Distributed Distortion Optimization for Correlated Sources with Network Coding

Tao Cui, Lijun Chen and Tracey Ho
Engineering & Applied Science Division
California Institute of Technology, USA
Email: {taocui@, chen@cds., tho@}caltech.edu

Abstract

We consider lossy data compression in capacity-constrained networks with correlated sources. We derive, using dual decomposition, a distributed algorithm that maximizes an aggregate utility measure defined in terms of the distortion levels of the sources. No coordination among sources is required; each source adjusts its distortion level according to distortion prices fed back by the sinks. The algorithm is developed for the case of squared error distortion and high resolution coding where the rate-distortion region is known, and can be easily extended to consider achievable regions that can be expressed in a related form. Our distributed optimization framework applies to unicast and multicast with and without network coding. Numerical example shows relatively fast convergence, allowing the algorithm to be used in time-varying networks.

Index Terms

Source coding, Rate allocation, Multicast, Network coding, Wyner-Ziv.

This work has been supported in part by DARPA grant N66001-06-C-2020, Caltech's Lee Center for Advanced Networking, the Okawa Foundation Research Grant and a gift from Microsoft Research. This paper has been presented in part at the International Symposium on Information Theory (ISIT), Nice, France, June 2007.

I. Introduction

In this paper, we consider a network that has multiple correlated sources with associated distortion measures. In such a situation, we can integrate source coding and rate control by adapting the distortion of the sources to network congestion. Specifically, we consider adaptive lossy source coding for multicast with network coding [1], where each multicast session contains a set of continuous and possibly correlated sources. We are interested in distributed algorithms, which are more scalable than centralized algorithms and can adapt to unknown and dynamically changing network topology.

For correlated sources, independent data compression is not an optimal strategy. Higher data compression efficiency can be obtained by using distributed source coding techniques. Existing approaches for network optimization with distributed source coding of correlated sources, e.g., [2], [3] for lossless coding and [4] for lossy coding, require coordination among the sources and do not admit fully distributed implementation.

Motivated by the optimization decomposition and utility maximization framework developed for TCP congestion control (see, e.g., [5], [6]), we consider the problem of maximizing an aggregate utility measure defined in terms of the distortion levels of the sources, e.g., minimum mean-square error (MMSE) distortion, and solve the problem to obtain a dual-based joint lossy source coding and network coding algorithm. The receiver-driven source coding algorithm adjusts distortion levels according to the distortion prices fed back from the sinks, and hence does not require coordination among the sources. With random network coding [7], our algorithm can be implemented in a fully distributed manner.

Our algorithm is developed for the case of squared error distortion and high resolution coding where the rate distortion region is known [8], and is easily extended to consider achievable regions that can be expressed in a related form. Our distributed optimization framework applies to unicast and multicast with and without network coding. Numerical examples show relatively fast convergence, allowing the algorithm to be used in time-varying networks.

II. Related Work

Joint optimization of source coding and routing/network coding for networks with correlated sources has been considered in a few recent works. In [2], joint optimization of lossless source coding and routing is proposed, where rate is allocated across sources to minimize a flow cost function under the constraint that the rates of the sources must lie in the Slepian-Wolf region. This approach is extended to lossy source coding in [4], where high-resolution source coding is assumed. A minimum cost subgraph construction algorithm for lossless source coding and network coding is proposed in [3], for the case of two sources.

Even though Slepian-Wolf coding is distributed, the optimization problems in [2]–[4] still require the coordination of the sources to guarantee that the source rates lie in the Slepian-Wolf region. Therefore, the algorithms in these works are not fully distributed.

In [9], [11], rate control for multicast with network coding has been studied for elastic traffic, with an aggregate utility maximization objective. The utility of each source is a function of its sending rate. In our work, the utility objective is defined in terms of distortion of each source. The rate-distortion region imposes a new type of constraint on the optimization.

III. Preliminaries

A. Network and Coding Model

Consider a network, denoted by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, with a set $\mathcal{N}$ of nodes and a set $\mathcal{L}$ of directed links. We denote a link either by a single index l or by the directed pair (i, j) of nodes it connects. Each link l has a fixed finite capacity c_l packets per second. A set of multicast sessions $\mathcal{M}$ is transmitted over the network. Each session $m \in \mathcal{M}$ is associated with a set $\mathcal{S}_m \subset \mathcal{N}$ of sources and a set of $\mathcal{T}_m \subset \mathcal{N}$ of sinks. For session m, each source $s \in \mathcal{S}_m$ multicasts x^{ms} bits to all the sinks in $\mathcal{T}_m$. By flow conservation, we have, for any i, m, $s \in \mathcal{S}_m$ and $t \in \mathcal{T}_m$,

$$\sum_{j:(i,j)\in\mathcal{L}} g_{i,j}^{mst} - \sum_{j:(j,i)\in\mathcal{L}} g_{j,i}^{mst} = \begin{cases} x^{ms} & \text{if } i = s \\ -x^{ms} & \text{if } i = t \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $g_{i,j}^{mst}$ is the rate of packets over link (i, j) from source $s \in \mathcal{S}_m$ to sink $t \in \mathcal{T}_m$ for session m. Note that the superscripts such as ^{ms} and ^{mst} are not polynomial power.

