MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Semismooth Equation Approach to Network Utility Maximization (NUM)

Bai, L.; Raghunathan, A.U.

TR2013-060 June 2013

Abstract

Popular approach to solving NUM utilizes dual decomposition and subgradient iterations, which are extremely slow to converge. Recently there has been investigation of barrier methods for the solution of NUM which have been shown to possess second order convergence. However, the question of accelerating dual decomposition based methods is still open. We propose a novel semismooth equation approach to solving the standard dual decomposition formulation of NUM. We show that under fairly mild assumptions that the approach converges locally superlinearly to the solution of the NUM. Globalization of the proposed algorithm using a linesearch is also described. Numerical experiments show that the approach is competitive with a state-of-the-art nonlinear programming solver which solves the NUM without decomposition.

American Control Conference (ACC)

Semismooth Equation Approach to Network Utility Maximization (NUM)

Lijie Bai[1,2] and Arvind U. Raghunathan[2]

Abstract— **Popular approach to solving NUM utilizes dual decomposition and subgradient iterations, which are extremely slow to converge. Recently there has been investigation of barrier methods for the solution of NUM which have been shown to posess second order convergence. However, the question of accelerating dual decomposition based methods is still open. We propose a novel semismooth equation approach to solving the standard dual decomposition formulation of NUM. We show that under fairly mild assumptions that the approach converges locally superlinearly to the solution of the NUM. Globalization of the proposed algorithm using a linesearch is also described. Numerical experiments show that the approach is competitive with a state-of-the-art nonlinear programming solver which solves the NUM without decomposition.**

I. Introduction

Network Utility Maximization (NUM) problems are characterized by a fixed network and a set of sources, which send information over the network along predetermined routes. Each source has a local utility function over the rate at which it sends information. The goal is to determine the source rates that maximize the sum of utilities subject to link capacity constraints. This work focusses on NUM problem which arise from rate control problem in wireline networks [1]. NUM has largely been solved by dual decomposition and subgradient iterations [2], [3], [4]. The popularity of this method stems from the distributed computational framework provided by dual decomposition. However, the method performs poorly in practice due to the sublinear rate of convergence of the subgradient method [5]. Low and Lapsley [2] proposed a modified subgradient method with improved convergence rates but well short of the second-order rates of convergence of Newton's method.

Recently, [6], [7], [8] considered solving NUM using a barrier approach. The barrier formulation converts the NUM into an equality constrained problem to which Newton's method is applied. The second order convergence follows as a consequence. The difference between the above papers is in the manner in which the step computation is distributed. Belief propagation is used to solve the linear system in [6]. There are no guarantees of convergence for this approach. On the other hand, [7], [8] employ matrix splitting techniques which have rigorous convergence guarantees.

A. Our Contribution

We propose a novel second order method that relies on the dual decomposition framework. In a significant departure from previous approaches, we pose the problem of computing the optimal dual multipliers as a complementarity system. We show that this complementarity system is *semismooth* under mild assumptions. The semismooth equation framework allows us to employ the generalized Newton's method which also yields superlinear rate of convergence. In order to compute the Newton direction we require that the individual subproblems provide not only the solution but also sensitivity of the solution to the chosen dual multiplier values. The sensitivities are shown to exist under standard assumptions and can be computed analytically. The case of distributed computations is not considered in this work and will be the subject of a future study.

B. Notation

All vectors will be assumed to column vectors. For a vector $x \in \mathbb{R}^n$, x_i will denote the i-th entry of the vector and and x^T will denote its transpose. For a matrix $A \in \mathbb{R}^{n\times m}$ A_i will denote the i-th row of the matrix and A_i^T its transpose. If $\alpha \subseteq \{1,\dots,n\}$ and $\beta \subseteq \{1.\dots,m\}$ then $A_{\alpha\beta} \in \mathbb{R}^{|\alpha|\times|\beta|}$ will denote the submatrix formed from elements of A in rows indexed by α and columns indexed by β. For a function $f : \mathbb{R}^n \to \mathbb{R}$, $\nabla_x f(x)$ and $\nabla_x^2 f(x)$ will denote the gradient and hessian of the function w.r.t x.

C. Organization of paper

The rest of the paper is organized as follows. Section II describes the NUM problem, our complementarity equation formulation and shows the equivalence between the two. Section III shows that the complementarity formulation is semismooth using nonlinear programming sensivity. Section IV outlines the algorithm and presents convergence results. Computational results are provided in Section V, followed by conclusions in Section VI.

II. NUM as Complementarity Equations

The NUM problem is formulated as,

$$\max_{x\ge 0} \sum_{i=1}^{N} U_i(x_i) \ \text{ s.t. } \ Rx \le c \tag{1}$$

where $x \in \mathbb{R}^S$ is the vector of source rates, $U_i : \mathbb{R}_+ \to \mathbb{R}$ is a twice continuously differentiable, strictly concave, monotonically increasing objective function, the matrix $R \in \mathbb{R}^{L\times N}$ is a binary matrix where the 1-entries in row R_l indicates the sources sharing the link l and $c_l > 0 : l \in \{1,\dots,L\}$ is the capacity of the link.

The dual decomposition approach for solving the NUM problem (1) dualizes the constraints $Rx \le c$ using multipliers

[1] Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180. bail@rpi.edu

[2] Mitsubishi Electric Research Laboratories, Cambridge,MA 02139. raghunathan@merl.com

$\lambda \in \mathbb{R}^L_+$ to obtain the dual function $g(\lambda)$,

$$g(\lambda) := \min_{x \geq 0} \sum_{i=1}^{N} f_i(x_i) + \lambda^T(Rx - c) \tag{2}$$

where $f_i(x_i) = -U_i(x_i)$. The dualization renders (2) separable in the x variables. We will denote the solution to (2) as $x(\lambda)$. To obtain the optimal value of λ^* that solves (1) the dual function needs to be maximized

$$\max_{\lambda \geq 0} g(\lambda). \tag{3}$$

The dual function $g(\lambda)$ is in general nonsmooth and hence, the subgradient method is employed for solving (3) [2]-[4].

