out-degree of node $i$ in the dual graph (i.e., the diagonal entry $(D_k + \hat{B}_k)_{ii}^{-1}B_{ii}$ of the Laplacian matrix) can be viewed as a measure of the congestion level of a link in the original network since the neighbors in the dual graph represent links that share flows in the original network. We next show that the spectral properties of the Laplacian matrix of the dual graph dictate the convergence speed of dual iteration (15). We will use the following lemma [27].

Lemma V.1: Let $M$ be an $n \times n$ matrix, and assume that its spectral radius, denoted by $\rho(M)$, satisfies $\rho(M) < 1$. Let $\{\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n\}$ denote the set of eigenvalues of $M$, with $1 > |\lambda_1| > |\lambda_2| > \ldots > |\lambda_n|$ and let $n_i$ denote the set of corresponding unit length right eigenvectors. Assume the matrix has $n$ linearly independent eigenvectors.7 Then, for the sequence $\bar{w}(t)$ generated by the following iteration:

$$w(t + 1) = Mu(t)$$

6 We adopt the following definition for the weighted Laplacian matrix of a graph. Consider a weighted undirected graph $G$ with weight $W_{ij}$ associated with the link from node $i$ to $j$. We let $W_{ii} = 0$ whenever the link is not present. These weights form a weighted adjacency matrix $W$. The weighted out-degree matrix $D$ is defined as a diagonal matrix with $D_{ii} = \sum_j W_{ij}$, and the weighted Laplacian matrix $L$ is defined as $L = D - W$. See [6] and [9] for more details on graph Laplacian matrices.

7 An alternative assumption is that the algebraic multiplicity of each $\lambda_i$ is equal to its corresponding geometric multiplicity, since eigenvectors associated with different eigenvalues are independent [19].
we have
\[ |w(t) - w^*|_2 \leq |\lambda_1|^t |w(0)|_2, \]
for some positive scalar $\alpha$, where $w^*$ is the limit of iteration (27) as $t \to \infty$.

We use $M$ to denote the $I \times I$ matrix $M = (I + B_k^{-1}B_k)^{-1}$ and $z$ to denote the vector $z = (D_k + B_k)^{-1}(I - M)^{-1}z$. We can rewrite iteration (15) as $w(t+1) = Mw(t) + z$, which implies $w(t+q) = M^qw(t) + \sum_{s=0}^{q-1} M^s (I - M)^{-1}z$. This alternative representation is possible since $\rho(M) < 1$, since the iteration converges as shown in Part I of this paper. After rearranging the terms, we obtain
\[ w(t+q) = M^q (w(t) - (I - M)^{-1}z) + (I - M)^{-1}z. \]

Therefore, starting from some arbitrary initial vector $w(0)$, the convergence speed of the sequence $w(t)$ coincides with the sequence $u(t)$, generated by $u(t+q) = M^q u(t) + (I - M)^{-1}z$, where $u(0) = w(0)$.

We next show that the matrix $M$ has $I$ linearly independent eigenvectors in order to apply the preceding lemma. We first note that since the non-negative matrix $A$ has full-row rank and the Hessian matrix $H$ has positive diagonal elements, the product matrix $AH_k^{-1}A'$ has positive diagonal elements and non-negative entries. This shows that the matrix $B_k$ [cf. (12)] has positive diagonal elements and the matrix $B$ [cf. (14)] has non-negative entries. Therefore, the matrix $(D_k + B_k)^{-1/2}$ is diagonal and nonsingular. Hence, using the relation $M = (I + B_k)^{-1/2}(D_k + B_k)^{-1}B_k^{-1/2}$, we see that the matrix $M = (D_k + B_k)^{-1}(B_k - B_k)(D_k + B_k)^{-1/2}$. From the definition of $B_k$ [cf. (13)] and the symmetry of the matrix $AH_k^{-1}A'$, we conclude that the matrix $B$ is symmetric. This shows that the matrix $M$ is symmetric and, hence, diagonalizable, which implies that the matrix $M$ is also diagonalizable and, therefore, it has $L$ linearly independent eigenvectors.8 We can use Lemma V.1 to infer that
\[ |w(t) - w^*|_2 \leq |\lambda_1|^t |w(0)|_2 \leq |\lambda_1|^t \alpha \]
where $\lambda_1$ is the eigenvalue of $M$ with the largest magnitude, and $\alpha$ is a constant that depends on the initial vector $w(0) = w(0) - (I - M)^{-1}z$. Hence, $\lambda_1$ determines the speed of convergence of the dual iteration.

We next analyze the relationship between $\lambda_1$ and the dual graph topology. First note that the matrix $M = (D_k + B_k)^{-1}(B_k - B_k)$ is the weighted Laplacian matrix of the dual graph [cf. Definition 1], and is therefore positive semidefinite [9]. We then have $\rho(M) = \lambda_1 = \lambda_1 > 0$. From graph theory [22] and the above analysis, we have
\[ \frac{4\text{mc}(M)}{L} \leq \lambda_1 \leq \min \left\{ 2 \max_{i \in L} \left[ (D_k + B_k)^{-1}B_k \right]_{ij}, 1 \right\}. \]

where $\text{mc}(M)$ is the weighted maximum cut of the dual graph, i.e.,
\[ \text{mc}(M) = \max_{\mathcal{S} \subseteq \mathcal{V}} \left\{ \sum_{i \in \mathcal{S}, j \notin \mathcal{S}} W_{ij} + \sum_{i \notin \mathcal{S}, j \in \mathcal{S}} W_{ji} \right\}, \]
where $W_{ij}$ is the weight associated with the link from node $i$ to $j$. The above relation suggests that a large maximal cut of the dual graph provides a large lower bound on $\lambda_1$, implying that the dual iteration cannot finish with very few iterations. When the maximum weighted out-degree (i.e., $\max_{i \in L} (D_k + B_k)^{-1}B_k_{ij}$) in the dual graph is small, the above relation provides a small upper bound on $\lambda_1$, hence suggesting that the dual iteration converges fast.