Network coding allows flows for different destinations of a multicast session to share network capacity by being coded together: for each multicast session m, with coding the actual physical flow on each link needs only be the maximum of the individual destinations' flows [1]. These constraints can be expressed as

$$g_{i,j}^{mst} \le f_{i,j}^{m}, \quad (i, j) \in \mathcal{L},\ m \in \mathcal{M},\ (s, t) \in (\mathcal{S}_m, \mathcal{T}_m), \tag{2}$$

where physical flow $f_{i,j}^m$ is the rate at which coded packets from session m are transmitted over link (i,j). For the case of multiple sessions sharing a network, achieving optimal throughput requires coding across sessions. However, designing such codes is a complex and largely open problem. Thus, we simply assume that coding is done only across packets of the same session, i.e., intra-session coding. In this case, the set of feasible flow vectors is specified by combining constraints (1)-(2) for each session m with the link capacity constraint:

$$\sum_{m \in \mathcal{M}} f_{i,j}^m \leq c_{i,j} \ , \quad \forall \, (i,j) \in \mathcal{L}. \tag{3}$$

In practice, the network codes can be designed using random linear network coding, see, e.g., [7], where for each node the data on outgoing links are random linear combination of the data on incoming links. If (1)-(2) holds, every sink can recover the transmitted packets with high probability. See [7] for a detailed description and discussion of overhead in random network coding and other practical implementation issues.

B. Lossy Source Coding

We consider multiterminal lossy source coding for continuous sources. Lossy source coding is data compression with a distortion measure. One technique for distributed lossy source coding is Wyner-Ziv coding [12], which is for a single source with uncoded side information at the sink. The general distributed rate-distortion region for coding correlated sources in the general setting is unknown even for Gaussian sources [13] (Recently, the rate-distortion region for quadratic Gaussian two-terminal source-coding is found in [14]).

It is still an open problem whether or not in general the optimal solution can be separated into a simple quantization for each source followed by Slepian-Wolf lossless coding, but such separation exists in the high-resolution limit: the optimal rate-distortion performance can be achieved by separately quantizing each source, e.g., by dithered lattice quantizers, and then applying Slepian-Wolf lossless encoding to the quantizers' outputs [8]. In the extreme of high resolution, it is shown in [8] that for squared-error distortion, the asymptotically achievable rate-distortion region for n correlated sources, $X_1, \ldots, X_n$, is given by

$$\sum_{X_i \in \mathcal{S}} R_i \geq h(\mathcal{S}|\mathcal{X} \backslash \mathcal{S}) - \log \left((2\pi e)^{|\mathcal{S}|} \prod_{X_i \in \mathcal{S}} D_i \right), \forall \mathcal{S} \subseteq \mathcal{X}, \tag{4}$$

where $\mathcal{X} = \{X_1, \ldots, X_n\}$, R_i and D_i are respectively the rate and MMSE distortion of X_i, and $h(\cdot)$ denotes differential entropy. A similar region is derived for more general difference distortion measures satisfying certain conditions. In general, the high-resolution region is an outer bound which becomes tighter as resolution increases. By using the results in [7], (4) can be readily extended to general networks by quantizing each source separately and then using random network coding.

For ease of exposition, we use the region defined in (4) in our subsequent development. Our results extend easily to the case where we have any achievable convex rate-distortion region, e.g., that in [13]. For instance,

$$\sum_{X_i \in \mathcal{S}} R_i \geq h(\mathcal{S}|\mathcal{X}\backslash\mathcal{S}) - \alpha_{\mathcal{S}} \log \prod_{X_i \in \mathcal{S}} D_i + \beta_{\mathcal{S}}, \forall \mathcal{S} \subseteq \mathcal{X}. \tag{5}$$

where $\alpha_{\mathcal{S}}$ and $\beta_{\mathcal{S}}$ are any constants. By appropriately choosing $\alpha_{\mathcal{S}}$ and $\beta_{\mathcal{S}}$, we can use (5) to approximate arbitrary achievable rate-distortion region. Note that $\alpha_{\mathcal{S}} > 0$, as the (minimum) transmission rate should be a decreasing function of the distortion.

IV. Distributed Algorithm

We assume for simplicity that each source transmits information over a single given multicast tree connecting it to its corresponding sink nodes. Such a multicast tree can be obtained by using protocols such as the distance vector multicast routing protocol [15]. These trees constitute an un-capacitated coding subgraph for each multicast session. Our distributed algorithm can be readily generalized to the case with multiple trees or without given trees (where the algorithm constructs coding subgraphs via back pressure) as in, e.g., [11].

Let T^{ms} denote the multicast tree for source s in session m. Each tree T^{ms} contains a set $\mathcal{L}_{ms} \subseteq \mathcal{L}$ of links, which defines an $|\mathcal{L}| \times 1$ vector ξ^{ms} whose l-th entry is given by

$$\xi_l^{ms} = \begin{cases} 1 & \text{if } \ l \in T^{ms} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Similar to (2) and (3), with intra-session network coding we have the following two constraints

$$\xi_l^{ms} x^{ms} \leq f_l^m, \qquad \forall l \in \mathcal{L},\, m \in \mathcal{M},\, s \in \mathcal{S}_m, \tag{7}$$

$$\sum_m f_l^m \leq c_l, \qquad \forall l \in \mathcal{L}. \tag{8}$$

With lossy source coding, the perceptual quality of an imperfect copy of a signal is determined by the human sensory system (visual, auditory, etc). It is reasonable to assume that the perceptual quality is determined by distortion. Thus, different from [5], we assume that each source s of session m attains a utility $U_{ms}^D(D^{ms})$ when it compresses data at a distortion level D^{ms}, rather than a rate-dependent utility. We assume that $U_{ms}^D(\cdot)$ is continuously differentiable, decreasing, and concave. This assumption is reasonable as the sources prefer smaller distortions. It also enforces some kind of fairness among the sources, as the marginal utility is decreasing when the distortion is further reduced. Examples of such utility functions are $\log(D_{\max} - D^{ms})$ and $-D^{ms}$, where $D_{\max}$ is the maximum tolerable distortion.

