A Distributed Newton Method for Network Optimization

Ali Jadbabaie[†], Asuman Ozdaglar[‡], and Michael Zargham[†]

***Abstract*—Most existing work uses dual decomposition and subgradient methods to solve network optimization problems in a distributed manner, which suffer from slow convergence rate properties. This paper proposes an alternative distributed approach based on a Newton-type method for solving minimum cost network optimization problems. The key component of the method is to represent the dual Newton direction as the solution of a discrete Poisson equation involving the graph Laplacian. This representation enables using an iterative consensus-based local averaging scheme (with an additional input term) to compute the Newton direction based only on local information. We show that even when the iterative schemes used for computing the Newton direction and the stepsize in our method are truncated, the resulting iterates converge superlinearly within an explicitly characterized error neighborhood. Simulation results illustrate the significant performance gains of this method relative to subgradient methods based on dual decomposition.**

I. Introduction

The standard approach to distributed optimization in networks is to use dual decomposition and subgradient (or first-order) methods, which for some classes of problems yields iterative algorithms that operate on the basis of local information (see [13], [14],[21], and [5]). However, a major shortcoming of this approach, particularly relevant in today's large-scale networks, is the slow convergence rate of the resulting algorithms. In this paper, we propose an alternative approach based on using Newton-type (or second-order) methods for minimum cost network optimization problems. We show that the proposed method can be implemented in a distributed manner and has faster convergence properties.

Consider a network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Each edge e in the network has a convex cost function $\phi_e(x^e)$, which captures the cost due to congestion effects as a function of the flow x^e on this edge. The total cost of a flow vector $x = [x^e]_{e\in\mathcal{E}}$ is given by the sum of the edge costs, i.e., $\sum_{e\in\mathcal{E}} \phi_e(x^e)$. Given an external supply b_i for each node $i \in \mathcal{N}$, the *minimum cost network optimization problem* is to find a minimum cost flow allocation vector that satisfies the flow conservation constraint at each node.[1] This problem can be formulated as a convex optimization problem with linear equality constraints. The application of dual decomposition together with a dual subgradient algorithm then yields a distributed iterative solution method. Instead, we propose a distributed primal-dual Newton-type method that achieves a superlinear convergence rate (to an error neighborhood).

The challenges in using Newton-type methods in this context are twofold. First, the superlinear convergence rate of this type of methods is achieved by using a backtracking stepsize rule, which relies on global information in the computation of the norm of a residual function (used in defining the stepsize). We solve this problem by using a consensus-based local averaging scheme for estimating the norm of the residual function.[2] Second, the computation of the dual Newton step involves a matrix inversion, which requires global information. Our main contribution in this regard is to develop a distributed iterative scheme for the computation of the dual Newton step. The key idea is to recognize that the dual Newton step is defined through a discrete Poisson equation (with constant input), which involves the Laplacian of the graph, and therefore can be solved using a consensus-based scheme in a distributed manner. We show that the convergence rate of this scheme is governed by the spectral properties of the underlying graph. Hence, for fast-mixing graphs (i.e., those with large spectral gaps), the dual Newton step can be computed efficiently using only local information.

Since our method uses consensus-based schemes to compute the stepsize and the Newton direction in each iteration, exact computation is not feasible. Another major contribution of our paper is to consider truncated versions of these consensus-schemes at each iteration and present convergence rate analysis of the constrained Newton method when the stepsize and the direction are estimated with some error. We show that when these errors are sufficiently small, the value of the residual function converges superlinearly to a neighborhood of the origin, whose size is explicitly quantified as a function of the errors and the parameters of the objective function and the constraints of the minimum cost network optimization problem. Our analysis relates to work on the convergence rate analysis of inexact Newton methods ([8], [12]). 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). Hence, our focus on providing convergence rate results to an error neighborhood may be of independent interest.

The rest of the paper is organized as follows: Section II defines the minimum cost network optimization problem and shows that the dual decomposition and subgradient method can be implemented in a distributed manner. This section also presents the constrained primal-dual Newton method for this problem and introduces a distributed iterative scheme for computing the dual Newton step. Section III presents a

A. Ozdaglar's research was supported by the National Science Foundation under Career grant DMI-0545910 and the DARPA ITMANET program. Ali Jadbabaie's research was supported by Army Research lab's MAST Collaborative Technology Alliance , DARPA/DSO SToMP, and ONR MURI N000140810747.

[†]Ali Jadbabaie (corresponding author,`jadbabai@seas.upenn.edu`) and Michael Zargham are with the Department of Electrical and Systems Engineering and GRASP Laboratory, University of Pennsylvania.

[‡] Asuman Ozdaglar is with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

[1]We focus on feasible problems, i.e., we assume that the total in-flow to the network is equal to the total out-flow, $\sum_{i\in\mathcal{N}} b_i = 0$.

[2]Consensus-based schemes have been used extensively in recent literature as distributed mechanisms for aligning (or computing the average of) the values held by multiple agents; see [10], [17],[18], and [19].

convergence rate analysis for an inexact Newton method, for which there are errors associated with computation of the step and the stepsize. In section IV, simulations demonstrate that the Newton's method outperforms the subgradient method with respect to runtime. Section V contains our concluding remarks.

