A Distributed Newton Method for Network Utility Maximization, II: Convergence*

Ermin Wei†, Asuman Ozdaglar†, and Ali Jadbabaie‡

May 11, 2011

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 slow rate of convergence. Recent work [25] 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.

*This work was supported by National Science Foundation under Career grant DMI-0545910, the DARPA ITMANET program, ONR MURI N000140810747 and AFOSR Complex Networks Program.

†Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology

‡Department of Electrical and Systems Engineering and GRASP Laboratory, University of Pennsylvania

1 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], [23], [7], [17], [19]). 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 ([16], [19], [21]). 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 [25]. 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 [4] and [13]. In [4], 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 [13] 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 [10], [15]). 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 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 the paper is organized as follows: Section 2 defines the problem formulation and related transformations. Section 3 describes the exact constrained primal-dual Newton method for this problem. Section 4 outlines the distributed inexact Newton-type algorithm developed in [25]. Section 5 contains the rate of convergence analysis for our algorithm. Section 6 contains our concluding remarks.

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 nonnegative real numbers, i.e., $\mathbb{R}_+ = [0, \infty)$. 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 kth element of a sequence. When $x_i \geq 0$ for all components i of a vector x, we write $x \geq 0$.

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 $|g'''(x)| \leq 2g''(x)^{\frac{3}{2}}$ 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, i.e., if the function $\tilde{g}(t) = g(x + tv)$ is self-concordant in t for all x and v. Operations that preserve self-concordance property include summing, scaling by a factor $\alpha \geq 1$, and composition with affine transformation (see [6] Chapter 9 for more details).

2 Network Utility Maximization Problem

We consider a network represented by a set $\mathcal{L} = \{1, ..., L\}$ of (directed) links of finite nonzero capacity given by $c = [c_l]_{l \in \mathcal{L}}$ and a set $\mathcal{S} = \{1, ..., S\}$ of sources, each of which transmits information along a predetermined route.[2] For each link l, let $S(l)$ denote the set of sources using it. For each source i, let $L(i)$ denote the set of links it uses. Let the nonnegative source rate vector be denoted by $s = [s_i]_{i \in \mathcal{S}}$. Let matrix R be the *routing matrix* of dimension $L \times S$, given by

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

For each i, we use $U_i : \mathbb{R}_+ \to \mathbb{R}$ to denote the utility function of source i. The *Network Utility Maximization (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

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{S} U_i(s_i) \\ \text{subject to} \quad & Rs \leq c, \quad s \geq 0. \end{aligned} \tag{2}$$

We adopt the following assumptions on the utility function.

[1]Self-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 $a > 0$, such that $|g'''(x)| \leq 2a^{-\frac{1}{2}} g''(x)^{\frac{3}{2}}$ for all x in its domain [22], [14]. Here we focus on the case $a = 1$ for notational simplification in the analysis.

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

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 functions $-U_i : \mathbb{R}_+ \to \mathbb{R}$ are self-concordant on the set of positive real numbers.

The self-concordance assumption is satisfied by standard utility functions considered in the literature, for instance logarithmic (i.e., weighted proportionally fair, see [23]) utility functions and concave quadratic 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), i.e., $h(x) = -\sum_{i=1}^{S} U_i(x_i)$, 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, i.e., there exists a feasible solution x with $x_i = \frac{\underline{c}}{S+1}$ for all $i \in \mathcal{S}$ with $\underline{c} > 0$.[4]

We reformulate the problem into one with only equality constraints by introducing nonnegative slack variables $[y_l]_{l\in\mathcal{L}}$, such that

$$\sum_{j=1}^{S} R_{lj} s_j + y_l = c_l \quad \text{for } l = 1, 2 \ldots L, \tag{3}$$

and using logarithmic barrier functions for the nonnegativity constraints (which can be done since the feasible set of (2) has a nonempty interior).[5] The new decision vector is $x = ([s_i]'_{i\in\mathcal{S}}, [y_l]'_{l\in\mathcal{L}})'$ and problem (2) can be rewritten as

$$\begin{aligned} \text{minimize} \quad & -\sum_{i=1}^{S} U_i(x_i) - \mu \sum_{i=1}^{S+L} \log(x_i) \\ \text{subject to} \quad & Ax = c, \end{aligned} \tag{4}$$

where A is the $L \times (S + L)$-dimensional matrix given by

$$A = [R \quad I(L)], \tag{5}$$

and μ is a nonnegative 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 μ.[6]

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 μ as $x(\mu)$. One can show that as the barrier function coefficient μ approaches 0, the optimal solution of problem (4) approaches that of problem (2), when the constraint set in (2) has a nonempty

---

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

[4] One possible value for $\underline{c}$ is $\underline{c} = \min_l\{c_l\}$.

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

[6] This problem has a feasible solution, hence f^* is upper bounded. Each of the variable x_i is upper bounded by $\bar{c}$, where $\bar{c} = \max_l\{c_l\}$, 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.

interior and is convex [2], [11]. 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 μ. 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 5.3, 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 $\frac{h(x(\mu))-h^*}{h^*} \leq a$ for any positive scalar a.

3 Exact Newton Method

For each fixed μ, 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 [3]). Moreover, since matrix A has full row rank, we can use a (feasible start) equality-constrained Newton method to solve problem (4)(see [6] Chapter 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.

3.1 Feasible Initialization

We initialize the algorithm with some feasible and strictly positive vector x^0. For example, one such initial vector is given by

$$x_i^0 = \frac{\underline{c}}{S+1} \quad \text{for } i = 1, 2 \ldots S, \tag{7}$$
$$x_{l+S}^0 = c_l - \sum_{j=1}^{S} R_{lj} \frac{\underline{c}}{S+1} \quad \text{for } l = 1, 2 \ldots L,$$

where c_l is the finite capacity for link l, $\underline{c}$ is the minimum (nonzero) link capacity, S is the total number of sources in the network, and R is routing matrix [cf. Eq. (1)].

3.2 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 $x^{k+1} = x^k + d^k \Delta x^k$, where d^k is a positive stepsize, Δx^k is the (primal) Newton direction given as

$$\Delta x^k = -H_k^{-1}\left(\nabla f(x^k) + A' w^k\right), \text{ and} \tag{8}$$
$$(A H_k^{-1} A') w^k = -A H_k^{-1} \nabla f(x^k), \tag{9}$$

where $w^k = [w_l^k]_{l \in \mathcal{L}}$ is the dual vector and the w_l^k are the dual variables for the link capacity constraints at primal iteration k. This system has a unique solution for all k. To see this, note that the matrix H_k is a diagonal matrix with entries

$$(H_k)_{ii} = \begin{cases} -\frac{\partial^2 U_i(x_i^k)}{\partial x_i^2} + \frac{\mu}{(x_i^k)^2} & 1 \leq i \leq S, \\ \frac{\mu}{(x_i^k)^2} & S+1 \leq i \leq S+L. \end{cases} \tag{10}$$

By Assumption 1, the functions U_i are strictly concave, which implies $\frac{\partial^2 U_i(x_i^k)}{\partial x_i^2} \leq 0$. Moreover, the primal vector x^k is bounded (since the method maintains feasibility) and, as we shall see in

Section 4.2, can be guaranteed to remain strictly positive by proper choice of stepsize. Therefore, the entries $(H_k)_{ii} > 0$ and are well-defined for all i, implying that the Hessian matrix H_k is invertible. Due to the structure of A [cf. Eq. (5)], the column span of A is the entire space $\mathbb{R}^L$, and hence the matrix $AH_k^{-1}A'$ is also invertible.[7] This shows that the preceding system of linear equations can be solved uniquely for all k.

The objective function f is separable in x_i, therefore given the vector w_l^k for l in $L(i)$, the Newton direction Δx_i^k can be computed by each source i using local information available to that source. However, the computation of the vector w^k at a given primal solution x^k cannot be implemented in a decentralized manner since the evaluation of the matrix inverse $(AH_k^{-1}A')^{-1}$ requires global information. This motivates using matrix splitting technique to compute the dual variables w^k in the inexact distributed Newton method developed in [25], which we briefly summarize in the following section.

4 Distributed Inexact Newton Method

This section describes the distributed Newton algorithm developed in [25]. We consider a broader class of algorithms which allow for errors in the stepsize computation. These algorithms use a distributed iterative scheme to compute the dual vector, which is then used to determine an inexact primal Newton direction that maintains primal feasibility. Section 4.1 summarizes the distributed finitely terminated dual vector computation procedure. Section 4.2 presents the distributed primal Newton direction computation and the stepsize rule, together with the bounds on the error level in the inexact algorithm.

4.1 Distributed Dual Variable Computation via Matrix Splitting

The computation of the dual vector w^k at a given primal solution x^k requires solving a linear system of equations [cf. Eq. (9)]. The dual variables can be computed using a distributed iterative scheme relying on novel ideas from matrix splitting (see [9] for a comprehensive review). We let D_k be a diagonal matrix with diagonal entries

$$(D_k)_{ll} = (AH_k^{-1}A')_{ll}, \tag{11}$$

B_k be a symmetric matrix given by

$$B_k = (AH_k^{-1}A') - D_k, \tag{12}$$

and $\bar{B}_k$ be a diagonal matrix with diagonal entries

$$(\bar{B}_k)_{ii} = \sum_{j=1}^{L} (B_k)_{ij}. \tag{13}$$

It was shown in [25] that we can use the matrix splitting

$$AH_k^{-1}A' = (\bar{B}_k + D_k) + (B_k - \bar{B}_k) \tag{14}$$

to compute the dual variables iteratively. We include the theorem statement here for completeness.

[7] If for some $x \in \mathbb{R}^L$, we have $AH_k^{-1}A'x = 0$, then $x'AH_k^{-1}A'x = \left|\left|H_k^{-\frac{1}{2}}A'x\right|\right|_2 = 0$, which implies $||A'x||_2 = 0$, because the matrix H is invertible. The rows of the matrix A' span $\mathbb{R}^L$, therefore we have $x = 0$. This shows that the matrix $AH_k^{-1}A'$ is invertible.

Theorem 4.1. For a given $k > 0$, let D_k, B_k, $\bar{B}_k$ be the matrices defined in Eqs. (11), (12) and (13). Let $w(0)$ be an arbitrary initial vector and consider the sequence $\{w(t)\}$ generated by the iteration

$$w(t+1) = (D_k + \bar{B}_k)^{-1}(\bar{B}_k - B_k)w(t) + (D_k + \bar{B}_k)^{-1}(-AH_k^{-1}\nabla f(x^k)), \tag{15}$$

for all $t \geq 0$. Then the spectral radius of the matrix $(D_k + \bar{B}_k)^{-1}(B_k - \bar{B}_k)$ is strictly bounded above by 1 and the sequence $\{w(t)\}$ converges as $t \to \infty$, and its limit is the solution to Eq. (9).

The results from [25] shows that iteration (15) can be implemented in a distributed way, where each source or link is viewed as a processor, information available at source i can be passed to the links it traverses, i.e., $l \in L(i)$, and information about the links along a route can be aggregated and sent back to the corresponding source using a feedback mechanism. The algorithm has comparable level of information exchange with the subgradient based algorithms applied to the NUM problem, see [25] for more implementation details.

