2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Accelerated Gradient Methods for Networked Optimization

Euhanna Ghadimi, Mikael Johansson and Iman Shames

Abstract— **This paper explores the use of accelerated gradient methods in networked optimization. Optimal algorithm parameters and associated convergence rates are derived for distributed resource allocation and consensus problems, and the practical performance of the accelerated gradient algorithms are shown to outperform alternatives in the literature. Since the optimal parameters for the accelerated gradient method depends on upper and lower bounds of the Hessian, we study how errors in these estimates influence the convergence rate of the algorithm. This analysis identifies, among other things, cases where erroneous estimates of the Hessian bounds cause the accelerated method to have slower convergence than the corresponding (non-accelerated) gradient method. An application to Internet congestion control illustrates these issues.**

I. Introduction

Distributed optimization has recently attracted a significant attention from several different research communities. Examples include the work on network utility maximization for resource allocation in communication networks [1], distributed coordination of multi-agent systems [2], and collaborative estimation and event detection in wireless sensor networks (WSNs) [3] and many others. The majority of these praxes rely on the application of gradient or subgradient methods to the dual formulation of the decision problem at hand (cf. [1]). Although gradient methods are easy to implement and require modest computations, they suffer from slow convergence rates. In certain special cases, such as the celebrated development of distributed power control algorithms for cellular phones [4], one can replace gradient methods by fixed-point iterations and achieve improved convergence rates. For other problems, such as average consensus [5], a number of heuristic methods have been proposed that improve the convergence rate of the standard method [6], [7]. However, we are not interested in techniques that apply only in special cases. We would like to develop general-purpose schemes that retain the simplicity and the applicability of the gradient method while improving the convergence rates.

In the optimization literature, there are essentially two ways of accelerating the convergence rate of the gradient methods. One is to use higher-order methods, such as Newton's method [8]. Although distributed Newton methods have recently been developed for special problem classes, they impose large communication overhead. Another way to improve convergence is to use multi-step methods [9], [8]. These methods only rely on gradient information (and can hence often be implemented based on local information) but use a history of past iterates to compute the next.

The authors are with the ACCESS Linnaeus Center, Electrical Engineering, Royal Institute of Technology, Stockholm, Sweden. E-mails: {euhanna, mikaelj, imansh}@ee.kth.se

This paper explores how multi-step methods can be used in networked optimization. Our initial focus is to attempt to accelerate the projected gradient method of [10]. We derive optimal parameters for the algorithm and show how these are directly related to the topology of the underlying communication graph. Contrary to the gradient descent method, which only needs an upper bound on the Hessian for finding a step size that ensures convergence, the multi-step method need both the upper and the lower bounds on the Hessian. Due to this fact, we formally analyze the convergence and compute the convergence rate in the presence of estimation errors in the Hessian bounds. Later, we illustrate how the technique allows us to derive accelerated consensus iterations and demonstrate improved convergence rates relative to other acceleration schemes proposed in the literature [11], [6]. Finally, we implement the techniques that we have developed to accelerate the network flow control algorithm described in [12]. We discuss how uncertainties in finding the bounds on Hessian can affect the convergence rate of the algorithm.

The rest of the paper is organized as follows. In Section II, we review distributed resource allocation and multi-step gradient techniques. In Section III we present our accelerated resource allocation algorithm with proofs of convergence and performance comparison with the basic scaled gradient method. In Section IV, robustness analysis of multi-step algorithm in the presence of uncertainty is presented. Section V is devoted to other applications and considers accelerated consensus and network flow control. Conclusion remarks and future work outlook are presented in Section VI.

II. Preliminaries

We consider constrained optimization problems on a network of nodes. The network is modeled as a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with vertices (nodes) in the set $\mathcal{V}=\{1,2,..,n\}$ and pairs of nodes as edges in the set $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$. We use $\mathcal{N}_i=\{j|(i,j)\in\mathcal{E}\}$ to denote the set of neighbors of node i.