Basic Notation and Notions:

A vector is viewed as a column vector, unless clearly stated otherwise. We denote by x^i the i-th component of a vector x. 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} or $[A]_{ij}$ to denote the matrix entry in the i-th row and j-th column. We write x' to denote the transpose of a vector x. The scalar product of two vectors $x, y \in \mathbb{R}^m$ is denoted by $x'y$. We use $\|x\|$ to denote the standard Euclidean norm, $\|x\| = \sqrt{x'x}$. For a vector-valued function $f : \mathbb{R}^n \to \mathbb{R}^m$, the gradient matrix of f at $x \in \mathbb{R}^n$ is denoted by $\nabla f(x)$.

A vector $a \in \mathbb{R}^m$ is said to be a *stochastic vector* when its components a_i, $i = 1, \ldots, m$, are nonnegative and their sum is equal to 1, i.e., $\sum_{i=1}^m a_i = 1$. A square $m \times m$ matrix A is said to be a *stochastic matrix* when each row of A is a stochastic vector. A stochastic matrix is called irreducible and aperiodic (also known as primitive) if all eigenvalues (except the trivial eigenvalue at 1) are subunit.

One can associate a discrete-time Markov chain with a stochastic matrix and a graph $\mathcal{G}$ as follows: The state of the chain at time $k \in \{1, 2, \cdots\}$, denoted by $X(k)$, is a node in $\mathcal{N}$ (the node set of the graph) and the weight associated to each edge in the graph is the probability with which X makes a transition between two adjacent nodes. In other words, the transition from state i to state j happens with probability p_{ij}, the weight of edge (i, j). If $\pi(k)$ with elements defined as $\pi_i(k) = \mathbb{P}(X(k) = i)$ is the probability distribution of the state at time k, the state distribution satisfies the recursion $\pi(k+1)^T = \pi(k)^T P$. If the chain is irreducible and aperiodic then for all initial distributions, π converges to the unique stationary distribution π^* [2].

II. Minimum Cost Network Optimization Problem

We consider a network represented by a directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with node set $\mathcal{N} = \{1, \ldots, N\}$, and edge set $\mathcal{E} = \{1, \ldots, E\}$. We denote the flow vector by $x = [x^e]_{e \in \mathcal{E}}$, where x^e denotes the flow on edge e. The flow conservation conditions at the nodes can be compactly expressed as$Ax = b$, where A is the $N \times E$ *node-edge incidence matrix* of the graph, i.e.,

$$A_{ij} = \begin{cases} 1 & \text{if edge } j \text{ leaves node } i \\ -1 & \text{if edge } j \text{ enters node } i \\ 0 & \text{otherwise,} \end{cases}$$

and the vector b denotes the external sources, i.e., $b_i > 0$ (or $b_i < 0$) indicates b_i units of external flow enters (or exits) node i. We associate a cost function $\phi_e : \mathbb{R} \to \mathbb{R}$ with each edge e, i.e., $\phi_e(x^e)$ denotes the cost on edge e as a function of the edge flow x^e. We assume that the cost functions ϕ_e are strictly convex and twice continuously differentiable. The minimum cost network optimization problem can be written as

$$\text{minimize} \sum_{e=1}^{E} \phi_e(x^e) \quad \text{subject to: } Ax = b \tag{1}$$

In this paper, our goal is to investigate *iterative distributed methods* for solving problem (1). In particular, we focus on two methods: first relies on solving the dual of problem (1) using a subgradient method; second uses a constrained Newton method, where, at each iteration, the Newton direction is computed iteratively using an averaging method.

A. Dual Subgradient Method

We first consider solving problem (1) using a dual subgradient method. To define the dual problem, we form the Lagrangian function of problem (1) $\mathcal{L} : \mathbb{R}^E \times \mathbb{R}^N \to \mathbb{R}$ given by $\mathcal{L}(x, \lambda) = \sum_{e=1}^E \phi_e(x^e) - \lambda'(Ax - b)$. The dual function $q(\lambda)$ becomes

$$\begin{aligned} q(\lambda) &= \inf_{x \in \mathbb{R}^E} \mathcal{L}(x, \lambda) = \inf_{x \in \mathbb{R}^E} \left(\sum_{e=1}^{E} \phi_e(x^e) - \lambda' A x \right) + \lambda' b \\ &= \sum_{e=1}^{E} \inf_{x^e \in \mathbb{R}} \left(\phi_e(x^e) - (\lambda' A)^e x^e \right) + \lambda' b. \end{aligned}$$

Hence, in view of the fact that the objective function and the constraints of problem (1) are separable in the decision variables x^e, the evaluation of the dual function decomposes into one-dimensional optimization problems. We assume that each of these optimization problems has an optimal solution, which is unique by the strict convexity of the functions ϕ_e and is denoted by $x^e(\lambda)$. Using the first order optimality conditions, it can be seen that for each e, $x^e(\lambda)$ is given by

$$x^e(\lambda) = (\phi_e')^{-1}(\lambda^i - \lambda^j), \tag{2}$$

where $i, j \in \mathcal{N}$ denote the end nodes of edge e. Thus, for each edge e, the evaluation of $x^e(\lambda)$ can be done based on local information about the edge cost function ϕ^e and the dual variables of the incident nodes i and j. We can write the dual problem as $\max_{\lambda \in \mathbb{R}^N} \; q(\lambda)$. The dual problem can be solved by using a subgradient method: given an initial vector λ_0, the iterates are generated by $\lambda_{k+1} = \lambda_k - \alpha_k g_k \quad$ for all $k \geq 0$, where g_k is a subgradient of the dual function $q(\lambda)$ at $\lambda = \lambda_k$ given by $g_k = Ax(\lambda_k) - b$, and $x(\lambda_k) = \operatorname{argmin}_{x \in \mathbb{R}^E} \mathcal{L}(x, \lambda_k)$, i.e., for all $e \in \mathcal{E}$, $x^e(\lambda_k)$ is given by Eq. (2) with $\lambda = \lambda_k$.