4.2 Distributed Primal Direction Computation

Given the dual variables computed using the above iteration, the distributed Newton method computes the primal Newton direction in two stages to maintain feasibility. In the first stage, the first S components of $\Delta\tilde{x}^k$ are computed via Eq. (8) using the dual vector obtained from iteration (15). Then in the second stage, the last L components of $\Delta\tilde{x}^k$, corresponding to the slack variables, are solved explicitly by the links to guarantee the condition $A\Delta\tilde{x}^k = 0$ is satisfied. The feasibility correction is given by

$$\left(\Delta\tilde{x}^k\right)_{\{S+1\ldots S+L\}} = -R\left(\Delta\tilde{x}^k\right)_{\{1\ldots S\}}. \tag{16}$$

Starting from an initial feasible vector x^0, the initialization in Eq. (7) for instance, the distributed Newton algorithm generates the primal vectors x^k as follows:

$$x^{k+1} = x^k + d^k\Delta\tilde{x}^k, \tag{17}$$

where d^k is a positive stepsize, and $\Delta\tilde{x}^k$ 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 equations (8) as the *exact Newton direction*, denoted by Δx^k. The inexact Newton direction $\Delta\tilde{x}^k$ computed by our algorithm is a feasible estimate of Δx^k. At a given primal vector x^k, we define the *exact Newton decrement* $\lambda(x^k)$ as

$$\lambda(x^k) = \sqrt{(\Delta x^k)'\nabla^2 f(x^k)\Delta x^k}. \tag{18}$$

Similarly, the *inexact Newton decrement* $\tilde{\lambda}(x^k)$ is given by

$$\tilde{\lambda}(x^k) = \sqrt{(\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k}. \tag{19}$$

Note that both $\lambda(x^k)$ and $\tilde{\lambda}(x^k)$ are nonnegative 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 θ^k as its approximate value. One possible procedure with finite termination yielding $\theta^k = \tilde{\lambda}(x^k)$ is described in [25]. However, other estimates θ^k can be used, which can potentially be obtained by exploiting the diagonal structure of the Hessian matrix, writing the inexact Newton decrement as

$$\tilde{\lambda}(x^k) = \sqrt{\sum_{i\in\mathcal{L}\bigcup\mathcal{S}} (\Delta\tilde{x}^k)_i^2 (H_k)_{ii}} = \sqrt{(L+S)\bar{y}},$$

where $\bar{y} = \frac{1}{S+L} \sum_{i \in \mathcal{S} \bigcup \mathcal{L}} (\Delta \tilde{x}^k)_i^2 (H_k)_{ii}$ and using iterative consensus-type algorithms.

Given the scalar θ^k, an approximation to the inexact Newton decrement $\tilde{\lambda}(x^k)$, 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 [22], we have

$$d^k = \begin{cases} \frac{b}{\theta^k + 1} & \text{if} \quad \theta^k \geq V \text{ for all previous } k, \\ 1 & \text{otherwise,} \end{cases} \tag{20}$$

where $\frac{V+1}{2V+1} < b \leq 1$. The upper bound on V will be used in 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. Theorem 4.3]. The lower bound on b will be used to guarantee a lower bounded improvement in the damped convergent phase. The stepsize rule in [25] uses $\theta^k = \tilde{\lambda}(x^k)$ 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 iterative scheme (15); two-stage computation of an approximate primal direction to maintain feasibility; 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 Δx^k and $\Delta \tilde{x}^k$ denote the exact and inexact Newton directions at x^k, and γ^k denote the error in the Newton direction computation, i.e.,

$$\Delta x^k = \Delta \tilde{x}^k + \gamma^k. \tag{21}$$

For all k, γ^k satisfies

$$|(\gamma^k)' \nabla^2 f(x^k) \gamma^k| \leq p^2 (\Delta \tilde{x}^k)' \nabla^2 f(x^k) \Delta \tilde{x}^k + \epsilon. \tag{22}$$

for some positive scalars $p < 1$ and ϵ.

This assumption imposes a bound on the weighted norm of the Newton direction error γ^k as a function of the weighted norm of $\Delta \tilde{x}^k$ and a constant ϵ. Note that without the constant ϵ, we would require this error to vanish when x^k is close to the optimal solution, *i.e.*, when $\Delta \tilde{x}^k$ 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 ϵ, the dual computation can terminate after certain number of iterations such that the resulting error γ^k satisfies this Assumption.

The recent papers [25] and [26] presented two different distributed methods to determine when to terminate the dual computation procedure such that the above error tolerance level is satisfied. The method in [25] 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 [26] 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 [26] 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 τ^k denote the error in the Newton decrement calculation, i.e.,

$$\tau^k = \tilde{\lambda}(x^k) - \theta^k. \tag{23}$$

For all k, τ^k satisfies

$$|\tau^k| \leq \left(\frac{1}{b} - 1\right)(1 + V).$$

This assumption will be used in establishing the strict positivity of the generated primal vectors x^k. When the method presented in [25] is used to compute θ^k, then we have $\tau^k = 0$ and $b = 1$ for all k and the preceding assumption is satisfied clearly. Throughout the rest of the paper, we assume the conditions in Assumptions 1-3 hold.

In [25] we have shown that the stepsize choice with $\theta^k = \tilde{\lambda}(x^k)$ 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 (20) will also guarantee strict positivity. We first establish a bound on the error in the stepsize under Assumption 3.

Lemma 4.2. Let θ^k be an approximation of the inexact Newton decrement $\tilde{\lambda}(x^k)$ defined in (19). For $\theta^k \geq V$, we have

$$(2b-1)/(\tilde{\lambda}(x^k)+1) \leq \frac{b}{\theta^k+1} \leq 1/(\tilde{\lambda}(x^k)+1), \tag{24}$$

where $b \in (0, 1]$ is the constant used in stepsize choice (20).

Proof. By Assumption 3 and the fact $\theta^k \geq V$, we have

$$|\tilde{\lambda}(x^k) - \theta^k| \leq \left(\frac{1}{b} - 1\right)(1 + V) \leq \left(\frac{1}{b} - 1\right)(1 + \theta^k). \tag{25}$$

By multiplying both sides by the positive scalar b, the above relation implies

$$b\theta^k - b\tilde{\lambda}(x^k) \leq (1-b)(1+\theta^k),$$

which yields

$$(2b-1)\theta^k + (2b-1) \leq b\tilde{\lambda}(x^k) + b.$$

By dividing both sides of the above relation by the positive scalar $(\theta^k+1)(\tilde{\lambda}(x^k)+1)$, we obtain the first inequality in Eq. (24).

Similarly, using Eq. (25) we can establish

$$b\tilde{\lambda}(x^k) - b\theta^k \leq (1-b)(1+\theta^k),$$

which can be rewritten as

$$b\tilde{\lambda}(x^k) + b \leq \theta^k + 1.$$

After dividing both sides of the preceding relation by the positive scalar $(\theta^k+1)(\tilde{\lambda}(x^k)+1)$, we obtain the second inequality in Eq. (24). □

With this bound on the stepsize error, we can show that starting with a strictly positive feasible solution, the primal vectors x^k generated by our algorithm remain positive for all k.

Proposition 4.3. Given a strictly positive feasible primal vector x^0, let $\{x^k\}$ be the sequence generated by the inexact distributed Newton method (17). Assume that the stepsize d^k is selected according to Eq. (20) and the constant b satisfies $\frac{V+1}{2V+1} < b \leq 1$. 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 $-\frac{\partial^2 U_i}{\partial x_i^2}(x_i^k) \geq 0$. Given the form of the Hessian matrix [cf. Eq. (10)], this implies $(H_k)_{ii} \geq \frac{\mu}{(x_i^k)^2}$ for all i, and therefore

$$\tilde{\lambda}(x^k) = \left(\sum_{i=1}^{S+L} (\Delta \tilde{x}_i^k)^2 (H_k)_{ii}\right)^{\frac{1}{2}} \geq \left(\sum_{i=1}^{S+L} \mu \left(\frac{\Delta \tilde{x}_i^k}{x_i^k}\right)^2\right)^{\frac{1}{2}} \geq \max_i \left|\frac{\sqrt{\mu}\Delta \tilde{x}_i^k}{x_i^k}\right|,$$

where the last inequality follows from the nonnegativity of the terms $\mu\left(\frac{\Delta \tilde{x}_i^k}{x_i^k}\right)^2$. By taking the reciprocal on both sides, the above relation implies

$$\frac{1}{\tilde{\lambda}(x^k)} \leq \frac{1}{\max_i \left|\frac{\sqrt{\mu}\Delta \tilde{x}_i^k}{x_i^k}\right|} = \frac{1}{\sqrt{\mu}} \min_i \left|\frac{x_i^k}{\Delta \tilde{x}_i^k}\right| \leq \min_i \left|\frac{x_i^k}{\Delta \tilde{x}_i^k}\right|, \tag{26}$$

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

We show the inductive step by considering two cases.

- Case i: $\theta^k \geq V$
  Since $0 < \frac{V+1}{2V+1} < b \leq 1$, we can apply Lemma 4.2 and obtain that the stepsize d^k satisfies

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

  Using Eq. (26), this implies $d^k < \min_i \left|\frac{x_i^k}{\Delta \tilde{x}_i^k}\right|$. Hence if $x^k > 0$, then $x^{k+1} = x^k + d^k \Delta \tilde{x}^k > 0$.

- Case ii: $\theta^k < V$
  By Assumption 3, we have $\tilde{\lambda}(x^k) < V + \left(\frac{1}{b} - 1\right)(1 + V)$. Using the fact that $b > \frac{V+1}{2V+1}$, we obtain

  $$\tilde{\lambda}(x^k) < V + \left(\frac{1}{b} - 1\right)(1 + V) < V + \left(\frac{2V+1}{V+1} - 1\right)(1 + V) = 2V \leq 1,$$

  where the last inequality follows from the fact that $V < 0.267$. Hence we have $d^k = 1 < \frac{1}{\tilde{\lambda}(x^k)} \leq \min_i \left|\frac{x_i^k}{\Delta \tilde{x}_i^k}\right|$, where the last inequality follows from Eq. (26). Once again, if $x^k > 0$, then $x^{k+1} = x^k + d^k \Delta \tilde{x}^k > 0$.

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

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

5 Convergence Analysis

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

5.1 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 $\mathcal{G} = \{\mathcal{L}, \mathcal{S}\}$, represented by a set $\mathcal{L} = \{1, ..., L\}$ of (directed) links, and a set $\mathcal{S} = \{1, ..., S\}$ of sources. The links form a strongly connected graph, and each source sends information along a predetermined route. The *weighted dual (routing) graph* $\tilde{\mathcal{G}} = \{\tilde{\mathcal{N}}, \tilde{\mathcal{L}}\}$, where $\tilde{\mathcal{N}}$ is the set of nodes, and $\tilde{\mathcal{L}}$ is the set of (directed) links defined by:
A. $\tilde{\mathcal{N}} = \mathcal{L}$;
B. A link is present between node L_i to L_j in $\tilde{\mathcal{G}}$ if and only if there is some common flow between L_i and L_j in $\mathcal{G}$.
C. The weight $\tilde{W}_{ij}$ on the link from node L_i to L_j is given by

$$\tilde{W}_{ij} = (D_k + \bar{B}_k)^{-1}_{ii}(B_k)_{ij} = (D_k + \bar{B}_k)^{-1}_{ii}(AH_k^{-1}A')_{ij} = (D_k + \bar{B}_k)^{-1}_{ii} \sum_{s \in S(i) \cap S(j)} H_{ss}^{-1},$$

where the matrices D_k, B_k, and $\bar{B}_k$ are defined in Eqs. (11), (12) and (13).