We propose to solve NUM by posing it as the following complementarity problem

$$c - Rx(\lambda) \geq 0 \perp \lambda \geq 0. \tag{4}$$

Lemma 1 shows the equivalence between problems (1) and (4). Prior to that we will state the assumptions on solution to (1). The first order stationary conditions [9] of (1) are

$$\begin{array}{l} \nabla_x f(x) + R^T\lambda - \nu = 0 \\ 0 \leq \nu \perp x \geq 0 \\ 0 \leq \lambda \perp c - Rx \geq 0. \end{array} \tag{5}$$

where $f(x) = \sum_{i=1}^{N} f_i(x_i)$ and (λ, ν) are respectively the multipliers corresponding to the inequality constraints and bounds. We will denote by (x^*, λ^*, ν^*) the solution to (1) with its associated multipliers.

Assumption 1: Each $f_i(x_i) = -U_i(x_i)$ from the objective function $\sum_{i=1}^{N} f_i(x_i)$ of the NUM is strictly convex. Further, f_i are at least twice, Lipschitz continuously differentiable.

Assumption 2: Linear Independence Constraint Qualification (LICQ) [9] holds at the solution point x^*.

Assumption 3: $x^* + \nu^* > 0$, which means that x^* and ν^* are strictly complementary to each other.

The following shows the equivalence between (1) and (4).

Lemma 1: Suppose Assumption 1 holds. Then, x^* solves (1) with multipliers (λ^*, ν^*) if and only if λ^* solves (4).

Proof: A solution $x(\lambda)$ to (2) satisfies,

$$\begin{array}{l} \nabla_x f(x(\lambda)) + R^T\lambda - \nu(\lambda) = 0 \\ 0 \leq \nu(\lambda) \perp x(\lambda) \geq 0. \end{array}$$

Since (2) is strictly convex (Assumption 1), we have that above first order conditions are also sufficient [9]. Consequently, if λ solves (4) then, it is easy to see that $(x(\lambda), \lambda, \nu(\lambda))$ also satisfies (5). Strict convexity of (1) (Assumption 1) implies that (5) are also sufficient for (1). The claim follows. ■

Under assumptions of strict convexity of f_i and LICQ (Assumptions 1-2), it is easy to show that (x^*, λ^*, ν^*) uniquely solves NUM (1)[9]. The following lemma shows that the solution uniqueness property also holds for the complementarity equations (4).

Lemma 2: Suppose (x^*, λ^*, ν^*) solves (1) and Assumptions 1-2 hold. Then, λ^* uniquely solves (4).

Proof: Suppose the claim does not hold. Then there exists $\hat{\lambda} \neq \lambda^*$ such that $\hat{\lambda}$ solves (4) with $\hat{x}, \hat{\nu}$ denoting the solution, multiplier to (2) for $\lambda = \hat{\lambda}$. From Lemma 1, we have that $(\hat{x}, \hat{\lambda}, \hat{\nu})$ also solves (1). Assumptions on strict convexity of f_i (Assumption 1) this implies $x^* = \hat{x}$ while satisfaction of LICQ at x^* implies that λ^*, ν^* are unique. Thus, $\hat{\lambda} = \lambda^*$ contradicting our hypothesis. ■

Lemma 2 shows that λ^* is an isolated solution of complementarity equations (4). However, the complementarity system in (4) is nonsmooth. We note that (4) is equivalent to solving either of the following systems:

$$\Phi^{\min}(\lambda) := \min(\lambda, c - Rx(\lambda)) = 0, \tag{6}$$

where $\min$ operator is applied componentwise; and

$$\Phi^{\text{FB}}(\lambda) \quad := \begin{pmatrix} \phi^{\text{FB}}(\lambda_1, c_1 - R_1 x(\lambda)) \\ \ldots \\ \phi^{\text{FB}}(\lambda_L, c_L - R_L x(\lambda)). \end{pmatrix} = 0, \tag{7}$$

where $\phi^{\text{FB}}(a, b)$ is the Fischer-Burmeister [10] function, which has the form

$$\phi^{\text{FB}}(a, b) = \sqrt{a^2 + b^2} - a - b \tag{8}$$

with a and b being scalars. Clearly $\phi(a, b)$ satisfies

$$\phi^{\text{FB}}(a, b) = 0 \Longleftrightarrow a \geq 0, b \geq 0 \,\text{and}\, ab = 0. \tag{9}$$

The above formulations are nondifferentiable at the point where both components are 0. However, under mild assumptions it is shown in the following that nonsmooth Newton methods can be applied to solve (6) and (7). In the rest of the paper, we will focus only on the Fischer-Burmeister function Φ^{FB} as globalization of generalized Newton's method is easy to show in this case.

III. Semismooth Equation Framework

We briefly review the semismooth Newton method, while interested readers are referred to [11] for a detailed survey of the recent developments of this method.

Definition 1 ([11]): The function $\Phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is said to be semismooth at $\lambda^* \in \mathbb{R}^L$ if Φ is locally Lipschitz, directionally differentiable at λ^* and

$$Vd - \Phi'(\lambda^*; d) = o(\|d\|)$$

where $V \in \partial\Phi(\lambda^* + d)$ (the set of generalized Jacobian of Φ at $\lambda^* + d$) and $\Phi'(\lambda^*; d)$ is the directional derivative of Φ at λ^* along the direction d.

The set of limiting Jacobians at λ or *B-subdifferential* of Φ, denoted as $\partial_B\Phi(\lambda)$, is given by

$$\partial_B\Phi(\lambda) = \left\{ \lim_{\lambda^j \to \lambda, \lambda^j \in D} |\Phi'(\lambda^j) \right\}$$

where $D = \{\lambda | \Phi \text{ is differentiable at } \lambda\}$. The limiting Jacobian is obtained by approching the nondifferentiable point through a sequence of differentiable points. The generalized Jacobian of Φ at λ is the convex hull of the limiting Jacobians at λ and denoted as $\partial\Phi(\lambda) = \text{conv}\partial_B\Phi(\lambda)$.