This method naturally lends itself to a distributed implementation: each node i updates its dual variable λ^i using local (subgradient) information g^i obtained from edges e incident to that node, which in turn updates its primal variables $x^e(\lambda)$ using dual variables of the incident nodes. Despite its simplicity and distributed nature, however, it is well-known that the dual subgradient method suffers from slow rate of convergence (see [16] and [15] for rate analysis and construction of primal solutions for dual subgradient methods), which motivates us to consider a Newton method for solving problem (1).

B. Equality-Constrained Newton Method

We next consider solving problem (1) using an (infeasible start) equality-constrained Newton method (see [4], Chapter 10). We let $f(x) = \sum_{e=1}^E \phi_e(x^e)$ for notational simplicity. Given an initial primal vector x_0, the iterates are generated by $x_{k+1} = x_k + \alpha_k v_k$ where v_k is the Newton step given as the solution to the following system of linear equations:[3]

$$\begin{pmatrix} \nabla^2 f(x_k) & A' \\ A & 0 \end{pmatrix} \begin{pmatrix} v_k \\ w_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) \\ Ax_k - b \end{pmatrix}.$$

[3]This is essentially a primal-dual method with the vectors v_k and w_k acting as primal and dual steps; see Section III.

We let $H_k = \nabla^2 f(x_k)$ and $h_k = Ax_k - b$ for notational convenience. Solving for v_k and w_k in the preceding yields $v_k = -H_k^{-1}(\nabla f(x_k) + A' w_k)$, and

$$(AH_k^{-1}A')w_k = h_k - AH_k^{-1}\nabla f(x_k). \tag{3}$$

Since the matrix H_k^{-1} is a diagonal matrix with entries $[H_k^{-1}]_{ee} = (\frac{\partial^2 \phi_e}{(\partial x^e)^2})^{-1}$, given the vector w_k, the Newton step v_k can be computed using local information. However, the computation of the vector w_k at a given primal vector x_k cannot be implemented in a decentralized manner in view of the fact that solving equation (3) $(AH_k^{-1}A')^{-1}$ requires global information. The following section provides an iterative scheme to compute the vector w_k using local information.

1) Distributed Computation of the Newton Direction: Consider the vector w_k defined in Eq. (3). The key step in developing a decentralized iterative scheme for the computation of the vector w_k is to recognize that the matrix $AH_k^{-1}A'$ is the weighted Laplacian of the underlying graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, denoted by L_k. Hence, L_k can be written as $L_k = AH_k^{-1}A' = D_k - B_k$. Here B_k is an $N \times N$ matrix with entries

$$(B_k)_{ij} = \begin{cases} \left(\frac{\partial^2 \phi_e}{(\partial x^e)^2}\right)^{-1} & \text{if } e = (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise,} \end{cases}$$

and D_k is an $N \times N$ diagonal matrix with entries $(D_k)_{ii} = \sum_{j \in \mathcal{N}_i} (B_k)_{ij}$, where $\mathcal{N}_i$ denotes the set of neighbors of node i, i.e., $\mathcal{N}_i = \{j \in \mathcal{N} \mid (i,j) \in \mathcal{E}\}$. Letting $s_k = h_k - AH_k^{-1}\nabla f(x_k)$ for notational convenience, Eq. (3) can be then rewritten as $(I - (D_k + I)^{-1}(B_k + I))w_k = (D_k + I)^{-1}s_k$. This motivates the following iterative scheme (known as splitting) to solve for w_k. For any $t \geq 0$, the iterates are generated by $w(t+1) = (D+I)^{-1}(B+I)w(t) + (D+I)^{-1}s$, (where we suppressed the indices k for notational convenience). Note that the matrix $P := (D+I)^{-1}(B+I)$ is row stochastic, and when the graph of the network is connected, it is irreducible and aperiodic (i.e., a primitive matrix) [9]. Furthermore, P is diagonally similar to a symmetric matrix, therefore all of its eigenvalues are real and by Perron-Frobenius theorem [2] they satisfy $1 = \lambda_1(P) > \lambda_2(P) \geq \cdots \geq \lambda_n(P) > -1$.Aside from one eigenvalue at 1 (corresponding to the all-one eigenvector $\mathbf{1}$), all other eigenvalues are subunit [9]. As a result, the projection of the dynamics to the orthogonal complement of the span of $\mathbf{1}$ is stable. Let V be the $n \times (n-1)$ dimensional matrix whose orthonormal $n-1$ columns span $\mathbf{1}^{\perp}$, the orthogonal subspace to $\mathbf{1}$. Denote $w(t) = V\bar{w}(t), s = V\bar{s}$, where $V'V = I_{n-1}$, and $VV' = I_n - \frac{\mathbf{1}\mathbf{1}^T}{n}$. The projected dynamics can now be written as $\bar{w}(t+1) = V'PV\bar{w}(t) + V'(D+I)^{-1}V\bar{s}$.

The above equation depicts the dynamics of a stable linear system with a constant (step) input, which from basic linear systems theory is known to converge to the solution of equation (3). The exponential convergence rate is determined by the convergence rate of the random walk $w(t+1) = Pw(t)$ to its stationary distribution. More precisely, the speed of convergence of $w(t)$ to the stationary distribution, depends on the eigenstructure of the probability transition (weight) matrix P.

