Approximate Augmented Lagrangians for Distributed Network Optimization

Nikolaos Chatzipanagiotis, Darinka Dentcheva and Michael M. Zavlanos

Abstract— **In this paper, we propose a distributed algorithm for optimal routing in wireless multi-hop networks. We build our approach on a recently proposed model for stochastic routing, whereby each node selects a neighbor to forward a packet according to a given probability distribution. Our solution relies on dual decomposition techniques with regularization, that can significantly improve on the slow convergence of subgradient methods. In particular, we employ the method of augmented Lagrangians (AL). While regularization introduces coupling of the primal variables, a recently proposed iterative approximation technique can be used to decouple the minimization problem in the augmented Lagrangian method (ALM). Once the approximation reaches a predetermined number of iterations it is terminated and followed by a novel update of the Lagrange multipliers, that differs from that in the standard ALM. We show that truncating the approximation is necessary to obtain a fully distributed approach, and that the proposed update of the Lagrange multipliers is critical to obtain convergence of our method. An additional advantage of our approach is that convergence is very fast even for sparse networks, where techniques that incorporate consensus iterations into the algorithm tend to be slow.**

I. Introduction

Communication networks are widely used to transport information between source nodes and their intended destinations. This is often achieved in a multi-hop fashion, where information reaches its destination following a path of intermediate nodes that are selected based on their ability to relay data. Upon defining desired optimization criteria, such as end-to-end rates or available resources, optimal data rates and routes can be determined as the solution of an appropriate Network Utility Maximization (NUM) problem [1], [2].

Since modern communication networks are often large-scale, centralized approaches to network optimization can incur large communication costs, entail significant delays and are vulnerable to failures, giving rise to the need for distributed optimization techniques. Dual decomposition has been extensively used for this purpose because it provides a simple, yet decentralized, subgradient algorithm that can ensure global optimality [3]–[5]. While subgradient (or first order) methods can provide computationally efficient and distributed algorithms, they typically suffer from slow convergence rates [6]–[8]. This has, recently, motivated the use of distributed, second order Newton methods [9]–[11], where the challenge now lies in computing the Newton direction and stepsize, which require global information. This is resolved in [9] using a belief propagation algorithm, while in [11] a Taylor expansion of the inverse Hessian is proposed to compute the Newton direction using local information. The common disadvantage of these approaches is the requirement that all utility functions are strictly concave. Moreover, distribution may involve consensus [10], [11], a method that behaves best for dense networks [12].

In this paper, we propose the use of augmented Lagrangians (AL) for network optimization. Unlike Newton's method, their use does not require a strictly concave utility function because strict convexity of the AL function is obtained by adding a quadratic penalty term to the ordinary Lagrangian [13]–[15]. AL methods converge very fast, especially compared to first order methods, however they lack the decomposability properties of the ordinary Lagrangian. Decomposition techniques have been proposed in [16]–[20]. In this paper, we focus on the technique proposed in [21], where the key idea is to replace each minimization step by minimizing a separable iteratively updated approximation of the AL function. We propose a novel update of the Lagrange multipliers that allows for convergence of the algorithm even when this approximation is not exact, which, as we show is necessary to obtain a fully distributed method. An additional advantage of our approach is that convergence is very fast even for sparse networks. Therefore, it is better suited for large problems, compared to approaches that utilize consensus for distributed computation. A similar AL approach to stochastic optimal routing was proposed in [22], wherein the recovered primal variables seem to exhibit oscillatory behavior.

The rest of the paper is organized as follows: In Section II, we discuss optimal routing in wireless networks and present a convex programming formulation. In Section III, we propose a solution based on ALs and in particular on the distributed method introduced in [21]. Finally, in Section IV, we present simulation results and discuss several ways to improve convergence speed of the proposed algorithm.

II. Problem Definition

In this paper, we employ a model of stochastic routing that guarantees deliverability, as proposed in [23] and further studied in [24], [25] in the context of mobile networks. According to this model, a set of J source nodes generate data at a normalized average rate $r_i \in [0, 1]$, measured in packets per unit time, and forward this information in a

This work is supported by the National Science Foundation under awards CNS #1054604 and CMMI #0965702.

Nikolaos Chatzipanagiotis and Michael M. Zavlanos are with the Dept. of Mechanical Engineering, Duke University, Durham, NC, 27708, USA {n.chatzip,michael.zavlanos}@duke.edu. Darinka Dentcheva is with the Dept. of Mathematics, Stevens Institute of Technology, Hoboken, NJ 07030, USA ddentche@stevens.edu.