The generalized Newton method for solving $\Phi(\lambda) = 0$ can be stated as the iteration

$$\lambda^{k+1} = \lambda^k - (V^k)^{-1}\Phi(\lambda^k) \tag{10}$$

where $V^k \in \partial_B \Phi(\lambda^k)$. In order for the above iteration to be well-defined we require the nonsingularity of V^k in a neighborhood of λ^* where $\Phi(\lambda^*) = 0$.

In the following we show that the NUM approach (4) is semismooth and that the generalized Newton method achieves fast local convergence (Therorem 1). We begin by showing that $x(\lambda)$ is Lipschitz in a neighborhood λ^* and so is $\Phi^{\mathrm{FB}}(\lambda)$. This allows us to show that $\Phi^{\mathrm{FB}}(\lambda)$ is semismooth by Definition 1. The result for $x(\lambda)$ follows from standard nonlinear programming sensitivity [12].

Lemma 3 (Corollary 1 in [12]): Let (x^*, λ^*, ν^*) be a solution to the NUM problem (1) satisfying Assumptions 1-3. Then, solutions $(x_i(\lambda), \nu_i(\lambda))$ to (2) are Lipschitz continuously differentiable in neighborhood of λ^*. The gradients with respect to λ satisfy the following linear system

$$\left[\begin{array}{ll} \nabla_i^2 f_i(x_i(\lambda^*)) & -1 \\ \nu_i(\lambda^*) & x_i(\lambda^*) \end{array}\right] \left(\begin{array}{c} \frac{dx_i(\lambda^*)}{d\lambda_l} \\ \frac{d\nu_i(\lambda^*)}{d\lambda_l} \end{array}\right) = \left[\begin{array}{l} -R_{li} \\ 0 \end{array}\right] \tag{11}$$

where $\nabla_i^2 f_i(x_i(\lambda^*))$ denotes the hessian of f_i at $x_i(\lambda^*)$. Hence, we have

$$\frac{dx_i(\lambda^*)}{d\lambda_l} = \begin{cases} \frac{-R_{li}}{\nabla_i^2 f_i(x_i(\lambda^*))} & \text{if } x_i(\lambda^*) > 0 \\ 0 & \text{if } \nu_i(\lambda^*) > 0. \end{cases}$$

Remark 1: In the case that $x_i(\lambda^*)$ and $\nu_i(\lambda^*)$ are degenerate (both are zeros), $x_i(\lambda^*)$ is continuous and locally Lipschitz but not differentiable (refer Theorem 2 in [12]).

The above lemma allows the semismooth characterization of $\Phi^{\mathrm{FB}}(\lambda)$ from which the fast local convergence follows.

Lemma 4: Let (x^*, λ^*, ν^*) solve the NUM problem (1) and Assumptions 1-3 hold. Then $\Phi^{\mathrm{FB}}(\lambda)$ is semismooth at λ^*.

Proof: The claim follows from properties of ϕ^{FB} [13], Definition 1 and Lemma 3. ■

To show fast local convergence of the generalized Newton iteration (10) we need to ensure that the iteration is well defined. In other words, that $\partial_B \Phi(\lambda^*)$ has nonsingular elements. We show this in the following. Prior to proving this result, we introduce some notation. At the solution point (x^*, λ^*, ν^*) of the NUM problem (1), we define the following index sets.

$$\begin{array}{ll} \alpha & = \{l | (c - Rx^*)_l = 0, \lambda_l^* > 0\} \\ \beta & = \{l | (c - Rx^*)_l = 0, \lambda_l^* = 0\} \\ \gamma & = \{l | (c - Rx^*)_l > 0, \lambda_l^* = 0\}. \end{array} \tag{12}$$

Let $R_{\alpha\cup\beta}$ denote the Jacobian of the active linear inequality constraints, $(c - Rx^*)_l$ for $l \in \alpha \cup \beta$ w.r.t. x. Let E_x denote the matrix with rows corresponding to the gradients of the active bounds $x_i^* = 0$ and E_γ refer to the rows corresponding to the constraint gradients of $\lambda_i^* = 0 \; \forall \; i \in \gamma$.

As noted previously, the function ϕ^{FB} is differentiable at all points λ except at points where both arguments vanish. The subdifferential of $\Phi^{\mathrm{FB}}(\lambda)$ at λ^* consists of matrices of the form (refer Theorem 7.1 in [13])

$$\partial_B \Phi^{\mathrm{FB}}(\lambda^*) = \left\{ D^a - D^b R \nabla_\lambda x(\lambda^*) \right\}$$

where D^a, D^b are diagonal matrices with nonpositive entries on the diagonal. In particular the diagonal matrices are,

$$D^a = \begin{bmatrix} 0 & & \\ & D_\beta^a & \\ & & -I_\gamma \end{bmatrix}, D^b = \begin{bmatrix} -I_\alpha & & \\ & D_\beta^b & \\ & & 0 \end{bmatrix}$$

where $I_\alpha \in \mathbb{R}^{|\alpha| \times |\alpha|}$ is the identity matrix and D_β^a, D_β^b are diagonal matrices with $D_\beta^a + D_\beta^b = -I_\beta$.

The following technical result will be utilized in showing the nonsingularity for all $V \in \partial_B \Phi^{\mathrm{FB}}(\lambda^*)$.