We finally illustrate the relationship between the dual graph topology and the underlying network properties by means of two simple examples that highlight how different network structures can affect the dual graph and, hence, the convergence rate of the dual iteration. In particular, we show that the dual iteration converges slower for a network with a more congested link. Consider once more the two networks given in Figs. 1(a) and 2(a), whose corresponding dual graphs are presented in Figs. 1(b) and 2(b), respectively. Both of these networks have three source-destination pairs and seven links. However, in Fig. 1(a), all three flows use the same link (i.e., $L_4$), whereas in Fig. 2(a), at most, two flows share the same link. This difference in network topology results in different degree distributions in the dual graphs as shown in Figs. 1(b) and 2(b). To be more concrete, let $U_i(c_i) = 15 \log_2(c_i)$ for all sources $i$ in both graphs and link capacity $c_i = 35$ for all links $i$. We apply our distributed Newton algorithm to both problems and for the primal iteration when all source rates are 10, the largest weighted out-degree in the dual graphs of the two examples is 0.46 for Fig. 1(b) and 0.095 for Fig. 2(b), which implies the upper bounds for $\lambda_1$ of the corresponding dual iterations are 0.92 and 0.19, respectively [cf. (29)]. The weighted maximum cut for Fig. 1(b) is obtained by isolating the node corresponding to $L_4$, with a weighted maximum cut value of 0.52. The maximum cut for Fig. 2(b) is formed by isolating the set $\{L_4, L_5\}$, with a weighted maximum cut value of 0.17. Based on (29), these graph cuts generate lower bounds for $\lambda_1$ of 0.30 and 0.096, respectively. By combining the upper and lower bounds, we obtain intervals for $\lambda_1$ of $[0.30, 0.92]$ and $[0.096, 0.19]$, respectively. Recall that a large spectral radius corresponds to slow convergence in the dual iteration [cf. (28)]; therefore, these bounds guarantee that the dual iteration for the network in Fig. 2(a), which is less congested, converges faster than for the one in Fig. 1(a). Numerical results suggest the actual largest eigenvalues are 0.47 and 0.12, respectively, which confirm the prediction.

B. Convergence in Primal Iterations

We next present our convergence analysis for the primal sequence $\{x^k\}$, generated by the inexact Newton method (18). For the $k$th iteration, we define the function $\hat{f}_k : \mathbb{R} \rightarrow \mathbb{R}$ as
\[ \hat{f}_k(t) = f(x^k + t\Delta x^k). \]
which is self-concordant, because the objective function \( f \) is self-concordant. Note that the value \( f_k(0) \) and \( f_k(d^k) \) are the objective function values at \( x^k \) and \( x^{k+1} \), respectively. Therefore, \( f_k(d^k) - f_k(0) \) measures the decrease in the objective function value at the \( k \)th iteration. We will refer to the function \( f_k \) as the objective function along the Newton direction.

Before proceeding further, we first introduce some properties of self-concordant functions and the Newton decrement, which will be used in our convergence analysis. We use the same notation in these lemmas as in (4)-(6) since these relations will be used in the convergence analysis of the inexact Newton method applied to problem (4).

1) Preliminaries: Using the definition of a self-concordant function, we have the following result (see [7] for the proof).

**Lemma V.2:** Let \( f : \mathbb{R} \rightarrow \mathbb{R} \) be a self-concordant function. Then, for all \( t \geq 0 \) in the domain of the function \( f \) with \( t f''(0)^{1/2} < 1 \), the following inequality holds:

\[
\tilde{f}(t) \leq f(0) + t f'(0) - t f''(0) \frac{\tilde{z}}{2} - \log \left( 1 - t f''(0) \frac{\tilde{z}}{2} \right). \tag{31}
\]

We will use the preceding lemma to prove a key relation in analyzing convergence properties of our algorithm [see Lemma V.8]. The next lemma will be used to relate the weighted norms of a vector \( z \), with weights \( \nabla^2 f(x) \) and \( \nabla^2 f(y) \), for some \( x \) and \( y \). This lemma plays an essential role in establishing properties for the Newton decrement. (See [15], [24], and [25] for more details.)

**Lemma V.3:** Let \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) be a self-concordant function. Suppose vectors \( x \) and \( y \) are in the domain of \( f \) and \( \tilde{\lambda} = ((x - y) \nabla^2 f(x)(x - y))^{1/2} < 1 \), then for any \( z \in \mathbb{R}^n \), the following inequality holds:

\[
(1 - \tilde{\lambda}^2) z' \nabla^2 f(x) z \leq \frac{1}{(1 - \tilde{\lambda})^2} z' \nabla^2 f(y) z. \tag{32}
\]

The next two lemmas establish properties of the Newton decrement generated by the equality-constrained Newton method. The first lemma extends results in [15] and [25] to allow inexactness in the Newton direction and reflects the effect of the error in the current step on the Newton decrement in the next step.⁹

**Lemma V.4:** Let \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) be a self-concordant function. Consider solving the equality-constrained optimization problem

\[
\text{minimize } f(x) \quad \text{subject to } Ax = c \tag{33}
\]

using an (unconstrained) Newton method. Let \( \Delta x \) be the exact Newton direction at \( x \) (i.e., \( \Delta x = -\nabla^2 f(x)^{-1} \nabla f(x) \)). Let \( \lambda(x) \) be the exact Newton decrement (i.e., \( \lambda(x) = \sqrt{(\Delta x)' \nabla^2 f(x) \Delta x} \)). Let \( f^* \) denote the optimal value of problem (35). If \( \lambda(x) \leq 0.68 \), then we have

\[
f^* \geq F(x) - \lambda(x)^2. \tag{36}
\]

Using the same elimination technique and isomorphism established for Lemma V.4, the next result follows immediately.