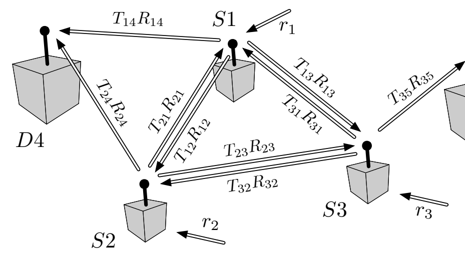


Fig. 1. Wireless network consisting of two destinations (D) and three sources (S). Shown are the packet rates r_i generated by every source as well as the rates $T_{ij}R_{ij}$ sent from source i and successfully decoded by node j, where T_{ij} is the probability that node i routes packets to node j and R_{ij} is the reliability of the channel between nodes i and j.

multi-hop fashion to a set of K destinations (or sinks). We denote by $\mathcal{J} = \{1, \dots, J\}$ and $\mathcal{K} = \{J+1, \dots, J+K\}$ the sets of sources and destinations, respectively. Let $R_{ij} \in [0,1]$ represent the probability that a packet transmitted from node i is correctly decoded by node j. Setting the transmission rate of the terminals' radios to unity, for simplicity of notation, the effective transmission rate from i to j is given by R_{ij}. Since destinations do not transmit information, we set $R_{ij} = 0$ for all $i \in \mathcal{K}$.

In the proposed framework, nodes can either route packets directly to a destination $j \in \mathcal{K}$ if $R_{ij} > 0$, or relay them to a nearby node $j \in \mathcal{J}$ for subsequent transmission. We model this process by defining routing probabilities T_{ij} denoting the probability with which node i selects node j, either source or sink, as a destination of its transmitted packets. The transmitted packets are then successfully received by node j with probability R_{ij}. Since these two events are independent, $T_{ij}R_{ij}$ denotes the normalized rate at which packets are sent from node i to node j; see Fig. 1.

To store packets between transmissions, every node i maintains a queue; see Fig. 2. Denoting by $\mathcal{N}_i = \{j \in \mathcal{J} \cup \mathcal{K} \mid R_{ij} > 0\}$ the set of neighbors of node i, the average rate of incoming packets at node i is $r_i + \sum_{j \in \mathcal{N}_i} T_{ji}R_{ji}$. Similarly, the average rate of packets departing the queue at node i is $\sum_{j \in \mathcal{N}_i} T_{ij}R_{ij}$. If the rate of packets entering is less than the rate of packets leaving the queue, i.e. $r_i + \sum_{j \in \mathcal{N}_i} T_{ji}R_{ji} \leq \sum_{j \in \mathcal{N}_i} T_{ij}R_{ij}$, then we say that the queue is stable. The goal of stochastic optimal routing is then to determine variables r_i and T_{ij} that maximize a given network utility $f(r_1, \dots, r_J) = \sum_{i \in \mathcal{J}} f_i(r_i)$ that quantifies the preference towards larger rates and possibly expresses risk aversion, subject to flow constraints at every node as well as bounds on the minimum rates of exogenous data acquisition $r_i^{min} \geq 0$ [23], i.e.

$$\begin{aligned} &\max \left[f(r_1, \dots, r_J) = \textstyle\sum_{i \in \mathcal{J}} f_i(r_i)\right] \\ &\text{s.t. } r_i + \textstyle\sum_{j \in \mathcal{N}_i} T_{ji}R_{ji} \leq \textstyle\sum_{j \in \mathcal{N}_i} T_{ij}R_{ij}, \quad \forall i \in \mathcal{J} \\ &\qquad \textstyle\sum_{j \in \mathcal{N}_i} T_{ij} \leq 1, \;\; r_i \geq r_i^{min}, \quad \forall i \in \mathcal{J}. \end{aligned} \tag{1}$$

Depending on the utility functions $f_i(r_i)$, problem (1) can attain a simple linear or concave form and can be solved efficiently using available techniques [6], [26]. However, centralized solutions to (1) can incur a large communication cost to collect information about the network topology, contained in the reliabilities R_{ij}, at a central location and communicate routes T_{ij} and rates r_i back to the nodes. They also entail significant delays and are vulnerable to failures. For this reason, distributed algorithms are most desirable, whereby each node i has access only to variables available to its neighbors in $\mathcal{N}_i$.

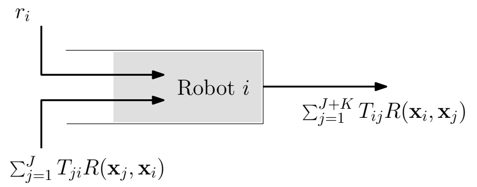

r_i
Robot i
$\sum_{j=1}^{J+K} T_{ij}R(\mathbf{x}_i, \mathbf{x}_j)$
$\sum_{j=1}^{J} T_{ji}R(\mathbf{x}_j, \mathbf{x}_i)$

Fig. 2. Incoming and departing flows at terminal i.

III. The Method of Augmented Lagrangians

While dual subgradient methods that exploit the separable structure of the ordinary Lagrangian have been successfully used in the past to obtain distributed solutions to problems such as ours, (e.g., [2]–[5]), they suffer from inherent disadvantages, such as slow convergence speed and instability when the dual function is non-differentiable at the optimal solution. This motivates alternative methods that take advantage of regularization techniques. A widely used primal regularization technique is the *augmented Lagrangian Method* (ALM) [6], [27], [28], which we present in what follows.