Two sample network - dual graph pairs are presented in Figures 1, 2 and 3, 4 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 Eq. (14) involves the matrix $(D_k + \bar{B}_k)^{-1}(\bar{B}_k - B_k)$, which is the weighted Laplacian matrix of the dual graph.[8] The weighted out-degree of node i in the dual graph, i.e., the diagonal entry $(D_k + \bar{B}_k)^{-1}_{ii}\bar{B}_{ii}$ 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 show next 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 [24].

Lemma 5.1. Let M be an $n \times n$ matrix, and assume that its spectral radius, denoted by $\rho(M)$, satisfies $\rho(M) < 1$. Let $\{\lambda_i\}_{i=1,\ldots,n}$ denote the set of eigenvalues of M, with $1 > |\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_n|$ and let v_i denote the set of corresponding unit length right eigenvectors. Assume the matrix has n linearly independent eigenvectors.[9] Then for the sequence $w(t)$ generated by the following iteration

$$w(t+1) = Mw(t), \tag{27}$$

we have

$$||w(t) - w^*||_2 \leq |\lambda_1|^t \alpha, \tag{28}$$

for some positive scalar α, where w^* is the limit of iteration (27) as $t \to \infty$.

We use M to denote the $L \times L$ matrix, $M = (D_k + \bar{B}_k)^{-1}(\bar{B}_k - B_k)$, and z to denote the vector $z = (D_k + \bar{B}_k)^{-1}(-AH_k^{-1}\nabla f(x^k))$. We can rewrite iteration (15) as $w(t+1) = Mw(t) + z$, which implies

$$w(t+q) = M^q w(t) + \sum_{i=0}^{q-1} M^i z = M^q w(t) + (I - M^q)(I - M)^{-1} z.$$

[8] We adopt the following definition for the weighted Laplacian matrix of a graph. Consider a weighted directed graph $\mathcal{G}$ with weight W_{ij} associated with the link from node i to j. We let $W_{ij} = 0$ 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 [5], [8] for more details on graph Laplacian matrices.

[9] An alternative assumption is that the algebraic multiplicity of each λ_i is equal to its corresponding geometric multiplicity, since eigenvectors associated with different eigenvalues are independent [18].

This alternative representation is possible since $\rho(M) < 1$, which follows from Theorem 4.1. 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 $u(t+q) = M^q u(0)$, where $u(0) = w(0) - M(I-M)^{-1}z$.

We next show that the matrix M has L linearly independent eigenvectors in order to apply the preceding lemma. We first note that since the nonnegative 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 nonnegative entries. This shows that the matrix D_k [cf. Eq. (11)] has positive diagonal elements and the matrix $\bar{B}$ [cf. Eq. (13)] has nonnegative entries. Therefore the matrix $(D_k+\bar{B}_k)^{-\frac{1}{2}}$ is diagonal and nonsingular. Hence, using the relation $\tilde{M} = (D_k+\bar{B}_k)^{\frac{1}{2}} M (D_k+\bar{B}_k)^{-\frac{1}{2}}$, we see that the matrix $M = (D_k+\bar{B}_k)^{-1}(\bar{B}_k - B_k)$ is similar to the matrix $\tilde{M} = (D_k+\bar{B}_k)^{-\frac{1}{2}}(\bar{B}_k - B_k)(D_k+\bar{B}_k)^{-\frac{1}{2}}$. From the definition of B_k [cf. Eq. (12)] and the symmetry of the matrix $AH_k^{-1}A'$, we conclude that the matrix B is symmetric. This shows that the matrix $\tilde{M}$ is symmetric and hence diagonalizable, which implies that the matrix M is also diagonalizable, and therefore it has L linearly independent eigenvectors.[10] We can use Lemma 5.1 to infer that

$$||w(t) - w^*||_2 = ||u(t) - u^*||_2 \leq |\lambda_1|^t \alpha,$$

where λ_1 is the eigenvalue of M with largest magnitude, and α is a constant that depends on the initial vector $u(0) = w(0) - (I-M)^{-1}z$. Hence λ_1 determines the speed of convergence of the dual iteration.

We next analyze the relationship between λ_1 and the dual graph topology. First note that the matrix $M = (D_k+\bar{B}_k)^{-1}(\bar{B}_k - B_k)$ is the weighted Laplacian matrix of the dual graph [cf. Section 4.1], and is therefore positive semidefinite [8]. We then have $\rho(M) = |\lambda_1| = \lambda_1 \geq 0$. From graph theory [20], Theorem 4.1 and the above analysis, we have

$$\frac{4\mathrm{mc}(M)}{L} \leq \lambda_1 \leq \min\left\{2\max_{l\in L}\left[(D_k+\bar{B}_k)^{-1}\bar{B}_k\right]_{ll}, 1\right\}, \tag{29}$$

where $\mathrm{mc}(M)$ is the weighted maximum cut of the dual graph, i.e.,

$$\mathrm{mc}(M) = \max_{S\subset\tilde{\mathcal{N}}}\left\{\sum_{i\in S, j\notin S}\tilde{W}_{ij} + \sum_{i\in S, j\notin S}\tilde{W}_{ji}\right\},$$

where $\tilde{W}_{ij}$ 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 λ_1, implying the dual iteration cannot finish with very few iterates. When the maximum weighted out-degree, i.e., $\max_{l\in L}\left[(D_k+\bar{B}_k)^{-1}\bar{B}_k\right]_{ll}$, in the dual graph is small, the above relation provides a small upper bound on λ_1 and 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 Figures 1 and 3, whose corresponding dual graphs

[10] If a square matrix A of size $n \times n$ is symmetric, then A has n linearly independent eigenvectors. If a square matrix B of size $n \times n$ is similar to a symmetric matrix, then B has n linearly independent eigenvectors [12].

are presented in Figures 2 and 4 respectively. Both of these networks have 3 source-destination pairs and 7 links. However, in Figure 1 all three flows use the same link, i.e., L_4, whereas in Figure 3 at most two flows share the same link. This difference in the network topology results in different degree distributions in the dual graphs as shown in Figures 2 and 4. To be more concrete, let $U_i(s_i) = 15\log(s_i)$ for all sources i in both graphs and link capacity $c_l = 35$ for all links l. We apply our distributed Newton algorithm to both problems, for the primal iteration when all the source rates are 10, the largest weighted out-degree in the dual graphs of the two examples are 0.46 for Figure 2 and 0.095 for Figure 4, which implies the upper bounds for λ_1 of the corresponding dual iterations are 0.92 and 0.19 respectively [cf. Eq. (29)]. The weighted maximum cut for Figure 2 is obtained by isolating the node corresponding to L_4, with weighted maximum cut value of 0.52. The maximum cut for Figure 4 is formed by isolating the set $\{L_4, L_6\}$, with weighted maximum cut value of 0.17. Based on (29) these graph cuts generate lower bounds for λ_1 of 0.30 and 0.096 respectively. By combining the upper and lower bounds, we obtain intervals for λ_1 as $[0.30, 0.92]$ and $[0.096, 0.19]$ respectively. Recall that a large spectral radius corresponds to slow convergence in the dual iteration [cf. Eq. (28)], therefore these bounds guarantee that the dual iteration for the network in Figure 3, which is less congested, converges faster than for the one in Figure 1. Numerical results suggest the actual largest eigenvalues are 0.47 and 0.12 respectively, which confirm with the prediction.

5.2 Convergence in Primal Iterations

We next present our convergence analysis for the primal sequence $\{x^k\}$ generated by the inexact Newton method (17). For the k^{th} iteration, we define the function $\tilde{f}_k : \mathbb{R} \to \mathbb{R}$ as

$$\tilde{f}_k(t) = f(x^k + t\Delta\tilde{x}^k), \tag{30}$$

which is self-concordant, because the objective function f is self-concordant. Note that the value $\tilde{f}_k(0)$ and $\tilde{f}_k(d^k)$ are the objective function values at x^k and x^{k+1} respectively. Therefore $\tilde{f}_k(d^k) - \tilde{f}_k(0)$ measures the decrease in the objective function value at the k^{th} iteration. We will refer to the function $\tilde{f}_k$ 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.[11]

5.2.1 Preliminaries

Using the definition of a self-concordant function, we have the following result (see [6] for the proof).

Lemma 5.2. Let $\tilde{f} : \mathbb{R} \to \mathbb{R}$ be a self-concordant function. Then for all $t \geq 0$ in the domain of the function $\tilde{f}$ with $t\tilde{f}''(0)^{\frac{1}{2}} < 1$, the following inequality holds:

$$\tilde{f}(t) \leq \tilde{f}(0) + t\tilde{f}'(0) - t\tilde{f}''(0)^{\frac{1}{2}} - \log(1 - t\tilde{f}''(0)^{\frac{1}{2}}). \tag{31}$$

We will use the preceding lemma to prove a key relation in analyzing convergence properties of our algorithm [see Lemma 5.8]. 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 [14], [22] for more details).

[11] 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).

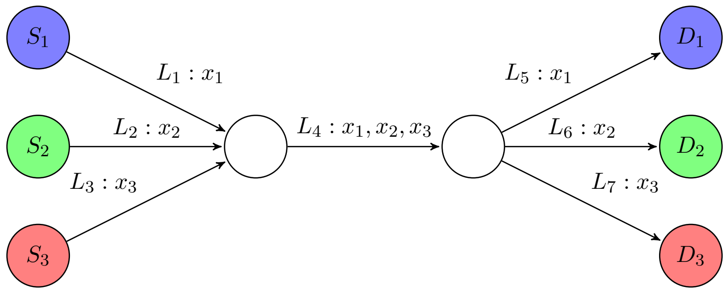

Figure 1: Each source-destination pair is displayed with the same color. We use x_i to denote the flow corresponding to the i^{th} source-destination pair and L_i to denote the i^{th} link. All 3 flows traverse link L_4.

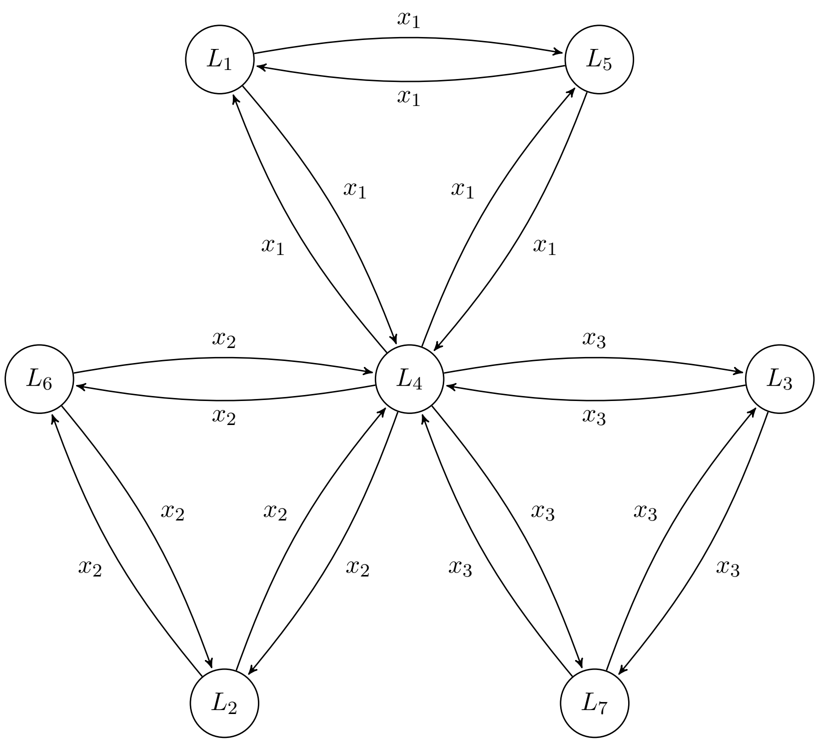

Figure 2: Dual graph for the network in Figure 1, each link in this graph corresponds to the flows shared between the links in the original network. The node corresponding to link L_4 has high unweighted out-degree equal to 6.