**Lemma V.5:** Let \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) be a self-concordant function. Consider solving the equality-constrained optimization problem

\[
\text{minimize } f(x) \quad \text{subject to } Ax = c \tag{37}
\]

using a constrained Newton method with feasible initialization. Let \( \Delta x \) be the exact (primal) Newton direction at \( x \), that is, \( \Delta x \) solves the system

\[
\begin{pmatrix}
\nabla^2 f(x) & A' \\
A & 0
\end{pmatrix}
\begin{pmatrix}
\Delta x \\
w
\end{pmatrix}
= \begin{pmatrix}
\nabla f(x) \\
0
\end{pmatrix}. \tag{34}
\]

Let \( \lambda(x) \) be the exact Newton decrement (i.e., \( \lambda(x) = \sqrt{(\Delta x)' \nabla^2 f(x) \Delta x} \)). Let \( f^* \) denote the optimal value of problem (37). If \( \lambda(x) \leq 0.68 \), then we have

\[
f^* \geq f(x) - \lambda(x)^2. \tag{38}
\]

Note that the relation on the optimality gap in the preceding lemma holds when the exact Newton decrement is sufficiently small (provided by the numerical bound 0.68, see [7]). We will use these lemmas in the subsequent sections for the convergence rate analysis of the distributed inexact Newton method applied to problem (4). Our analysis is comprised of two parts: The first part is the **damped convergent phase**, in which we provide a lower bound on the improvement in the objective function value at each step by a constant. The second part is the **quadratically convergent phase**, in which the optimality gap in the objective function value diminishes quadratically to an error level.

2) Basic Relations: We first introduce some key relations, which provides a bound on the error in the Newton direction.
computation. This will be used for both phases of the convergence analysis.

**Lemma V.7:** Let \( \{x^k\} \) be the primal sequence generated by the inexact Newton method (18). Let \( \lambda(x^k) \) be the exact Newton decrement at \( x^k \) [cf. (20)]. For all \( k \), we have

\[
\langle \gamma_k \rangle \nabla^2 f(x^k) \Delta x^k \leq p \lambda(x^k)^2 + \lambda(x^k) \sqrt{\epsilon},
\]

where \( \gamma_k \), \( p \), and \( \epsilon \) are the scalars defined in Assumption 2.

Using the preceding lemma, the following basic relation can be established, which will be used to measure the improvement in the objective function value.

**Lemma V.8:** Let \( \{x^k\} \) be the primal sequence generated by the inexact Newton method (18). Let \( f^k \) be the objective function along the Newton direction and \( \lambda(x^k) \) be the exact Newton decrement [cf. (30) and (20)] at \( x^k \), respectively. For all \( k \) with \( 0 \leq t < 1/\lambda(x^k) \), we have

\[
\tilde{f}_k(t) \leq \tilde{f}_k(0) - t(1-p)\lambda(x^k)^2 - \frac{(1-\sqrt{\epsilon})t\lambda(x^k) - \log \left(1 - t\lambda(x^k)\right)}{(1/2 - p - \alpha(1+p))},
\]

(39)

where \( p \) and \( \epsilon \) are the non-negative scalars defined in Assumption 2.

The preceding lemma shows that a careful choice of the stepsize \( t \) can guarantee a constant lower bound on the improvement in the objective function value at each iteration. We present the convergence properties of our algorithm in the following two sections.

3) **Damped Convergent Phase:** In this section, we consider the case when \( b^k \geq V \) and stepsize \( d^k = t/\sqrt{b^k + 1} \) [cf. (21)]. We will provide a constant lower bound on the improvement in the objective function value in this case. To this end, we first establish the improvement bound for the exact stepsize choice of \( t = 1/\lambda(x^k) + 1 \).

**Theorem V.9:** Let \( \{x^k\} \) be the primal sequence generated by the inexact Newton method (18). Let \( f^k \) be the objective function along the Newton direction and \( \lambda(x^k) \) be the exact Newton decrement at \( x^k \) [cf. (30) and (20)]. Consider the scalars \( p \) and \( \epsilon \) defined in Assumption 2 and assume that \( 0 < p < 0 \), \( \epsilon \leq 1/2 \), and \( \epsilon < (0.5 - p)/(2Vb - V + b - 1)/b^2 \), where \( b \) is the constant used in the stepsize rule [cf. (21)]. For \( b^k \geq V \) and \( t = 1/\lambda(x^k) + 1 \), there exists a scalar \( \alpha > 0 \) such that

\[
\tilde{f}_k(t) - \tilde{f}_k(0) \leq -\alpha(1 + p) \frac{(2Vb - V + b - 1)^2}{(1 + 2Vb - V + b - 1)^2},
\]

(40)

**Proof:** For notational simplicity, let \( y = \lambda(x^k) \) in this proof. We will show that for any positive scalar \( \alpha \) with \( 0 < \alpha < (1/2) - p - \sqrt{\epsilon}/(2Vb - V + b - 1)/b^2 \), (40) holds. Note that such \( \alpha \) exists since \( \epsilon < (0.5 - p)/(2Vb - V + b - 1)/b^2 \).