A. Resource Allocation Under Total Budget Constraint

Consider the following resource allocation problem

$$\begin{array}{ll}\min_x & \sum_{i=1}^n f_i(x_i)\\ \text{subject to} & \sum_{i=1}^n x_i = x_{tot}\end{array} \tag{1}$$

Where $\sum_{i=1}^n x_i = x_{tot}$ is called the budget constraint, and f_i are real-valued strictly convex and twice differentiable functions whose second derivatives satisfy

$$L_i \le f_i''(x_i) \le U_i, \quad i=1,..,n, \tag{2}$$

for known constants $U_i \ge L_i > 0$. A distributed resource allocation mechanism that maintains the budget constraint

at all times was developed in [10], [13]. nodes iteratively exchange resources via the scaled gradient method

$$x_{k+1} = x_k - W\nabla f(x_k) \tag{3}$$

The weight matrix W needs to satisfy $\mathbf{1}^T W = 0, \quad W\mathbf{1} = 0$ for the total resource constraint to be always satisfied, and needs to have the same sparsity pattern as the underlying graph to ensure that nodes only exchange resources with neighbors. One matrix that satisfies these constraints on W is the Laplacian of the underlying graph $\mathscr{L} = A(\mathscr{G})A(\mathscr{G})^T$ [14]. Here, $A(\mathscr{G})$ is the *adjacency matrix* of $\mathscr{G}$ with entries 1 when $(i,j) \in \mathscr{E}$, -1 when $(j,i) \in \mathscr{E}$ and 0 otherwise. To ensure convergence of the iteration (3), the Laplacian has to be appropriately scaled $W = -\delta\mathscr{L}$ for some $\delta > 0$.

The Laplacian weights relevant to a specific node can be determined using local topology information. Specifically,

$$[W]_{ij} = \begin{cases} \delta, & (i,j) \in \mathscr{E}, \\ -\delta d_i & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Where d_i is the degree of node i. Following the notation in [13], we call these *constant* weights, since all edges are assigned a constant weight and then W_{ii} is adjusted to make sure that $W\mathbf{1} = 0$. Several heuristic weight choices are introduced in [13] . For the special case when $U_i = 1$ for all $i \in \mathscr{N}$, this includes: the *maximum degree* weights where $\delta = 1/\max_i d_i$; the *best constant* weights, for which $\delta = 2/(\lambda_2(\mathscr{L}) + \lambda_n(\mathscr{L}))$; and the *Metropolis-Hasting* weights

$$[W]_{ij} = \begin{cases} -\min\{1/d_i, 1/d_j\} & (i,j) \in \varepsilon, \\ -\sum_{(i,k)\in\varepsilon} W_{ik} & i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

We will return to this method in Section III and see how accelerated gradient techniques, described next, allow to achieve improved convergence rates compared to (3).

B. Multi-step Gradient Methods

The classical gradient method for minimization of a convex function f takes the form

$$x_{k+1} = x_k - \alpha\nabla f(x_k) \tag{6}$$

for some step size parameter $\alpha > 0$. This method is a suitable choice for distributed optimization due to its simplicity and ease of implementation. However, gradient based algorithms often exhibit slow convergence rate [9]. The convergence rate can be improved by accounting for the history of the process which is already obtained in the preceding iterations. Methods in which the new approximation depends on the preceding ones are called *multi-step methods*. In this paper, we use two-step iterations of the form

$$x_{k+1} = x_k - \alpha\nabla f(x_k) + \beta(x_k - x_{k-1}) \tag{7}$$

where $\alpha > 0, \beta \geq 0$ are fixed step sizes. This method, proposed for centralized applications by Polyak [9] is called the Heavy Ball (HB) method and can be tuned to have smoother trajectory toward the local minimum point compared with traditional the gradient iterations [9]. The smoother trajectory translates to faster convergence rates. The reason why we focus on this method is that it is computationally simple and does not need higher order information which might not be locally available in distributed applications.

III. Accelerated Resource Allocation

Our first contribution in this paper is to consider the application of multi-step methods to the distributed resource allocation problem (1). To this end, consider the iteration

$$x_{k+1} = x_k - \alpha W\nabla f(x_k) + \beta(x_k - x_{k-1}) \tag{8}$$

Let $x^\star$ be an optimal point of $f(x)$. Using Taylor expansion

$$\nabla f(x_k) = \nabla^2 f(x^\star)(x_k - x^\star) + o(x_k - x^\star)^2$$

Letting $z^{k+1} = (x_{k+1} - x^\star, x_k - x^\star)$, we can rewrite (8) as

$$z^{k+1} = \mathscr{A} z^k + o(z^k) \tag{9}$$

where the $2n \times 2n$-square matrix $\mathscr{A}$ is given by

$$\mathscr{A} = \begin{bmatrix} (1+\beta)I - \alpha WH & -\beta I \\ I & 0 \end{bmatrix}, \qquad H = \nabla^2 f(x^\star) \tag{10}$$

Let $L = \lambda_1(H) \leq \lambda_2(H) \leq ... \leq \lambda_n(H) = U$ be the eigenvalues of H. In what follows we study the local convergence of the iterative process described by (8).