Lemma 5: Suppose (x^*, λ^*, ν^*) solves the NUM (1) and Assumptions 1-2 hold. Then, the following matrices are nonsingular for all $\beta_1 \subseteq \beta$:

(i) $\begin{bmatrix} \nabla_x^2 f(x^*) & R_{\alpha\cup\beta_1}^T & -E_x^T \\ R_{\alpha\cup\beta_1} & 0 & 0 \\ E_x & 0 & 0 \end{bmatrix}$, (ii) $\begin{bmatrix} \nabla_x^2 f(x^*) & -E_x^T \\ E_x & 0 \end{bmatrix}$

and (iii) $\begin{bmatrix} R_{\alpha\cup\beta_1} & 0 \end{bmatrix} \begin{bmatrix} \nabla_x^2 f(x^*) & -E_x^T \\ E_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} R_{\alpha\cup\beta_1}^T \\ 0 \end{bmatrix}$.

Proof: Suppose that the matrix in (i) is singular. Then there exists $(u, v, w) \neq 0$ such that

$$\begin{array}{l} \nabla^2 f(x^*) u + R_{\alpha\cup\beta_1}^T v - E_x w = 0 \\ R_{\alpha\cup\beta_1} u = 0 \\ E_x u = 0 \end{array} \tag{13}$$

Left multiplying the first equation by u^T we have that, $u^T(\nabla_x^2 f(x^*) u + R_{\alpha\cup\beta_1}^T v - E_x w) = 0$ which can be simplified using $R_{\alpha\cup\beta_1} u = 0, E_x u = 0$ to $u^T \nabla_x^2 f(x^*) u = 0$. From strict convexity of f we have that $\nabla_x^2 f(x)$ is positive definite for all x. Consequently, $u = 0$ for the condition $u^T \nabla_x^2 f(x^*) u = 0$ to hold. Substituting $u = 0$ in to the first equation of (13) we obtain that $R_{\alpha\cup\beta_1}^T v - E_x w = 0$. From the assumption of LICQ, we have that $[R_{\alpha\cup\beta_1}^T - E_x]$ is full column rank which yields that $(v, w) = 0$. Collecting results, we have $(u, v, w) = 0$ which contradicts our assumption. Hence, the matrix in (i) is non-singular. The proof for the matrix in (ii) follows along identical lines.

Suppose that the matrix in statement (iii) is singular. If this were true then there exists $v \neq 0$ such that

$$\left(\begin{bmatrix} R_{\alpha\cup\beta_1} & 0 \end{bmatrix} \begin{bmatrix} \nabla_x^2 f(x^*) & -E_x \\ E_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} R_{\alpha\cup\beta_1}^T \\ 0 \end{bmatrix} \right) v = 0$$

Also, define the vector (u, w) as,

$$\begin{bmatrix} u \\ w \end{bmatrix} = \left(\begin{bmatrix} \nabla_x^2 f(x^*) & -E_x \\ E_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} R_{\alpha\cup\beta_1}^T \\ 0 \end{bmatrix} \right) v.$$

We have constructed $(u, v, w) \neq 0$ such its right product with matrix in (i) is 0. This is in contradiction with non-singularity of the matrix in (i) and hence, the matrix in (iii) is non-singular. ■

The next lemma proves positive semidefiniteness of the matrices in (ii) and (iii) which are not symmetric.

Lemma 6: Suppose (x^*, λ^*, ν^*) solves the NUM (1) and Assumptions 1-2 hold. Then, matrices in (ii) and (iii) of Lemma 5 are positive semidefinite for all $\beta_1 \subseteq \beta$.

Proof: Consider the matrix in (ii) of Lemma of 5. For any $(u, v) \neq 0$,

$$\begin{bmatrix} u^T v^T \end{bmatrix} \underbrace{\begin{bmatrix} \nabla_x^2 f(x^*) & -(E_x)^T \\ E_x & 0 \end{bmatrix}}_{=:K} \begin{bmatrix} u \\ v \end{bmatrix} = u^T \nabla_x^2 f(x^*) u \geq 0$$

where the last inequality follows from the strict convexity of f. This proves the positive semidefiniteness of the matrix in (i). Consider the matrix in (ii), for some $w \neq 0$

$$w^T \begin{bmatrix} R_{\alpha\cup\beta_1} & 0 \end{bmatrix} K^{-1} \underbrace{\begin{bmatrix} R^T_{\alpha\cup\beta_1} \\ 0 \end{bmatrix} w}_{z} = z^T K^{-1} z.$$

Since LICQ holds we have that $R^T_{\alpha\cup\beta_1}$ has full column rank and $z \neq 0$ whenever $w \neq 0$. Hence,

$$z^T K^{-1} z = z^T (K K^{-1})^T K^{-1} z = z^T (K^{-1})^T K^T \underbrace{K^{-1} z}_{\hat{z}}$$
$$= \hat{z}^T K^T \hat{z} = \hat{z}^T K \hat{z} \geq 0$$

where the last inequality follows from positive semidefiniteness of K and $\hat{z} \neq 0$. This completes the proof. ■

The following proves the result for $\partial_B \Phi^{\min}(\lambda^*)$.

Lemma 7: Suppose (x^*, λ^*, ν^*) solves the NUM (1) and Assumptions 1-3 hold. Then all $V \in \partial_B \Phi^{\mathrm{FB}}(\lambda^*)$ are nonsingular.

Proof: From Lemma 3 it can be shown that,

$$\nabla_\lambda x(\lambda^*) = \underbrace{\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \nabla_x^2 f(x^*) & -E_x^T \\ E_x & 0 \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix}}_{=:M} (-R^T)$$

where $I \in \mathbb{R}^{S\times S}$. Any element in subdifferential $\partial_B \Phi^{\mathrm{FB}}(\lambda^*)$ can be expressed as

$$D^a - D^b R M(-R^T) = D^a + D^b R M R^T.$$

For any such element, consider a partition of $\beta_1 \cup \beta_2 = \beta$, $\beta_1 \cap \beta_2 = \emptyset$ such that $D^b_{\beta_2} = 0$. In other words, $D^a_{\beta_2}$ has nonzero diagonal elements since $D^a_\beta + D^b_\beta = -I_\beta$. Also, $D^b_{\beta_1}$ has nonzero diagonal elements by the definition of β_2. Using this partition an element V of subdifferential can be written as,

$$V = \begin{bmatrix} D^a_{\alpha\cup\beta_1} & 0 \\ 0 & -I_{\beta_2\cup\gamma} \end{bmatrix} + \begin{bmatrix} D^b_{\alpha\cup\beta_1} R_{\alpha\cup\beta_1} M R \\ 0 \end{bmatrix}$$