By Assumption 3, we have for \( b^k \geq V \)

\[
y > \theta^k - \frac{1}{b} \implies \frac{1}{b} > \frac{1}{b} - 1 \implies V - \frac{1}{b} - 1 \implies \frac{2Vb - V + b - 1}{b}.
\]

(41)

Using \( b^k > ((V + 1)/(2V + 1)) \), we have \( y \geq V - ((1/b) - 1)/(1 + V) > 0 \), which implies \( 2Vb - V + b - 1 > 0 \). Together with \( 0 < \alpha \leq ((1/2) - p - \sqrt{\epsilon}/(2Vb - V + b - 1))/b \), this yields

\[
\sqrt{\epsilon} \leq \frac{2Vb - V + b - 1}{b} \left( \frac{1}{2} - p - \alpha(1 + p) \right).
\]

Combining the previous two relations, we obtain

\[
\sqrt{\epsilon} \leq \frac{1}{(1/2) - p - \alpha(1 + p)}.
\]

From (41), we have \( y > 0 \). We can therefore multiply by \( y \) and divide by \( 1 + y \) both sides of the previous inequality to obtain

\[
\frac{1 - p}{1 + y} y^2 - \frac{1 - \sqrt{\epsilon}}{1 + y} y^2 - y^2 \leq -\alpha \frac{(1 + p)y^2}{1 + y}.
\]

Using second-order Taylor expansion on \( \log(1 + y) \), we have

\[
\log(1 + y) \leq y - \frac{y^2}{2(1 + y)}.
\]

Using this relation in (42) yields

\[
\frac{1 - p}{1 + y} y^2 - \frac{1 - \sqrt{\epsilon}}{1 + y} y^2 - \log(1 + y) \leq -\alpha \frac{(1 + p)y^2}{1 + y}.
\]

Substituting the value of \( t = 1/(y + 1) \), the above relation can be rewritten as

\[
-(1 - p)^2y^2 - (1 - \sqrt{\epsilon})ty^2 - (1 - ty^2) \leq -\alpha \frac{(1 + p)y^2}{1 + y}.
\]

Using (39) from Lemma V.8 and the definition of \( y = \lambda(x^k) \), we obtain

\[
\tilde{f}_k(t) - \tilde{f}_k(0) \leq -\alpha(1 + p) \frac{y^2}{y + 1}.
\]

Observe that the function \( h(y) = y^2/(y + 1) \) is monotonically increasing in \( y \), and for \( b^k \geq V \) by (41), we have \( y \geq (2Vb - V + b - 1)/b \). Therefore

\[
-\alpha(1 + p) \frac{y^2}{y + 1} \leq -\alpha \frac{(2Vb - V + b - 1)^3}{(1 + 2Vb - V + b - 1)^2}.
\]

Combining the preceding two relations completes the proof. ■

Note that our algorithm uses the stepsize \( d^k = 1/(\lambda(x^k) + 1) \) in the damped convergent phase, which is an approximation to the stepsize \( t = 1/\lambda(x^k) + 1 \) used in the previous theorem. The error between the two is bounded by (25) as shown in Lemma IV.4. We next show that with this error in the stepsize computation, the improvement in the objective function value in the inexact algorithm is still lower bounded at each iteration.
Let $\beta = d^k/t$, where $t = 1/(\hat{\lambda}(x_k) + 1)$. By the convexity of $f$, we have $f(x^k + \beta \Delta x^k) - f(x^k) \leq \beta f(x^k + \Delta x^k) + (1 - \beta) f(x^k)$ for all $\beta \in [0, 1]$. Therefore, the objective function value improvement is bounded by $f(x^k + \Delta x^k) - f(x^k) \leq \beta f(x^k + \Delta x^k) + (1 - \beta) f(x^k)$.

Hence, in the damped convergent phase, we can guarantee a lower bound on the objective function value improvement at each iteration. This bound is monotone in $\beta$ (i.e., the closer the scalar $\beta$ is to 1, the faster the objective function value improves); however, this also requires the error in the inexact Newton decrement calculation to diminish to 0 [cf. Assumption 3].

4) Quadratically Convergent Phase: In this phase, there exists $\bar{k}$ with $\theta^\bar{k} < V$ and the stepsize choice is $d^k = 1$ for all $k > \bar{k}$. We show that the optimality gap in the primal objective function value diminishes quadratically to a neighborhood of an optimal solution. We proceed by first establishing the following lemma for relating the exact and the inexact Newton decrements.

5) Lemma V.10: Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (18) and $\hat{\lambda}(x_k)$ be the exact and inexact Newton decrements at $x^k$ [cf. (19) and (20)]. Let $p$ and $\epsilon$ be the non-negative scalars defined in Assumption 2, we have

\begin{equation}
(1-p)\hat{\lambda}(x_k) - \sqrt{\epsilon} \leq \lambda(x_k) \leq (1-p)\hat{\lambda}(x_k) + \sqrt{\epsilon}.
\end{equation}

Before proceeding to establish quadratic convergence in terms of the primal iterations to an error neighborhood of the optimal solution, we need to impose the following bound on the errors in our algorithm in this phase. Recall that $\bar{k}$ is an index such that $\hat{\lambda}(x_k) < V$ and $d^k = 1$ for all $k \geq \bar{k}$.

Assumption 4: Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (18) and $\hat{\lambda}(x_k)$ be the exact and inexact Newton decrements at $x^k$ [cf. (19) and (20)]. Letting $p$ and $\epsilon$ be the non-negative scalars defined in Assumption 2, we have

\begin{equation}
(1-p)\hat{\lambda}(x_k) - \sqrt{\epsilon} \leq \lambda(x_k) \leq (1-p)\hat{\lambda}(x_k) + \sqrt{\epsilon}.
\end{equation}

Letting $p$ and $\epsilon$ be the non-negative scalars defined in Assumption 2, we have

\begin{equation}
(1-p)\hat{\lambda}(x_k) - \sqrt{\epsilon} \leq \lambda(x_k) \leq (1-p)\hat{\lambda}(x_k) + \sqrt{\epsilon}.
\end{equation}

Before proceeding to establish quadratic convergence in terms of the primal iterations to an error neighborhood of the optimal solution, we need to impose the following bound on the errors in our algorithm in this phase. Recall that $\bar{k}$ is an index such that $\hat{\lambda}(x_k) < V$ and $d^k = 1$ for all $k \geq \bar{k}$.