Theorem 1: Let $\omega = WH$ and $\lambda_n(\omega)$ be the largest eigenvalue of ω and $x^\star$ be a nonsingular minimum point of $f(x), x \in R^n$. Then for

$$0 \leq \beta < 1, \quad 0 < \alpha < \frac{2(1+\beta)}{\lambda_n(\omega)}, \quad LI \leq \nabla^2 f(x^\star) \leq UI \tag{11}$$

the method (8) converges to $x^\star$ with the rate of geometric progression:

$$\frac{||x_{k+1} - x^\star||}{||x_k - x^\star||} \leq q_1 \quad 0 \leq q_1 < 1.$$

Proof: For brevity we omit the proofs. The interested reader may refer to [15]. ■

The next theorem gives the optimal step sizes as well as the optimum convergence rate of (8).

Theorem 2: The convergence rate of (8) is given by

$$q_1 = \max\left\{2\sqrt{\beta}, |1+\beta-\alpha\lambda_2(\omega)|, |1+\beta-\alpha\lambda_n(\omega)|\right\} - \sqrt{\beta} \tag{12}$$

The minimal value of q_1,

$$q_1^\star = \frac{\sqrt{\lambda_n(\omega)} - \sqrt{\lambda_2(\omega)}}{\sqrt{\lambda_n(\omega)} + \sqrt{\lambda_2(\omega)}} \tag{13}$$

is obtained for the step sizes $\alpha = \alpha^\star, \beta = \beta^\star$ with

$$\alpha^\star = \frac{4}{\left(\sqrt{\lambda_n(\omega)} + \sqrt{\lambda_2(\omega)}\right)^2}$$
$$\beta^\star = \left(\frac{\sqrt{\lambda_n(\omega)} - \sqrt{\lambda_2(\omega)}}{\sqrt{\lambda_n(\omega)} + \sqrt{\lambda_2(\omega)}}\right)^2 \tag{14}$$

It is known that the convergence rate of the non-accelerated method is (cf. [9], [8], [13])

$$q_2 = \frac{\lambda_n(\omega) - \lambda_2(\omega)}{\lambda_n(\omega) + \lambda_2(\omega)} \tag{15}$$

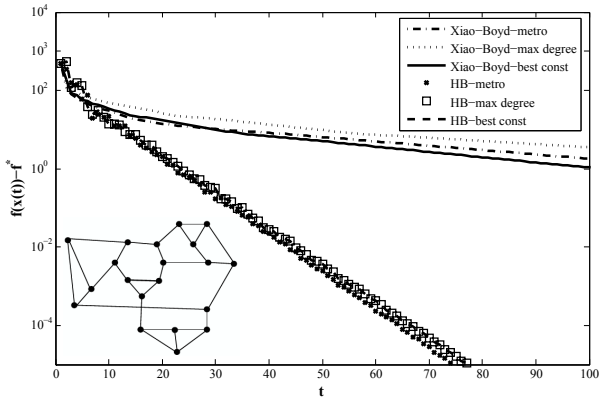

Fig. 1. Convergence behavior of HB and Xiao and Boyd method using randomly generated network and all the heuristic weights. plot shows the objective function values $f(x(t))-f^\star$ versus iteration number t.

One can verify that $q_1 \le q_2$, and that the improvement in convergence rate is depends on the quantity $\kappa = \lambda_n(\omega)/\lambda_2(\omega)$. In particular, when κ is large, then the speed-up is roughly proportional to $\sqrt{\kappa}$ (cf. the analysis for the centralized heavy-ball method in [9]). Large values of κ essentially appear for two reasons: one is a large spread in the values U_i between nodes, and the other is the topology of the underlying graph. Assume for simplicity that $U_i = 1$ for all i, so that $\omega = W$ and consider Laplacian weights. It is well-known from spectral graph theory [14] that the complete graph with n vertices has $\lambda_2(\mathscr{L}) = \lambda_n(\mathscr{L}) = n/(n-1)$, so $\kappa = 1$ and there is no real advantage of using the accelerated scheme. On the other hand, for a ring network of n nodes, the eigenvalues of the Laplacian are $1-\cos\frac{2\pi k}{n}$ for $k = 0,....,n-1$, which means that κ grows quickly with n, and the performance improvements of the accelerated methods can be substantial.