2) Convergence Rates: It is well known that the rate of convergence to the stationary distribution of a Markov chain is governed by the second largest eigenvalue modulus of matrix P defined as $\mu(P) = \max_{i=2,\cdots,n}\{|\lambda_i(P)|\} = \max\{\lambda_2(P), |\lambda_n(P)|\}$. To make this statement more precise, let i be the initial state and define the *total variation distance* between the distribution at time k and the stationary distributionπ^* as $\Delta_i(k) = \frac{1}{2}\sum_{j \in \mathcal{V}} |P_{ij}^k - \pi_j^*|$. The rate of convergence to the stationary distribution is measured using the following quantity known as the *mixing time*:

$$T_{mix} = \max_i \min\{k : \Delta_i(k') < e^{-1} \text{ for all } k' \geq k\}.$$

The following theorem indicates the relationship between the mixing time of a Markov chain and the second largest eigenvalue modulus of its transition matrix [20], [1].

Theorem 1: The mixing time of a reversible Markov chain with transition probability matrix W and second largest eigenvalue modulus μ satisfies $\frac{\mu}{2(1-\mu)}(1 - \ln 2) \leq T_{mix} \leq \frac{1+\log n}{1-\mu}$.

Therefore, the speed of convergence of the Markov chain to its stationary distribution is determined by the value of $1 - \mu$ known as the *spectral gap*; the larger the spectral gap, the faster the convergence. This suggests that for each iteration k, the dual Newton step w_k can be computed faster on graphs with large spectral gap.

The above convergence results indicate that the dual Newton step can be computed in a decentralized fashion with a diffusion scheme, or more precisely, with a discrete Poisson equation. Since our distributed solution involves an iterative scheme, exact computation is not feasible. In what follows, we show that the Newton method has desirable convergence properties even when the Newton direction is computed with some error provided that the errors are sufficiently small.

III. Inexact Newton Method

In this section, we consider the following convex optimization problem with equality constraints:

$$\text{minimize } f(x) \quad \text{subject to: } Ax = b \tag{4}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable convex function, and A is an $m \times n$ matrix. The minimum cost network optimization problem (1) is a special case of this problem with $f(x) = \sum_{e=1}^{E} \phi_e(x^e)$ and A is the $N \times E$ node-edge incidence matrix. We denote the optimal value of this problem by f^*. Throughout this section, we assume that the value f^* is finite and problem (4) has an optimal solution, which we denote by x^*.

We consider an *inexact (infeasible start) Newton method* for solving problem (4) (see [4]). In particular, we let $y = (x, \nu) \in \mathbb{R}^n \times \mathbb{R}^m$, where x is the primal variable and ν is the dual variable, and study a primal-dual method which updates the vector y at iteration k as follows:

$$y_{k+1} = y_k + \alpha_k d_k, \tag{5}$$

where α_k is a positive stepsize, and the vector d_k is an *approximate constrained Newton direction* given by

$$Dr(y_k)d_k = -r(y_k) + \epsilon_k. \tag{6}$$

Here, the residual function $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ is defined as

$$\begin{aligned} r(x,\nu) &= (r_{dual}(x,\nu), r_{pri}(x,\nu)), & (7)\\ r_{dual}(x,\nu) &= \nabla f(x) + A'\nu & (8)\\ r_{pri}(x,\nu)) &= Ax - b. & (9) \end{aligned}$$

Moreover, $Dr(y) \in \mathbb{R}^{(n+m)\times(n+m)}$ is the gradient matrix of r evaluated at y, and the vector ϵ_k is an error vector at iteration k. We assume that the error sequence $\{\epsilon_k\}$ is uniformly bounded from above, i.e., there exists a scalar $\epsilon \geq 0$ such that

$$\|\epsilon_k\| \leq \epsilon \text{ for all } k \geq 0. \qquad (10)$$

We adopt the following standard assumption:

Assumption 1: Let $r : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n \times \mathbb{R}^m$ be the residual function defined in Eqs. (7)-(9). Then, we have:

(a) *(Lipschitz Condition) There exists some constant $L > 0$ such that $\|Dr(y) - Dr(\bar{y})\| \leq L\|y - \bar{y}\| \forall y, \bar{y} \in \mathbb{R}^n \times \mathbb{R}^m$.*
(b) *There exists some constant $M > 0$ such that*
$$\|Dr(y)^{-1}\| \leq M \qquad \forall y \in \mathbb{R}^n \times \mathbb{R}^m.$$

A. Basic Relation

We use the norm of the residual vector $\|r(y)\|$ to measure the progress of the algorithm. In the next proposition, we present a relation between the iterates $\|r(y_k)\|$, which holds for any stepsize rule. The proposition follows from a multi-dimensional extension of the descent lemma (see [3]).

Proposition 1: Let Assumption 1 hold. Let $\{y_k\}$ be a sequence generated by the method (5). For any stepsize rule α_k, we have $\|r(y_{k+1})\| \leq (1 - \alpha_k)\|r(y_k)\| + M^2 L\alpha_k^2 \|r(y_k)\|^+ \alpha_k\|\epsilon_k\| + M^2 L\alpha_k^2\|\epsilon_k\|^2$.