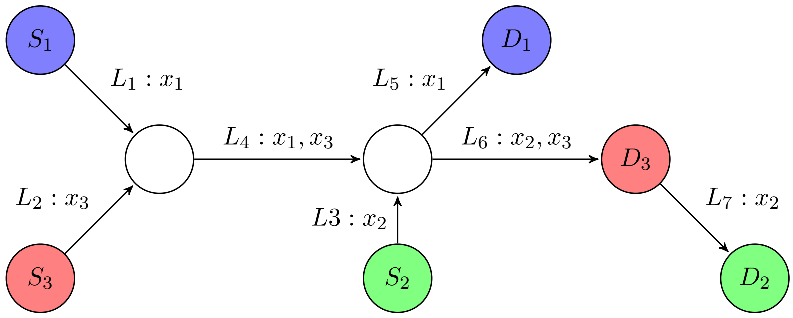

Figure 3: Each source-destination pair is displayed with the same color. We use x_i to denote the flow corresponding to the i^{th} source-destination pair and L_i to denote the i^{th} link. Each link has at most 2 flows traversing it.

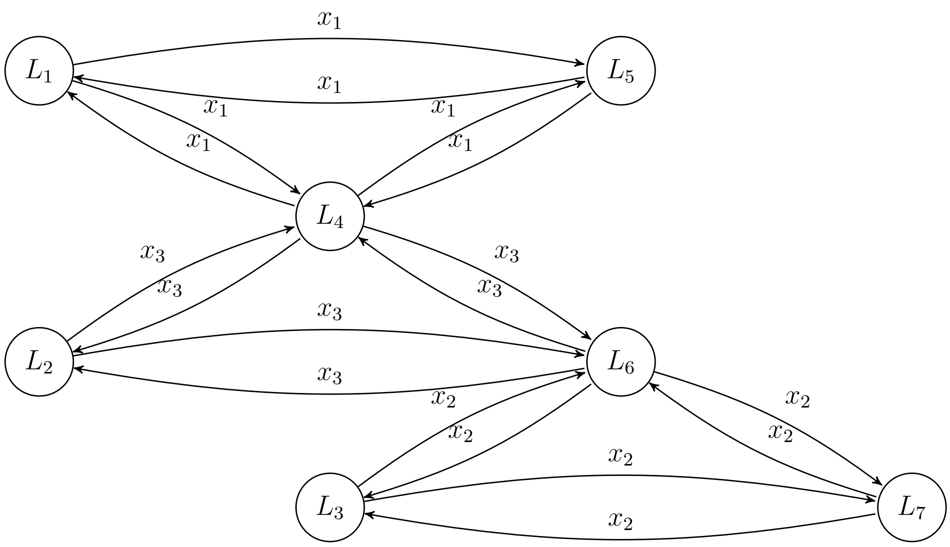

Figure 4: Dual graph for the network in Figure 4, each link in this graph corresponds to the flows shared between the links in the original network. Both nodes corresponding to links L_4 and L_6 has relatively high out-degree equal to 4.

Lemma 5.3. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a self-concordant function. Suppose vectors x and y are in the domain of f and $\tilde{\lambda} = ((x-y)'\nabla^2 f(x)(x-y))^{\frac{1}{2}} < 1$, then for any $z \in \mathbb{R}^n$, the following inequality holds:

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

The next two lemmas establish properties of the Newton decrement generated by the equality-constrained Newton method. The first lemma extends results in [14] and [22] 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.[12]

Lemma 5.4. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a self-concordant function. Consider solving the equality constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & Ax = c, \end{aligned} \tag{33}$$

using an (exact) Newton method with feasible initialization, where the matrix A is in $\mathbb{R}^{L\times(L+S)}$ and has full column rank, i.e., $\text{rank}(A) = L$. Let Δx be the exact Newton direction at x, i.e., Δx solves the following system of linear equations,

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

Let $\Delta\tilde{x}$ denote any direction with $\gamma = \Delta x - \Delta\tilde{x}$, and $x(t) = x + t\Delta\tilde{x}$ for $t \in [0,1]$. Let z be the exact Newton direction at $x + \Delta\tilde{x}$. If $\tilde{\lambda} = \sqrt{\Delta\tilde{x}'\nabla^2 f(x)\Delta\tilde{x}} < 1$, then we have

$$z\nabla^2 f(x+\Delta\tilde{x})'z \leq \frac{\tilde{\lambda}^2}{1-\tilde{\lambda}}\sqrt{z'\nabla^2 f(x)z} + |\gamma'\nabla^2 f(x)'z|.$$

Proof. See Appendix A. □

One possible matrix K in the above proof for problem (4) is given by $K = \begin{pmatrix} I(S) \\ -R \end{pmatrix}$, whose corresponding unconstrained domain consists of the source rate variables. In the unconstrained domain, the source rates are updated and then the matrix K adjusts the slack variables accordingly to maintain the feasibility, which coincides with our inexact distributed algorithm in the primal domain. The above lemma will be used to guarantee quadratic rate of convergence for the distributed inexact Newton method (17)]. The next lemma plays a central role in relating the optimality gap in the objective function value to the exact Newton decrement (see [6] for more details).

Lemma 5.5. Let $F : \mathbb{R}^n \to \mathbb{R}$ be a self-concordant function. Consider solving the unconstrained optimization problem

$$\text{minimize}_{x\in\mathbb{R}^n} \; F(x), \tag{35}$$

using an (unconstrained) Newton method. Let Δx 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

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

[12]We use the same notation in the subsequent lemmas as in problem formulation (4) despite the fact that the results hold for general optimization problems with self-concordant objective functions and linear equality constraints.

Using the same elimination technique and isomorphism established for Lemma 5.4, the next result follows immediately.

Lemma 5.6. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a self-concordant function. Consider solving the equality constrained optimization problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & Ax = c, \end{aligned} \tag{37}$$

using a constrained Newton method with feasible initialization. Let Δx be the exact (primal) Newton direction at x, i.e., Δx solves the system

$$\begin{pmatrix} \nabla^2 f(x) & A' \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ w \end{pmatrix} = - \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix}.$$

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 $\lambda(x) \le 0.68$, then we have

$$f^* \ge f(x) - \lambda(x)^2. \tag{38}$$

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 [6]). 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 comprises 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.

5.2.2 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 5.7. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17). Let $\tilde{\lambda}(x^k)$ be the inexact Newton decrement at x^k [cf. Eq. (19)]. For all k, we have

$$|(\gamma^k)' \nabla^2 f(x^k) \Delta \tilde{x}^k| \le p \tilde{\lambda}(x^k)^2 + \tilde{\lambda}(x^k)\sqrt{\epsilon},$$

where γ^k, p, and ϵ are nonnegative scalars defined in Assumption 2.

Proof. By Assumption 1, the Hessian matrix $\nabla^2 f(x^k)$ is positive definite for all x^k. We therefore can apply the generalized Cauchy-Schwarz inequality and obtain

$$\begin{aligned} |(\gamma^k)' \nabla^2 f(x^k) \Delta \tilde{x}^k| &\le \sqrt{((\gamma^k)' \nabla^2 f(x^k) \gamma^k)((\Delta \tilde{x}^k)' \nabla^2 f(x^k) \Delta \tilde{x}^k)} \\ &\le \sqrt{(p^2 \tilde{\lambda}(x^k)^2 + \epsilon) \tilde{\lambda}(x^k)^2} \\ &\le \sqrt{(p^2 \tilde{\lambda}(x^k)^2 + \epsilon + 2p\tilde{\lambda}(x^k)\sqrt{\epsilon}) \tilde{\lambda}(x^k)^2}, \end{aligned} \tag{39}$$

where the second inequality follows from Assumption 2 and definition of $\tilde{\lambda}(x^k)$, and the third inequality follows by adding the nonnegative term $2p\sqrt{\epsilon}\tilde{\lambda}(x^k)^3$ to the right hand side. By the nonnegativity of the inexact Newton decrement $\tilde{\lambda}(x^k)$, it can be seen that relation (39) implies

$$|(\gamma^k)' \nabla^2 f(x^k) \Delta \tilde{x}^k| \le \tilde{\lambda}(x^k)(p\tilde{\lambda}(x^k) + \sqrt{\epsilon}) = p\tilde{\lambda}(x^k)^2 + \tilde{\lambda}(x^k)\sqrt{\epsilon},$$

which proves the desired relation. □

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 5.8. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17). Let $\tilde{f}_k$ be the objective function along the Newton direction and $\tilde{\lambda}(x^k)$ be the inexact Newton decrement [cf. Eqs. (30) and (19)] at x^k respectively. For all k with $0 \leq t < 1/\tilde{\lambda}(x^k)$, we have

$$\tilde{f}_k(t) \leq \tilde{f}_k(0) - t(1-p)\tilde{\lambda}(x^k)^2 - (1-\sqrt{\epsilon})t\tilde{\lambda}(x^k) - \log(1 - t\tilde{\lambda}(x^k)), \tag{40}$$

where p, and ϵ are the nonnegative scalars defined in Assumption 2.

Proof. Recall that Δx^k is the exact Newton direction, which solves the system (8). Therefore for some w^k, the following equation is satisfied,

$$\nabla^2 f(x^k)\Delta x^k + A'w^k = -\nabla f(x^k).$$

By left multiplying the above relation by $(\Delta\tilde{x}^k)'$, we obtain

$$(\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta x^k + (\Delta\tilde{x}^k)'A'w^k = -(\Delta\tilde{x}^k)'\nabla f(x^k).$$

Using the facts that $\Delta x^k = \Delta\tilde{x}^k + \gamma^k$ from Assumption 2 and $A\Delta\tilde{x}^k = 0$ by the design of our algorithm, the above relation yields

$$(\Delta\tilde{x}^k)'\nabla^2 f(x)\Delta\tilde{x}^k + (\Delta\tilde{x}^k)'\nabla^2 f(x^k)\gamma^k = -(\Delta\tilde{x}^k)'\nabla f(x^k).$$

By Lemma 5.7, we can bound $(\Delta\tilde{x}^k)'\nabla^2 f(x^k)\gamma^k$ by,

$$p\tilde{\lambda}(x^k)^2 + \tilde{\lambda}(x^k)\sqrt{\epsilon} \geq (\Delta\tilde{x}^k)'\nabla^2 f(x^k)\gamma^k \geq -p\tilde{\lambda}(x^k)^2 - \tilde{\lambda}(x^k)\sqrt{\epsilon}.$$

Using the definition of $\tilde{\lambda}(x^k)$ [cf. Eq. (19)] and the preceding two relations, we obtain the following bounds on $(\Delta\tilde{x}^k)'\nabla f(x^k)$:

$$-(1+p)\tilde{\lambda}(x^k)^2 - \tilde{\lambda}(x^k)\sqrt{\epsilon} \leq (\Delta\tilde{x}^k)'\nabla f(x^k) \leq -(1-p)\tilde{\lambda}(x^k)^2 + \tilde{\lambda}(x^k)\sqrt{\epsilon}.$$