. The structure above implies that nonsingularity of V will follow from nonsingularity of

$$D^a_{\alpha\cup\beta_1} + D^b_{\alpha\cup\beta_1} R_{\alpha\cup\beta_1} M R^T_{\alpha\cup\beta_1}.$$

To show this, suppose for some $u \neq 0$ that,

$$\left(D^a_{\alpha\cup\beta_1} + D^b_{\alpha\cup\beta_1} R_{\alpha\cup\beta_1} M R^T_{\alpha\cup\beta_1} \right) u = 0$$
$$\implies \left((D^b_{\alpha\cup\beta_1})^{-1} D^a_{\alpha\cup\beta_1} + R_{\alpha\cup\beta_1} M R^T_{\alpha\cup\beta_1} \right) u = 0$$

which follows from nonsingularity from $D^b_{\alpha\cup\beta_1}$. Hence

$$\begin{bmatrix} 0 \\ (D_b^{\beta_1})^{-1} D_a^{\beta_1} u^{\beta_1} \end{bmatrix} + \left(R_{\alpha\cup\beta_1} M (R_{\alpha\cup\beta_1})^T \right) u = 0$$

since $D^a_\alpha = 0$. If $D^a_{\beta_1} u_{\beta_1} = 0$ then the matrix above reduces to requiring singularity of the matrix in (iii) of Lemma 5 which is a contradiction. Consider $D_a^{\beta_1} u^{\beta_1} \neq 0$ then premultiplying by u^T we obtain,

$$(u_{\beta_1})^T (D^b_{\beta_1})^{-1} D^a_{\beta_1} u_{\beta_1} + u^T \left(R_{\alpha\cup\beta_1} M (R_{\alpha\cup\beta_1})^T \right) u = 0$$

which again is a contradiction since by our assumption on $D_a^{\beta_1} u^{\beta_1}$ and Lemma 6 the first term is positive while the second term is nonnegative. Hence, there exists no such $u \neq 0$ and the claim follows. ■

Combining the previous results we obtain the fast convergence of the iteration in (10) in a neighborhood of λ^*.

Theorem 1 ([11]): Suppose (x^*, λ^*, ν^*) solves the NUM (1) and Assumptions 1-3 hold. Then the generalized Newton method (10) using $V \in \partial_B \Phi^{\mathrm{FB}}(\cdot)$ is superlinearly convergent in a neighborhood of λ^*.

Proof: The proof of this result relies on the semismoothness of $\Phi^{\mathrm{FB}}(\lambda^*)$ and the nonsingularity of elements in $\partial_B \Phi(\lambda^*)$ (Lemma 7). We omit the proof for sake of brevity and refer the interested reader to [11]. ■

IV. Algorithm

The main step in the generalized Newton method is the solution of the linear system

$$V d\lambda = -\Phi^{\mathrm{FB}}(\lambda) \text{ for some } V \in \partial_B \Phi^{\mathrm{FB}}(\lambda) \qquad (14)$$

at each iteration. In our case, the calculation of V inolves solving the subproblems (2). Critical to this calculation is the existence of the sensitivity matrix $\nabla_\lambda x(\lambda)$. As stated in Lemma 3 we require the strict complementarity condition to hold. For the local analysis it was sufficient that this hold at the solution to NUM. However, for our algorithm to compute steps from iterates that are far removed from the solution we will make the following assumption.

Assumption 4: For all iterates $\{\lambda^k\}$ produced by the algorithm, the strict complementarity condition $x(\lambda^k) + \nu(\lambda^k) > 0$ holds.

Assumption 4 implies that $x(\lambda)$ is differentiable as a function of λ, $c - Rx(\lambda)$ is differentiable as well. To calculate the derivatives of $c - Rx(\lambda)$, define $D(\lambda) = \mathrm{diag}(\frac{1_{\{x_1(\lambda)>0\}}}{(\nabla_1^2 f_1(x_1(\lambda)))^2}, \ldots, \frac{1_{\{x_N(\lambda)>0\}}}{(\nabla_N^2 f_N(x_N(\lambda)))^2})$. Therefore

$$\nabla_\lambda (c - Rx(\lambda)) = -R \nabla_\lambda x(\lambda) = R D(\lambda) R^T. \qquad (15)$$

For the puprposes of our algorithm it is sufficient to obtain one element $V \in \partial_B \Phi^{\mathrm{FB}}(\lambda)$ at each iteration. We choose V as follows (refer Section 7 of [13]),

$$\nabla_\lambda \phi^{\mathrm{FB}}(\lambda_l, c_l - R_l x) = \begin{cases} \left(\frac{\lambda_l}{\|(\lambda_l, c_l - R_l x)\|} - 1 \right) e_l + \left(\frac{c_l - R_l x}{\|(\lambda_l, c_l - R_l x)\|} - 1 \right) (R D(\lambda) R_l^T) \\ \text{if } (\lambda_l, c_l - R_l x) \neq 0 \\ \left(\frac{\xi_l}{\|(\xi_l, R_l D(\lambda) R^T \xi)\|} - 1 \right) e_l + \\ \left(\frac{R_l D(\lambda) R^T \xi}{\|(\lambda_l, R_l D(\lambda) R^T \xi)\|} - 1 \right) (R D(\lambda) R_l^T) \\ \text{otherwise.} \end{cases}$$