1) Numerical Examples: In this section we numerically compare the performance of the accelerated and non-accelerated resource allocation method under various weight matrices described in the previous section. To make a fair comparison, we consider a random graph generated in a similar way as in [13]. The network shown in the Fig. 1 consists of 20 nodes, each of degree three. Edges are bidirectional and the objective function at each node has the form $f_i(x_i) = \frac{1}{2}a_i(x_i-c_i)^2 + \log[1+\exp(b_i(x_i-d_i))]$, $i = 1,..,n$. The coefficients $a_i, b_i, c_i,$ and d_i are drawn uniformly from the intervals $[0,2],[-2,2],[-10,10]$ and $[-10,10]$, respectively, and kept the same for all simulations. As initial values, we fix the sum of variables to zero (i.e., $\sum_{i=1}^n x_i = 0$). The second derivative of the functions f_i are positive and lower and upper bounded are given by $l_i = a_i$, $u_i = a_i + \frac{1}{4}b_i^2$, $i = 1,...,n$. Using the techniques in [13], we set the parameters $\delta = -0.1251$ for maximum degree weight and $\delta^\star = -0.2030$ for best constant weight scheme. Fig. 1 shows the objective value minus the optimal value as function of iteration count. The plot shows that the accelerated gradient method yields a significantly increased convergence rate.

IV. Robustness Analysis

When the upper and lower bounds on the Hessian are known and their ratio is significant, multi-step methods give a considerable increase in convergence rate over the standard gradient iteration. However such upper and lower bounds are sometimes hard to estimate accurately in practice. It is therefore important to analyze the sensitivity of multi-step methods to errors in the Hessian bounds to assess if the performance benefits prevail if the bounds are inaccurate. Such a robustness analysis will be performed next.

Let L and U be the true upper and lower bounds of the Hessian of the objective function, and let $\hat{L}$ and $\hat{U}$ be the estimated bounds used when tuning the gradient and accelerated gradient methods. We would like to observe for which values of $\hat{L}$ and $\hat{U}$ the two methods converge under their "optimal" step-size rules and compare the associated convergence rates. In [9] sufficient conditions for the convergence of gradient iterates of smooth functions are given. According to [9, Theorem 3], for fixed step size $0 < \alpha < 2/U$, the gradient algorithm converges with rate

$$q_2 = \max\{|1-\alpha L|, |1-\alpha U| < 1\}.$$

The minimum value $q_2^\star = (U-L)/(U+L)$ is attained by the optimal step size $\alpha^\star = 2/(L+U)$. Together with the analysis in Theorem 1 this yields the following observation:

Proposition 1: Consider $0 < \hat{L} < \hat{U}$ to be the erroneous estimates of the Hessian bounds L, U respectively. For all values of $\hat{L}, \hat{U}$ fulfilling the condition $U < \hat{U} + \hat{L}$ both the gradient iteration (6) with step size selection

$$\hat{\alpha} = 2/(\hat{L}+\hat{U})$$

and the HB algorithm (7) with parameters

$$\hat{\alpha} = 4/(\sqrt{\hat{L}}+\sqrt{\hat{U}})^2,\ \hat{\beta} = ((\sqrt{\hat{U}}-\sqrt{\hat{L}})/(\sqrt{\hat{U}}+\sqrt{\hat{L}}))^2$$

converge to the optimum of $f(x)$.

According to this proposition, the convergence of both methods is rather similar. To compare convergence rates, we start by presenting the following lemma.

Lemma 1: For parameters $\hat{L}, \hat{U}$ satisfying $0 < U < \hat{L} + \hat{U}$ the convergence rate of gradient algorithm is given by

$$\hat{q}_2 = \begin{cases} 2U/(\hat{L}+\hat{U}) - 1 & \hat{L}+\hat{U} < L+U, \\ 1 - 2L/(\hat{L}+\hat{U}) & L+U < \hat{L}+\hat{U}. \end{cases} \tag{16}$$

It is easy to check $\hat{q}_2 > q_2^\star$ for either of two cases, *i.e.* the gradient method with misestimated Hessian bounds has a slower convergence rate than the optimally tuned one. The best step size choice unexpectedly happens when $\hat{L}+\hat{U} = L+U$, for which $\hat{q}_2 = q_2^\star$.

On the other hand, Theorem 2 establishes the convergence rate of HB with arbitrary (uncertain) step sizes to be

$$\hat{q}_1 = \max\left\{\sqrt{\hat{\beta}}, |1+\hat{\beta}-\hat{\alpha}L| - \sqrt{\hat{\beta}}, |1+\hat{\beta}-\hat{\alpha}U| - \sqrt{\hat{\beta}}\right\}. \tag{17}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the values of the optimal step size rule when the erroneous Hessian bounds are used. It is interesting to note that for $\hat{L} = \hat{U}$ and the convergence constraint, $\hat{L}+$

$\hat{U} > U$, both methods converges with rate $1 - L/\hat{L}$. However, when the Hessian bounds are not known, the Heavy Ball method can either perform better or worse than the gradient method. We have the following results

Proposition 2: Assuming parameters $\hat{L} > L, \hat{U} > U$. then convergence rate of the Heavy Ball method is $\hat{q}_1 = 1 + \hat{\beta} - \hat{\alpha} l - \hat{\beta}^{1/2}$ which is faster than the gradient alternative.