By differentiating the function $\tilde{f}_k(t)$, and using the preceding relation, this yields,

$$\begin{aligned}\tilde{f}_k'(0) &= \nabla f(x^k)'\Delta\tilde{x}^k \\ &\leq -(1-p)\tilde{\lambda}(x^k)^2 + \tilde{\lambda}(x^k)\sqrt{\epsilon}.\end{aligned} \tag{41}$$

Moreover, we have

$$\begin{aligned}\tilde{f}_k''(0) &= (\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k \\ &= \tilde{\lambda}(x^k)^2.\end{aligned} \tag{42}$$

The function $\tilde{f}_k(t)$ is self-concordant for all k, therefore by Lemma 5.2, for $0 \leq t < 1/\tilde{\lambda}(x^k)$, the following relations hold:

$$\begin{aligned}\tilde{f}_k(t) &\leq \tilde{f}_k(0) + t\tilde{f}_k'(0) - t\tilde{f}_k''(0)^{\frac{1}{2}} - \log(1 - t\tilde{f}_k''(0)^{\frac{1}{2}}) \\ &\leq \tilde{f}_k(0) - t(1-p)\tilde{\lambda}(x^k)^2 + t\tilde{\lambda}(x^k)\sqrt{\epsilon} - t\tilde{\lambda}(x^k) - \log(1 - t\tilde{\lambda}(x^k)) \\ &= \tilde{f}_k(0) - t(1-p)\tilde{\lambda}(x^k)^2 - (1-\sqrt{\epsilon})t\tilde{\lambda}(x^k) - \log(1 - t\tilde{\lambda}(x^k)),\end{aligned}$$

where the second inequality follows by Eqs. (41) and (42). This proves Eq. (40). □

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.

5.2.3 Damped Convergent Phase

In this section, we consider the case when $\theta^k \geq V$ and stepsize $d^k = \frac{b}{\theta^k+1}$ [cf. Eq. (20)]. 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 $t = 1/(\tilde{\lambda}(x^k)+1)$.

Theorem 5.9. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17). Let $\tilde{f}_k$ be the objective function along the Newton direction and $\tilde{\lambda}(x^k)$ be the inexact Newton decrement at x^k [cf. Eqs. (30) and (19)]. Consider the scalars p and ϵ defined in Assumption 2 and assume that $0 < p < \frac{1}{2}$ and $0 < \epsilon < \left(\frac{(0.5-p)(2Vb-V+b-1)}{b}\right)^2$, where b is the constant used in the stepsize rule [cf. Eq. (20)]. For $\theta^k \geq V$ and $t = 1/\left(\tilde{\lambda}(x^k)+1\right)$, there exists a scalar $\alpha > 0$ such that

$$\tilde{f}_k(t) - \tilde{f}_k(0) \leq -\alpha\,(1+p)\left(\frac{2Vb-V+b-1}{b}\right)^2 \Big/ \left(1 + \frac{2Vb-V+b-1}{b}\right). \tag{43}$$

Proof. For notational simplicity, let $y = \tilde{\lambda}\left(x^k\right)$ in this proof. We will show that for any positive scalar α with $0 < \alpha \leq \left(\frac{1}{2} - p - \frac{\sqrt{\epsilon}b}{(2Vb-V+b-1)}\right)/(p+1)$, Eq. (43) holds. Note that such α exists since $\epsilon < \left(\frac{(0.5-p)(2Vb-V+b-1)}{b}\right)^2$.

By Assumption 3, we have for $\theta^k \geq V$,

$$y \geq \theta^k - \left(\frac{1}{b} - 1\right)(1+V) \geq V - \left(\frac{1}{b} - 1\right)(1+V) = \frac{2Vb-V+b-1}{b}. \tag{44}$$

Using $b > \frac{V+1}{2V+1}$, we have $y \geq V - \left(\frac{1}{b}-1\right)(1+V) > 0$, which implies $2Vb-V+b-1 > 0$. Together with $0 < \alpha \leq \left(\frac{1}{2} - p - \frac{\sqrt{\epsilon}b}{2Vb-V+b-1}\right)/(p+1)$ and $b > \frac{V+1}{2V+1}$, this shows

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

Combining the above, we obtain

$$\sqrt{\epsilon} \leq y\left(\frac{1}{2} - p - \alpha\,(1+p)\right),$$

which using algebraic manipulation yields

$$-(1-p)\,y - \left(1-\sqrt{\epsilon}\right) + (1+y) - \frac{y}{2} \leq -\alpha\,(1+p)\,y.$$

From Eq. (44), we have $y > 0$. We can therefore multiply by y and divide by $1+y$ both sides of the above inequality to obtain

$$-\frac{1-p}{1+y}y^2 - \frac{1-\sqrt{\epsilon}}{1+y}y + y - \frac{y^2}{2\,(1+y)} \leq -\alpha\frac{(1+p)\,y^2}{1+y} \tag{45}$$

Using second order Taylor expansion on $\log(1+y)$, we have for $y \geq 0$

$$\log(1+y) \leq y - \frac{y^2}{2\,(1+y)}.$$

Using this relation in Eq. (45) yields,

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

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

$$-(1-p)\,ty^2 - \left(1-\sqrt{\epsilon}\right)ty - \log(1-ty) \leq -\alpha\frac{(1+p)\,y^2}{1+y}.$$

Using Eq. (40) from Lemma 5.8 and definition of y in the preceding, we obtain

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

Observe that the function $h(y) = \frac{y^2}{y+1}$ is monotonically increasing in y, and for $\theta^k \geq V$ by relation (44) we have $y \geq \frac{2Vb-V+b-1}{b}$. Therefore

$$-\alpha(1+p)\frac{y^2}{y+1} \leq -\alpha(1+p)\left(\frac{2Vb-V+b-1}{b}\right)^2 \Big/ \left(1+\frac{2Vb-V+b-1}{b}\right).$$

Combining the preceding two relations completes the proof. □

Note that our algorithm uses the stepsize $d^k = \frac{d}{\theta^k+1}$ in the damped convergent phase, which is an approximation to the stepsize $t = 1/(\tilde{\lambda}(x^k)+1)$ used in the previous theorem. The error between the two is bounded by relation (24) as shown in Lemma 4.2. 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 $\beta = \frac{d^k}{t}$, where $t = 1/(\tilde{\lambda}(x^k)+1)$. By the convexity of f, we have

$$f(x^k + \beta t\Delta x^k) = f(\beta(x^k + t\Delta x^k) + (1-\beta)(x^k)) \leq \beta f(x^k + t\Delta x^k) + (1-\beta)f(x^k).$$

Therefore the objective function value improvement is bounded by

$$\begin{aligned} f(x+\beta t\Delta x^k) - f(x^k) &\leq \beta f(x^k + t\Delta x^k) + (1-\beta)f(x^k) - f(x^k) \\ &= \beta(f(x^k + t\Delta x^k) - f(x^k)) \\ &= \beta(\tilde{f}_k(t) - \tilde{f}_k(0)), \end{aligned}$$

where the last equality follows from the definition of $\tilde{f}_k(t)$. Since $0 < \frac{V+1}{2V+1} < b \leq 1$, we can apply Lemma 4.2 and obtain bounds on β as $2b-1 \leq \beta \leq 1$. Hence combining this bound with Theorem 5.9, we obtain

$$f(x^{k+1}) - f(x^k) \leq -(2b-1)\alpha(1+p)\frac{\left(\frac{2Vb-V+b-1}{b}\right)^2}{\left(1+\frac{2Vb-V+b-1}{b}\right)}. \tag{46}$$

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

5.2.4 Quadratically Convergent Phase

In this phase, there exists $\bar{k}$ with $\theta^{\bar{k}} < V$ and the step size choice is $d^k = 1$ for all $k \geq \bar{k}$.[13] We show that the optimality gap in the primal objective function value diminishes quadratically to a neighborhood of optimal solution. We proceed by first establishing the following lemma for relating the exact and the inexact Newton decrements.

Lemma 5.10. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17) and $\lambda(x^k)$, $\tilde{\lambda}(x^k)$ be the exact and inexact Newton decrements at x^k [cf. Eqs. (18) and (19)]. Let p and ϵ be the nonnegative scalars defined in Assumption 2. We have

$$(1-p)\tilde{\lambda}(x^k) - \sqrt{\epsilon} \leq \lambda(x^k) \leq (1+p)\tilde{\lambda}(x^k) + \sqrt{\epsilon}. \tag{47}$$

Proof. By Assumption 1, for all k, $\nabla^2 f(x^k)$ is positive definite. We therefore can apply the generalized Cauchy-Schwarz inequality and obtain

$$\begin{aligned} |(\Delta x^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k| &\leq \sqrt{((\Delta x^k)'\nabla^2 f(x^k)\Delta x^k)((\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k)} \\ &= \lambda(x^k)\tilde{\lambda}(x^k), \end{aligned} \tag{48}$$

where the equality follows from definition of $\lambda(x^k)$ and $\tilde{\lambda}(x^k)$. Note that by Assumption 2, we have $\Delta x^k = \Delta\tilde{x}^k + \gamma^k$, and hence

$$\begin{aligned} |(\Delta x^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k| &= |(\Delta\tilde{x}^k + \gamma^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k| \\ &\geq (\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k - |(\gamma^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k| \\ &\geq \tilde{\lambda}(x^k)^2 - p\tilde{\lambda}(x^k)^2 - \tilde{\lambda}(x^k)\sqrt{\epsilon}, \end{aligned} \tag{49}$$

where the first inequality follows from a variation of triangle inequality, and the last inequality follows from Lemma 5.8. Combining the two inequalities (48) and (49), we obtain

$$\lambda(x^k)\tilde{\lambda}(x^k) \geq \tilde{\lambda}(x^k)^2 - p\tilde{\lambda}(x^k)^2 - \sqrt{\epsilon}\tilde{\lambda}(x^k),$$

By canceling the nonnegative term $\tilde{\lambda}(x^k)$ on both sides, we have

$$\lambda(x^k) \geq \tilde{\lambda}(x^k) - p\tilde{\lambda}(x^k) - \sqrt{\epsilon}.$$

This shows the first half of the relation (47). For the second half, using the definition of $\lambda(x^k)$, we have

$$\begin{aligned} \lambda(x^k)^2 &= (\Delta x^k)'\nabla^2 f(x^k)\Delta x^k \\ &= (\Delta\tilde{x}^k + \gamma^k)'\nabla^2 f(x^k)(\Delta\tilde{x}^k + \gamma^k) \\ &= (\Delta\tilde{x}^k)'\nabla^2 f(x^k)\Delta\tilde{x}^k + (\gamma^k)'\nabla^2 f(x^k)\gamma^k + 2(\Delta\tilde{x}^k)'\nabla^2 f(x^k)\gamma^k, \end{aligned}$$

where the second equality follows from the definition of γ^k [cf. Eq. (21)]. By using the definition of $\tilde{\lambda}(x^k)$, Assumption 2 and Lemma 5.7, the preceding relation implies,

$$\begin{aligned} \lambda(x^k)^2 &\leq \tilde{\lambda}(x^k)^2 + p^2\tilde{\lambda}(x^k)^2 + \epsilon + 2p\tilde{\lambda}(x^k)^2 + 2\sqrt{\epsilon}\tilde{\lambda}(x^k) \\ &\leq \tilde{\lambda}(x^k)^2 + p^2\tilde{\lambda}(x^k)^2 + 2p\tilde{\lambda}(x^k)^2 + 2\sqrt{\epsilon}(1+p)\tilde{\lambda}(x^k) + \epsilon \\ &= ((1+p)\tilde{\lambda}(x^k) + \sqrt{\epsilon})^2, \end{aligned}$$

where the second inequality follows by adding a nonnegative term of $2\sqrt{\epsilon}p\tilde{\lambda}(x^k)$ to the right hand side. By nonnegativity of p, ϵ, λ and $\tilde{\lambda}(x^k)$, we can take the square root of both sides and this completes the proof for relation (47). □

[13] Note that once the condition $\theta^{\bar{k}} < V$ is satisfied, in all the following iterations, we have stepsize $d^k = 1$ and no longer need to compute θ^k.