Proof: We consider two vectors $w \in \mathbb{R}^n \times \mathbb{R}^m$ and $z \in \mathbb{R}^n \times \mathbb{R}^m$. We let ξ be a scalar parameter and define the function $g(\xi) = r(w + \xi z)$. From the chain rule, it follows that $\nabla g(\xi) = Dr(w + \xi z)z$. Using the Lipschitz continuity of the residual function gradient [cf. Assumption 1(a)], we obtain: $r(w+z) - r(w) = g(1) - g(0) = \int_0^1 \nabla g(\xi) d\xi = \int_0^1 Dr(w + \xi z) z d\xi$

$$\begin{aligned}
&\leq \left| \int_0^1 (Dr(w + \xi z) - Dr(w)) z d\xi \right| + \int_0^1 Dr(w) z d\xi \\
&\leq \int_0^1 \|Dr(w + \xi z) - Dr(w)\| \, \|z\| d\xi + Dr(w) z \\
&\leq \|z\| \int_0^1 L\xi \|z\| d\xi + Dr(w) z = \frac{L}{2}\|z\|^2 + Dr(w) z.
\end{aligned}$$

We apply the preceding relation with $w = y_k$ and $z = \alpha_k d_k$ and obtain $r(y_k + \alpha_k d_k) - r(y_k) \leq \alpha_k Dr(y_k) d_k + \frac{L}{2}\alpha_k^2 \|d_k\|^2$. By Eq. (6), we have $Dr(y_k) d_k = -r(y_k) + \epsilon_k$. Substituting this in the previous relation, this yields $r(y_k + \alpha_k d_k) \leq (1 - \alpha_k) r(y_k) + \alpha_k \epsilon_k + \frac{L}{2}\alpha_k^2\|d_k\|^2$. Moreover, using Assumption 1(b), we have $\|d_k\|^2 = \|Dr(y_k)^{-1}(-r(y_k) + \epsilon_k)\|^2$
$\leq \|Dr(y_k)^{-1}\|^2 \, \|-r(y_k) + \epsilon_k\|^2 \leq M^2 \Big(2\|r(y_k)\|^2 + 2\|\epsilon_k\|^2\Big)$

Combining the above relations, we obtain $\|r(y_{k+1})\| \leq (1 - \alpha_k)\|r(y_k)\| + M^2 L\alpha_k^2 \|r(y_k)\|^2 + \alpha_k \|\epsilon_k\| + M^2 L \alpha_k^2 \|\epsilon_k\|^2$, establishing the desired relation. ■

B. Inexact Backtracking Stepsize Rule

We use a backtracking stepsize rule in our method to achieve the superlinear local convergence properties of the Newton method. However, this requires computation of the norm of the residual function $\|r(y)\|$. In view of the distributed nature of the residual vector $r(y)$ [cf. Eq. (7)], this norm can be computed using a distributed consensus-based scheme. Since this scheme is iterative, in practice the residual norm can only be estimated with some error.

Let $\{y_k\}$ be a sequence generated by the inexact Newton method (5). At each iteration k, we assume that we can compute the norm $\|r(y_k)\|$ with some error, i.e., we can compute a scalar $n_k \geq 0$ that satisfies

$$\Big| n_k - \|r(y_k)\| \Big| \leq \gamma/2, \qquad (11)$$

for some constant $\gamma \geq 0$. Hence, n_k is an approximate version of $\|r(y_k)\|$, which can be computed for example using distributed iterative methods. For fixed scalars $\sigma \in (0, 1/2)$ and $\beta \in (0, 1)$, we set the stepsize α_k equal to $\alpha_k = \beta^{m_k}$, where m_k is the smallest nonnegative integer that satisfies

$$n_{k+1} \leq (1 - \sigma\beta^m) n_k + B + \gamma. \qquad (12)$$

Here, γ is the maximum error in the residual function norm [cf. Eq. (11)], and B is a constant given by

$$B = \epsilon + M^2 L \epsilon^2, \qquad (13)$$

where ϵ is the upper bound on the error sequence in the constrained Newton direction [cf. Eq. (10)] and M and L are the constants in Assumption 1.

C. Global Convergence of the Inexact Newton Method

The next two sections provide convergence rate estimates for the damped Newton phase and the local convergence phase of the inexact Newton method when there are errors in the Newton direction and the backtracking stepsize.

1) Convergence Rate for Damped Newton Phase: We first show a strict decrease in the norm of the residual function if the errors ϵ and γ are sufficiently small, as quantified in the following assumption.

Assumption 2: The errors B and e [cf. Eqs. (13) and (11)] satisfy

$$B + 2\gamma \leq \frac{\beta}{16 M^2 L},$$

where β is the constant used in the inexact backtracking stepsize rule, and M and L are the constants defined in Assumption 1.

Under this assumption, the next proposition establishes a strict decrease in the norm of the residual function as long as $\|r(y)\| > \frac{1}{2M^2 L}$.