Proposition 3: Assuming parameters $\hat{L} < L, \hat{U} > U$ and $\hat{L} + \hat{U} = L + U$. Then the convergence rate of the Heavy Ball method is given by $\hat{q}_1 = \hat{\beta}^{1/2}$. Moreover, if $(\hat{U}/\hat{L})^{1/2} > U/L$, then this rate is slower than the gradient alternative.

From our simulation experience, the situation described in Proposition 3 is rather singular. In fact, we have not yet experienced this situation in non-contrived scenarios.

V. Further Applications

A. Accelerated Consensus

Distributed algorithms for consensus seeking, first proposed in Tsitsiklis et al. [5], have been heavily researched during the last few of years. We consider consensus via linear iterations on the form

$$x_i(k+1) = W_{ii} x_i(k) + \sum_{j \in \mathcal{N}_i} W_{ij} x_j(k), \quad i = 1, ..., n, \qquad (18)$$

Here, W_{ij} is the weight on x_j assigned in node i. In [16] necessary and sufficient conditions on W for (18) to converge to the average of initial values are given. More specifically, it is shown that such matrices W have their largest eigenvalue equal to 1 while their second largest eigenvalue is strictly less than 1 and determines the asymptotic convergence factor towards consensus [16]. For symmetric weights, [16] derived a semi-definite program (SDP) for finding the weight matrix W with maximized convergence rate and proposed several simple heuristics for finding suboptimal weights, including *constant* and *Metropolis-Hastings* weights (cf. [16], [2]). In what follows we use dual decomposition to develop multi-step consensus iterations with accelerated convergence.

1) Consensus Algorithm Using Dual Decomposition: It is possible to achieve similar iterations as primal consensus using networked minimization of quadratic functions

$$\begin{array}{ll} \text{min} & \sum_{i=1}^n \frac{1}{2}(x_i - c_i)^2 \\ \text{subject to} & x_i = x_j, \forall (x_i, x_j) \in \mathscr{E} \end{array} \qquad (19)$$

Any distributed method for solving this problem is also a distributed averaging (or average consensus) algorithm. A primal-dual based algorithm combined with a subgradient method to solve (19) is presented in [17], and an alternating direction multiplier to cast the optimization problem in distributed fashion is discussed in [18]. The iterations in the latter are shown to be resilient to communication noises. One can re-write (19) in vector notation and apply Lagrange duality to the coupling constraint to find the Lagrangian

$$L(x, \mu) = \frac{1}{2}(x - c)^T (x - c) + \mu^T A x \qquad (20)$$

By the first-order optimality conditions for (20) we can define the dual problem as following unconstrained minimization

$$\text{minimize} \quad -g(\mu) = -\tfrac{1}{2}\mu^T A A^T \mu - \mu^T A c \qquad (21)$$

For given Lagrange multiplier μ, the primal variable x will be updated by minimizing the Lagrangian (20). The accelerated gradient method hence suggests the multi-step iteration

$$\begin{array}{l} \mu_{k+1} = \mu_k - \alpha(AA^T \mu_k - Ac) + \beta(\mu_k - \mu_{k-1}) \\ x_{k+1} = c - A^T \mu_{k+1} \end{array} \qquad (22)$$

Note that the μ-iterations in the dual scheme has the same form as the distributed resource allocation iterations under Laplacian weight selection. Hence, our analysis and design rules for selecting optimal algorithm parameters apply immediately. To understand the relationship between this method and alternative schemes in the literature, it is useful to try to eliminate the μ-update and only consider the dynamics of the primal variables. To this end, multiplying A^T on both sides of (22) yields

$$A^T \mu_{k+1} = A^T \mu_k - \alpha A^T (AA^T \mu_k - Ac) + \beta A^T (\mu_k - \mu_{k-1}) \qquad (23)$$

Using $A^T \mu_k = c - x_k$ and letting $W = A^T A$ gives

$$x_{k+1} = ((1+\beta)I - \alpha W) x_k - \beta x_{k-1} \qquad (24)$$

As argued previously, W is positive semidefinite with a simple eigenvalue at 0 and fulfills $W\mathbf{1} = 0, \mathbf{1}^T W = 0$. With this definition, the consensus iterations coincide with the general results of Theorem 2 (the Hessian is identity in (10)).