We formulate the source coding and network resource allocation problem as a utility maximization problem with the rate constraints (7)-(8) and the rate-distortion constraint (4) as follows.

$$\begin{aligned} \max_{D,x,f} \quad & \sum_{m\in\mathcal{M}, s\in\mathcal{S}_m} U_{ms}^D(D^{ms}) \\ \text{s.t.} \quad & \xi_l^{ms} x^{ms} \leq f_l^m, \ \forall l \in \mathcal{L},\ m \in \mathcal{M},\ s \in \mathcal{S}_m, \\ & \sum_{m\in\mathcal{M}} f_l^m \leq c_l,\ \forall l \in \mathcal{L}, \\ \sum_{s\in\mathcal{S}} x^{ms} \geq h(\mathcal{S}|\mathcal{S}_m\backslash\mathcal{S}) - \log & \left((2\pi e)^{|\mathcal{S}|} \prod_{s\in\mathcal{S}} D^{ms}\right),\ \forall \mathcal{S} \subseteq \mathcal{S}_m. \end{aligned} \tag{9}$$

Note that (28) is a convex problem and can be solved in polynomial time if all the utility and constraint information is given. However, a distributed algorithm is preferred in practice.

A. Algorithm

One way to derive a distributed solution is to consider its Lagrangian dual. However, in the rate-distortion constraint in (28), the source rates and distortions are not coupled at a single entity such as a node or link. We thus could not obtain a distributed algorithm by directly relaxing the rate-distortion constraint, which would still require source coordination. For the same reason, the algorithm in [4] is not fully distributed. In order to obtain a distributed solution, we consider the following equivalent problem

$$
\begin{aligned}
\max_{D,Z,x,y,f} \quad & \sum_{m,s} U^D_{ms}(D^{ms}) \\
\text{s.t.} \quad & \xi^{ms}_l x^{ms} \le f^m_l,\ \sum_m f^m_l \le c_l,\ \forall l,\ s, \\
& y^{mst} \le x^{ms}, Z^{mst} \le D^{ms},\ \forall l,\ m,\ s,\ t, \\
\sum_{s \in \mathcal{S}} y^{mst} \ge h(\mathcal{S}|\mathcal{S}_m \backslash \mathcal{S}) & - \log\left((2\pi e)^{|\mathcal{S}|} \prod_{s\in\mathcal{S}} Z^{mst} \right), \forall \mathcal{S} \subseteq \mathcal{S}_m,
\end{aligned} \tag{10}
$$

where by introducing auxiliary variables y^{mst} and Z^{mst} at each sink $t \in \mathcal{T}_m$, we remove the troublesome coupling among the sources in the rate-distortion constraint. We will see later that these auxiliary variables admit physical interpretation and enable a distributed receiver-driven source coding algorithm.

Consider the Lagrangian dual to problem (10)

$$
\min_{p\ge 0, q \ge 0, \lambda \ge 0} \phi(p, q, \lambda) \tag{11}
$$

with partial dual function

$$
\begin{aligned}
\phi(p,q,\lambda) = \max & \sum_{m,s} U^D_{ms}(D^{ms}) - \sum_{l,m,s} p^{ms}_l \left(\xi^{ms}_l x^{ms} - f^m_l\right) \\
& - \sum_{m,s,t} q^{ms}_t \left(y^{mst} - x^{ms}\right) - \sum_{m,s,t} \lambda^{ms}_t \left(Z^{mst} - D^{ms}\right) \\
\text{s.t.} \quad & \sum_{m\in\mathcal{M}} f^m_l \le c_l, \\
& \sum_{s \in \mathcal{S}} y^{mst} \ge h(\mathcal{S}|\mathcal{S}_m \backslash \mathcal{S}) - \log\left((2\pi e)^{|\mathcal{S}|} \prod_{s\in\mathcal{S}} Z^{mst} \right),
\end{aligned} \tag{12}
$$

where we relax only the first and the third constraints in (10) by introducing Lagrange multiplier p^{ms}_l at link l for source s in session m, and q^{ms}_t and λ^{ms}_t at sink t for source s in session m.

The dual function $\phi(p, q, \lambda)$ has a nice decomposition structure into four separate subproblems

$$\phi_1(q, \lambda) = \min_{y,Z} \sum_{m,s,t} q_t^{ms} y^{mst} + \lambda_t^{ms} Z^{mst}, \tag{13}$$

$$\text{s.t.} \sum_{s \in \mathcal{S}} y^{mst} \geq h(\mathcal{S}|\mathcal{S}_m \backslash \mathcal{S}) - \log \left((2\pi e)^{|\mathcal{S}|} \prod_{s \in \mathcal{S}} Z^{mst} \right),$$

$$\phi_2(\lambda) = \max_{D} \sum_{m,s} U_{ms}^D (D^{ms}) + \sum_{m,s} \left(\sum_t \lambda_t^{ms} \right) D^{ms}, \tag{14}$$

$$\phi_3(p, q) = \min_{x} \sum_{m,s} x^{ms} \left(\sum_l p_l^{ms} \xi_l^{ms} - \sum_t q_t^{ms} \right), \tag{15}$$

$$\phi_4(p) = \max_{f} \sum_{l,m,s} p_l^{ms} f_l^m, \text{ s.t.} \sum_{m \in \mathcal{M}} f_l^m \leq c_l. \tag{16}$$

The first subproblem is the minimum weighted rate and distortion problem for virtual lossy source coding at each sink. The second subproblem is distortion control. The third one is rate allocation. The fourth one is joint network coding and session scheduling. Thus, by dual decomposition, the problem decomposes into separate "local" optimization problems of application, transport, and network/link layers, respectively. The four problems interact through dual variables $p, \ q, \ \lambda$.

