Distributed Subgradient Methods for Delay Tolerant Networks

Riccardo Masiero
Department of Information Engineering
University of Padova
Via G. Gradenigo 6/B, 35131 Padova (PD), Italy
E-mail: riccardo.masiero@dei.unipd.it

Giovanni Neglia
EPI Maestro
INRIA Sophia-Antipolis Méditerranée
2004 route des Lucioles, 06902 Sophia-Antipolis, France
E-mail: giovanni.neglia@inria.fr

***Abstract*—In this paper we apply distributed sub-gradient methods to optimize global performance in Delay Tolerant Networks (DTNs). These methods rely on simple local node operations and consensus algorithms to average neighbours' information. Existing results for convergence to optimal solutions can only be applied to DTNs in the case of synchronous operation of the nodes and memory-less random meeting processes. In this paper we address both these issues. First, we prove convergence to the optimal solution for a more general class of mobility models. Second, we show that, under asynchronous operations, a direct application of the original sub-gradient method would lead to suboptimal solutions and we propose some adjustments to solve this problem. Further, at the end of the paper, we illustrate a possible DTN application to demonstrate the validity of this optimization approach.**

***Index Terms*—delay tolerant networks, distributed optimization, consensus, sub-gradient method**

I. Introduction

In this paper we consider that nodes in a Delay Tolerant Network (DTN) may collaborate to optimize global network performance, for example tuning the values of some local parameters. The work in [1], later extended in [2], presents a distributed solution to this problem when the global optimization target f can be expressed as sum of M convex functions f_i and each node i only knows the corresponding function f_i, referred to as the *local objective function*.

In this framework, nodes optimize their own local objective functions through a *sub-gradient method*, and at the same time they try to reach agreement on their local estimates of the optimal solution by occasionally exchanging and averaging them, like in a consensus problem [3], [4]. Within this approach, referred to as the *distributed sub-gradient method*, the local estimate of each node is proven to converge to the optimal solution under certain assumptions. In a DTN scenario these assumptions correspond to impose deterministic bounds on the inter-meeting times among nodes [1] or a memory-less meeting process [2]. Neither of these conditions is in general satisfied in a real network. Further, all nodes should update their estimates at the same time, but synchronicity is difficult to achieve in a disconnected scenario.

After reviewing the distributed sub-gradient method in Sec. II and motivating our work in Sec. III, as original contribution in this paper we relax the above assumptions. In particular, in Sec. IV we extend the results in [1] and [2] proving that the distributed sub-gradient method also converges under a more general Markovian mobility model with memory in the meeting process. In Sec. V, we study how the presented framework needs to be extended to cope with asynchronous node operations. Further, in Sec. VI we illustrate a possible DTN application for the distributed sub-gradient method. Sec. VII concludes the paper.

An extended version of this paper is available as an INRIA research report [5].

II. Distributed Sub-gradient Method's Overview

In this section we review the main results in [1], [2] on convergence and optimality of the distributed sub-gradient method when a random network scenario is considered.

Let us consider a set of M nodes (agents), that want to cooperatively solve the following optimization problem:

Problem 1 (Global Optimization Problem)**.** *Given M convex functions $f_i(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}$, determine:*

$$\mathbf{x}^* \in \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x}) \ .$$

Clearly, for the above problem we assume that a feasible solution exists. The difficulty of the task arises from the fact that agent i, for $i = 1, 2, \cdots M$, only knows the corresponding function $f_i(\mathbf{x})$, namely its *local objective function*. A time slotted system is also assumed, where, at the end of a slot, each node i communicates its local estimate to a subset of all the other nodes, and then updates the estimate according to[1]:

$$\mathbf{x}^i(k+1) = \sum_{j=1}^{M} a_{ij}(k)\mathbf{x}^j(k) - \gamma(k)\mathbf{d}^i(k) \ , \tag{1}$$

where the vector $\mathbf{d}^i(k) \in \mathbb{R}^M$ is a sub-gradient[2] of agent i's objective function $f_i(\mathbf{x})$ computed at $\mathbf{x} = \mathbf{x}^i(k)$, the scalar $\gamma(k) > 0$ is the step-size of the sub-gradient algorithm at iteration k, and $a_{ij}(k)$ are non-negative weights, such that $a_{ij}(k) > 0$ if and only if node i has received node j's estimate at the step k and $\sum_{j=1}^{M} a_{ij}(k) = 1$. We denote by $\mathbf{A}(k)$ the matrix whose elements are the weights, i.e. $[\mathbf{A}(k)]_{ij} = a_{ij}(k)$.