We compare this simple technique with two alternative acceleration methods; one from the literature on accelerated consensus, and the other one from the literature on (centralized) first-order techniques for convex optimization.

2) Accelerated Consensus via Shift Registers: Shift registers can be used to speed up convergence in stochastic form of (18). In [6] it is demonstrated by simulations that the consensus algorithm can be accelerated substantially by using shift-registers in each node. The shift register in each node stores the most recent history of the node's iterates. Consider at iteration k two nodes i, j who decide to update their values. Node i changes its current value as follows (node j also updates similarly)

$$\begin{cases} x_{k+1}^1(i) = a_1(\frac{1}{2}x_k^1(i) + \frac{1}{2}x_k^1(j)) + \sum_{l=2}^n a_l x_k^l(i), \\ x_{k+1}^l(i) = x_k^{l-1}(i) \qquad l = 2, .., n. \end{cases} \qquad (25)$$

Where a_l are constant and $\sum_{l=1}^n a_l = 1$. Johansson et al., [7] investigated necessary and sufficient conditions as well as optimal parameters for convergence of such algorithms using symmetric weights. More recently, [19] analyzes the accelerated convergence rate of shift-registers. For the consensus problem, shift registers result in iterations on the following form (assuming two entries in each shift register)

$$x_{k+1} = \zeta W x_k + (1 - \zeta) x_{k-1} \qquad (26)$$

where ζ is a constant scalar, W also is weight matrix (the one from (18) can be used). For symmetric W, the average convergence factor is minimized by letting [20]

3) Nesterov Method: In [21], Nesterov discusses the complexity bounds of various optimization methods. It is shown that no gradient-based method can achieve ε accuracy (of the optimal point) in less than $o(1/\sqrt{\varepsilon})$ iterations. Moreover, accelerated methods (which called *optimal* methods) to achieve the lower bounds for minimizing smooth convex and strongly convex functions are developed. Specially for the convex functions with bounded Hessian, the following iterations are proposed

$$\begin{cases} \hat{x}_{k+1} = x_k - \frac{1}{U}\nabla f(x_k), \\ x_{k+1} = \hat{x}_{k+1} + \frac{\sqrt{U}-\sqrt{L}}{\sqrt{L}+\sqrt{U}}(\hat{x}_{k+1} - \hat{x}_k), \\ \text{where} \qquad \hat{x}_0 = x_0. \end{cases} \tag{27}$$

Now we apply this method for consensus by dual decomposition of the optimization formulation of (19). Following the same procedure as above, we arrive at the iterations

$$\begin{cases} x_{k+1} = (I - \frac{1}{U}W)\,(x_k + b(x_k - x_{k-1})) \\ \text{where} \qquad b = \frac{\sqrt{U}-\sqrt{L}}{\sqrt{L}+\sqrt{U}} \end{cases} \tag{28}$$

Note that here again (as in (24)) $W = A^T A$. For the consensus case, we can compute lower and upper bounds of Hessian (21) with respect to μ. i.e., $\nabla^2 g(\mu) = AA^T$. So L and U relate to the smallest and the largest eigenvalues of *Laplacian*. Via techniques similar to those introduced earlier, and observing that the *Laplacian* has a simple smallest eigenvalue equal to zero, one finds that L corresponds to the second smallest eigenvalue of the Laplacian.

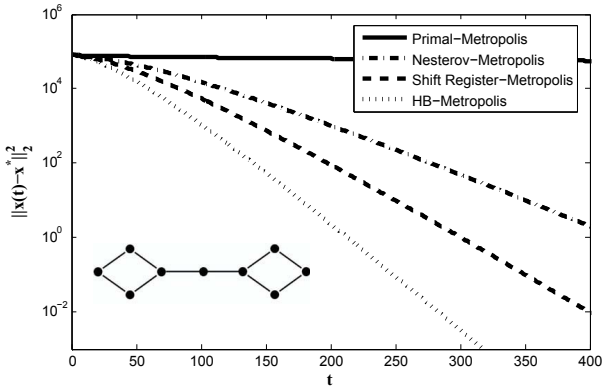

Fig. 2. Comparison of different consensus algorithms using Metropolis weight. simulation on a Dumbbell of 100 nodes: log scale of Euclidean distance from optimal point $\|x(t) - x^\star\|_2^2$ versus iteration number t.

4) Numerical Examples: Fig. 2 compares the different consensus algorithms on a dumbbell topology using a weight matrix designed using the *Metropolis-Hastings* scheme. Here, the accelerated gradient and shift register solutions converge much faster than the standard iterations and the iterations derived using Nesterov's order-optimal technique. Furthermore, the accelerated gradient method still outperforms the shift register techniques.