Assumption 4: Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (18). Let $\phi$ be a positive scalar with $\phi \leq 0.267$. Let $\xi$ and $\epsilon$ be non-negative scalars defined in terms of $\phi$ as $\xi = (\phi p + \sqrt{\epsilon})/(1 - p - \phi - \sqrt{\epsilon})$. By the fact that $\hat{\lambda}(x_k) \leq V$, we have $\lambda(x_k) < V$ and $\hat{\lambda}(x_k) < V$. The above relation implies $\hat{\lambda}(x_k) \leq 1/(1 - \hat{\lambda}(x_k))$.}

where $\xi$ and $v$ are the scalars defined in Assumption 4 and $p$ and $\epsilon$ are defined as in Assumption 2.

Proof: Given $\lambda(x_k) < 1$, we can apply Lemma V.4 by letting $z = \Delta x^{k+1}$. We then have

\begin{equation}
\lambda(x^{k+1})^2 = (\Delta x^{k+1})^2 f(x + \Delta x^{k+1}) \leq \frac{\hat{\lambda}(x_k)^2}{1 - \hat{\lambda}(x_k)} \sqrt{\Delta x^{k+1}}^2 f(x) \Delta x^{k+1} + \frac{\lambda(x_k)^2}{1 - \lambda(x_k)} \sqrt{\Delta x^{k+1}}^2 f(x) \Delta x^{k+1} + \sqrt{(\gamma_k)^2 f(x) \Delta x^{k+1}}^2 f(x) \Delta x^{k+1}
\end{equation}

where the last inequality follows from the generalized Cauchy–Schwarz inequality. Using Assumption 2, the above relation implies $\lambda(x^{k+1})^2 \leq \left(\frac{\lambda(x_k)^2}{1 - \lambda(x_k)} + \sqrt{p^2 \lambda(x_k)^2 + \epsilon} \right) \sqrt{(\Delta x^{k+1})^2 f(x) \Delta x^{k+1}}$. By the fact that $\hat{\lambda}(x_k) < \theta^k$, we can apply Lemma V.3 and obtain

\begin{equation}
\lambda(x^{k+1})^2 \leq \frac{1}{1 - \lambda(x_k)^2} \left(\frac{\lambda(x_k)^2}{1 - \lambda(x_k)} + \frac{p^2 \lambda(x_k)^2 + \epsilon}{1 - \lambda(x_k)} \right) \sqrt{(\Delta x^{k+1})^2 f(x + \Delta x^{k+1})} \Delta x^{k+1}
\end{equation}

Note that once the condition $\theta^k < V$ is satisfied, in all of the following iterations, we have stepsize $d^k = 1$ and no longer need to compute $\theta^k$.
By dividing the last line by $\lambda(x^{k+1})$, this yields

$$
\lambda(x^{k+1}) \leq \frac{\lambda(x^*)^2}{(1 - \lambda(x^*))^2} + \frac{\sqrt{p^2\lambda(x^*)^2 + \epsilon}}{1 - \lambda(x^*)}.
$$

From (44), we have $\tilde{\lambda}(x^{k+1}) \leq ((\lambda(x^*) + \sqrt{e})/(1 - p))$. Therefore, the above relation implies

$$
\lambda(x^{k+1}) \leq \frac{\lambda(x^*) + \sqrt{e}}{1 - p - \lambda(x^*) - \sqrt{e}}.
$$

By (51), we have $\lambda(x^{k+1}) \leq \phi$ and, therefore, the above relation can be relaxed to $\lambda(x^{k+1}) \leq (\lambda(x^*)/(1 - p - \phi - \sqrt{e}))^2 + p\lambda(x^*) + \sqrt{e}$. Hence, by definition of $\xi$ and $v$, we have

$$
\lambda(x^{k+1}) \leq v\lambda(x^*)^2 + \xi.
$$

Thus, we have established the desired relation.

In the next theorem, building upon the preceding lemma, we apply relation (38) to bound the optimality gap in our algorithm (i.e., $f(x^k) - f^*$), using the exact Newton decrement. We show that under the above assumption, the objective function value $f(x^k)$ generated by our algorithm converges quadratically in terms of the primal iterations to an explicitly characterized error neighborhood of the optimal value $f^*$. Theorem V.12: Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (18) and $\lambda(x^k)$, $\tilde{\lambda}(x^k)$ be the exact and inexact Newton decrements at $x^k$ [cf. (19) and (20)]. Let $f(x^k)$ be the corresponding objective function value at the $k$th iteration and $f^*$ denote the optimal objective function value for problem (4). Let Assumption 4 hold, and $\xi$ and $v$ be the scalars defined in Assumption 4. Assume that for some $\delta \in [0, 1/2)$

$$
\xi + v\xi \leq \frac{\delta}{2v}.
$$

Then for all $m \geq 1$, we have

$$
\lambda(x^{k+m}) \leq \frac{1}{22^m} + \xi + \frac{\delta 2^{2m-1} - 1}{22^m}
$$

and

$$
\limsup_{m \to \infty} f(x^{k+m}) - f^* \leq \xi + \frac{\delta}{2v},
$$

where $k$ is the iteration index with $\theta^k < 1$.