[1]In this paper all the real valued vectors are assumed to be column vectors.

[2]$\mathbf{d}^i \in \mathbf{R}^N$ is a sub-gradient of the function f_i at $\mathbf{x}^i \in \mathrm{dom}(f_i)$ iif $f_i(\mathbf{x}^i) + (\mathbf{d}^i)^T(\mathbf{x} - \mathbf{x}^i) \le f_i(\mathbf{x})$ for all $\mathbf{x} \in \mathrm{dom}(f_i)$.

We observe that the first addend in the right-hand side of (1) corresponds to averaging according to a consensus algorithm [3].

In [1] it is procen that the iterations (1) generate sequences converging to a minimum of f under the following set of conditions: 1) $\gamma(k)$ is such that $\sum_{k=1}^{\infty} \gamma(k) = \infty$ and $\sum_{k=1}^{\infty} \gamma(k)^2 < \infty$; 2) f_i is bounded for all i; 3) $\mathbf{A}(k)$ is symmetric (then doubly stochastic) for all k; 4) $\exists\ \eta > 0$, such that $a_{ii}(k) > \eta$ and, if $a_{ij}(k) > 0$, then $a_{ij}(k) > \eta$; 5) the information of each agent i reaches every other agent j (directly or indirectly) infinitely often; $6'$) there is a deterministic bound for the intercommunication interval between two nodes. In [2], condition $6'$ is replaced by: $6''$) matrices $\mathbf{A}(k)$ are i.i.d. random matrices. Both papers address also the case when the gradient step-size is kept constant ($\gamma(k) = \gamma$). In this case, the sequence of estimates $\mathbf{x}^i$ does not converge in general to a point of minimum of f, but it may keep oscillating around one of such points, in a neighbourhood that shrinks as γ gets smaller.

To illustrate the considered framework, we present now a simple toy example that we are going to use different times across this paper. Consider three nodes, labeled as 1, 2 and 3. Their local objective functions are $f_1(x) = f_2(x) = x(x-1)/2$ and $f_3(x) = 2x^2$, where $x \in \mathbb{R}$. Then the global function is $f(x) = \sum_i f_i(x) = 3x^2 - x$, it has minimum value equal to $-1/12$ and a unique point of minimum in $x = 1/6$. The weight matrices $\mathbf{A}(k)$ are i.i.d. random matrices. At each step $\mathbf{A}(k)$ is equal to one of the following three matrices

$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}, \qquad (2)$$

with probability $2/3$, $1/6$ and $1/6$, respectively. Figure 1 shows the evolution of the estimates at the 3 nodes, when the algorithm is applied with $\gamma(k) = 1/k$. We can see that state's estimates tends to couple and then converge to the optimal value. We obtained similar results with $\gamma(k) = \gamma \ll 1$.

The proofs of the convergence in [1] and [2] share mainly the same outline. In detail, under $6'$, [1] proves that the *backward matrix product*, i.e., $\mathbf{A}_{(s)}^{(k)} = \mathbf{A}(k)\mathbf{A}(k-1)\cdots\mathbf{A}(s)$, surely converges to the matrix $\mathbf{J} = 1/M\mathbf{1}\mathbf{1}^T$, and that there are two positive constants C and β such that $\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\|_\infty \leq C\beta^{k-s}$ for all $k \geq s$. Under $6''$, [2] proves almost surely convergence of $\mathbf{A}_{(s)}^{(k)}$ to $\mathbf{J}$, and an exponential convergence rate in expectation, i.e. $E[\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\|_\infty] \leq C\beta^{k-s}$. Then, similar bounds are established for the distance between $\mathbf{x}^*$ and the average of all the estimates $\mathbf{y}(k)$ (i.e. $\mathbf{y}(k) \stackrel{def}{=} 1/M \sum_i \mathbf{x}^i(k)$), and between $\mathbf{x}^i(k)$ and $\mathbf{y}(k)$. Convergence results (or asymptotic bounds when $\gamma(k)$ is constant) follow.