As the distributed version of this method is developed for affine equality constraints, we define the slack variables $s_i \geq 0, \ \forall i \in \mathcal{J}$ and rewrite the queue flow constraints as

$$\sum_{j \in \mathcal{N}_i} T_{ij}R_{ij} - \sum_{j \in \mathcal{N}_i} T_{ji}R_{ji} - r_i = s_i, \quad \forall i \in \mathcal{J}. \tag{2}$$

Denoting by $\mathbf{z}_i = \left[s_i, r_i, T_{i1}, T_{i2}, \dots, T_{i(J+K)}\right]^T \in \mathbb{R}^{J+K+2}$ the vector of decision variables of node i, and by $\mathbf{R}_i \in \mathbb{R}^{J \times (J+K+2)}$ the matrix

$$\mathbf{R}_i = \begin{bmatrix} 0 & 0 & -R_{i1} & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 0 & -R_{i2} & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \dots & \vdots \\ -1 & -1 & R_{i1} & R_{i2} & \dots & R_{iJ} & \dots & R_{i(J+K)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -R_{iJ} & \dots & 0 \end{bmatrix},$$

the flow constraints (2) can be compactly written as $\sum_{i \in \mathcal{J}} \mathbf{R}_i \mathbf{z}_i = 0$. We convert problem (1) to a minimization problem and associate Lagrange multiplies $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_J]^T \in \mathbb{R}^J$ with constraints (2). The AL Λ for this problem takes the form

$$\Lambda(\mathbf{z}, \boldsymbol{\lambda}) = \overbrace{\sum_{i \in \mathcal{J}} \Big(-f_i(\mathbf{z}_i) + \boldsymbol{\lambda}^T \mathbf{R}_i \mathbf{z}_i\Big)}^{\text{Ordinary Lagrangian}} + \overbrace{\frac{1}{2}\rho \Big\| \sum_{i \in \mathcal{J}} \mathbf{R}_i \mathbf{z}_i \Big\|_2^2}^{\text{Penalty term}} \tag{3}$$

Algorithm 1 Augmented Lagrangian Method (ALM)

```
1: For fixed Lagrange multipliers $\boldsymbol{\lambda}^k$, find primal variables
   $\mathbf{z}^k$ that solve the problem:
       $$\begin{array}{ll} \mathbf{z}^k & = \arg\min_{\mathbf{z}} \Lambda(\mathbf{z}, \boldsymbol{\lambda}^k) \\ \text{s.t.} & \mathbf{z}_i \in \mathcal{Z}_i \end{array} \quad . \qquad (5)$$
2: If the constraints $\sum_{i\in\mathcal{J}} \mathbf{R}_i \mathbf{z}_i = \mathbf{0}$ are satisfied, then stop
   (optimal solution found). Otherwise, set :
       $$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho \left( \sum_{i\in\mathcal{J}} \mathbf{R}_i \mathbf{z}_i^k \right) \qquad (6)$$
   increase $k$ by one and return to Step 1.
```

where $\rho \in \mathbb{R}_+$ is a user defined penalty parameter and $\mathbf{z} = [\mathbf{z}_1, \ldots, \mathbf{z}_J]^T$. The remaining constraints in (1) are local at each node and we do not include them in the Lagrangian (3). We denote the set of points satisfying these local constraints by $\mathcal{Z} \triangleq \bigcup_{i\in\mathcal{J}} \mathcal{Z}_i$, where

$$\mathcal{Z}_i = \left\{ \mathbf{z}_i \in \mathbb{R}^{J+K+2} | \text{ s.t. } \mathbf{e}_i^T \mathbf{z}_i \le 1, \ \mathbf{z}_i \ge 0 \right\} \qquad (4)$$

with $\mathbf{e}_i$ a suitably defined 0-1 vector. Note that the explicit constraint $\mathbf{z}_i \ge 0$ can be easily manipulated to include $r_i \ge r_i^{min}$, if needed. The ALM is summarized in Alg. 1. Recall that the subgradient of the dual function is given by the amount of constraint violations emerging from the update of primal variables at each step. For a more detailed discussion see [6], [28].

A. A Distributed Algorithm

The ALM is enticing, however, a major drawback stems from the fact that (3) is not in a separable form, due to the quadratic penalty term and the underlying inner products $\langle \mathbf{R}_i \mathbf{z}_i, \mathbf{R}_j \mathbf{z}_j \rangle$ that form, when expanded. To address this issue, we employ the approach developed in [21] to iteratively decouple the first step of ALM and propose a novel modification of the dual update that allows full distribution of our method; see Sec. III-B. In particular, as in [21], define the *local augmented Lagrangian* associated with node i by

$$\Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}, \boldsymbol{\lambda}) = -f_i(\mathbf{z}_i) + \boldsymbol{\lambda}^T \mathbf{R}_i \mathbf{z}_i + \frac{1}{2}\rho \left\| \mathbf{R}_i \mathbf{z}_i + \sum_{j\neq i} \mathbf{R}_j \tilde{\mathbf{z}}_j \right\|_2^2 \qquad (7)$$