Lossy source coding: The virtual joint rate allocation and data compression problem (13) can further decompose into separate optimization problems at each sink $t \in |\mathcal{T}_m|$,

$$\min_{y,Z} \quad \sum_{s} q_t^{ms} y^{mst} + \lambda_t^{ms} Z^{mst} \tag{17}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} y^{mst} \geq h(\mathcal{S}|\mathcal{S}_m \backslash \mathcal{S}) - \log \left((2\pi e)^{|\mathcal{S}|} \prod_{s \in \mathcal{S}} Z^{mst} \right).$$

For fixed Z^{mst}, it can be readily verified that the polyhedron described by the constraint in (17) is a contra-polymatroid [16].[1] From Lemma 3.3 in [16], a greedy algorithm solves (17) optimally. Let π^* be any permutation of $\mathcal{S}_m$ such that $q_t^{m\pi^*(1)} \leq q_t^{m\pi^*(2)} \leq \cdots \leq q_t^{m\pi^*(|\mathcal{T}_m|)}$.

[1]Let $\mathcal{E} = \{1, \ldots, N\}$ and $f : 2^{\mathcal{E}} \rightarrow \mathbb{R}^+$ be a set function. The polyhedron $\mathcal{P}(f) = \{(x_1, \ldots, x_N) : \sum_{i \in \mathcal{S}} x_i \geq f(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{E}\}$ is a *contra-polymatroid* if f satisfies: 1) $f(\emptyset) = 0$ (normalized); 2) $f(S) \leq f(T)$ if $\mathcal{S} \subset \mathcal{T}$ (nondecreasing); and 3) $f(S) + f(T) \leq f(\mathcal{S} \cup \mathcal{T}) + f(\mathcal{S} \cap \mathcal{T})$ (supermodular).

Then, by Lemma 3.3 in [16], the solution of (17) with given Z is given by

$$\begin{aligned} y^{m\pi^*(1)t}(q) =& h\left(\pi^*(1)\right) - \log\left(2\pi e Z^{m\pi^*(1)t}\right), \\ y^{m\pi^*(2)t}(q) =& h\left(\pi^*(2)|\pi^*(1)\right) - \log\left(2\pi e Z^{m\pi^*(2)t}\right), \\ & \dots \\ y^{m\pi^*(|\mathcal{T}_m|)t}(q) =& h\left(\pi^*(|\mathcal{T}_m|)|\pi^*(|\mathcal{T}_m|-1), \dots, \pi^*(1)\right) - \log\left(2\pi e Z^{m\pi^*(|\mathcal{T}_m|)t}\right). \end{aligned} \tag{18}$$

Substituting (18) into (17) and minimizing (17) over Z^{mst}, we get

$$Z^{mst}(q, \lambda) = \frac{q_t^{ms}}{\lambda_t^{ms}}. \tag{19}$$

Substituting (19) into (18), we obtain the optimal $y^{mst}(q, \lambda)$.

Now, consider the distortion control problem (14). At source s, at each time slot τ, instead of solving (14) directly for D^{ms}, we update D^{ms} using a primal subgradient algorithm according to

$$D^{ms}(\tau+1) = \left[D^{ms}(\tau) + \epsilon_\tau \left(U_{ms}^{D\,\prime}\left(D^{ms}(\tau)\right) + \sum_t \lambda_t^{ms}\right)\right]^+, \tag{20}$$

where $U_{ms}^{D\,\prime}$ is the derivative of U_{ms}^D, ϵ_τ is a positive scalar stepsize, and $^+$ denotes the projection on the set of non-negative real numbers. We will see that λ_t^{ms} can be interpreted as the price resulting from the mismatch between the source distortion and virtual source distortion at the sink. The source distortion is adjusted according to the aggregate *distortion price* $\sum_t \lambda_t^{ms}$ due to virtual source coding, which is fed back from the sinks of session m.

Rate allocation: To recover the source rate, instead of solving (15) directly, we update the source rate using a primal subgradient algorithm. At time $\tau+1$, the source rate $x^{ms}(\tau+1)$ is updated according to

$$x^{ms}(\tau+1) = \left[x^{ms}(\tau) - \epsilon_\tau \left(\sum_l p_l^{ms} \xi_l^{ms} - \sum_t q_t^{ms}\right)\right]^+. \tag{21}$$

Each source then compresses data according to rate $x^{ms}(\tau+1)$ by using dithered lattice quantizers [8] and randomized linear network coding. This source coding and rate allocation mechanism has the desired price structure and is an end-to-end congestion control mechanism.

Session scheduling and network coding: For each link l, find the session $m_l^* = \arg\max_m \sum_s p_l^{ms}$. A random linear combination of packets from all the sources in session m_l^* is sent at the rate of c_l. This is equivalent to solving (16) by the following assignment

$$f_l^m(p) = \begin{cases} c_l & \text{if } m = m_l^* \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

Dual variable update: By using the first order Lagrangian method [17], at time $\tau + 1$, the dual variables are updated according to

$$p_l^{ms}(\tau+1) = \left[p_l^{ms}(\tau) + \gamma_\tau \left(\xi_l^{ms} x^{ms}(p(\tau), q(\tau)) - f_l^m(p(\tau))\right)\right]^+, \tag{23}$$

$$q_t^{ms}(\tau+1) = \left[q_t^{ms}(\tau) + \gamma_\tau \left(y^{mst}(q(\tau), \lambda(\tau)) - x^{ms}(p(\tau), q(\tau))\right)\right]^+, \tag{24}$$

$$\lambda_t^{ms}(\tau+1) = \left[\lambda_t^{ms}(\tau) + \gamma_\tau \left(Z^{mst}(q(\tau), \lambda(\tau)) - D^{ms}(\lambda(\tau))\right)\right]^+, \tag{25}$$

where γ_τ is a positive scalar stepsize. Note that (23)-(25) are distributed and can be implemented by individual links and sinks using only local information. The algorithm (18)-(25) is a distributed primal-dual subgradient algorithm for problem (10) and its dual. By using Lyapunov method and extending the techniques for the dual subgradient method as in, e.g., [11], we can prove that the algorithm (18)-(25) converges to within an arbitrarily small neighborhood of the optimal by using suitable stepsizes $\epsilon_\tau, \gamma_\tau$.