In the above, $\xi \in \mathbb{R}^L$ with $\xi_l = 0$ for the indices such that $(\lambda_l, c_l - R_l x) \neq 0$ and 1 otherwise and $e_l \in \mathbb{R}^L$ with 1 for l-th entry. Solving the non-linear system $\Phi^{\mathrm{FB}}(\lambda) = 0$ is closely related to the unconstrained minimization problem

$$\min \quad \Psi^{\mathrm{FB}}(\lambda) := \tfrac{1}{2}\Phi^{\mathrm{FB}}(\lambda)^T\Phi^{\mathrm{FB}}(\lambda). \tag{16}$$

For the Fischer function it has been shown that $\Psi^{\mathrm{FB}}(\lambda)$ is continuously differentiable and stationary points of $\Psi^{\mathrm{FB}}(\lambda)$ correspond to zeros of $\Phi^{\mathrm{FB}}(\lambda)$. Hence, the above function is an ideal choice for use as a merit function. In particular, progress towards satisfaction of the $\Phi^{\mathrm{FB}}(\lambda) = 0$ can be gauged by reduction in $\Psi^{\mathrm{FB}}(\lambda)$. Given a choice of direction $d\lambda$ the steplength is determined as the smallest natural number r such that:

$$\Psi^{\mathrm{FB}}(\lambda + d\lambda) \leq \Psi^{\mathrm{FB}}(\lambda) + \sigma\rho^r\nabla_\lambda\Psi^{\mathrm{FB}}(\lambda)^T d\lambda \tag{17}$$

where $\nabla\Psi^{\mathrm{FB}}(\lambda) = \nabla\Phi^{\mathrm{FB}}(\lambda)^T\Phi^{\mathrm{FB}}(\lambda)$ is well-defined.

Algorithm 1 details the steps of the algorithm under satisfaction of Assumption 4. At each iteration the algorithm first attempts to compute the Newton step by solving (14). If the linear cannot be solved or if the step size required to satisfy the Armijo conditon (17) is too small then, we attempt to make progress with the gradient step. If the stepsize for satisfying (17) is also short then reset step is invoked. This is a feature that is similar to the crash restart discussed in [14]. The algorithm attempts to find an iterate that can make progress towards solving the problem. We control the number of reset steps that are taken in the algorithm using the $n^{\text{reset,max}}$. The algorithm terminates when steps are too small or if the current iterate solves the NUM to tolerance.

To provide intuition for the requirement of reset steps, consider for instance the case where at some iteration $x(\lambda^k) = 0$. In this case $\nabla_\lambda x(\lambda^k) = 0$ and hence, $V^k = 0$ and (14) cannot be solved. Further, the gradient step cannot make progress towards reducing the merit function. In this case, it is necessary to invoke a reset step. As long as the reset steps are not invoked often the algorithm should converge to a solution of the NUM.

The requirement of Assumption 4 is restrictive. To relax the assumption, we consider reformulating (2) as,

$$\min \sum_{i=1}^{S} (f_i(x_i) - \mu\ln(x_i)) + \lambda^T(Rx - c) \tag{18}$$

The above problem has no inequality constraints and hence, the solution $x(\lambda, \mu)$ is always continuously differentiable as a function λ. Algorithm 1 can be applied to this formualtion for a fixed value of barrier parameter, μ. The complementarity equations are solved approximately to tolerance that is a multiple of μ and then, the barrier parameter is decreased further. This process continues until the NUM is solved. The steps of the barrier based algorithm are provided in Algorithm 2. In this algorithm, we denote by SN$(f_i - \mu\ln(x_i), \mu, \lambda^l)$ the call to Algorithm 1 for solving the NUM with (18) to a tolerance of μ starting with initial iterate λ^l. Reusing the solution of the previous barrier problem as an intial guess for the current barrier parameter improves the convergence of the algorithm. This feature is also exploited in interior point algorithms such as Ipopt [15]. The choice of barrier parameter decrease ensures that the sequence of solutions to (18) converge superlinearly as well.

Algorithm 1: Semismooth Newton method - SN$(f_i, \epsilon, \bar{\lambda})$

```
Let $\sigma, \rho, \omega, \eta, \tau \in (0,1), n^{\text{reset,max}} \geq 0$ be given scalars
Set $k = 0$, $\lambda^0 = \bar{\lambda}, n^{\text{reset}} = 0$
repeat
    /* Newton Step */
    if linear system (14) solvable then
        Set $d\lambda_N^k = -(V^k)^{-1}\Phi^{\mathrm{FB}}(\lambda^k)$
        Find smallest $r_N \geq 0$ such that (17) holds
        Set $d\lambda^k = \rho^{r_N} d\lambda_N^k$
    end
    /* Gradient Step */
    if linear system (14) unsolvable or $\rho^{r_N} < \tau$ then
        Set $d\lambda_G^k = -\nabla_\lambda\Phi^{\mathrm{FB}}(\lambda^k)$
        Find smallest $r_G \geq 0$ such that (17) holds
        Set $d\lambda^k = \rho^{r_G} d\lambda_G^k$
    end
    /* Reset Step */
    if (14) unsolvable or ($r_N < \tau$ and $r_G < \tau$) then
        if $n^{reset} < n^{reset,max}$ then
            Set $n^{\text{reset}} = n^{\text{reset}} + 1$
            Set $d\lambda^k = (\min(\omega, \|\lambda^k\|^{1+\eta}) - 1)\lambda^k$
        end
    else
        Set $d\lambda^k = 0.$
    end
    /* Update iterate */
    Set $\lambda^{k+1} = \lambda^k + d\lambda^k$
    Set $k = k + 1$
until $\|\lambda^{k+1} - \lambda^k\| \leq \epsilon$ or
$\|\min(c - Rx(\lambda^k), \lambda^k)\|_\infty \leq \epsilon$
```

For lack of space, we omit global convergence results for the above algorithms. Instead we provide extensive numerical results that demonstrate the efficiency of the algorithms.