where $\tilde{\mathbf{z}}_j \in \mathbb{R}^{J+K+2}$ denotes the primal variables that are local to node j but communicated to node i for optimization of its local Lagrangian Λ_i. With respect to node i, these are considered fixed parameters. The main idea behind the proposed approach is to replace problem (5) by J parallel problems of the form $\mathbf{z}_i^k = \arg\min_{\mathbf{z}_i} \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}^k, \boldsymbol{\lambda}^k)$ and to iteratively update the estimates $\tilde{\mathbf{z}}$ by making steps towards a correct approximation of (5). Now, the objective is to devise a suitable procedure for updating and minimizing $\tilde{\Lambda}(\mathbf{z}, \tilde{\mathbf{z}}, \boldsymbol{\lambda}) = \sum_{i=1}^J \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}, \boldsymbol{\lambda})$, which will ensure the minimization of the true AL $\Lambda(\mathbf{z}, \boldsymbol{\lambda})$. This can be achieved by introducing an inner loop in place of step 1 of Alg. 1, indexed by m, wherein at the m-th inner iteration each node i computes

$$\mathbf{z}_i^{k,m} = \arg\min_{\mathbf{z}_i \in \mathcal{Z}_i} \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}^{k,m}, \boldsymbol{\lambda}^k) \qquad (11)$$

based on the values of $\tilde{\mathbf{z}}^{k,m}$ and $\boldsymbol{\lambda}^k$ communicated by its neighbors at the end of the previous iteration. Then, the minimizer $\mathbf{z}_i^{k,m}$ is subsequently used to update the local estimate $\tilde{\mathbf{z}}_i^{k,m}$ according to the equation

$$\tilde{\mathbf{z}}_i^{k,m+1} = \tilde{\mathbf{z}}_i^{k,m} + \tau \left(\mathbf{z}_i^{k,m} - \tilde{\mathbf{z}}_i^{k,m} \right), \qquad (12)$$

where $\tau > 0$ is an appropriately chosen parameter (see Sec. III-B). The variables $\tilde{\mathbf{z}}_i^{k,m+1}$ can now be transmitted to the neighbors for the next inner iteration. As in [21], the stopping criterion for the inner loop is satisfied when $\mathbf{R}_i \mathbf{z}_i^{k,m} = \mathbf{R}_i \tilde{\mathbf{z}}_i^{k,m}$ for all i and the algorithm then moves to the next step of updating the dual variables $\boldsymbol{\lambda}^k$. Nevertheless, unlike [21], we propose to update the dual variables using the $\tilde{\mathbf{z}}$ variables instead of the variables $\mathbf{z}$ and allow the updates even if the test in step 3 is not satisfied. As we discuss in Sec. III-B, this modification ensures that our algorithm converges even if the synchronization in the entire network (corresponding to the test in step 3) was not completed, which is critical for a fully distributed implementation. Our proposed method is illustrated in Alg. 2.

Algorithm 2 Accelerated Distributed AL (ADAL)

```
Require: Define $M$, set $k = 1$, choose initial Lagrange
   multipliers $\boldsymbol{\lambda}^1$ and variables $\mathbf{z}^0$.
1: Set $\tilde{\mathbf{z}}^{k,0} = \mathbf{z}^{k-1}$ and $m = 0$.
2: If $m = M$ then set $\mathbf{z}^k = \tilde{\mathbf{z}}^{k,m}$ and go to Step 4.
   Otherwise, for all nodes $i \in \mathcal{J}$ determine $\mathbf{z}_i^{k,m}$ as a
   solution of
       $$\begin{array}{ll} \min_{\mathbf{z}_i} & \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}^{k,m}, \boldsymbol{\lambda}^k) \\ \text{s.t.} & \mathbf{z}_i \in \mathcal{Z}_i \end{array} \quad . \qquad (8)$$
3: If $\mathbf{R}_i \mathbf{z}_i^{k,m} = \mathbf{R}_i \tilde{\mathbf{z}}_i^{k,m}, \forall i \in \mathcal{J}$, then go to Step 4.
   Otherwise, for $i \in \mathcal{J}$ set
       $$\tilde{\mathbf{z}}_i^{k,m+1} = \tilde{\mathbf{z}}_i^{k,m} + \tau \left( \mathbf{z}_i^{k,m} - \tilde{\mathbf{z}}_i^{k,m} \right), \qquad (9)$$
   communicate $\tilde{\mathbf{z}}_i^{k,m+1}$ to adjacent nodes, increase $m$ by
   1, and go to Step 2.
4: If $\sum_{i\in\mathcal{J}} \mathbf{R}_i \mathbf{z}_i^k = \mathbf{0}$, then stop (optimal solution found).
   Otherwise, set
       $$\boldsymbol{\lambda}_i^{k+1} = \boldsymbol{\lambda}_i^k + \rho \left[ \sum_{j\in\mathcal{J}} \mathbf{R}_j \tilde{\mathbf{z}}_j^k \right]_i \qquad (10)$$
   communicate $\lambda_i^{k+1}$ to adjacent nodes, increase $k$ by 1,
   and go to Step 1.
```