Proof: We prove (50) by induction. First, for $m = 1$, from Assumption 3, we have $\lambda(x^{k+1}) \leq \theta^k + \tau^k$. Relation (45) implies $\theta^k + \tau^k < \phi < 1$; hence, we have $\lambda(x^k) < 1$ and we can apply Lemma V.11 and obtain

$$
\lambda(x^{k+1}) \leq v\lambda(x^k)^2 + \xi.
$$

By Assumption 4 and (44), we have

$$
\lambda(x^*) \leq (1 + p)\{\theta^k + \tau^k\} + \sqrt{e} \leq \phi.
$$

The above two relations imply

$$
\lambda(x^{k+1}) \leq v\phi^2 + \xi.
$$

The right-hand side is monotonically increasing in $\phi$. Since $\phi \leq 0.68$, we have with (46), $\lambda(x^{k+1}) \leq 0.68$. By relation (48), we obtain $(1 - p - \phi - \sqrt{e})^2 \geq 4\phi^2$. Using the definition of $v$ (i.e., $v = 1/(1 - p - \phi - \sqrt{e})$), the above relation implies $v\phi^2 \leq (1/4v)$. Hence, we have

$$
\lambda(x^{k+1}) \leq \frac{1}{4v} + \xi.
$$

This establishes relation (50) for $m = 1$.

We next assume that (50) holds and $\lambda(x^{k+m}) \leq 0.68$ for some $m > 0$, and show that these also hold for $m + 1$. From (44) and (47), we have

$$
\tilde{\lambda}(x^{k+m+1}) \leq \frac{\lambda(x^{k+m}) + \sqrt{e}}{1 - p} \leq \frac{0.68 + \sqrt{e}}{1 - p} \leq 1.
$$

where, in the second inequality, we used the inductive hypothesis that $\lambda(x^{k+m}) \leq 0.68$. Hence, we can apply (49) and obtain

$$
\lambda(x^{k+m+1}) \leq v\lambda(x^{k+m})^2 + \xi.
$$

We use (46) and $\lambda(x^{k+m}) \leq 0.68$ once more and have $\lambda(x^{k+m+1}) \leq 0.68$. From our inductive hypothesis that (50) holds for $m$, the above relation also implies

$$
\lambda(x^{k+m+1}) \leq \frac{1}{22^m} + \xi + \frac{\delta 2^{2m-1} - 1}{22^m} + \xi
$$

which completes the induction and the proof of relation (50).

The induction proof above suggests that the condition $\lambda(x^{k+m}) \leq 0.68$ holds for all $m > 0$, we can therefore apply Lemma V.6, and obtain an upper bound on the optimality gap as follows:

$$
f(x^{k+m}) - f^* \leq \left(\lambda(x^{k+m})\right)^2 \leq \lambda(x^{k+m}).
$$

Combining this with (50), we obtain

$$
f(x^{k+m}) - f^* \leq \frac{1}{22^m} + \xi + \frac{\delta 2^{2m-1} - 1}{22^m}.
$$

Taking limit superior on both sides of the preceding relation establishes the final result.

\[11\]Note that we do not need monotonicity in $\tilde{\lambda}(x^*)$. Instead, the error-level assumption from relation (47) enables us to use Lemma V.11 to establish the quadratic rate of convergence.
The above theorem shows that the objective function value \( f(x^k) \) generated by our algorithm converges in terms of the primal iterations quadratically to a neighborhood of the optimal value \( f^* \), with the neighborhood of size \( \xi + (\mu/2v) \), where \( \xi = (\phi p + \sqrt{v})/(1 - p - \phi - \sqrt{v}) + (2\phi\sqrt{v} + 1)/(1 - p - \phi + \sqrt{v}) \), \( v = 1/((1 - p - \phi - \sqrt{v})^2) \), and the condition \( \xi + \sqrt{\xi} \leq (\mu/2v) \) is satisfied. Note that with the exact Newton algorithm, we have \( p - \epsilon = 0 \), which implies \( \xi = 0 \) and we can choose \( \epsilon = 0 \) which, in turn, leads to the size of the error neighborhood being 0. This confirms the fact that the exact Newton algorithm converges quadratically to the optimal objective function value.

Note that the analysis is independent of how the dual variables are obtained. We can consider any algorithm for problem (4), where the update rule is given as (18) with stepsize \( d^k \) defined as in (21) and an inexact Newton direction \( \Delta \hat{x}^k \) defined as an inexact solution to the system (7). If Assumptions 1–4 are satisfied, then the preceding analysis can be applied and the sequence of the objective function value generated by the algorithm converges quadratically to an error neighborhood of the optimal value.

C. Convergence With Respect to the Design Parameter \( \mu \)

In the preceding development, we have restricted our attention to developing an algorithm for a given logarithmic barrier coefficient \( \mu \). We next study the convergence properties of the optimal objective function value as a function of \( \mu \) and develop a method that enables us to bound the error introduced by the logarithmic barrier functions to be arbitrarily small. We utilize the following result from [25].

**Lemma V.13:** Let \( G \) be the non-negative orthant in \( \mathbb{R}^n \), and function \( g : \mathbb{R}^n \to \mathbb{R} \) be a logarithmic barrier for \( G \) (i.e., \( g(x) = -\sum_{i=1}^n \log(x_i) \)). Then, for any \( x, y \) in interior of \( G \), we have \( (y - x)^T \nabla g(x) \leq 1.12 \)

Using this lemma and an argument similar to that in [25], we can establish the following result, which bounds the suboptimality as a function of \( \mu \).

**Theorem V.14:** Given \( \mu > 0 \), let \( x(\mu) \) denote the optimal solution of problem (4) and \( h(x(\mu)) = \sum_{i=1}^n -U_i(x_i(\mu)) \).