V. Results

Algorithms 1 and 2 were coded in MATLAB. For all computational experiments the following values were used for the scalars in Algorithm 1, $\sigma = 10^{-4}$, $\rho = 0.9$, $\omega = 0.5$, $\eta = 0.5$, $\tau = 10^{-8}$ and $n^{\text{reset,max}} = 2$. All problems were solved to a convergence tolerance of $\epsilon = 10^{-8}$. The codes were executed in MATLAB 7.11 on a machine with 2.4GHz Intel Core 2 CPU and 2 GB of RAM. For comparison we also solved the problems using Ipopt [15] through a custom MEX interface linked to Ipopt's precompiled DLLs [16].

The utility functions were chosen as $U_i(x_i) = -f_i(x_i) = w_i\ln(x_i)$ where $w_i > 0$ for which Assumption 1 holds. In the Algorithms 1 and 2, the iterates λ^k can be become negative. In which case $x_i(\lambda^k) \to \infty$. To avoid this, we impose an upper bound of $x^{\max} = \max(c) + 1$ on all x_i which is valid for any NUM solution x^*. Consequently, the subproblems

Algorithm 2: Semismooth Newton method (Barrier)

```
Initialization: Let $\sigma \in (0,1)$ and $\rho \in (0,1)$ be given scalars
Set $l = 0$, pick $\mu > 0$, $\lambda^0 \in \mathbb{R}^L_+$ and convergence tolerance $\epsilon$
repeat
    Solve SN($f_i - \mu \ln(x_i), \mu, \lambda^l$) to obtain solution $\lambda^{l+1}$
    Set $\mu = \min(\mu^{1.5}, 0.1\mu)$
    Set $l = l + 1$
until $\mu \leq \epsilon$
```

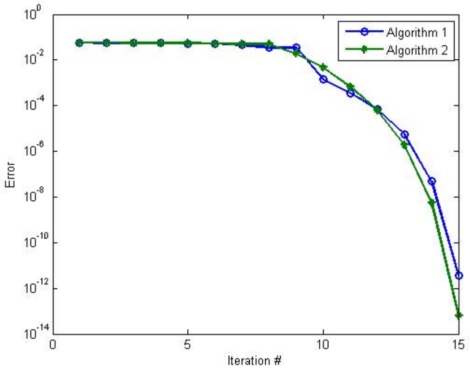

Fig. 1. Plot of error versus iterations for the two algorithms on a 100 source, 100 links instance.

that are solved in the two algorithms are respectively

$$\min_{0 \leq x \leq x^{\max}} \min \sum_{i=1}^{S} f_i(x_i) + \lambda^T(Rx - c)$$
$$\min \sum_{i=1}^{S} (f_i(x_i) - \mu \ln(x_i) - \mu \ln(x^{\max} - x_i)) + \lambda^T(Rx - c).$$

Data for the problem instances were randomly generated as suggested in [7]. The link-source matrix R was generated from a Bernoulli distribution. Any instance constained a row with all zeros was discarded. The link capacities and source weights were generated randomly using MATLAB's `rand` command as $c = 3*\texttt{rand}(L, 1)$ and $w = \texttt{rand}(S, 1)$.

A. Fast local convergence of the algorithms

Theorem 1 shows the asymptotically the algorithm converges at a superlinear rate. Figure 1 plots the error as function of iteration number for the Algorithms 1 and 2 for a randomly generated problem instance with $S = 100, L = 100$. Both algorithms were provided an initial point of all 1's, $\lambda^0 = 1$. Clearly the last few iterates converge rapidly to a solution. This behavior was observed in all our examples validating the theory.

B. Comparison with smooth second order algorithm

To demonstrate the computational efficiency of the computational efficiency of the algorithms we compare our implementation against an algorithm that directly solves the NUM (1). Ipopt [15] is used to solve (1). Ipopt is a state-of-the-art interior point method for solving nonlinear programming (NLP) problems. Ipopt also employs a second order algorithm which achieves superlinear convergence in the neighborhood of a solution satisfying the assumptions stated in this work. In the following, Ipopt is initialized with $x^0 = 1$, Algorithm 1 is intialized with $\lambda^0 = 1$ and Algorithm 2 is initialized with $\lambda^0 = Rw$. Ipopt is fairly insensitive to the choice of initial point. On the other, the choices for our algorithms are empirically seen to provide very good performance. We emphasize the algorithms are robust for other choices of λ^0 as well.

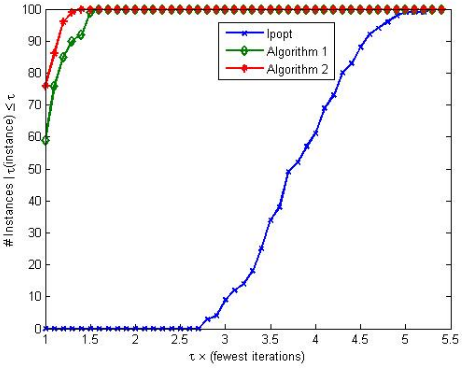

Fig. 2. Performance profiles of algorithms based on number of iterations for convergence.

We compare the algorithms in three aspects - number of iterations, number of function evaluations and CPU time. Figures 2,3,4 show the performance profiles of the three algorithms on a randomly generated set of 100 instances of size $S = L = 100$. Performance profiles were introduced in [17] as a means of comparing algorithm behaviors on different aspects. To compare the algorithms based on number of iterations for convergence, we determine for each algorithm j and each problem instance p

$$t(j,p) = \frac{\text{\# iterations by algorithm } j \text{ on } p}{\text{least \# iterations among all algorithms on } p}$$

which is a normalization against best algorithm for each problem instance. The performance profile is essentially a cumulative plot of $\{\#p | t(j,p) \leq \tau\}$ against τ. Figure 2 shows that Algorithm 2 requires the least number for iterations for convergence on more than 75 instances, while Algorithm 1 has the least iteration count on 60 instances. The number of iterations for Algorithm 2 is sum of iteratons required for each barrier parameter. Both algorithms solve all problems in about 1.5 times the number of iterations requires for the best algorithm. Ipopt on the other hand requires as much as 2.5 times that required by best algorithm on all problem instances. Algorithms 1 and 2 are essentially active set type algorithms and hence, are likely to have excellent convergent behavior when started with a good initial guess. Yet, this is certainly remarkable given that we employ a dual decomposition framework and Ipopt is