Note that p_l^{ms} results from the rate constraint and thus can be interpreted as a virtual congestion price at the link. q_t^{ms} can be interpreted as the price resulting from the mismatch between the physical source rate and virtual source rate at the sink, and λ_t^{ms} as the price resulting from the mismatch between the source distortion and virtual source distortion at the sink. Our adaptive source coding is a receiver-driven scheme. Since the sink receives information from all the sources, it can estimate the rate-distortion region of correlated sources, and solve a virtual joint rate allocation and data compression problem. By adapting to the prices q_t^{ms} and λ_t^{ms}, the source tries to match the virtual rate and distortion. The source rate also adapts to p_l^{ms} to avoid congestion.

Also note that in our algorithm the sink does not feedback any information about the source distributions to the sources. To feedback this information may change the rate-distortion region. When this happens, the whole system might be improved, and the current solution is then not optimal.

B. Performance Analysis

The above distributed source coding and rate allocation algorithm is a (partial) primal-dual subgradient algorithm. By extending the standard results on the convergence of the subgradient method [17], we can show that, for constant stepsize, the algorithm is guaranteed to converge to within a small neighborhood of the optimum. For diminishing stepsizes, the algorithm is guaranteed to converge to the optimum.

Before analyzing the performance of the above algorithm, let us first introduce some variables that will be used. Let $\bar{p}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} p(i)$, $\bar{\lambda}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} \lambda(i)$ and $\bar{q}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} q(i)$ be the average dual variables till time τ, $\bar{D}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} D(i)$ be the average source distortion and $\bar{x}(\tau) = \frac{1}{\tau}\sum_{i=1}^{\tau} x(i)$ be the average data rate injected into the network till time τ. Denote by $\bar{p}(\infty), \bar{q}(\infty), \bar{\lambda}(\infty), \bar{D}(\infty)$ and $\bar{x}(\infty)$ the corresponding averages at time $\tau = \infty$. Denote by $P(D) = \sum_{m\in\mathcal{M}, s\in\mathcal{S}_m} U_{ms}^D(D^{ms})$ the primal function and by $L(D, x, Z, y, f, p, q, \lambda)$ the Lagrangian function of problem (10). Define

$$\tilde{L}(D, x, p, q, \lambda) = L(D, x, Z(q, \lambda), y(q), f(p), p, q, \lambda).$$

Denote by $\nabla_D \tilde{L}, \nabla_x \tilde{L}, \nabla_p \tilde{L}, \nabla_q \tilde{L}$ and $\nabla_\lambda \tilde{L}$ the subgradients of $\tilde{L}$ with respect to D, x, p, q and λ, respectively. In practice, it is reasonable to assume that the norm of the subgradient is uniformly bounded, i.e., there exist constants $G_1 > 0$ and $G_2 > 0$ such that for the primal subgradient $\|\nabla_D \tilde{L}\|^2 + \|\nabla_x \tilde{L}\|^2 \leq G_1^2$ and for the dual subgradient $\|\nabla_p \tilde{L}\|^2 + \|\nabla_q \tilde{L}\|^2 + \|\nabla_\lambda \tilde{L}\|^2 \leq G_2^2$. The following theorem, proved in Appendix for general primal-dual subgradient algorithms, guarantees the convergence of the distributed source coding and rate allocation algorithm to the optimum.

Theorem 1: Let p^*, λ^*, q^* denote optimal values of the dual variables and D^*, x^* denote the optimal values of the primal variables. For constant stepsize $\epsilon_\tau = \epsilon$ and $\gamma_\tau = \gamma$, we have

$$\tilde{L}(D^*, x^*, p(\infty), q(\infty), \lambda(\infty)) \geq \tilde{L}(D^*, x^*, p^*, q^*, \lambda^*) - \frac{\epsilon G_1^2 + \gamma G_2^2}{2}, \tag{26}$$

and

$$P(D(\infty)) \leq P(D^*) + \frac{\epsilon G_1^2 + \gamma G_2^2}{2}. \tag{27}$$

Note that by optimality condition, $\tilde{L}(D^*, x^*, p(\infty), q(\infty), \lambda(\infty)) \leq \tilde{L}(D^*, x^*, p^*, q^*, \lambda^*)$. Since $\tilde{L}$ is a continuous function, inequality (26) implies that the average price $p(\infty)$, $q(\infty)$ and $\lambda(\infty)$

are bounded, and thus the average distortion $\bar{D}(\infty)$ and data rate $\bar{x}(\infty)$ is within the achievable region defined by equation (4). So, $P(D(\infty)) \geq P(D^*)$. Inequality (27) implies that the average distortion and transmission rates approach the optimum when the stepsize ϵ and γ is small enough.

In practice, the proposed distributed algorithm can be implemented as either an offline or an online algorithm. If it is implemented as an offline algorithm, the algorithm will be run until it converges; and sources then send data using the converged average D^{ms} and x^{ms}. If it is implemented as an online algorithm, decoding errors in the transient phase can be reduced if the sources code at a distortion level slightly higher than the average D^{ms} up to that point, and if the coding block length spans a number of oscillations in the optimization algorithm. It is difficult to entirely remove the possibility of decoding failure, since in the transient stage the algorithm is probing and learning the network.

C. Choice of stepsize

The bound on the norm of the subgradient scales with the size of the system according to $G_1^2 \sim 2\sum_m |\mathcal{S}_m|$ and $G_2^2 \sim 2\sum_m |\mathcal{S}_m||\mathcal{T}_m| + \sum_{m,s} |\mathcal{L}_{ms}|$. For a given performance gap δ (i.e., we require $P(D(\infty)) \leq P(D^*) + \delta$), we can choose any stepsize ϵ and λ such that $\epsilon G_1^2 + \gamma G_2^2 \leq 2\delta$. One convenient choice is $\epsilon = \delta / G_1^2 \sim \frac{1}{2\sum_m |\mathcal{S}_m|}$ and $\gamma = \delta / G_2^2 \sim \frac{1}{2\sum_m |\mathcal{S}_m||\mathcal{T}_m| + \sum_{m,s} |\mathcal{L}_{ms}|}$.