Note that every step in Alg. 2 requires that each node has access only to variables readily available in its neighborhood. To see this, expand the quadratic terms in (7) to get

$$\begin{aligned} \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}, \boldsymbol{\lambda}) = -f_i(\mathbf{z}_i) + \sum_{j\in\mathcal{J}} \lambda_j [\mathbf{R}_i]_j^T \mathbf{z}_i + & \qquad (13) \\ + \frac{1}{2}\rho \sum_{j\in\mathcal{J}} \left([\mathbf{R}_i]_j^T \mathbf{z}_i + \sum_{k\neq i} [\mathbf{R}_k]_j^T \tilde{\mathbf{z}}_k \right)^2, & \end{aligned}$$

where the notation $[\mathbf{R}_i]_j$ stands for the j-th row of matrix $\mathbf{R}_i$. Noticing that $[\mathbf{R}_i]_j \neq \mathbf{0}$ if and only if $j \in \mathcal{N}_i \cup \{i\}$, we obtain

$$\sum\nolimits_{j\in\mathcal{J}} \lambda_j [\mathbf{R}_i]_j^T \mathbf{z}_i = \sum\nolimits_{j\in\mathcal{N}_i\cup\{i\}} \lambda_j [\mathbf{R}_i]_j^T \mathbf{z}_i. \qquad (14)$$

Similarly,

$$\sum_{j\in\mathcal{J}} \Big([\mathbf{R}_i]_j^T \mathbf{z}_i + \sum_{k\neq i}[\mathbf{R}_k]_j^T \tilde{\mathbf{z}}_k\Big)^2 = \sum_{j\notin\mathcal{N}_i\cup\{i\}} \Big(\sum_{k\neq i}[\mathbf{R}_k]_j^T \tilde{\mathbf{z}}_k\Big)^2$$

$$+ \sum\nolimits_{j\in\mathcal{N}_i\cup\{i\}} \Big([\mathbf{R}_i]_j^T \mathbf{z}_i + \sum\nolimits_{k\in\mathcal{N}_j\cup\{j\}}[\mathbf{R}_k]_j^T \tilde{\mathbf{z}}_k\Big)^2 \qquad (15)$$

since $[\mathbf{R}_k]_j \neq \mathbf{0}$ if and only if $k \in \mathcal{N}_j \cup \{j\}$. Substituting (14) and (15) in (13) and observing that the first term on the right-hand-side of (15) does not depend on $\mathbf{z}_i$, we obtain

$$\min_{\mathbf{z}_i} \Lambda_i(\mathbf{z}_i, \tilde{\mathbf{z}}, \boldsymbol{\lambda}) = \min_{\mathbf{z}_i} \Big\{ - f_i(\mathbf{z}_i) + \sum_{j\in\mathcal{N}_i\cup\{i\}} \lambda_j [\mathbf{R}_i]_j^T \mathbf{z}_i +$$

$$+ \frac{1}{2}\rho \sum\nolimits_{j\in\mathcal{N}_i\cup\{i\}} \Big([\mathbf{R}_i]_j^T \mathbf{z}_i + \sum\nolimits_{k\in\mathcal{N}_j\cup\{j\}}[\mathbf{R}_k]_j^T \tilde{\mathbf{z}}_k\Big)^2 \Big\},$$

which reveals that calculation depends on receiving information from the 2-hop neighbors, thus requiring minimal communication overhead. Note also that the term $\Big[\sum_{j\in\mathcal{J}} \mathbf{R}_j \tilde{\mathbf{z}}_j^k\Big]_i$ in (10) is the flow constraint violation (2) at each node i. This can be easily computed by each node, since it only requires knowledge of its neighbors' variables.

B. Discussion on Distribution & Convergence

As discussed in Sec. III-A, the main novelty of our algorithm, compared to the one proposed in [21], consists in the rule for for updating the dual variables and in the step for that update in terms of the variables $\tilde{\mathbf{z}}$ instead of $\mathbf{z}$. This allows for convergence of Alg. 2 even if the inner loop is truncated to exit after a pre-determined maximum number of iterations M is reached, instead of being executed until convergence as in [21]. This, in turn, leads to great improvements in the behavior of the algorithm. More specifically, note that the termination criterion $\mathbf{R}_i\mathbf{z}_i^{k,m} = \mathbf{R}_i\tilde{\mathbf{z}}_i^{k,m}, \forall i \in \mathcal{J}$ of the inner loop in step 3 of Alg. 2 requires global information, since it must be satisfied by all agents concurrently. This requires a synchronization in the network, which is a serious bottleneck in distributed applications, especially for large networks, and can be avoided by being able to pre-determine M. Although theoretical results that characterize the performance of Alg. 2 for fixed M are the subject of future work, numerical analysis in Section IV reveals that convergence is, surprisingly, greatly accelerated by decreasing the value of M, with the best performance obtained for $M = 1$. Note that if we let $M \to \infty$, then the original algorithm in [21] is obtained. In that case the inner loop termination criterion $\mathbf{R}_i\mathbf{z}_i = \mathbf{R}_i\tilde{\mathbf{z}}_i, \forall i \in \mathcal{J}$ becomes satisfied and the update (10) becomes equivalent to (6). In this case, convergence of Alg. 2 follows directly form [21]. Also note that, in most practical applications, the algorithm runs throughout the whole life of the network. Thus, the outer loop termination criterion involving all constraints ($\sum_{i\in\mathcal{J}} \mathbf{R}_i\mathbf{z}_i = \mathbf{0}$) is satisfied for the stationary routing policy.