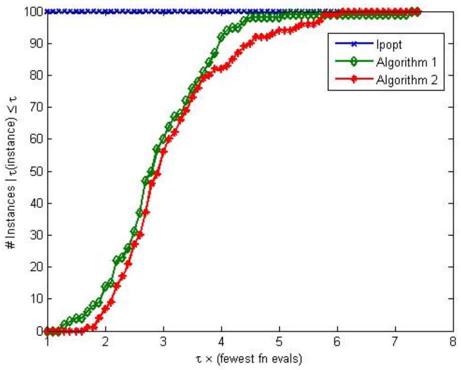

Fig. 3. Performance profiles of algorithms based on number of function evaluations for convergence.

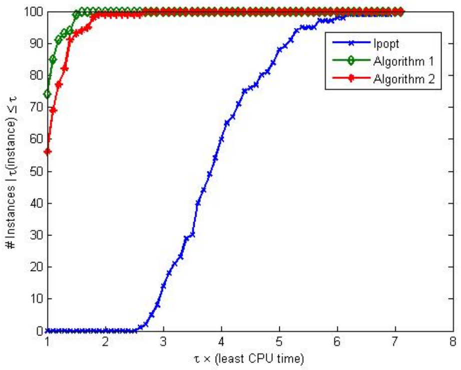

Fig. 4. Performance profiles of algorithms based on CPU time for convergence.

a robust, efficient algorithm. In the context of function evaluations (refer Figure 3), Ipopt clearly surpasses our algorithms on all problems. Both our algorithms have almost identical performance profiles and require more than 3 times the number of function evaluations of Ipopt on more than 50 instances. This is possibly due to line search cutting back on the given search directions. We have also noticed that the Jacobian $\nabla_\lambda \Phi^{\mathrm{FB}}(\lambda^k)$ also becomes ill-conditioned at some early iterations. In terms of CPU time, our algorithms are much faster than Ipopt as seen in Figure 4.

C. *Frequency of reset steps*

In the previous section, we alluded to the possibility of having to invoke reset steps quite often. In our computational experience, we never had to use the reset steps in any of the 100 instances for the given initialization of both algorithms. However, when Algorithm 2 is initialized with $\lambda^0 = 1$ we have noticed that there are instances where the reset step is invoked at the first iteration only. This behavior is quite encouraging and shows robustness of our approach.

VI. Conclusion

Dual decomposition for NUM have long been plagued by the poor convergence properties of the subgradient algorithm. We presented two algorithms based on a novel reformulation of the NUM to semismooth complementarity equations. Both algorithms enjoy fast local convergence properties near the solution to NUM under mild assumptions. Numerical experience confirmed the excellent local and global convergence properties of the algorithms. The performance of the algorithms is superior to even a state-of-the-art nonlinear programming algorithm in certain aspects. We will investigate the global convergence proof of the algorithms and extension to a distributed computational setting in a future work. We believe the linear system solution can be distributed using iterative linear algebra techniques along the lines of [7]. This allows to develop a distributed computational approach which will retain the fast local convergence shown here.

References

[1] F. P. Kelly, A. K. Malulloo, and D. K. H. Tan, "Rate control in communications networks: Shadow prices, proportional fairness and stability," *Journal of the Operations Research Society*, vol. 49, pp. 237–252, 1998.

[2] S. H. Low and D. E. Lapsley, "Optimization flow control, I: basic algorithm and convergence," *IEEE/ACM Transaction on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[3] S. Low and R. Srikant, "A mathematical framework for designing a low-loss low-delay internet," *Network and Spatial Economics*, vol. 4, no. 1, pp. 75–101, 2004.

[4] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architecture," *Proceedings of IEEE*, vol. 95, no. 1, pp. 255–312, 2007.

[5] N. Z. Shor, *Minimization methods for nondifferentiable functions*. Berlin: Springer, 1985, Translated from Russian by K.C. Kiwiel and A. Ruszczynski.

[6] D. Bickson, Y. Tock, A. Zyrnnis, S. Boyd, and D. Dolev, "Distributed large scale network utility maximization," *Proceedings of IEEE International Symposium on Information Theory*.

[7] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A Distributed Newton Method for Network Utility Maximization, I: Algorithm," Massachussetts Institute of Technology, Tech. Rep.

[8] J. Liu and H. Sherali, "A Distributed Newton Approach for Joint Multi-Hop Routing and Flow Control: Theory and Algorithm," Tech. Rep.

[9] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2000.

[10] A. Fischer, "A special Newton-type optimization method," *Optimization*, vol. 24, pp. 269–284, 1992.

[11] L. Qi and D. Sun, "A survey of some nonsmooth equations and smoothing Newton methods," in *Progress in optimization : contributions from Australasia Vol. 30 of Applied optimization*, A. Eberhard, B. Glover, R. Hill, D. Ralph, Ed. Kluwer Academic Publications, pp. 121–146.

[12] K. Jittorntrum, "Solution Point Differentiability without Strict Complementarity in Nonlinear Programming," *Mathematical Progamming*, vol. 21, pp. 127–138, 1984.

[13] T. D. Luca, F. Facchinei, and C. Kanzow, "A semismooth equation approach to the solution of nonlinear complementarity problems," *Mathematical Programming*, vol. 75, pp. 407–439, 1996.

[14] T. S. Munson, F. Facchinei, M. C. Ferris, A. Fischer, and C. Kanzow, "The Semismooth algorithm for large scale complementarity problems," *INFORMS Journal on Computing*, vol. 13, pp. 294–311, 2001.

[15] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[16] "Ipopt pre-compiled binaries." [Online]. Available: http://www.coin-or.org/Binaries/Ipopt/

[17] E. D. Dolan and J. J. More, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213.