Proposition 2: Let Assumptions 1 and 2 hold. Let $\{y_k\}$ be a sequence generated by the method (5) when the stepsize sequence $\{\alpha_k\}$ is selected using the inexact backtracking stepsize rule [cf. Eq. (12)]. Assume that $\|r(y_k)\| > \frac{1}{2M^2L}$. Then, we have

$$\|r(y_{k+1})\| \leq \|r(y_k)\| - \frac{\beta}{16 M^2 L}.$$

Proof: For any $k \geq 0$, we define $\bar{\alpha}_k = \frac{1}{2M^2 L(n_k + \gamma/2)}$. In view of the condition on n_k [cf. Eq. (11)], we have

$$\frac{1}{2M^2 L(\|r(y_k)\| + \gamma)} \leq \bar{\alpha}_k \leq \frac{1}{2M^2 L\|r(y_k)\|} < 1, \qquad (14)$$

where the last inequality follows by the assumption $\|r(y_k)\| > \frac{1}{2M^2L}$. Using the preceding relation and substituting $\alpha_k = \bar{\alpha}_k$ in the basic relation in Proposition 1, we obtain:

$$\begin{aligned}
\|r(y_{k+1})\| &\leq \|r(y_k)\| + \bar{\alpha}_k\|\epsilon_k\| + M^2 L\bar{\alpha}_k^2\|\epsilon_k\|^2 \\
&\quad -\bar{\alpha}_k\|r(y_k)\|\Big(1 - M^2 L\bar{\alpha}_k\|r(y_k)\|\Big) \\
&\leq \|r(y_k)\| + \bar{\alpha}_k\|\epsilon_k\| + M^2 L\bar{\alpha}_k^2\|\epsilon_k\|^2 \\
&\quad -\bar{\alpha}_k\|r(y_k)\|\left(1 - M^2 L\frac{\|r(y_k)\|}{2M^2 L\|r(y_k)\|}\right) \\
&\leq \bar{\alpha}_k\|\epsilon_k\| + M^2 L\bar{\alpha}_k^2\|\epsilon_k\|^2 + \left(1 - \frac{\bar{\alpha}_k}{2}\right)\|r(y_k)\| \\
&\leq B + \left(1 - \frac{\bar{\alpha}_k}{2}\right)\|r(y_k)\|,
\end{aligned}$$

where the second inequality follows from the definition of $\bar{\alpha}_k$ and the third inequality follows by combining the facts $\bar{\alpha}_k < 1$, $\|\epsilon_k\| \leq \epsilon$ for all k, and the definition of B. The constant σ used in the definition of the inexact backtracking line search satisfies $\sigma \in (0, 1/2)$, therefore, it follows from the preceding relation that $\|r(y_{k+1})\| \leq (1 - \sigma\bar{\alpha}_k)\|r(y_k)\| + B$. Using condition (11) once again, this implies $n_{k+1} \leq (1 - \sigma\bar{\alpha}_k)n_k + B + \gamma$, showing that the steplength α_k selected by the inexact backtracking line search satisfies $\alpha_k \geq \beta\bar{\alpha}_k$. From condition (12), we have $n_{k+1} \leq (1 - \sigma\alpha_k)n_k + B + \gamma$, which implies $\|r(y_{k+1})\| \leq (1 - \sigma\beta\bar{\alpha}_k)\|r(y_k)\| + B + 2\gamma$. Combined with Eq. (14), this yields

$$\|r(y_{k+1})\| \leq \left(1 - \frac{\sigma\beta}{2M^2L(\|r(y_k)\| + \gamma)}\right)\|r(y_k)\| + B + 2\gamma.$$

By Assumption 2, we also have $\gamma \leq B + 2\gamma \leq \frac{\beta}{16M^2L}$, which in view of the assumption $\|r(y_k)\| > \frac{1}{2M^2L}$ implies that $\gamma \leq \|r(y_k)\|$. Substituting this in the preceding relation and using the fact $\alpha \in (0, 1/2)$, we obtain $\|r(y_{k+1})\| \leq \|r(y_k)\| - \frac{\beta}{8M^2L} + B + 2\gamma$. Combined with Assumption 2, this yields the desired result. ■

The preceding proposition shows that, under the assumptions on the size of the errors, at each iteration, we obtain a minimum decrease (in the norm of the residual function) of $\frac{\beta}{16M^2L}$, as long as $\|r(y_k)\| > 1/2M^2L$. This establishes that we need at most $\frac{16\|r(y_0)\|M^2L}{\beta}$, iterations until we obtain $\|r(y_k)\| \leq 1/2M^2L$.

2) Convergence Rate for Local Convergence Phase: In this section, we show that when $\|r(y_k)\| \leq 1/2M^2L$, the inexact backtracking stepsize rule selects a full step $\alpha_k = 1$ and the norm of the residual function $\|r(y_k)\|$ converges quadratically within an error neighborhood, which is a function of the parameters of the problem (as given in Assumption 1) and the error level in the constrained Newton direction.

Proposition 3: Let Assumption 1 hold. Let $\{y_k\}$ be a sequence generated by the method (5) when the stepsize sequence $\{\alpha_k\}$ is selected using the inexact backtracking stepsize rule [cf. Eq. (12)]. Assume that there exists some k such that $\|r(y_k)\| \leq \frac{1}{2M^2L}$. Then, the inexact backtracking stepsize rule selects $\alpha_k = 1$. We further assume that for some $\delta \in (0, 1/2)$, $B + M^2LB^2 \leq \frac{\delta}{4M^2L}$, where B is the constant defined in Eq. (13). Then, we have

$$\|r(y_{k+m})\| \leq \frac{1}{2^{2^m}M^2L} + B + \frac{\delta}{M^2L}\frac{(2^{2^m-1} - 1)}{2^{2^m}} \qquad (15)$$