While a detailed convergence analysis of the algorithm proposed in [21] for general convex optimization problems can be found in the aforementioned reference, here we discuss some results from [21] that are particularly relevant to network optimization. The first result states that convergence is guaranteed if $0 \leq \tau \leq \frac{1}{N}$ (step 3 of Alg. 2), where $N = \max_i |\mathcal{N}_i|$ is the cardinality of the largest neighborhood in the network. The second results states that the progress of the inner loop at each iteration is proportional to $q = 1 - \frac{\tau(1-\tau N)}{2\rho\alpha^2 N^2\gamma^{-1}+1}$, where $\alpha = \max_{1\leq i\leq J} \|\mathbf{R}_i\|_2$. Therefore, convergence speed depends on q, τ, α and N, all of which are easily seen to be affected by sparsity. In fact, convergence speed is increased for sparse networks. This particularly interesting feature of the algorithm, also verified in simulations, removes the necessity of consensus iterations to propagate information in the network, which is present in recently proposed, distributed approaches [10], [11]. Consensus algorithms are known to be efficient for dense network topologies and exhibit slow convergence in sparse networks. This disadvantage of consensus also affects problems that incorporate interference into the formulation [29]. Interestingly, simulations suggest that ADAL with $M = 1$ converges very fast not only for sparse, but also for dense networks, see Sec. IV.

IV. Numerical Simulations

Numerical experiments were performed in order to assess the performance of the Distributed AL (DAL) [21] and Accelerated Distributed AL (ADAL) methods under different sets of parameters. In these simulations, the maximum residual, i.e., the maximum constraint violation among all agents, was monitored as a metric of convergence. Subsequent simulations refer to networks consisted of 50 sources and 2 sinks, wherein the agents were randomly, uniformly distributed in rectangular boxes. The ADAL algorithm is implemented for $M = 1$, unless otherwise stated.

Implementation of the DAL and ADAL algorithms necessitates appropriate selection of the, performance defining, parameters ρ and τ. The penalty parameter ρ is user defined, based on the effect it has on the behavior of the algorithm, and we have found that fastest convergence is obtained for $\rho \in [10, 30]$, while preventing ill-conditioning. On the other hand, while in [21] the optimal value for $\tau = \frac{1}{2\max|\mathcal{N}_i|}$ is suggested, in ADAL simulations this could be increased to $\tau = \frac{1.5}{\max|\mathcal{N}_i|}$ without compromising convergence, while at the same time accelerating the procedure.

The requirement on the utility functions $f_i(r_i)$ is that they are monotonically non-decreasing expressing preference to larger transmission rates, i.e. an increase in the rate of one node does not decrease the value of the total utility function to be maximized. In our simulations, we choose $f(\mathbf{r}) = \prod_{i=1}^{J}(r_i)$ in order to maximize the product of rates, which can be recast as the sum of logarithms $f(\mathbf{r}) = \sum_{i=1}^{J} \log(r_i)$. This choice is typical in NUM problems and aims to produce a fairer distribution of rates among all nodes in the network. Note that the choice $f(\mathbf{r}) = w^T\mathbf{r}$, with w a vector of weights, results in a trivial problem, in which the nodes in communication range of the sinks are rewarded with the maximum rate 1 and the rest with 0. Fig. 3 shows the

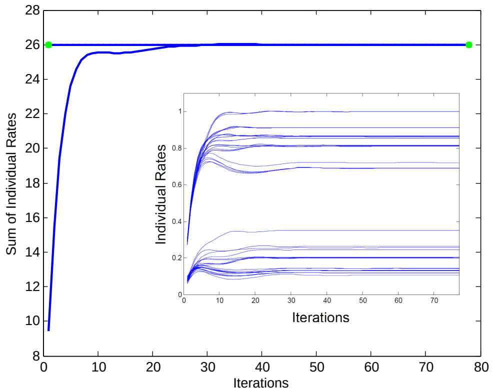


Fig. 3. Evolution of the sum of rates $\sum_{i\in\mathcal{J}} r_i$ during implementation of ADAL. The horizontal line depicts the value obtained by solving the centralized problem. We observe that the distributed utility function converges to the optimal solution very fast. Also included is the subfigure illustrating the evolution of individual rates for every source.

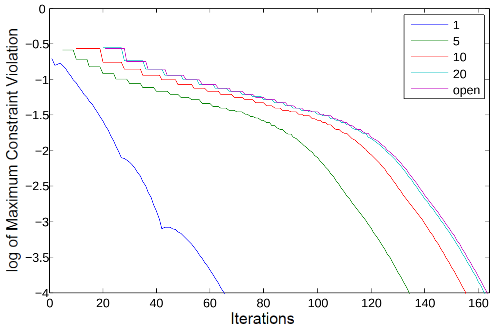


Fig. 4. Convergence speed for different exit criteria from the inner loop. The line labeled 'open' accounts for repeating the inner loop until convergence (10^{-3}), while in the other instances we force exit if the indicated amount of iterations M has been surpassed. We number the horizontal axis with respect to inner loop iterations. The step shape is caused because the constraint violations are updated at each outer loop iteration.

evolution of the individual and total rates, r_i and $\sum_{i=1}^{J} r_i$, respectively, corresponding to maximization of the utility $f(\mathbf{r}) = \sum_{i=1}^{J} \log(r_i)$ for fair allocation. We observe that the utility converges to its optimal value in only about 25 iterations.