III. Application to Optimization in DTNs

DTNs, see e.g., [6] and [7], are sparse and/or highly mobile wireless ad hoc networks where no continuous connectivity guarantee can be assumed. This intrinsically leads to the impossibility of collecting, at low cost and at a single data processing point, the information needed to solve network optimization problems in a centralized fashion. For this reason, in the present paper we advocate the use of the distributed subgradient method presented in Sec. II.

From a general perspective, we can proceed as follows. Nodes exchange their local estimates every time they meet. thus originating the weight matrices $\mathbf{A}(k)$. We define the contact matrix $\mathbf{C}(k)$, where $c_{ij}(k) = 1$ if node i has met node j since the last state's update performed by i , and $c_{ij}(k) = 0$ otherwise. Call $\mathcal{C}$ the (finite) set of all possible $M \times M$ matrices describing the contacts among M nodes. At the update time instants, each node i can calculate its own weights $a_{ij}(k)$, for $j = 1, \ldots M$, from the contacts it had (i.e.! from c_{ij}, for $j = 1, \ldots M$) in one of these two ways (which build matrix $\mathbf{A}(k)$ as doubly stochastic):

Rule 1 (Updates independent from meetings) For $j \neq i$, set[3] $a_{ij}(k) = 1/M$ if $c_{ij}(k) = 1$, otherwise set $a_{ij}(k) = 0$. Set $a_{ii}(k) = 1 - \sum_{j \neq i} a_{ij}(k)$. All the nodes update their estimate at a given sequence of time instants $\{t_k\}_{k \geq 1}$.

Rule 2 (Updates driven by meetings) When node i and node j meet, they update their estimate. In this case, set $a_{ij}(k) = a$ with $0 < a < 1$, $a_{ii}(k) = 1 - a$ and $a_{ih}(k) = 0$ for $h \neq j, i$.

Next, we discuss two key issues that can negatively impact the convergence of the distributed optimization process in a DTN scenario. The first one is related to the validity of assumptions $6'$ and $6''$. In fact, condition $6'$ is essentially equivalent to assume that there is a deterministic bound for the inter-meeting times of two nodes (that meet infinitely often), and this is not the case for all the random mobility models usually considered, see e.g., [8]. Condition $6''$ relaxes $6'$, but requires the independence of the meetings among nodes, that under realistic mobility are instead correlated[4]. The second issue is related to the synchronicity of the updates: in [1], [2] all the nodes update their estimates at the same time instants as in Rule 1, but this is not always feasible in a disconnected and distributed scenario like a DTN. Coming back to the toy example presented in Sec. II, we now show that we cannot simply ignore the issue of synchronicity. We observe that we can think our three matrices in (2) as generated according to Rule 2, when the meeting process has the following characteristics: at each time slot, node 1 and node 2 meet with probability $2/3$, node 1 and 3 meet with probability $1/6$ and node 2 and 3 meet with probability $1/6$. Fig. 2 shows the evolution of the estimates when the step-size is constant and equal to $25 \cdot 10^{-4}$, both for the synchronous case, where all the nodes update their estimates when a meeting occurs (even the node that is not involved in the meeting), and for the asynchronous case, where only the nodes involved in the meeting perform the update. The curves represents the average estimates over 100 different simulations with different meeting sequences. We note that in the synchronous case (top graph) all nodes agree on the optimal value to set x, whereas, in the asynchronous case (bottom graph) the estimates still converge, but not to the minimum of the global function f.

We consider more general mobility models in Sec. IV and address asynchronous operation in Sec. V. These extensions to

[3]Each node has to know M, i.e., the total number of nodes in the system.

[4]For example if in the recent past i has met j and j has met h, then i has a higher probability to meet h in the near future.

the basic framework proposed in [1], [2], while motivated in this paper by the DTN scenario, are of wide interest for other possible applications such as mobile wireless ad-hoc and sensor networks.