for all $m > 0$. As a particular consequence, we obtain

$$\limsup_{m\to\infty} \|r(y_m)\| \leq B + \frac{\delta}{2M^2L}.$$

Proof: We first show that if $\|r(y_k)\| \leq \frac{1}{2M^2L}$ for some $k > 0$, then the inexact backtracking stepsize rule selects $\alpha_k = 1$. Replacing $\alpha_k = 1$ in the basic relation of Proposition 1 and using the definition of the constant B, we obtain $\|r(y_{k+1})\| \leq M^2L\|r(y_k)\|^2 + B \leq \frac{1}{2}\|r(y_k)\| + B \leq (1 - \sigma)\|r(y_k)\| + B$, where to get the last inequality, we used the fact that the constant σ used in the inexact backtracking stepsize rule satisfies $\sigma \in (0, 1/2)$. Using the condition on n_k [cf. Eq. (11)], this yields $n_{k+1} \leq (1 - \sigma)n_k + B + \gamma$, showing that the steplength $\alpha_k = 1$ satisfies condition (12) in the inexact backtracking stepsize rule.

We next show Eq. (15) using induction on the iteration m. Using $\alpha_k = 1$ in the basic relation of Proposition 1, we obtain

$$\|r(y_{k+1})\| \leq \frac{1}{2}\|r(y_k)\| + B \leq \frac{1}{4M^2L} + B,$$

where the second inequality follows from the assumption $\|r(y_k)\| \leq \frac{1}{2M^2L}$. This establishes relation (15) for $m = 1$.

We next assume that (15) holds for some $m > 0$, and show that it also holds for $m + 1$. Eq. (15) implies that

$$\|r(y_{k+m})\| \leq \frac{1}{4M^2L} + B + \frac{\delta}{4M^2L}.$$

Using the assumption $B + M^2LB^2 \leq \frac{\delta}{4M^2L}$, this yields

$$\|r(y_{k+m})\| \leq \frac{1 + 2\delta}{4M^2L} < \frac{1}{2M^2L},$$

where the strict inequality follows from $\delta \in (0, 1/2)$. Hence, the inexact backtracking stepsize rule selects $\alpha_{k+m} = 1$. Using $\alpha_{k+m} = 1$ in the basic relation, we obtain

$$M^2L\|r(y_{k+m+1})\| \leq \left(M^2L\|r(y_{k+m})\|\right)^2 + B.$$

Using Eq. (15), this implies that $M^2L\|r(y_{k+m+1})\|$

$$\begin{aligned}
&\leq \left(\frac{1}{2^{2^m}} + M^2LB + \frac{\delta(2^{2^m-1} - 1)}{2^{2^m}}\right)^2 + M^2LB \\
&= \frac{1}{2^{2^{m+1}}} + \frac{M^2LB}{2^{2^m-1}} + \delta\,\frac{2^{2^m-1} - 1}{2^{2^{m+1}-1}} \\
&\quad + M^2L\left(B + \frac{\delta}{M^2L}\frac{(2^{2^m-1} - 1)}{2^{2^m}}\right)^2 + M^2LB.
\end{aligned}$$

Using algebraic manipulations and the assumption $B + M^2LB^2 \leq \frac{\delta}{4M^2L}$, this yields

$$\|r(y_{k+m+1})\| \leq \frac{1}{2^{2^{m+1}}M^2L} + B + \frac{\delta}{M^2L}\frac{(2^{2^{m+1}-1} - 1)}{2^{2^{m+1}}},$$

completing the induction and therefore the proof of relation (15). Taking the limit superior in Eq. (15) establishes the final result. ■

IV. Simulation Results

Our simulation results demonstrate that the decentralized Newton significantly outperforms the dual subgradient method algorithm in terms of runtime. Simulations were conducted as follows: Network flow problems with conservation constraints and the cost function $\Phi(x) = \sum_{e=1}^{E}\phi_e(x^e)$ where $\phi_e(x^e) = 1 - \sqrt{1 - (x^e)^2}$ [4] were generated on Erdös-Rényi random graphs with $n = 10$, 20, 80, and 160 nodes and an expected node degree, $np = 5$. We limit the simulations to optimization problems which are well behaved in the sense that the Hessian matrix remains well conditioned, defined as $\frac{\lambda_{\max}}{\lambda_{\min}} \leq 200$. For this subclass of problem, the runtime of the Newton's method algorithm is significantly less than the subgradient method for all trials. Note that the stopping criterion is also tested in a distributed manner for both algorithms.

In any particular experiment the Newton's method algorithm discovers the feasible direction in one iteration and proceeds to find the optimal solution in two to four additional iterations. One such experiment is shown in Figure 1. A sample runtime distribution for 150 trials is presented in Figure 2. The fact that of course Newton's method outperform the subgradient scheme is not surprising. Perhaps the surprising fact is that Newton's method outperforms subgradient scheme, despite the

[4]the cost function is motivated by the Kuramoto model of coupled nonlinear oscillators [11].

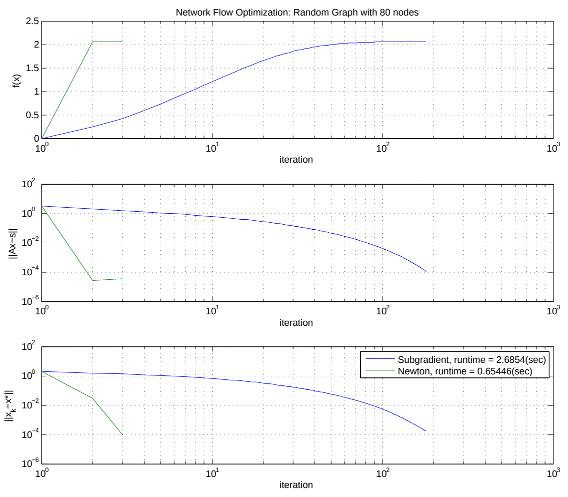

Fig. 1. Sample convergence trajectory for a random graph with 80 nodes and mean node degree 5.

fact the computation of the dual Newton step is based on a discrete Poisson equation (essentially diffusion with input).