Then, the following relation holds:

\[ h(x(\mu)) \geq h(x^* + \mu \nabla h(x^*)) - (x^* - x(\mu))^T \nabla h(x^*) \geq \mu \]

where \( h(x^*) \) is the value obtained from our algorithm for problem (4), and \( h^* \) is the optimal objective function value for problem (2). We achieve the above bound by implementing our algorithm twice. The first time involves running the algorithm for problem (4) with some arbitrary \( \mu \). This leads to a sequence of \( x^k \) converging to some \( x(\mu) \). Let \( h(x(\mu)) = \sum_{i=1}^n -U_i(x_i(\mu)) \). By Theorem V.14, we have

\[ h(x(\mu)) \geq h^* - \mu \leq h^* \tag{53} \]

Let scalar \( M \) be such that \( M = \{a[h(x^* + \mu \nabla h(x^*)) - h(x^* - x(\mu))] \}^{-1} \) and implement the algorithm one more time for problem (4), with \( \mu = 1 \) and the objective function multiplied by \( M \) (i.e., the new objective is to minimize \( -M \sum_{i=1}^n U_i(x_i(\mu)) - \mu \sum_{i=1}^n \log(x_i) \)), subject to link capacity constraints. We obtain a sequence of \( \tilde{x}_i(\mu) \) which converges to some \( \hat{x}(\mu) \). Denote the objective function value as \( h(\hat{x}(\mu)) \), then by applying the preceding theorem one more time, we have \( M \geq \mu \leq h(\hat{x}(\mu)) \). In view of relation (52), this establishes the desired result (i.e., \( h(x(\mu)) \leq h^* \)).

C. Convergence With Respect to the Design Parameter \( \mu \)

In the preceding development, we have restricted our attention to developing an algorithm for a given logarithmic barrier coefficient \( \mu \). We next study the convergence properties of the optimal objective function value as a function of \( \mu \) and develop a method that enables us to bound the error introduced by the logarithmic barrier functions to be arbitrarily small. We utilize the following result from [25].

**Lemma V.13:** Let \( G \) be the non-negative orthant in \( \mathbb{R}^n \), and function \( g : \mathbb{R}^n \to \mathbb{R} \) be a logarithmic barrier for \( G \) (i.e., \( g(x) = -\sum_{i=1}^n \log(x_i) \)). Then, for any \( x, y \) in interior of \( G \), we have \( (y - x)^T \nabla g(x) \leq 1.12 \)

Using this lemma and an argument similar to that in [25], we can establish the following result, which bounds the suboptimality as a function of \( \mu \).

**Theorem V.14:** Given \( \mu > 0 \), let \( x(\mu) \) denote the optimal solution of problem (4) and \( h(x(\mu)) = \sum_{i=1}^n -U_i(x_i(\mu)) \).

Then, the following relation holds:

\[ h(x(\mu)) \geq h(x^* + \mu \nabla h(x^*)) - (x^* - x(\mu))^T \nabla h(x^*) \geq \mu \]

where \( h(x^*) \) is the value obtained from our algorithm for problem (4), and \( h^* \) is the optimal objective function value for problem (2). We achieve the above bound by implementing our algorithm twice. The first time involves running the algorithm for problem (4) with some arbitrary \( \mu \). This leads to a sequence of \( x^k \) converging to some \( x(\mu) \). Let \( h(x(\mu)) = \sum_{i=1}^n -U_i(x_i(\mu)) \). By Theorem V.14, we have

\[ h(x(\mu)) \leq h^* \tag{54} \]

VI. Conclusion

This paper presents a convergence analysis for the distributed Newton-type algorithm for network utility maximization problems proposed in Part I of this paper, which uses an information-exchange mechanism similar to that involved in first-order methods applied to this problem. We utilize the property of self-concordant functions and show that even when the Newton direction and stepsize are computed with some error, the method converges globally and achieves a local superlinear convergence rate in terms of primal iterations to an
error neighborhood, the size of which can be specified explicitly using the error tolerance level and the parameters of the algorithm. Possible future directions include a more detailed analysis of the relationship between the rate of convergence of the dual iterations and the underlying topology of the network and extension to non-self-concordant objective functions.

REFERENCES

[1] S. Athuraliya and S. Low, “Optimization flow control with Newton-like algorithms,” J. Telecommun. Syst., vol. 15, pp. 345–358, 2000.

[2] A. Berman and R. J. Plemmons, Nonnegative Matrices in the Mathematical Sciences. London, U.K.: Academic Press, 1979.

[3] D. P. Bertsekas, Convex Optimization Theory. Belmont, MA: Athena Scientific, 2009.

[4] D. P. Bertsekas, A. Nedic, and A. Ozdaglar, Convex Analysis and Optimization. Belmont, MA: Athena Scientific, 2003.

[5] D. Bickson, Y. Tock, A. Zymnis, S. Boyd, and D. Dolev, “Distributed large scale network utility maximization,” in Proc. IEEE Int. Conf. Commun., 2008, pp. 2188–2192.

[6] N. Biggs, Algebraic Graph Theory, 2nd ed. Cambridge, UK.: Cambridge University Press, 1993.

[7] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge University Press, 2004.

[8] M. Chiang, S. Low, A. R. Calderbank, and J. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” Proc. IEEE, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[9] F. R. K. Chung, Spectral Graph Theory (CBMS Regional Conference Series in Mathematics). Providence, RI: American Mathematical Society, 1997, vol. 92.

[10] R. Cottle, J. Pang, and R. Stone, The Linear Complementarity Problem. London, U.K.: Academic Press, 1992.

[11] R. Dembo, S. Eisenstat, and T. Steihaug, “Inexact Newton methods,” SIAM J. Numerical Anal., vol. 19, pp. 400–408, 1982.