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 $\theta^{\bar{k}} < V$ 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 (17). Let ϕ be a positive scalar with $\phi \leq 0.267$. Let ξ and v be nonnegative scalars defined in terms of ϕ as

$$\xi = \frac{\phi p + \sqrt{\epsilon}}{1 - p - \phi - \sqrt{\epsilon}} + \frac{2\phi\sqrt{\epsilon} + \epsilon}{(1 - p - \phi - \sqrt{\epsilon})^2}, \qquad v = \frac{1}{(1 - p - \phi - \sqrt{\epsilon})^2},$$

where p and ϵ are the scalars defined in Assumption 2. The following relations hold

$$(1+p)(\theta^{\bar{k}} + \tau^{\bar{k}}) + \sqrt{\epsilon} \leq \phi, \tag{50}$$

$$v(0.68)^2 + \xi \leq 0.68, \tag{51}$$

$$\frac{0.68 + \sqrt{\epsilon}}{1 - p} \leq 1, \tag{52}$$

$$p + \sqrt{\epsilon} \leq 1 - (4\phi^2)^{\frac{1}{4}} - \phi, \tag{53}$$

where $\tau^{\bar{k}} > 0$ is a bound on the error in the Newton decrement calculation at step $\bar{k}$ [cf. Assumption 3].

The upper bound of 0.267 on ϕ is necessary here to guarantee relation (53) can be satisfied by some nonnegative scalars p and ϵ. Relation (50) can be satisfied by some nonnegative scalars p, ϵ and $\tau^{\bar{k}}$, because we have $\theta^{\bar{k}} < V < 0.267$. Relation (50) and (51) will be used to guarantee the condition $\lambda(x^k) \leq 0.68$ is satisfied throughout this phase, so that we can use Lemma 5.6 to relate the optimality gap with the Newton decrement, and relation (52) and (53) will be used for establishing the quadratic rate of convergence of the objective function value, as we will show in the Theorem 5.12. This assumption can be satisfied by first choosing proper values for the scalars p, ϵ and τ such that all the relations are satisfied, and then adapt both the consensus algorithm for $\theta^{\bar{k}}$ and the dual iterations for w^k according to the desired precision (see the discussions following Assumption 2 and 3 for how these precision levels can be achieved).

To show the quadratic rate of convergence for the primal iterations, we need the following lemma, which relates the exact Newton decrement at the current and the next step.

Lemma 5.11. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17) and $\lambda(x^k)$, $\tilde{\lambda}(x^k)$ be the exact and inexact Newton decrements at x^k [cf. Eqs. (18) and (19)]. Let θ^k be the computed inexact value of $\tilde{\lambda}(x^k)$ and let Assumption 4 hold. Then for all k with $\tilde{\lambda}(x^k) < 1$, we have

$$\lambda(x^{k+1}) \leq v\lambda(x^k)^2 + \xi, \tag{54}$$

where ξ and v are the scalars defined in Assumption 4 and p and ϵ are defined as in Assumption 2.

Proof. Given $\tilde{\lambda}(x^k) < 1$, we can apply Lemma 5.4 by letting $z = \Delta x^{k+1}$, we have

$$\begin{aligned}
\lambda(x^{k+1})^2 &= (\Delta x^{k+1})' \nabla f^2(x + \Delta\tilde{x}) \Delta x^{k+1} \\
&\leq \frac{\tilde{\lambda}(x^k)^2}{1 - \tilde{\lambda}(x^k)} \sqrt{(\Delta x^{k+1})' \nabla^2 f(x) \Delta x^{k+1}} + \left| (\gamma^k)' \nabla^2 f(x)' \Delta x^{k+1} \right| \\
&\leq \frac{\tilde{\lambda}(x^k)^2}{1 - \tilde{\lambda}(x^k)} \sqrt{(\Delta x^{k+1})' \nabla^2 f(x) \Delta x^{k+1}} + \sqrt{(\gamma^k)' \nabla^2 f(x) \gamma^k} \sqrt{(\Delta x^{k+1})' \nabla^2 f(x) \Delta x^{k+1}},
\end{aligned}$$

where the last inequality follows from the generalized Cauchy-Schwarz inequality. Using Assumption 2, the above relation implies

$$\lambda(x^{k+1})^2 \le \left(\frac{\tilde{\lambda}(x^k)^2}{1-\tilde{\lambda}(x^k)} + \sqrt{p^2\tilde{\lambda}(x^k)^2+\epsilon}\right)\sqrt{(\Delta x^{k+1})'\nabla^2 f(x)\Delta x^{k+1}}.$$

By the fact that $\tilde{\lambda}(x^k) \le \theta^k + \tau \le \phi < 1$, we can apply Lemma 5.3 and obtain,

$$\begin{aligned}\lambda(x^{k+1})^2 &\le \frac{1}{1-\tilde{\lambda}(x^k)}\left(\frac{\tilde{\lambda}(x^k)^2}{1-\tilde{\lambda}(x^k)} + \sqrt{p^2\tilde{\lambda}(x^k)^2+\epsilon}\right)\sqrt{(\Delta x^{k+1})'\nabla^2 f(x+\Delta\tilde{x})\Delta x^{k+1}} \\ &= \left(\frac{\tilde{\lambda}(x^k)^2}{(1-\tilde{\lambda}(x^k))^2} + \frac{\sqrt{p^2\tilde{\lambda}(x^k)^2+\epsilon}}{1-\tilde{\lambda}(x^k)}\right)\lambda(x^{k+1}).\end{aligned}$$

By dividing the last line by $\lambda(x^{k+1})$, this yields

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

From Eq. (47), we have $\tilde{\lambda}(x^k) \le \frac{\lambda(x^k)+\sqrt{\epsilon}}{1-p}$. Therefore the above relation implies

$$\lambda(x^{k+1}) \le \left(\frac{\lambda(x^k)+\sqrt{\epsilon}}{1-p-\lambda(x^k)-\sqrt{\epsilon}}\right)^2 + \frac{p\lambda(x^k)+\sqrt{\epsilon}}{1-p-\lambda(x^k)-\sqrt{\epsilon}}.$$

By Eq. (56), we have $\lambda(x^k) \le \phi$, and therefore the above relation can be relaxed to

$$\lambda(x^{k+1}) \le \left(\frac{\lambda(x^k)}{1-p-\phi-\sqrt{\epsilon}}\right)^2 + \frac{\phi p+\sqrt{\epsilon}}{1-p-\phi-\sqrt{\epsilon}} + \frac{2\phi\sqrt{\epsilon}+\epsilon}{(1-p-\phi-\sqrt{\epsilon})^2}.$$

Hence, by definition of ξ and v, we have

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

□

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 5.12. Let $\{x^k\}$ be the primal sequence generated by the inexact Newton method (17) and $\lambda(x^k)$, $\tilde{\lambda}(x^k)$ be the exact and inexact Newton decrements at x^k [cf. Eqs. (18) and (19)]. Let $f(x^k)$ be the corresponding objective function value at k^{th} iteration and f^* denote the optimal objective function value for problem (4). Let Assumption 4 hold, and ξ and v be the scalars defined in Assumption 4. Assume that for some $\delta \in [0, 1/2)$,

$$\xi + v\xi \le \frac{\delta}{4v}.$$

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

$$\lambda(x^{\bar{k}+m}) \leq \frac{1}{2^{2^m}v} + \xi + \frac{\delta}{v}\frac{2^{2^m-1}-1}{2^{2^m}}, \tag{55}$$

and

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

where $\bar{k}$ is the iteration index with $\theta^{\bar{k}} < V$.

Proof. We prove Eq. (55) by induction. First for $m = 1$, from Assumption 3, we have $\tilde{\lambda}(x^{\bar{k}}) \leq \theta^{\bar{k}} + \tau^{\bar{k}}$. Relation (50) implies $\theta^{\bar{k}} + \tau^{\bar{k}} \leq \phi < 1$, hence we have $\tilde{\lambda}(x^{\bar{k}}) < 1$ and we can apply Lemma 5.11 and obtain

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

By Assumption 4 and Eq. (47), we have

$$\lambda(x^{\bar{k}}) \leq (1+p)(\theta^{\bar{k}} + \tau^{\bar{k}}) + \sqrt{\epsilon} \leq \phi. \tag{56}$$

The above two relations imply

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

The right hand side is monotonically increasing in ϕ. Since $\phi \leq 0.68$, we have by Eq. (51), $\lambda(x^{\bar{k}+1}) \leq 0.68$. By relation (53), we obtain $(1-p-\phi-\sqrt{\epsilon})^4 \geq 4\phi^2$. Using the definition of v, i.e., $v = \frac{1}{(1-p-\phi-\sqrt{\epsilon})^2}$, the above relation implies $v\phi^2 \leq \frac{1}{4v}$. Hence we have

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

This establishes relation (55) for $m = 1$.

We next assume that Eq. (55) holds and $\lambda(x^{\bar{k}+m}) \leq 0.68$ for some $m > 0$, and show that these also hold for $m+1$. From Eqs. (47) and (52), we have

$$\tilde{\lambda}(x^{\bar{k}+m}) \leq \frac{\lambda(x^{\bar{k}+m}) + \sqrt{\epsilon}}{1-p} \leq \frac{0.68 + \sqrt{\epsilon}}{1-p} \leq 1,$$ [14]

where in the second inequality we used the inductive hypothesis that $\lambda(x^{\bar{k}+m}) \leq 0.68$. Hence we can apply Eq. (54) and obtain

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

using Eq. (51) and $\lambda(x^{\bar{k}+m}) \leq 0.68$ once more, we have $\lambda(x^{\bar{k}+m+1}) \leq 0.68$. From our inductive hypothesis that (55) holds for m, the above relation also implies

$$\begin{aligned}\lambda(x^{\bar{k}+m+1}) &\leq v\left(\frac{1}{2^{2^m}v} + \xi + \frac{\delta}{v}\frac{2^{2^m-1}-1}{2^{2^m}}\right)^2 + \xi \\ &= \frac{1}{2^{2^{m+1}}v} + \frac{\xi}{2^{2^m-1}} + \frac{\delta}{v}\frac{2^{2^m-1}-1}{2^{2^{m+1}-1}} + v\left(\xi + \frac{\delta}{v}\frac{2^{2^m-1}-1}{2^{2^m}}\right)^2 + \xi,\end{aligned}$$

[14] Note that we do not need monotonicity in $\tilde{\lambda}(x^k)$, instead the error level assumption from relation (52) enables us to use Lemma 5.11 to establish quadratic rate of convergence.

Using algebraic manipulations and the assumption that $\xi + v\xi \leq \frac{\delta}{4v}$, this yields

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

completing the induction and therefore the proof of relation (55).