On average the Newton's method terminates in less than half of the subgradient runtime and exhibits a tighter variance. This is a representative sample with respect to varying the number of nodes. As shown in Figure 3, the Newton's method algorithm completes in significantly less time on average for all of the graphs evaluated.

V. Conclusions

This paper develops a distributed Newton-type method for solving minimum cost network optimization problems that can achieve the superlinear convergence rate within some error neighborhood. We show that due to the sparsity structure of the incidence matrix of a network, the computation of the dual Newton step can be performed using a discrete Poisson equation. This enables using distributed consensus schemes to compute the dual Newton direction. We show that even when the Newton direction and stepsize are computed with some error, the method achieves superlinear convergence rate to an error neighborhood. Our simulation experiments on different graphs suggested the superiority of the proposed Newton scheme to standard dual subgradient methods. In ongoing work, we extend this approach to Network Utility Maximization (NUM) problems. Furthermore, we believe new results in [7] will enable us to solve the discrete Poisson equation for the dual Newton step using modified PageRank algorithms in a more scalable fashion [6].

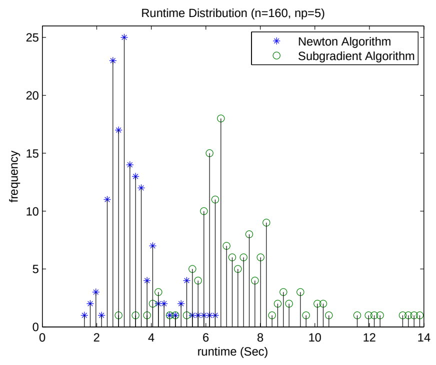

Fig. 2. Runtime Histogram, 160 node graphs with mean node degree 5

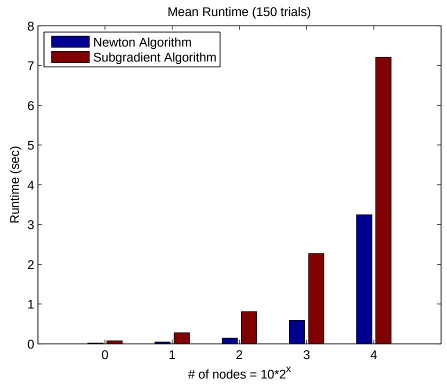

Fig. 3. Average Runtime, 150 samples each

References

[1] D. Aldous, *Some inequalities for reversible Markov chains*, Journal of the London Mathematical Society **25** (1982), 564–576.

[2] A. Berman and R. J. Plemmons, *Nonnegative matrices in the mathematical sciences*, Academic Press, New York, 1979.

[3] D.P. Bertsekas, A. Nedić, and A.E. Ozdaglar, *Convex analysis and optimization*, Athena Scientific, Cambridge, Massachusetts, 2003.

[4] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, Cambridge, UK, 2004.

[5] 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** (2007), no. 1, 255–312.

[6] F. Chung-Graham, *PageRank as a discrete Green's function*, preprint.

[7] ______, *The heat kernel as the pagerank of a graph*, Proceedings of the National Academy of Sciences **104** (2007), no. 50, 19735.

[8] R. Dembo, S. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM Journal on Numerical Analysis **19** (1982), 400–408.

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

[10] A. Jadbabaie, J. Lin, and S. Morse, *Coordination of groups of mobile autonomous agents using nearest neighbor rules*, IEEE Transactions on Automatic Control **48** (2003), no. 6, 988–1001.

[11] A. Jadbabaie, N. Motee, and M Barahona, *On the stability of Kuramoto model of coupled nonlinear oscillators*, Proceedings of the American Control Conference, June 2004.

[12] C.T. Kelley, *Iterative methods for linear and nonlinear equations*, SIAM, Philadelphia, PA, 1995.

[13] 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** (1998), 237–252.

[14] S. Low and D.E. Lapsley, *Optimization flow control, I: Basic algorithm and convergence*, IEEE/ACM Transactions on Networking **7** (1999), no. 6, 861–874.

[15] A. Nedić and A. Ozdaglar, *Approximate primal solutions and rate analysis for dual subgradient methods*, SIAM Journal on Optimization, forthcoming (2008).

[16] ______, *Subgradient methods in network resource allocation: Rate analysis*, Proc. of CISS, 2008.

[17] R. Olfati-Saber and R.M. Murray, *Consensus problems in networks of agents with switching topology and time-delays*, IEEE Transactions on Automatic Control **49** (2004), no. 9, 1520–1533.

[18] A. Olshevsky and J.N. Tsitsiklis, *Convergence rates in distributed consensus averaging*, Proceedings of IEEE CDC, 2006.

[19] ______, *Convergence speed in distributed consensus and averaging*, SIAM Journal on Optimization, forthcoming (2006).

[20] A. J. Sinclair, *Improved bounds for mixing rates of Markov chains and multicommodity flow*, Combinatorics, Probability and Computing **1** (1992), 351–370.

[21] R. Srikant, *Mathematics of Internet congestion control*, Birkhauser, 2004.