B. Accelerated Network Flow Control

To demonstrate the general applicability of accelerated gradient techniques to network optimization, we also present results from an attempt to develop accelerated methods for Internet congestion control. Network utility maximization (NUM) as a powerful framework for studying Internet congestion control and related problems has been attracted considerable attention during the last decade; see, e.g., [1], [12]. Almost all the congestion control protocols have been designed are adapting variations of the dual decomposition techniques employed in [12]. The optimal bandwidth sharing can be found by solving the following nonlinear program

$$\begin{array}{ll} \max\limits_{x_s \in \mathscr{I}_s} & \sum\limits_s u_s(x_s) \\ \text{subject to} & Rx \le c, \end{array} \tag{29}$$

where x_s is the source rate for client s, R is the Link-Source routing matrix and c is the vector of fixed link capacities. $\mathscr{I}_s = [m_s, M_s]$ imposes lower and upper bounds on the source rates. The utilities $u_s(x_s)$ are typically monotone increasing and strictly concave functions. This as well as the linear constrains cast the overall problem as a convex optimization. The analysis in [12] assumes that the utility functions are twice continuously differentiable with $0 < L < -u_s''(x_s) < U$ for $x_s \in \mathscr{I}_s$. Instead of solving primal problem which is not decomposable in network peers, the technique in the literature is to use dual problem as

$$q(\lambda) = \max_{x_s \in \mathscr{I}_s} \sum_s \left\{ u_s(x_s) - x_s \sum_l r_{ls}\lambda_l \right\} + \sum_l \lambda_l c_l \tag{30}$$

Note that the dual is separable in the end-to-end rates x_s. A solution for the dual problem can be found by letting each source optimize its own rate based on the knowledge of (the sum of) link prices λ_l which it uses to route the traffics. Meanwhile, link l updates its price by using a projected gradient iteration. To design an accelerated congestion control mechanism using the multi-step gradient techniques, we need to find upper and lower bounds on the Hessian. Following lemma offers required bounds

Lemma 2: The Hessian of the dual function (30) satisfies

$$\frac{L\bar{l}_{\max}\bar{s}_{\max}}{\sqrt{N_l}} \le \nabla^2 q(\lambda) \le U l_{\max} s_{\max} \tag{31}$$

Where N_l is the total number of links. Also $\bar{l}_{\max}$ and $\bar{s}_{\max}$ are lower bounds on $l_{\max}$ and $s_{\max}$, respectively.

Our attempt to accelerate the network flow problem results in a pricing algorithm on the form

$$\lambda_l^{k+1} = \mathscr{P}_\Lambda \left[\lambda_l^k + \alpha \left(\sum_l r_{ls} x_s^k - c_l \right) + \beta \left(\lambda_l^k - \lambda_l^{k-1} \right) \right] \tag{32}$$

The last term in above equation differs from the original formulation [12] and comes from the momentum term of the multi-step method. Disregarding the projection, the classical analysis of the heavy ball method [9] reveals that the iterations (32) with step size parameters satisfying

$$0 \le \beta < 1, \quad 0 < \alpha < \frac{2(1+\beta)}{U l_{\max} s_{\max}}$$

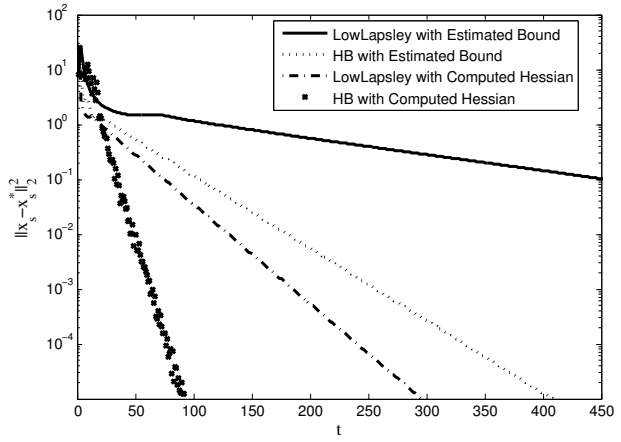

Fig. 3. Convergence behavior of Low-Lapsley versus HB. Plot shows log scale of the euclidean distance from optimal source rates $\|x_s - x_s^\star\|_2^2$ versus iteration number t.