The induction proof above suggests that the condition $\lambda(x^{\bar{k}+m}) \leq 0.68$ holds for all $m > 0$, we can therefore apply Lemma 5.6, and obtain an upper bound on optimality gap as follows,

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

Combining this with Eq. (55), we obtain

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

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

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 $\xi + \frac{\delta}{2v}$, where

$$\xi = \frac{\phi p + \sqrt{\epsilon}}{1 - p - \phi - \sqrt{\epsilon}} + \frac{2\phi\sqrt{\epsilon} + \epsilon}{(1 - p - \phi - \sqrt{\epsilon})^2}, \qquad v = \frac{1}{(1 - p - \phi - \sqrt{\epsilon})^2},$$

and the condition $\xi + v\xi \leq \frac{\delta}{4v}$ is satisfied. Note that with the exact Newton algorithm, we have $p = \epsilon = 0$, which implies $\xi = 0$ and we can choose $\delta = 0$, 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. Any algorithm for problem (4) where the update rule is given as Eq. (17) with stepsize d^k defined as in Eq. (20) and inexact Newton direction $\Delta\tilde{x}^k$ defined as an inexact solution to the system (8), if Assumptions 2-4 are satisfied, then the preceding analysis can be extended and the sequence of objective function value generated by the algorithm converges quadratically to an error neighborhood of the optimal value.

5.3 Convergence with respect to Design Parameter μ

In the preceding development, we have restricted our attention to developing an algorithm for a given logarithmic barrier coefficient μ. We next study the convergence properties of the optimal object function value as a function of μ 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 [22].

Lemma 5.13. Let G be a closed convex domain, and function g be a self-concordant barrier function for G, then for any x, y in interior of G, we have $(y - x)'\nabla g(x) \leq 1$.

Using this lemma and an argument similar to that in [22], we can establish the following result, which bounds the sub-optimality as a function of μ.

Theorem 5.14. Given $\mu \geq 0$, let $x(\mu)$ denote the optimal solution of problem (4) and $h(x(\mu)) = \sum_{i=1}^{S} -U_i(x_i(\mu))$. Similarly, let x^* denote the optimal solution of problem (2) together with corresponding slack variables (defined in Eq. (3)), and $h^* = \sum_{i=1}^{S} -U_i(x_i^*)$. Then, the following relation holds,

$$h(x(\mu)) - h^* \leq \mu.$$

Proof. For notational simplicity, we write $g(x) = -\sum_{i=1}^{S+L} \log(x_i)$. Therefore the objective function for problem (4) can be written as $h(x) + \mu g(x)$. By Assumption 1, we have that the utility functions are concave, therefore the negative objective functions in the minimization problems are convex. From convexity, we obtain

$$h(x^*) \geq h(x(\mu)) + (x^* - x(\mu))' \nabla h(x(\mu)). \tag{57}$$

By optimality condition for $x(\mu)$ for problem (4) for a given μ, we have,

$$(\nabla h(x(\mu)) + \mu \nabla g(x(\mu)))'(x - x(\mu)) \geq 0,$$

for any feasible x. Since x^* is feasible, we have

$$(\nabla h(x(\mu)) + \mu \nabla g(x(\mu)))'(x^* - x(\mu)) \geq 0,$$

which implies

$$\nabla h(x(\mu))'(x^* - x(\mu)) \geq -\mu \nabla g(x(\mu))'(x^* - x(\mu)).$$

For any μ, we have $x(\mu)$ belong to the interior of the feasible set, and by Lemma 5.13, we have for all $\tilde{\mu}$, $\nabla g(x(\mu))'(x(\tilde{\mu}) - x(\mu)) \leq 1$. By continuity of $x(\mu)$ and the fact that the convex set $Ax \leq c$ is closed, for A and c defined in problem (4), we have $x^* = \lim_{\mu \to 0} x(\mu)$, and hence

$$\nabla g(x(\mu))'(x^* - x(\mu)) = \lim_{\tilde{\mu} \to 0} \nabla g(x(\mu))'(x(\tilde{\mu}) - x(\mu)) \leq 1.$$

The preceding two relations imply

$$\nabla h(x(\mu))'(x^* - x(\mu)) \geq -\mu.$$

In view of relation (57), this establishes the desired result, i.e.,

$$h(x(\mu)) - h^* \leq \mu.$$

□

By using the above theorem, we can develop a method to bound the sub-optimality between the objective function value our algorithm provides for problem (4) and the exact optimal objective function value for problem (2), i.e, the sub-optimality introduced by the barrier functions in the objective function, such that for any positive scalar a, the following relation holds,

$$\frac{h(x(\mu)) - h^*}{h^*} \leq a, \tag{58}$$

where the value $h(x(\mu))$ 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 μ. This leads to a sequence of x^k converging to some $x(\mu)$. Let $h(x(\mu)) = \sum_{i=1}^{S} -U_i(x_i(\mu))$. By Theorem 5.14, we have

$$h(x(\mu)) - \mu \leq h^*. \tag{59}$$

Let scalar M be such that $M = (a[h(x(\mu)) - \mu])^{-1}$ 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 $-M\sum_{i=1}^{S}U_i(x_i)-\sum_{i=1}^{S+L}\log(x_i)$, subject to link capacity constraints.[15] We obtain a sequence of $\tilde{x}^k$ converges to some $\tilde{x}(1)$. Denote the objective function value as $h(\tilde{x}(1))$, then by applying the preceding theorem one more time we have

$$Mh(\tilde{x}(1))-Mh^*\leq\mu=1,$$

which implies

$$h(\tilde{x}(1))-h^*\leq a[h(x(\mu))-\mu]\leq ah^*$$

where the first inequality follows by definition of the positive scalar M and the second inequality follows from relation (59). Hence we have the desired bound (58).

Therefore even with the introduction of the logarithmic barrier function, the relative error in the objective function value can be bounded by an arbitrarily small positive scalar at the cost of performing the fast Newton-type algorithm twice.

6 Conclusions

This paper presents a convergence analysis for the distributed Newton-type algorithm for Network Utility Maximization problems proposed in [25], 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 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 investigating convergence properties for a fixed finite truncation of dual iterations.

A Proof of Lemma 5.4

We first transform problem (33) into an unconstrained one via elimination technique, establish equivalence in the Newton decrements and the Newton primal directions between the two problems following the lines in [6], then derive the results for the unconstrained problem and lastly we map the result back to the original constrained problem.

Since the matrix A has full column rank, i.e., $\text{rank}(A)=L$, in order to eliminate the equality constraints, we let matrix $K\in\mathbb{R}^{(S+L)\times S}$ be any matrix whose range is null space of A, with $\text{rank}(K)=S$, vector $\hat{x}\in\mathbb{R}^{S+L}$ be a feasible solution for problem (33), i.e., $A\hat{x}=c$. Then we have the parametrization of the affine feasible set as

$$\{x|Ax=c\}=\{Ky+\hat{x}|y\in\mathbb{R}^S\}.$$

The eliminated equivalent optimization problem becomes

$$\text{minimize}_{y\in\mathbb{R}^S}\quad F(y)=f(Ky+\hat{x}). \tag{60}$$

We next show the Newton primal direction for the constrained problem (33) and unconstrained problem (60) are isomorphic, where a feasible solution x for problem (33) is mapped to

[15] When $M<0$, we can simply add a constant to the original objective function to shift it upward. Therefore the scalar M can be assumed to be positive without loss of generality. If no estimate on M is available apriori, we can implement the distributed algorithm one more time in the beginning to obtain an estimate to generate the constant accordingly.

y in problem (60) with $Ky + \hat{x} = x$. We start by showing that each Δy in the unconstrained problem corresponds uniquely to the Newton direction in the constrained problem.

For the unconstrained problem, the gradient and Hessian are given by

$$\nabla F(y) = K'\nabla f(Ky + \hat{x}), \quad \nabla^2 F(y) = K'\nabla^2 f(Ky + \hat{x})K. \tag{61}$$

Note that the objective function f is three times continuously differentiable, which implies its Hessian matrix $\nabla^2 f(Ky + \hat{x})$ is symmetric, and therefore we have $\nabla^2 F(y)$ is symmetric, i.e., $\nabla^2 F(y)' = \nabla^2 F(y)$.

The Newton direction for problem (60) is given by

$$\Delta y = -\left(\nabla^2 F(y)\right)^{-1} \nabla F(y) = -(K'\nabla^2 f(x)K)^{-1}K'\nabla f(x).^{16} \tag{62}$$

We choose

$$w = -(AA')^{-1}A(\nabla f(x) + \nabla^2 f(x)\Delta x), \tag{63}$$

and show that $(\Delta x, w)$ where

$$\Delta x = K\Delta y \tag{64}$$

is the unique solution pair for the linear system (34) for the constrained problem (33). To establish the first equation, i.e., $\nabla^2 f(x)\Delta x + A'w = -\nabla f(x)$, we use the property that $\begin{pmatrix} K' \\ A \end{pmatrix} u = \begin{pmatrix} K'u \\ Au \end{pmatrix} = 0$ for some $u \in \mathbb{R}^{S+L}$ implies $u = 0$.[17] We have

$$\begin{aligned} &\begin{pmatrix} K' \\ A \end{pmatrix} \left(\nabla^2 f(x)\Delta x + A'w + \nabla f(x) \right) \\ &= \begin{pmatrix} K'\nabla^2 f(x)K(-(K'\nabla^2 f(x)K)^{-1}K'\nabla f(x)) + K'A'w + K'\nabla f(x) \\ A\nabla^2 f(x)\Delta x - A(\nabla f(x) + \nabla^2 f(x)\Delta x) + A\nabla f(x) \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \end{aligned}$$

where the first equality follows from definition of Δx, Δy and w [cf. Eqs. (64), (62) and (63)] and the second equality follows the fact that $K'A'w = 0$ for any w.[18] Therefore we conclude that the first equation in (34) holds. Since the range of matrix K is the null space of matrix A, we have $AKy = 0$ for all y, therefore the second equation in (34) holds, i.e., $A\Delta x = 0$.

For the converse, given a Newton direction Δx defined as solution to the system (34) for the constrained problem (33), we can uniquely recover a vector Δy, such that $K\Delta y = \Delta x$. This is because $A\Delta x = 0$ from (34), and hence Δx is in the null space of the matrix A, i.e., column space of the matrix K. The matrix K has full rank, thus there exists a unique Δy. Therefore the (primal) Newton directions for problems (60) and (33) are isomorphic under the mapping K. In what follows, we perform our analysis for the unconstrained problem (60) and then use isomorphic transformations to show the result hold for the equality constrained problem (33).

[16]The matrix $K\nabla^2 f(x)K$ is invertible. If for some $y \in \mathbb{R}^S$, we have $K\nabla^2 f(x)K'y = 0$, then $y'K\nabla^2 f(x)K'y = \left|\left|(\nabla^2 f(x))^{\frac{1}{2}}K'y\right|\right|_2 = 0$, which implies $||K'x||_2 = 0$, because the matrix $\nabla^2 f(x)$ is strictly positive for all x. The rows of the matrix K' span $\mathbb{R}^S$, therefore we have $y = 0$. This shows that the matrix $K\nabla^2 f(x)K'$ is invertible.

[17]If $K'u = 0$, then the vector u is orthogonal to the row space of the matrix K', and hence column space of the matrix K, i.e., null space of the matrix A. If $Au = 0$, then u is in the null space of the matrix A. Hence the vector u belongs to the set $\text{nul}(A) \cap (\text{nul}(A))^{\perp}$, which implies $u = 0$.