IV. Extension to more General Mobility Models

In our DTN scenario, we consider that the weights are determined from the contact matrix through a bijective function. Then condition 5 and $6''$ of [2], can be expressed as follows: the contact matrices $\mathbf{C}(k)$ are i.i.d. and $\mathrm{E}[\mathbf{C}(k)]$ is an irreducible aperiodic matrix. In this section, we extend the convergence results to the following, more general, mobility model. All the proofs are in [5].

Assumption 1 (Mobility model)**.** *It exists an irreducible, aperiodic and stationary Markov chain Φ with a finite or countable set of states S and a function $g : S \to \mathcal{C}$, such that $\mathbf{C}(k) = g(\Phi_k)$, for each $\Phi_k \in S$. Moreover, $E[\mathbf{C}(k)]$ is an irreducible aperiodic matrix.*

Since there is a bijective correspondence among weight and contact matrices, we observe that under assumption 1, it also exists a function $\hat{g} : S \to \mathcal{A}$, such that $\mathbf{A}(k) = \hat{g}(\Phi_k)$. The case when the contact matrices (and then the weight matrices) are i.i.d. is a particular case of our mobility model.

Our proof follows the same outline of [1], [2]: the main issue is to prove the exponential rate of convergence of the backward product $\mathbf{A}_{(1)}^{(k)}$ to $\mathbf{J} = 1/M\mathbf{1}\mathbf{1}^T$. To this end we need the following results:

Proposition 1. *Given an irreducible aperiodic and stationary Markov chain Φ with finite or countable states, the shift operator θ is measure-preserving and ergodic together with all its powers θ^k, where $k \in \mathbb{N}$.*

Lemma 1 (Windowing a Markov chain)**.** *Let $\Phi = \{\Phi_n, n \in \mathbb{N}\}$ be an irreducible, aperiodic and stationary Markov chain. Consider the stochastic process $\Psi = \{\Psi_n, n \in \mathbb{N}\}$, where $\Psi_n = (\Phi_n, \Phi_{n+1}, \cdots, \Phi_{n+h-1})$ with h a positive integer. Ψ is also an irreducible aperiodic stationary Markov chain.*

Convergence of $\mathbf{A}_{(1)}^{(k)}$ to $\mathbf{J}$ is then a corollary of results in [9]:

Proposition 2 (Convergence of the backward product)**.** *Under assumption 1 $\lim_{k\to+\infty} \mathbf{A}_{(1)}^{(k)} = \mathbf{J}$ almost surely (a.s.) .*

Now we are ready to prove our main result

Proposition 3. *Under assumption 1 on the mobility models, if the matrices are doubly stochastic, then for almost all the sequences there exist $C > 0$ and $0 < \beta < 1$ (with C in general depending of the sequence) such that for $k > s$*

$$\left\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\right\|_{\max} \leq C\beta^{k-s} .$$

In [2] a different result it is proven, i.e., that there exist $\hat{C}$ and $\hat{\beta}$ such that $\mathrm{E}\left[\left\|\mathbf{A}_{(s)}^{(k)} - \mathbf{J}\right\|_{\max}\right] \leq \hat{C}\hat{\beta}^{k-s}$. Then a series of inequalities for the expected values of $\|\mathbf{y}(k) - \mathbf{x}^i(k)\|_2$ are obtained for all i. Using Fatou's Lemma, along with the non-negativeness of distances, it is possible to derive inequalities that hold with probability 1. Using Proposition 3, instead, it is possible to obtain the same inequalities directly without the need to consider the expectation.

V. Asynchronous Updates

In this section, we study the case when nodes asynchronously update their status. First, we consider the case of decreasing step-sizes with the sequence $\{\gamma_i(k)\}_{k\geq 1}$ satisfying $\sum_{k=1}^{\infty} \gamma_i(k) = \infty$ and $\sum_{k=1}^{\infty} \gamma_i(k)^2 < \infty$.