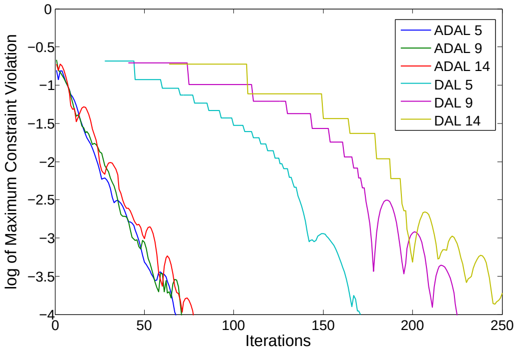


Fig. 5. Speed of convergence for different network densities for the ADAL and DAL methods. The results correspond to networks of 50 sources and 2 sinks with maximum degrees of 5,9 and 14 respectively.

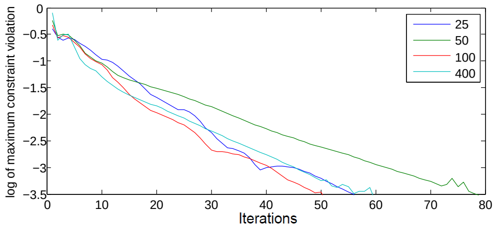


Fig. 6. Convergence speed of ADAL for different network sizes. The ratio of sources-to-sinks is kept at 25/1 and the maximum degree is 6 for all cases.

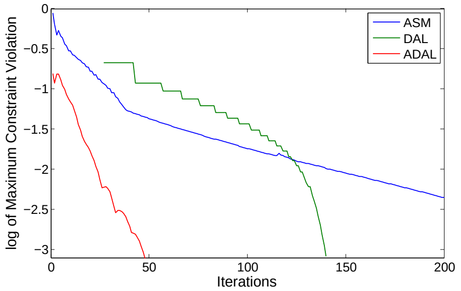
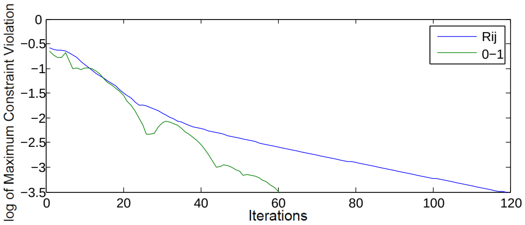


Fig. 7. Comparison between the ASM, DAL and ADAL methods, for a network of 50 sources, 2 sinks and maximum degree of 7.

In what follows, we illustrate convergence of ADAL for different values of inner loop iterations M, different network structures and different network sizes. Note first that, for the same network topology, the convergence speed of ADAL increases as M decreases with the best performance obtained for a single inner iteration $M = 1$; see Fig. 4. Therefore, not only is ADAL fully distributed compared to DAL (see Section III-B), but it is also faster. Moreover, while faster convergence for sparser networks, as are many modern large-scale wireless networks, is a unique feature of DAL, the convergence speed of ADAL is shown not to be affected by sparsity, while still being faster than DAL; see Fig. 5. A third important observation is that network size does not affect speed of convergence dramatically; see Fig. 6. Repeated simulations have shown that convergence speed remains at this level of magnitude. Note that, in all cases, the ratio of sources-to-sinks has been maintained the same at 25/1, in an effort to keep the randomly generated networks as similar as possible.

In order to compare our approach with other methods in the literature, we conducted additional simulations, wherein we implemented a version of the so called *Alternating Directions Method of Multipliers* (ADMM), due to its fast convergence speed, for small though accuracies [30]. More specifically, we compared our algorithm with the *Alternating Step Method* (ASM) presented in [31], which is an adaptation of the Generalized ADMM for monotropic programming problems, such as ours. As can be shown in Fig. 7, both DAL and ADAL outperform ASM with ADAL converging the fastest, in only 50 iterations.

Finally, we evaluated performance of our algorithm for different values of the channels reliabilities R_{ij}. In particular,



(a)

Fig. 8. Comparative diagram of the evolution of two simulations conducted on the same network topology. The first simulation uses R_{ij} as described in text and the second takes channel reliabilities to be of the form 0-1.

we modeled R_{ij} as a twice differentiable, decreasing function of the inter-node distance, ranging from 1 to 0, as with polynomial fitting of curves found in literature [32]. Fig. 8 depicts that, if the R_{ij} attain values in the interval $[0, 1]$, convergence is slower. This is an interesting observation that requires further investigation.