The choice of the stepsize also determines the convergence speed: the larger the stepsize the faster the convergence, and the smaller the stepsize the slower the convergence. This can be seen from the first term in the right hand side of inequalities (41) and (47) in the Appendix. Inequalities (41) and (47) also give the tradeoff between optimality (dictated by the second term in the right hand side of the inequalities) and the convergence speed (dictated by the first term in the right hand side of the inequalities).

While the above analysis shows the approximate scaling of the step sizes based on system parameters, in practice we can fine-tune the stepsize by observing the evolution of the source rates. For instance, we can first choose large stepsizes to ensure fast convergence, and subsequently, the stepsizes can be reduced as the source rates start oscillating around some mean value. On the other hand, in some distributed network scenarios, a constant stepsize may be more convenient to implement. A numerical example comparing different stepsizes is given in the following section.

D. Numerical Example

In this subsection, we provide numerical examples to complement the analysis in previous subsections. We consider a simple network as shown in Figure 1. For simplicity, we assume that there is only one multicast session with two correlated sources s_1 and s_2, and two sinks t_1 and t_2. The capacity of link $(s_1, 1)$ is 0.4 and the capacity of link $(s_2, 1)$ is 0.3. All the other links have unit capacity. We assume that all the sources have the same utility function $U^D(D) = \log(1 - D)$. We also assume that $h(s_1|s_2) = h(s_2|s_1) = 0.2$ and $h(s_1) = h(s_2) = 0.5$.

The multicast tree for source s_1 is chosen as $\{(s_1, 1), (1, 2), (2, t_2), (s_1, t_1)\}$, and for source s_2 is chosen as $\{(s_2, 1), (1, 2), (2, t_1), (s_2, t_2)\}$. Figure 2 shows the evolution of source distortions versus the number of iterations for lossy source coding with stepsizes $\epsilon = 1$ and $\gamma = 0.01$, and Figure 3 shows the evolution of source rates versus the number of iterations. We see that both source distortions and rates converge to a neighborhood of the corresponding optimum and oscillate around them. Figure 4 shows the evolution of the congestion prices at two of the links. Again, we see that the congestion prices converge to a neighborhood of the corresponding optimum and oscillate around them.

In order to study the impact of different choices of the stepsize on the convergence of the algorithms, we have run simulations with different stepsizes, see, as an illustrative example, Figure 5 for the evolution of a source rate with 3 different stepsizes. We found that the smaller the stepsize, the slower the convergence, and the larger the stepsize, the faster the convergence, which is a general characteristic of any gradient based method. However, when the stepsize is too large, the system may only approach to within a certain neighborhood of the equilibrium. This can also be seen from the proof of Theorem 1, which gives the relation between the size of this neighborhood and the stepsize. So, there is a tradeoff between convergence speed and optimality.

V. Extensions and Practical Considerations

1) Networks without given multicast trees: If there are no predefined multicast trees, we can use the flow constraints (1)-(3) in place of (7)-(8) in the optimization problem (28). The resulting problem can be solved in a similar way as the algorithm (18)-(25), except that the session scheduling component becomes similar to the back-pressure scheduling in [18].

2) Multicast without Network Coding: Network coding shows up in constraint (7). Many current networks do not employ network coding. In routing based multicast, (7) is replaced by $\sum_s \xi_l^{ms} x^{ms} \leq f_l^m$. The rest of the algorithm is essentially unchanged.

3) Practical Source Codes and Network Codes: As mentioned in Section III.B, our optimization approach works for any achievable convex rate-distortion region. While separate source coding and network coding is in general suboptimal compared to network-source coding, lower complexity can generally be achieved, e.g. using separate random [7] or deterministic [19] network codes with dithered lattice quantizers and the LDPC based Slepian-Wolf encoders [20].

4) Layered Source Coding: In heterogeneous networks, different sinks may demand different distortion levels. A practical approach is to encode information hierarchically in layers $i = 1, \ldots, n$ and to have each sink subscribe to a subset $1, \ldots, k \leq n$ of the layers, starting from the base layer [22].

Our optimization framework can be extended to this scenario as follows. We define a multicast tree for each layer of each session, which connects to a subset of the sinks. Specifically, let n_m be the number of layers in session m, and let T^{msi} denote the layer i multicast tree for source s in session m. Let x^{msi} be the transmission rate over tree T^{msi}. Let D^{msj} be the distortion achieved by subscribing to layers 1 to j, and let $U_{msj}(D^{msj})$ be the associated utility. The source coding and network resource allocation problem becomes

$$\begin{aligned} \max_{D,x,f} \quad & \sum_{m \in \mathcal{M}, s \in \mathcal{S}_m, 1 \leq j \leq n_m} U_{msj}(D^{msj}) \\ \text{s.t.} \quad & \xi_l^{msi} x^{msi} \leq f_l^{mi}, \forall l \in \mathcal{L}, m \in \mathcal{M}, s \in \mathcal{S}_m, 1 \leq i \leq n_m, \\ & \sum_{m \in \mathcal{M}, 1 \leq i \leq n_m} f_l^{mi} \leq c_l, \forall l \in \mathcal{L}, \\ & (x^{msi}, D^{msj} : \ 1 \leq i, j \leq n_m, s \in \mathcal{S}_m) \in \mathcal{R}, \forall m \in \mathcal{M} \end{aligned} \tag{28}$$

for some given convex achievable region $\mathcal{R}$ of a successively refinable rate-distortion code. The same optimization technique applies, since we can introduce auxiliary variables to make the convex achievable region a local rate-distortion constraint at each sink, as what has been done in problem (10).