We can go over the rationale in [2] and prove similar results for the new system description, see [5]. In particular, we have

Proposition 4 (Convergence of Agent Estimates)**.** *Under assumption 1, the estimate of each node converges almost surely to the vector $\mathbf{y}(k)$, i.e., $\lim_{k\to+\infty} \|\mathbf{y}(k) - \mathbf{x}^i(k)\|_2 = 0$ a.s. , for all i .*

The following step is to use bounds for the distance between $\mathbf{y}(k)$ and $\mathbf{x}^*$ (a point of minimum of f) to show that $\lim_{k\to+\infty} \mathbf{y}(k) = \mathbf{x}^*$. In detail in [2] is shown that

$$0 \leq \sum_{s=1}^{\infty} \gamma(s) \left[f(\mathbf{y}(s)) - f(\mathbf{x}^*)\right] < \infty \quad \text{a.s. ,} \tag{3}$$

from (3), we can conclude that $\liminf_{k\to\infty} f(\mathbf{y}(k)) = f(\mathbf{x}^*)$ a.s., since $\sum_{s=1}^{\infty} \gamma(s) = \infty$, and thus $\lim_{k\to\infty} \mathbf{x}^i(k) = \mathbf{x}^*$ a.s. .

In [5], a similar derivation is carried on, leading to the following generalization of (3):

$$\sum_{s=1}^{\infty} \sum_{i=1}^{M} \gamma_i(s) \left[f_i(\mathbf{y}(s)) - f_i(\mathbf{x}^*)\right] < \infty \quad \text{a.s. .} \tag{4}$$

Unfortunately, due to the asynchronous updates, the values of $\gamma_i(s)$ are now different for each i and this does not allow us to formulate the inequality above in terms of the global function f as in (3). However, (4) suggests us the following conjecture, that we support later with some examples:

Conjecture 1. *When updates are asynchronous, convergence results to optimality for sub-gradient methods hold if $E[\gamma_i(k)] = E[\gamma_j(k)]$ for each i and j.*

We can guarantee this condition in different cases. Consider Rule 2 in Sec. III and $\gamma_i(k) = 1/n_i(k)$, where $n_i(k)$ is the total number of updates node i has performed until the time instant t_k. If the meeting process follows a Poisson process with total rate λ and at each instant node i meets another node with probability p_i, we expect that by time k, node i has $p_i k$ meetings (and an equal number of updates). Then the expected value of its step-size is $\mathrm{E}[\gamma_i(k)] = \mathrm{E}[1/n_i(k)] = p_i/(p_i k) = 1/k$, for all i and we forecast the asynchronous sub-gradient mechanism to converge to the optimal solution. Fig. 3 (top graph) shows that this is true for our toy example. The simulations for the application considered in Sec. VI confirm such convergence.

Let us now revisit the example in Sec. III showing that the estimates were not converging to a point of minimum (Fig. 2, bottom graph). Here step-sizes were constant, i.e. $\gamma_i(k) = \gamma$. Now, reasoning as above we can conclude that $\mathrm{E}[\gamma_i(k)] = p_i\gamma$. Hence the expected values are not equal as far as node meeting rates (and then update rates) are not equal: this was the case of our example, where $p_1 = 5/6$, $p_2 = 5/6$ and $p_3 = 1/3$. Intuitively, we expect convergence to be biased

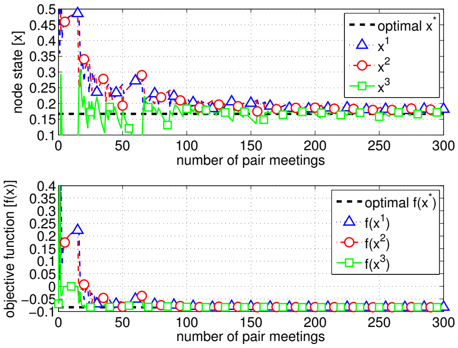

Fig. 1. Toy example. Top graph: state's estimates. Bottom graph: objective function value computed in the state's estimates.