V. Conclusions

In this paper, we consider the problem of optimal stochastic routing in wireless data networks, based on a model proposed in [23]. A provably correct algorithm was proposed to solve the emerging network optimization problem in a distributed fashion. The advantages of the proposed approach are that it is a first order method utilizing augmented Lagrangians, thus, it combines low computational complexity with the robustness and convergence speed properties of regularization. Moreover, it requires minimal communication overhead and is also known to converge faster for sparse networks, qualities which are sought after in applications such as mobile ad hoc networks. Finally, numerical analysis has revealed that great improvement can be achieved by the proposed modification of the method from [21] . Future research will aim towards providing theoretical verification of this observation.

References

[1] F. P. Kelly, A. Maulloo, and D. Tan, “Rate control for communication networks: shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[2] S. H. Low and D. E. Lapsley, “Optimization flow control I: basic algorithm and convergence,” *IEEE/ACM Trans. Netw.*, vol. 7, 1998.

[3] L. Xiao, M. Johansson, and S. Boyd, “Simultaneous routing and resource allocation via dual decomposition,” *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, July 2004.

[4] 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*, vol. 9, no. 1, pp. 255–312, 2007.

[5] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Trans. on Automatic Control*, vol. 54, no. 1, pp. 58–61, 2009.

[6] A. Ruszczynski, *Nonlinear Optimization*. Princeton University Press, 2006.

[7] D. Bertsekas, A. Nedic, and A. Ozdaglar, “Convex analysis and optimization,” in *Series in Computational Mathematics*. Athena Scientific, 2003.

[8] A. Nedic and A. E. Ozdaglar, “Subgradient methods in network resource allocation: Rate analysis,” in *Proc. of CISS*, 2008, pp. 1189–1194.

[9] D. Bickson, Y. Tock, A. Zymnis, S. Boyd, and D. Dolev, “Distributed large scale network utility maximization,” in *Proceedings of the 2009 IEEE international conference on Symposium on Information theory*, 2009.

[10] E. Wei, A. Ozdaglar, and A. Jadbabaie, “A distributed newton method for network utility maximization,” in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010.

[11] M. Zargham, A. Ribeiro, A. Jadbabaie, and A. Ozdaglar, “Accelerated dual descent for network optimization.” in *Proceedings of the American Control Conference*, San Francisco, CA, 2011.

[12] A. Jadbabaie, J. Lin, and A. S. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, February 2005.

[13] M. Hestenes, “Multiplier and gradient methods,” *Journal of Optimization Theory and Applications*, vol. 4, pp. 303–320, 1969.

[14] M. J. D. Powell, “A method for nonlinear constraints in minimization problems.” Optimization, Academic press, London, 1969.

[15] R. Rockafellar, “Augmented Lagrange multiplier functions and duality in nonconvex programming,” *SIAM Journal on Control*, vol. 12, pp. 268–285, 1973.

[16] P. Tatjewski, “New dual-type decomposition algorithm for nonconvex separable optimization problems,” *Automatica*, vol. 25, no. 2, pp. 233–242, 1989.

[17] N. Watanabe, Y. Nishimura, and M. Matsubara, “Decomposition in large system optimization using the method of multipliers,” *Journal of Optimization Theory and Applications*, vol. 25, no. 2, pp. 181–193, 1978.

[18] J. Mulvey and A. Ruszczynski, “A diagonal quadratic approximation method for large scale linear programs,” *Operations Research Letters*, vol. 12, pp. 205–215, 1992.

[19] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximations,” *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[20] J. Eckstein and D. P. Bertsekas, “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators,” *Mathematical Programming,*, vol. 55, pp. 293–318, 1992.

[21] A. Ruszczynski, “On convergence of an Augmented Lagrangian decomposition method for sparse convex optimization,” *Mathematics of Operations Research*, vol. 20, pp. 634–656, 1995.

[22] A. Ribeiro, N. Sidiropoulos, and G. Giannakis, “Optimal distributed stochastic routing algorithms for wireless multihop networks,” *IEEE Trans. Wireless Commun.*, vol. 7, pp. 4261–4272, 2008.

[23] A. Ribeiro, T. Luo, N. Sidiropoulos, and G. Giannakis, “A general optimization framework for stochastic routing in wireless multi-hop networks,” in *Proc. Asilomar Conf. on Signals Systems Computers*, Pacific Grove CA, 2006.

[24] M. M. Zavlanos, A. Ribeiro, and G. J. Pappas, “Network integrity in mobile robotic networks,” *IEEE Transactions on Automatic Control*, 2012, (accepted).

[25] ——, “Distributed control of mobility and routing in networks of robots,” in *Proc. 12th IEEE International Workshop on Signal Processing Advances in Wireless Communications*, San Francisco, CA, 2011, pp. 236–240.

[26] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge Univeristy Press, 2004.

[27] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[28] D. P. Bertsekas, “Constrained optimization and Lagrange multiplier methods.” Athena Scientific, 1982.

[29] S. Vanka, M. Haenggi, and V. Gupta, “Convergence speed of the consensus algorithm with interference and sparse long-range connectivity,” *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, no. 99, 2011.

[30] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers,” *Foundations and Trends in Machine Learning*, 2011.

[31] J. Eckstein and D. P. Bertsekas, “An alternating direction method for linear programming.” LIDS, MIT, 1990.

[32] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Linklevel measurements from an 802.11b mesh network,” *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 121–132, 2004.