[12] A. V. Fiacco and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Philadelphia, PA: SIAM, 1990.

[13] R. Horn and C. R. Johnson, Matrix Analysis. New York: Cambridge University Press, 1985.

[14] A. Jadbabaie, A. Ozdaglar, and M. Zargham, “A distributed Newton method for network optimization,” in Proc. CDC, 2009, pp. 2736–2741.

[15] F. Jarre, “Interior-point methods for convex programming,” Appl. Math. Optimiz., vol. 26, pp. 287–311, 1992.

[16] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations. Philadelphia, PA: SIAM, 1995.

[17] F. P. Kelly, “Charging and rate control for elastic traffic,” Eur. Trans. Telecommun., vol. 8, pp. 33–37, 1997.

[18] F. P. Kelly, A. K. Maulloo, and D. K. Tan, “Rate control for communication networks: Shadow prices, proportional fairness, and stability,” J. Oper. Res. Soc., vol. 49, pp. 237–252, 1998.

[19] D. C. Lay, Linear Algebra and Its Applications, 3rd ed. Reading, MA: Addison-Wesley, 2006.

[20] S. H. Low and D. E. Lapsley, “Optimization flow control: I: Basic algorithm and convergence,” IEEE/ACM Trans. Netw., vol. 7, no. 6, pp. 861–874, Dec. 1999.

[21] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” IEEE/ACM Trans. Netw., vol. 8, no. 5, pp. 556–567, 2000.

[22] B. Mohar, G. Hahn, and G. Sabidussi, Eds., “Some applications of Laplace eigenvalues of graphs,” Graph Symmetry: Algebraic Methods and Applications, ser. NATO ASI ser. C, vol. 497, pp. 227–275, 1997.

[23] A. Nedic and A. Ozdaglar, “Cooperative distributed multi-agent optimization,” in Convex Optimization in Signal Processing and Communication Networks, Y. Eldar and D. Palomar, Eds. Cambridge, UK.: Cambridge University Press, 2009.

[24] Y. Nesterov, Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization). Norwell, MA: Kluwer, 2004.

[25] Y. Nesterov and A. Nemirovski, Interior-Point Polynomial Algorithms in Convex Programming. Philadelphia, PA: SIAM, 2001.

[26] R. Srikant, The Mathematics of Internet congestion Control (Systems and Control: Foundations and Applications). Boston, MA: Birkhauser, 2004.

[27] R. Varaiya, Matrix Iterative Analysis. Upper Saddle River, NJ: Prentice-Hall, 1965.

[28] R. Varga, Geršgorin and His Circles, ser. Springer Ser. Comput. Math. New York: Springer, 2004.

[29] E. Wei, A. Ozdaglar, and A. Jadbabaie, A distributed Newton method for network utility maximization I, algorithm, LIDS Rep. 2832, 2011. [Online]. Available: http://www.mit.edu/~erminwei/Publication.html

[30] E. Wei, A. Ozdaglar, and A. Jadbabaie, A distributed Newton method for network utility maximization II, convergence, LIDS Rep. 2870, 2011. [Online]. Available: http://www.mit.edu/~erminwei/Publication.html

[31] E. Wei, M. Zargham, A. Ozdaglar, and A. Jadbabaie, “On global and local convergence of the distributed Newton method for network utility maximization,” in Proc. 50th IEEE Conf. Dec. Control Eur. Control Conf., 2011, pp. 6612–6617.

Asuman Ozdaglar (M’02) received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, in 1996, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1998 and 2003, respectively.

Currently, she is a Professor in the Electrical Engineering and Computer Science Department, MIT. She is also a member of the Laboratory for Information and Decision Systems and the Operations Research Center. She was an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL in 2012. She is currently the area co-editor for a new area of the journal Operations Research, entitled “Games, Information and Networks” She is the co-author of the book Convex Analysis and Optimization (Athena Scientific, 2003). Her research expertise includes optimization theory, with an emphasis on nonlinear programming and convex analysis, game theory, with applications in communication, social, and economic networks, distributed optimization and control, and network analysis with a special emphasis on contagious processes, systemic risk, and control.

Prof. Ozdaglar is the recipient of a Microsoft fellowship, the MIT Graduate Student Council Teaching award, the National Science Foundation Career award, the 2008 Donald P. Eckman Award of the American Automatic Control Council, the Clyde L. Klienerman Award, the NSF Career Award, and the Kavli Fellowship of the National Academy of Sciences, and the inaugural Steen and Renee Innovation Fellowship. She served on the Board of Governors of the Control System Society in 2010 and is currently the Chair of the Control System Society Technical Committee Networks and Communication Systems.

Ali Jadbabaie (SM’07) received the B.S. degree (Hons.) in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1995, the M.Sc. degree in electrical engineering from the University of New Mexico, Albuquerque in 1997, and the Ph.D. degree in control and dynamical systems from the California Institute of Technology (Caltech), Pasadena, in 2001.

After spending a year as a Postdoctoral Associate at Yale University, New Haven, CT, he joined the University of Pennsylvania (Penn), Philadelphia, PA, in 2002, where he is currently a Professor of Electrical and Systems Engineering with secondary appointments in the Department of Computer and Information Sciences and the Department of Operations and Information Management. He is the founding Co-director of the Singh Program in Networked and Social Systems (NETS), a new undergraduate degree program at Penn Engineering that blends operations research, economics, and computer science with decision theory and network science. His research interests are network science, control and coordination of multagent systems, and network economics.

Prof. Jadbabaie received the following awards: National Science Foundation Career Award, ONR YIP, as well as the O. Hugo Schuck Best Paper Award of the American Automatic Control Council, and the George Axelby Best Paper award of the IEEE Control Systems Society.