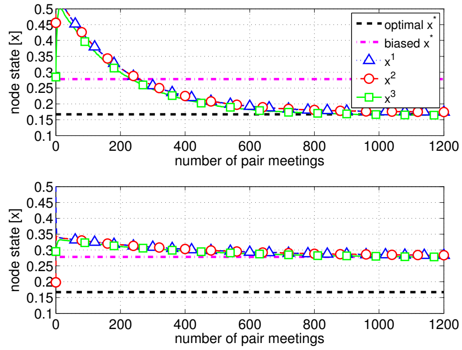

Fig. 2. Toy example, step-size $\gamma = 25 \cdot 10^{-4}$. Top graph: synchronous updates. Bottom graph: asynchronous updates.

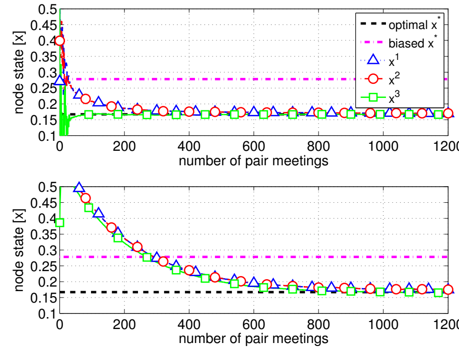

Fig. 3. Toy example, asynchronous updates. Top graph: step-size $\gamma_i(k) = 1/n_i(k)$. Bottom graph: weighted fixed step-size $\gamma_i(k) = p_i^{-1}\gamma$.

towards values closer to the optimum of the local functions of those nodes that perform the updates more often. Equation (4) suggests us that the naive asynchronous implementation of the sub-gradient method is actually minimizing the function $\sum_i p_i f_i = (3/2)x^2 - (5/6)x$ rather then $f = \sum_i f_i = 3x^2 - x$. This is the case, being that the estimates are converging to $5/18$ (dot-dashed line in Figs. 2 and 3). If now we want to correct the bias, it is sufficient to consider that each node selects its step-size inversely proportional to its meeting rate. Fig. 3 (bottom graph) shows that this correction leads the estimates to converge to the correct result.

VI. Application in DTNs: a Case Study

In this section we apply the distributed sub-gradient method to a DTN scenario inspired by the work in [10]. In detail, we consider a mobile network with M nodes, indexed from 1 to M. All the nodes are interested in the same dynamic information content that has for them a non-increasing value in time. They can get information updates directly from the Service Provider (SP) through a cellular network, but also share them through opportunistic transmissions whenever they meet. The SP is supposed to inject fresh information in the network according to a Poisson process of parameter μ updates/s. The goal of the SP is to optimize

$$f(\mathbf{x}) = \sum_{i=1}^{M} f_i(\mathbf{x}) = \sum_{i=1}^{M} E_{\mathbf{x}}[u_i] \ , \tag{5}$$

where u_i is the utility of the content for node i and $\mathbf{x} \in \mathbb{R}^M$ is the rate vector that indicates how the SP distributes its bandwidth among all the nodes. Then $\sum_{i=1}^{M} x_i \leq \mu$ and $x_i \geq 0$ for all i. In [10], eq. (5) is proved to be concave and therefore the optimal $\mathbf{x}$ can be obtained by the SP using standard optimization techniques, see e.g., [11]. In general, a closed formula for $f(\mathbf{x})$ is not known; thus, the gradient needs to be estimated as explained in [10]. Being that our interest is on the distributed sub-gradient method, here we consider the simple case where updates can travel at most two hops, thus avoiding to address the gradient estimation's issue. In fact, in this case, assuming also that a) for all nodes i, any information is worthless after τ seconds from its injection into the network by the SP (i.e. the information expires after τ) and b) the meeting process among node pairs is Poisson distributed, we can compute the local utility function for each node as (see [5])

$$f_i(\mathbf{x}) = 1 - \left[\prod_{j \in \mathcal{N}_i} \frac{x_j e^{-\lambda_{ij}\tau} - \lambda_{ij} e^{-x_j \tau}}{x_j - \lambda_{ij}} \right] e^{-x_i \tau} \ , \tag{6}$$

where λ_{ij} is the meeting rate between i and j and $\mathcal{N}_i \stackrel{\text{def}}{=} \{j : \lambda_{ij} > 0\}$. The global utility function in (5) is then simply obtained summing (6) over $i = 1, 2, \ldots, M$.