[18]Let $K'A'w = u$, then we have $||u||_2^2 = u'K'A'w = w'AKu$. Since the range of matrix K is the null space of matrix A, we have $AKu = 0$ for all u, hence $||u||_2^2 = 0$, suggesting $u = 0$.

Consider the unconstrained problem (33), let Δy denote the exact Newton direction at y [cf. Eq. (61)], vector $\Delta\tilde{y}$ denote any direction in $\mathbb{R}^S$, $y(t) = y + t\Delta\tilde{y}$ and $\tilde{\lambda} = \sqrt{\Delta\tilde{y}'\nabla^2 F(y)\Delta\tilde{y}}$. Note that with the isomorphism established earlier, we have

$$\tilde{\lambda} = \sqrt{\Delta\tilde{y}'\nabla^2 F(y)\Delta\tilde{y}} = \sqrt{\Delta\tilde{y}'K'\nabla^2 f(Ky+\hat{x})K\Delta\tilde{y}} = \sqrt{\Delta\tilde{x}'\nabla^2 f(x)\Delta\tilde{x}},$$

where $x = Ky + \hat{x}$ and $\Delta\tilde{x} = K\Delta\tilde{y}$. From the assumption in the theorem, we have $\tilde{\lambda} < 1$. For any $t < 1$, $(y - y(t))'\nabla^2 F(y)(y - y(t)) = t^2\tilde{\lambda}^2 < 1$ and by Lemma 5.3 for any z_y in $\mathbb{R}^S$, we have

$$(1 - t\tilde{\lambda})^2 z_y'\nabla^2 F(y)z_y \le z_y'\nabla^2 F(y(t))z_y \le \frac{1}{(1 - t\tilde{\lambda})^2} z_y'\nabla^2 F(y)z_y$$

which implies

$$z_y'(\nabla^2 F(y(t)) - \nabla^2 F(y))z_y \le \left(\frac{1}{(1 - t\tilde{\lambda})^2} - 1\right) z_y'\nabla^2 F(y)z_y, \tag{65}$$

and

$$z_y'(\nabla^2 F(y) - \nabla^2 F(y(t)))z_y \le \left(1 - (1 - t\tilde{\lambda})^2\right) z_y'\nabla^2 F(y)z_y.$$

Using the fact that $1 - (1 - t\tilde{\lambda})^2 \le \frac{1}{(1-t\tilde{\lambda})^2} - 1$, the preceding relation can be rewritten as

$$z_y'(\nabla^2 F(y) - \nabla^2 F(y(t)))z_y \le \left(\frac{1}{(1 - t\tilde{\lambda})^2} - 1\right) z_y'\nabla^2 F(y)z_y. \tag{66}$$

Combining relations (65) and (66) yields

$$\left|z_y'(\nabla^2 F(y) - \nabla^2 F(y(t)))z_y\right| \le \left(\frac{1}{(1 - t\tilde{\lambda})^2} - 1\right) z_y'\nabla^2 F(y)z_y. \tag{67}$$

Since the function F is convex, the Hessian matrix $\nabla^2 F(y)$ is positive semidefinite. We can therefore apply the generalized Cauchy-Schwarz inequality and obtain

$$\begin{aligned} &\left|(\Delta\tilde{y})'(\nabla^2 F(y(t)) - \nabla^2 F(y))z_y\right| \\ &\le \sqrt{(\Delta\tilde{y})'(\nabla^2 F(y(t)) - \nabla^2 F(y))\Delta\tilde{y}'}\sqrt{z_y'(\nabla^2 F(y(t)) - \nabla^2 F(y))z_y} \\ &\le \left(\frac{1}{(1 - t\tilde{\lambda})^2} - 1\right)\sqrt{(\Delta\tilde{y})'\nabla^2 F(y)\Delta\tilde{y}}\sqrt{z_y'\nabla^2 F(y)z_y} \\ &= \left(\frac{1}{(1 - t\tilde{\lambda})^2} - 1\right)\tilde{\lambda}\sqrt{z_y'\nabla^2 F(y)z_y}, \end{aligned} \tag{68}$$

where the second inequality follows from relation (67), and the equality follows from definition of $\tilde{\lambda}$.

Define the function $\kappa : \mathbb{R} \to \mathbb{R}$, as $\kappa(t) = \nabla F(y(t))'z_y + (1 - t)(\Delta\tilde{y})'\nabla^2 F(y)'z_y$, then

$$\left|\frac{d}{dt}\kappa(t)\right| = \left|(\Delta\tilde{y})'\nabla^2 F(y(t))'z_y - (\Delta\tilde{y})'\nabla^2 F(y)z_y\right| = \left|(\Delta\tilde{y})'(\nabla^2 F(y(t)) - \nabla^2 F(y))z_y\right|,$$

which is the left hand side of (68).

Define $\gamma_y = \Delta y - \Delta\tilde{y}$, which by the isomorphism, implies $\gamma = \Delta x - \Delta\tilde{x} = K\gamma_y$. By rewriting $\Delta\tilde{y} = \Delta y - \gamma_y$ and observing the exact Newton direction Δy satisfies $\Delta y = -\nabla^2 F(y)^{-1}\nabla F(y)$ [cf.

Eq. (61)] and hence by symmetry of the matrix $\nabla^2 F(y)$, we have $\Delta y' \nabla^2 F(y) = \Delta y' \nabla^2 F(y)' = -\nabla F(y)'$, we obtain

$$\kappa(0) = \nabla F(y)' z_y + (\Delta\tilde{y})' \nabla^2 F(y)' z_y = \nabla F(y)' z_y - \nabla F(y)' z_y - \gamma_y' \nabla^2 F(y) z_y = -\gamma_y' \nabla^2 F(y) z_y.$$

Hence by integration, we obtain the bound

$$\begin{aligned} |\kappa(t)| &\leq \tilde{\lambda} \sqrt{z_y' \nabla^2 F(y) z_y} \int_0^t \left(\frac{1}{(1 - s\tilde{\lambda})^2} - 1 \right) ds + |\gamma_y' \nabla^2 F(y) z_y| \\ &= \frac{\tilde{\lambda}^2 t^2}{1 - \tilde{\lambda} t} \sqrt{z_y' \nabla^2 F(y) z_y} + |\gamma_y' \nabla^2 F(y) z_y|. \end{aligned}$$

For $t = 1$, $y(t) = y + \Delta\tilde{y}$, above equation implies

$$|\kappa(1)| \;\; = |\nabla F(y + \Delta\tilde{y})' z_y| \;\; \leq \frac{\tilde{\lambda}^2}{1 - \tilde{\lambda}} \sqrt{z_y' \nabla^2 F(y) z_y} + |\gamma_y' \nabla^2 F(y) z_y|.$$

We now specify z_y to be the exact Newton direction at $y + \Delta\tilde{y}$, then z_y satisfies $z_y' \nabla^2 F(y + \Delta\tilde{y}) z_y = |\nabla F(y + \Delta\tilde{y})' z_y|$, by using the definition of Newton direction at $y + \Delta\tilde{y}$ [cf. Eq. (62)], which proves

$$z_y \nabla^2 F(y + \Delta\tilde{y}) z_y \leq \frac{\tilde{\lambda}^2}{1 - \tilde{\lambda}} \sqrt{z_y' \nabla^2 F(y) z_y} + |\gamma_y' \nabla^2 F(y)' z_y|.$$

We now use the isomorphism once more to transform the above relation to the equality constrained problem domain. We have $z = K z_y$, the exact Newton direction at $x + \Delta\tilde{x} = \hat{x} + Ky + K\Delta\tilde{y}$. The left hand side becomes

$$z_y' \nabla^2 F(y + \Delta\tilde{y}) z_y = z_y' K' \nabla^2 f(x + \Delta\tilde{x}) K z_y = z' \nabla^2 f(x + \Delta\tilde{x}) z.$$

Similarly, we have the right hand sand satisfies

$$\begin{aligned} \frac{\tilde{\lambda}^2}{1 - \tilde{\lambda}} \sqrt{z_y' \nabla^2 F(y) z_y} + |\gamma_y' \nabla^2 F(y)' z_y| &= \frac{\tilde{\lambda}^2}{1 - \tilde{\lambda}} \sqrt{z_y' K' \nabla^2 f(x) K z_y} + |\gamma_y' K' \nabla^2 f(x) K z_y| \\ &= \frac{\tilde{\lambda}^2}{1 - \tilde{\lambda}} \sqrt{z' \nabla^2 f(x) z} + |\gamma' \nabla^2 f(x)' z|. \end{aligned}$$

By combining the above two relations, we have established the desired relation.

References

[1] S. Athuraliya and S. Low. Optimization flow control with Newton-like algorithm. *Journal of Telecommunication Systems*, 15:345–358, 2000.

[2] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, 2009.

[3] D. P. Bertsekas, A. Nedic, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, Cambridge, MA, 2003.

[4] D. Bickson, Y. Tock, A. Zymnis, S. Boyd, and D. Dolev. Distributed large scale network utility maximization. *Proceedings of the 2009 IEEE International Conference on Symposium on Information Theory*, 2, 2009.

[5] N. Biggs. *Algebraic Graph Theory.* Cambridge University Press, second edition, 1993.

[6] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[7] M. Chiang, S. H. Low, A. R. Calderbank, and J.C. Doyle. Layering as optimization decomposition: a mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[8] F. R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics).* No. 92, American Mathematical Society, 1997.

[9] R. Cottle, J. Pang, and R. Stone. *The Linear Complementarity Problem.* Academic Press, 1992.

[10] R. Dembo, S. Eisenstat, and T. Steihaug. Inexact Newton Methods. *SIAM Journal on Numerical Analysis*, 19, 1982.

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

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

[13] A. Jadbabaie, A. Ozdaglar, and M. Zargham. A Distributed Newton method for network optimization. *Proc. of CDC*, 2009.

[14] F. Jarre. Interior-point methods for convex programming. *Applied Mathematics and Optimization*, 26:287–311, 1992.

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

[16] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[17] F. P. Kelly, A. K. Maulloo, and D. K. Tan. Rate control for communication networks: shadow prices, proportional fairness, and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.

[18] D. C. Lay. *Linear Algebra and Its Applications.* Person Education, third edition, 2006.

[19] S. H. Low and D. E. Lapsley. Optimization flow control, I: basic algorithm and convergence. *IEEE/ACM Transaction on Networking*, 7(6):861–874, 1999.

[20] B. Mohar. Some applications of Laplace eigenvalues of graphs. *In: Hahn, G. and Sabidussi, G. (Eds.) Graph Symmetry: Algebraic Methods and Applications*, NATO ASI Series C 497:227–275, 1997.

[21] A. Nedic and A. Ozdaglar. *Convex Optimization in Signal Processing and Communications*, chapter Cooperative distributed multi-agent optimization. Eds., Eldar, Y. and Palomar, D., Cambridge University Press, 2008.

[22] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming.* SIAM, 2001.

[23] R. Srikant. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications).* Birkhäuser Boston, 2004.

[24] R. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Inc, Englewood Cliffs, NJ, 1965.

[25] E. Wei, A. Ozdaglar, and A. Jadbabaie. A distributed Newton method for Network Utility Maximization I, Algorithm. *LIDS Report 2832*, 2010.

[26] E. Wei, M. Zargham, A. Ozdaglar, and A. Jadbabaie. On dual convergence of the distributed Newton method for Network Utility Maximization. *LIDS Report 2868*, 2011.