However, the capacity region of successively refinable rate-distortion coding is in general unknown. As an alternative, distortion optimization can be realized by having sinks subscribe adaptively to more or fewer layers based on the congestion prices. If a sink observes that the congestion price converges, it subscribes to an additional layer. If the congestion price does not converge (i.e., congestion is built up unboundedly), the sink drops a layer. Slepian-Wolf coding can be applied to sources in the same layer.

5) Entropy and Probability Density Estimation: State-of-the-art distributed source codes need the knowledge of joint probability density function (pdf) of all the sources in each session for both encoding and decoding. It is hard for all the sources to learn this information. Our proposed framework relaxes this constraint by requiring that only sinks need it. A possible approach for estimating the joint pdf at the sinks is for the sources to initially transmit quantized data without Slepian-Wolf coding. On receiving this data, the sinks estimate the joint pdf by using well-developed techniques in multivariate density estimation [23]. Later, the estimated pdf can be refined by the decompressed data. Cyclic redundancy checks can be used to detect errors in the decoded data, in which case the rate-distortion region is conservatively modified such that the next data frame can be decoded correctly. However, as pdf estimation is complicated, it is desirable to have universal distributed source codes.

VI. Conclusion

We have presented a fully distributed algorithm for adaptive lossy source coding for multicast with network coding, where each session contains a set of correlated sources. Based on the utility maximization framework and its decomposition, we proposed a distributed algorithm for joint optimization of source coding and network coding. The resulting receiver-driven algorithm adjusts distortion levels according to the distortion prices fed back from the sinks, and hence does not require coordination among the sources. With random network coding, the algorithm can be implemented in a fully distributed manner. In this work we have used the known rate-distortion region for high resolution lossy source coding; our work easily extends to achievable regions that can be expressed in a related form. It would be interesting to extend our work to other achievable rate-distortion regions.

References

[1] R. Ahlswede, N. Cai, S. Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[2] R. Cristescu, B. Beferull-Lozano, and M. Vetterli, "Networked Slepian-Wolf: Theory, algorithms and scaling laws," *IEEE Trans. Inform. Theory*, vol. 51, no. 12, pp. 4057–4073, Dec. 2005.

[3] A. Lee, M. Médard, K. Haigh, S. Gowan, and P. Rubel, "Minimum-cost subgraphs for joint distributed source and network coding," in *Proc. of Workshop on Network Coding, Theory and Applications*, Jan. 2007.

[4] R. Cristescu and B. Beferull-Lozano, "Lossy network correlated data gathering with high-resolution coding," *IEEE/ACM Trans. Networking*, vol. 14, pp. 2817–2824, June 2006.

[5] F. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, March 1998.

[6] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition," in *Proc. of IEEE*, Jan. 2007.

[7] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[8] R. Zamir and T. Berger, "Multiterminal source coding with high resolution," *IEEE Trans. Inform. Theory*, vol. 45, pp. 106–117, Jan. 1999.

[9] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: A shortest path approach," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1475–1488, Aug. 2006.

[10] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proc. of IEEE Infocom*, Apr. 2006.

[11] L. Chen, T. Ho, S. H. Low, M. Chiang, and J. C. Doyle, "Optimization Based Rate control for multicast with network coding," in *Proc. of IEEE Infocom*, 2007.

[12] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 22, no. 1, pp. 1–10, Jan. 1976.

[13] T. Berger, "Multiterminal source coding," in *The Information Theory Approach to Communications*, G. Longo, Ed. Springer-Verlag, 1977.

[14] A. Wagner, S. Tavildar, and P. Viswanath, "The rate region of the quadratic gaussian two-terminal source-coding problem," arXiv:cs.IT/0510095.

[15] D.Waitzman, S. Deering, and C. Partridge, "Distance vector multicast routing protocol," RFC 1075, Nov.1988.

[16] D. Tse and S. Hanly, "Multiaccess fading channels-part I: Polymatroid structure, optimal resource allocation and throughput capacities," *IEEE Trans. Inform. Theory*, vol. 44, no. 7, pp. 2796–2815, Nov. 1998.

[17] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1995.

[18] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," in *Proc. of Allerton Conference on Communication, Control, and Computing*, Sept. 2005.

[19] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inform. Theory*, vol. 51, no. 6, pp. 1973–1982, June 2005.

[20] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 440–442, Oct. 2002.

[21] A. Ramamoorthy, K. Jain, P. Chou, and M. Effros, "Separating distributed source coding from network coding," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2785–2795, June 2006.

[22] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE J. Select. Areas Commun.*, vol. 15, no. 6, pp. 983–1001, Aug. 1997.

[23] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

[24] T. Berger, "Multiterminal source coding," *The Information Theory Approach to Communications*, pp. 172–231, CISM Courses and Lectures, Springer-Verlag, 1977.

APPENDIX: PROOF OF THEOREM 1

In this appendix, we consider the following general problem

$$\begin{aligned} &\min_{x,y}\ f(x)+g(y) \\ &\text{subject to } h(x)+l(y)\le 0 \\ &x\ge 0,\ y\in\mathcal{Y}, \end{aligned} \tag{29}$$

where $f(\cdot), g(\cdot), h(\cdot)$ and $l(\cdot)$ are convex functions, and $\mathcal{Y}$ is certain convex set. It can be readily verified that problem (10) can be reformulated as (29). As in the paper, we do not relax all the constraints and we obtain the partial Lagrangian function of (29) as

$$L(x,y,\lambda)=f(x)+g(y)+\lambda^T(h(x)+l(y)),\ \text{subject to}\, x\ge 0,\ y\in\mathcal{Y}, \tag{30}$$

where λ is the dual variable. Define

$$\tilde{L}(x,\lambda)=f(x)+\lambda^T h(x)+\min_{y\in\mathcal{Y}}(g(y)+\lambda^T l(y)),\ \text{subject to}\, x\ge 0. \tag{31}$$