To optimize $f(\mathbf{x})$ in a distributed fashion we can use the framework presented in Sec. II. The local gradient function needed in (1) can be computed directly from (6), where nodes only need to estimate simple statistics on their own meeting rates. Clearly, (5) can be optimized also in a centralized fashion by the SP as in [10]. However, our distributed approach may reduce the amount of information to exchange between the SP and the nodes. Moreover, it does not force the nodes to disclose information about their meetings or their utility function u_i.

In detail, when two nodes i and j meet they: i) update $\mathbf{x}^i$ and $\mathbf{x}^j$ as in (1) and, ii) project the result so obtained onto the feasible set $\sum_{l=1}^{M} x_l \leq \mu$ and $x_l \geq 0$ for all $l \in \{1, \ldots, M\}$. Eventually all the $\mathbf{x}^i$ converge to the optimum $\mathbf{x}^*$ of (5). Henceforth, the SP can retrieve the optimal transmission rates collecting $\mathbf{x}^i$ from every node and obtaining the rate allocation vector as $\mathbf{x} = (\sum_{i=1}^{M} \mathbf{x}^i)/M$.

To test the performance achievable by the distributed sub-gradient method under traces with memory, we generated a meeting process that is both stationary and ergodic as explained in [5]. Nodes have diverse contact rates; in particular node 1 has the highest contact rate, whilst node 10 the lowest. Given that nodes have different contact rates and asynchronous updates are performed, we know that a direct application of the distributed sub-gradient algorithm may lead to sub-optimal results, see the toy example in Sec. III.

Figures 4–6 show simulation results for a given mobility model when $\tau = 20$s and μ varies between 0.1 and 25s^{-1}. Fig. 4 shows the optimal allocation rate for each user. When the bandwidth μ available to the SP is very low, the best solution is that the SP uniquely sends updates to the node that has the

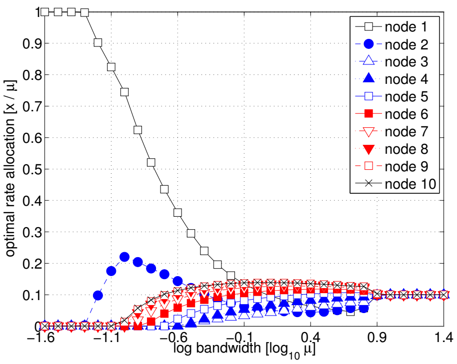

Fig. 4. Optimal bandwidth allocation for a network of 10 nodes.

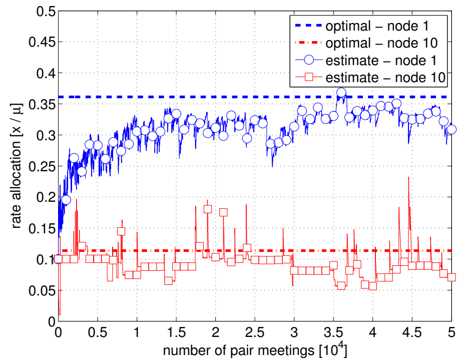

Fig. 5. Example of convergence of estimate for two nodes when the sub-gradient method is used.

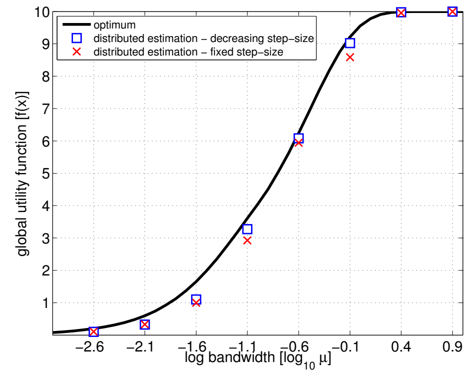

Fig. 6. Performance comparison: centralized solver vs distributed method.

higher contact rate, i.e., node 1; for large values of μ, instead, the SP can evenly send updates to all the nodes in the network. Interestingly, as already observed in [10], for some values of μ (in our case μ around $10^{0.7}$ update/sec) the optimal choice for the SP is to allocate more bandwidth (i.e., a larger fraction of μ) to the node with the lowest contact rate, namely, node 10. For these values of μ, in fact, those nodes with a high contact rate such as node 1 achieve high values for their utility functions just by collecting information from the large number of nodes they meet. In Fig. 5 we show the mean trajectory towards the optimal for two elements in $\mathbf{x} = (\sum_{i=1}^{M} \mathbf{x}^i)/M$, where the vectors $\mathbf{x}^i$ have been obtained along a sequence of $5 \cdot 10^4$ meetings considering $\mu = 10^{-1.1}$ update/sec. We note that the estimates provided through the distributed sub-gradient method converge[5] to the theoretical optimal allocations in Fig. 4.