converge to a nonsingular optimum point. The best step sizes corresponding to the suboptimal Hessian bounds (31) in the absence of the projection are given by

$$\beta^\star = \left(\frac{\sqrt{U l_{\max} s_{\max}} - \sqrt{L \bar{l}_{\max} \bar{s}_{\max} N_l^{-1/2}}}{\sqrt{U l_{\max} s_{\max}} + \sqrt{L \bar{l}_{\max} \bar{s}_{\max} N_l^{-1/2}}} \right)^2,$$
$$\alpha^\star = \frac{4}{(\sqrt{U l_{\max} s_{\max}} + \sqrt{L \bar{l}_{\max} \bar{s}_{\max} N_l^{-1/2}})^2}. \qquad (33)$$

1) Numerical Example: We consider a network with a routing matrix R of dimension 10×10. This matrix is full rank (providing unique equilibrium point) and is generated randomly. The utility functions of the sources are set to $a_s \log(1 + x_s)$, with $a_s = 1 \times 10^2$ for all sources. The bounds in (31) are computed from the routing matrix and the utility functions. Fig. 3 compares the performance of the classical optimization flow control and its accelerated variant. As it is shown in the plot, convergence of both algorithms with estimated step sizes are slower than what could be achieved if we could compute the Hessian in every iteration and adjust the step sizes accordingly. Note that also in this case the accelerated gradient method provides significant performance benefits over the gradient method.

VI. Conclusions

In this paper, we studied accelerated gradient methods for networked optimization problems. In particular, we investigated multi-step methods to accelerate center free resource allocation, distributed consensus and network flow control problems. We demonstrated both theoretically and numerically that our method outperforms existing algorithms. As a future direction, we would like to explore the effects of the uncertain parameters on the performance of gradient and multi-step methods and extend our analysis to cover the projected gradient methods as well.

Acknowledgment

The authors wish to thank Björn Johansson for helpful discussions. This work was sponsored in part by the European Commission project WIDE and the Swedish Foundation for Strategic Research project RAMCOORAN.

References

[1] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. of Operational Research Society*, vol. 49, pp. 237–252, 1998.

[2] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc of the IEEE*, vol. 95 Issue: 1, pp. 215–233, 2007.

[3] S. Barbarossa and G. Scutari, "Decentralized maximum likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems," *IEEE Trans on Signal Processing*, vol. 55, pp. 3456–3470, 2007.

[4] D. Goodman and N. Mandayam, "Power control for wireless data," *Personal Communications, IEEE*, vol. 7 Issue:2, pp. 48–54, 2000.

[5] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans on Automatic Control*, vol. 31, pp. 803–812, 1986.

[6] M. Cao, D. A. Spielman, and E. M. Yeh, "Accelerated gossip algorithms for distributed computation," in *44th Allerton Conf on Communication, Control, and Computation*, 2006.

[7] B. Johansson, "On distributed optimization in networked systems," Ph.D. dissertation, Royal Institute of Technology, 2008.

[8] D. P. Bertsekas., *Nonlinear Programming*. Athena Scientific, 1999.

[9] B. Polyak, *Introduction to Optimization*. ISBN 0-911575-14-6, 1987.

[10] Y. C. Ho, L. Servi, and R. Suri, "A class of center-free resource allocation algorithms," *Large Scale Systems*, vol. 1, pp. 51–62, 1980.

[11] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans on Inf Theory*, vol. 52, pp. 2508–2530, 2006.

[12] S. H. Low and D. E. Lapsley, "Optimization flow control - i: Basic algorithm and convergence." *IEEE/ACM Trans on Networking*, vol. 7 Issue: 6, pp. 861–874, 1999.

[13] L. Xiao and S. Boyd, "Optimal scaling of a gradient method for distributed resource allocation," *J. Opt. Theory and Applications*, vol. 129 Issue:3, pp. 469–488, 2006.

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

[15] E. Ghadimi, M. Johansson, and I. Shames, "Accelerated gradient methods for networked optimization," KTH, 2010, Tech. Rep.

[16] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53 Issue: 1, pp. 65–78, 2004.

[17] M. Rabbat and R. Nowak, "Generalized consensus computation in networked systems with erasure links," in *IEEE SPAWC*, 2005.

[18] I. Schizas, A. Ribeiro, and B. Giannakis, "Consensus-based distributed parameter estimation in ad hoc wireless sensor networks with noisy links," in *IEEE ICASSP*, 2007.

[19] J. Liu, B. D. O. Anderson, M. Cao, and A. S. Morse, "Analysis of accelerated gossip algorithms," in *48th IEEE CDC*, 2009, pp. 871–876.

[20] G. H. Golub and R. S. Varga, "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods," *Numerische Matematik*, vol. 3, pp. 147–156, 1961.

[21] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer-Verlag New York, LCC, 2003.