Finally, in Fig. 6 we draw with a solid-line the maximum of $f(\mathbf{x})$ corresponding to the optimal rate allocations in Fig. 4, which was obtained using a centralized solver [11]. For eight different values of the available bandwidth μ, we plot with squares the utility function values corresponding to the rate allocations obtained by the distributed sub-gradient method. With crosses we show the performance of the method with a fixed step size, which neglects the asynchronous update issue. As expected, the latter algorithm achieves worse results. Most importantly, the solutions achieved with our approach are close to the actual optimum for all values of μ. This confirms the validity of this distributed framework.

VII. Conclusions

In this paper we considered the recent optimization framework based on distributed sub-gradient methods proposed in [1], and later extended in [2]. We pointed out that existing convergence results for this framework can be applied to DTNs only in the case of synchronous node operation and in the presence of simple random meeting processes without memory. Therefore, we addressed both these issues: first, we proved convergence to optimality of the sub-gradient optimization technique under a more general class of mobility processes and second, we proposed some modifications to the original sub-gradient algorithm so as to avoid bias problems (i.e., convergence towards sub-optimal solutions) when nodes operate asynchronously. Finally, as a case study, we applied the presented framework to the optimization of dynamic content dissemination in a DTN. All the provided results confirmed that the distributed sub-gradient method is an effective and promising tool for optimization in distributed contexts.

[5]Concerns about the convergence rate of such estimates are out of the scope of the present paper and will be addressed in the future research.

VIII. Acknowledgments

Authors thank Dr. Philippe Nain and Dr. Michele Rossi for their precious comments and suggestions.

References

[1] A. Nedic and A. Ozdaglar, "Distributed Subgradient Methods for Multi-Agent Optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] I. Lobel and A. Ozdaglar, "Distributed Subgradient Methods for Convex Optimization over Random Networks," *Automatic Control, IEEE Transactions on*.

[3] R. Wei, R. Beard, and E. Atkins, "A survey of Consensus Problems in Multi-agent Coordination," in *Proc. of the American Control Conference*, vol. 3, Jun. 2005, pp. 1859–1864.

[4] P. Denantes, F. Bénézit, P. Thiran, and M. Vetterli, "Which Distributed Averaging Algorithm Should I Choose for my Sensor Network?" in *Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, 2008.

[5] R. Masiero and G. Neglia, "Distributed Sub-gradient Method for Delay Tolerant Networks," INRIA, Tech. Rep. RR-7345, 2010.

[6] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant Networking: an Approach to Interplanetary Internet," *Communications Magazine, IEEE*, vol. 41, no. 6, pp. 128–136, Jun. 2003.

[7] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic Networking: Data Forwarding in Disconnected Mobile Ad Hoc Networks," *Communications Magazine, IEEE*, vol. 44, no. 11, pp. 134–141, Nov. 2006.

[8] C. Bettstetter, "Mobility Modeling in Wireless Networks: Categorization, Smooth Movement, and Border Effects," *Mobile Computing and Communications Review, ACM SIGMOBILE*, vol. 5, no. 3, pp. 55–67, 2001.

[9] A. Tahbaz-Salehi and A. Jadbabaie, "Consensus Over Ergodic Stationary Graph Processes," *Automatic Control, IEEE Transactions on*, vol. 55, no. 1, pp. 225–230, Jan. 2010.

[10] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and Scalable Distribution of Content Updates over a Mobile Social Network," in *In Proc. of the IEEE Int. Conf. on Computer Communications (INFOCOM)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 1422–1430.

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