The generic primal-dual subgradient algorithm is written as

$$\begin{aligned} x(\tau+1)&=\left[x(\tau)-\epsilon\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right]^+ \\ &=\left[x(\tau)-\epsilon\left(f'(x(\tau))+\lambda^T(\tau)h'(x(\tau))\right)\right]^+, \end{aligned} \tag{32}$$

$$\begin{aligned} y(\tau)&=\arg\min_{y\in\mathcal{Y}} L(x(\tau),y,\lambda(\tau)) \\ &=\arg\min_{y\in\mathcal{Y}} g(y)+\lambda^T l(y), \end{aligned} \tag{33}$$

and

$$\begin{aligned} \lambda(\tau+1)&=\left[\lambda(\tau)+\gamma\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right]^+ \\ &=\left[\lambda(\tau)+\gamma(h(x(\tau))+l(y(\tau)))\right]^+, \end{aligned} \tag{34}$$

where $\nabla_x\tilde{L}$ and $\nabla_\lambda\tilde{L}$ denote the gradients of $\tilde{L}$ with respect to x and λ, respectively. Let x^*, y^*, λ^* denote the optimal values of x, y, λ, respectively. By duality, we have $x^*=\arg\min_{x\ge 0}(f(x)+{\lambda^*}^T h(x))$, $y^*=\arg\min_{y\in\mathcal{Y}}(g(y)+{\lambda^*}^T l(y))$, $\lambda^*=\arg\max_\lambda\tilde{L}(x^*,\lambda)=\arg\max_\lambda L(x^*,y^*,\lambda)$, and $\tilde{L}(x^*,\lambda^*)=L(x^*,y^*,\lambda^*)$.

By equation (32), we have

$$\begin{aligned}\frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2 \leq & \frac{1}{\epsilon}\left\|x(\tau)-\gamma\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)-x^*\right\|^2 \\ = & \frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2 \\ & -2(x(\tau)-x^*)^T\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right) \\ \leq & \frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2 \\ & -2\left(\tilde{L}\left(x(\tau),\lambda(\tau)\right)-\tilde{L}\left(x^*,\lambda(\tau)\right)\right),\end{aligned} \tag{35}$$

where the last inequality comes from the fact that $\tilde{L}(x,\lambda)$ is a convex function in x for given λ. Similarly, we can obtain

$$\begin{aligned}\frac{1}{\gamma}\left\|\lambda(\tau+1)-\lambda^*\right\|^2 \leq & \frac{1}{\gamma}\left\|\lambda(\tau)-\lambda^*\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2 \\ & +2\left(\tilde{L}\left(x(\tau),\lambda(\tau)\right)-\tilde{L}\left(x(\tau),\lambda^*\right)\right).\end{aligned} \tag{36}$$

By adding equations (35) and (36) together, we get

$$\begin{aligned}& \frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau+1)-\lambda^*\right\|^2 \\ \leq & \frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau)-\lambda^*\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2 \\ & -2\left(\tilde{L}\left(x(\tau),\lambda^*\right)-\tilde{L}\left(x^*,\lambda(\tau)\right)\right) \\ \leq & \frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau)-\lambda^*\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2 \\ & -2\left(\tilde{L}\left(x^*,\lambda^*\right)-\tilde{L}\left(x^*,\lambda(\tau)\right)\right),\end{aligned} \tag{37}$$

where the last inequality follows from the relation $\tilde{L}\left(x(\tau),\lambda^*\right) \geq \tilde{L}\left(x^*,\lambda^*\right)$. Applying the above inequality recursively, we get

$$\begin{aligned}& \frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau+1)-\lambda^*\right\|^2 \\ \leq & \frac{1}{\epsilon}\left\|x(1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(1)-\lambda^*\right\|^2+\sum_{i=1}^{\tau}\left(\epsilon\left\|\nabla_x\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2\right) \\ & -2\sum_{i=1}^{\tau}\left(\tilde{L}\left(x^*,\lambda^*\right)-\tilde{L}\left(x^*,\lambda(i)\right)\right).\end{aligned} \tag{38}$$

Since $\frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau+1)-\lambda^*\right\|^2 \geq 0$, we have

$$\begin{aligned}
&2\sum_{i=1}^{\tau}\left(\tilde{L}\left(x^*,\lambda^*\right)-\tilde{L}\left(x^*,\lambda(i)\right)\right)\\
\leq&\frac{1}{\epsilon}\left\|x(1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(1)-\lambda^*\right\|^2+\sum_{i=1}^{\tau}\left(\epsilon\left\|\nabla_x\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2\right)\\
\leq&\frac{1}{\epsilon}\left\|x(1)-x^*\right\|_2^2+\frac{1}{\gamma}\left\|\lambda(1)-\lambda^*\right\|_2^2+\tau(\epsilon G_1^2+\gamma G_2^2),
\end{aligned} \tag{39}$$

where $\left\|\nabla_x\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2 \leq G_1^2$ and $\left\|\nabla_\lambda\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2 \leq G_2^2$ by assumption. From equation (39), we obtain

$$\frac{1}{\tau}\sum_{i=1}^{\tau}\left(\tilde{L}\left(x^*,\lambda^*\right)-\tilde{L}\left(x^*,\lambda(i)\right)\right) \leq \frac{\left\|x(1)-x^*\right\|^2/\epsilon+\left\|\lambda(1)-\lambda^*\right\|^2/\gamma}{2\tau}+\frac{\epsilon G_1^2+\gamma G_2^2}{2}. \tag{40}$$

Since $\tilde{L}(x^*,\lambda)$ is a concave function in λ, by Jensen's inequality, we have

$$\tilde{L}\left(x^*,\lambda^*\right)-\tilde{L}\left(x^*,\bar{\lambda}(\tau)\right) \leq \frac{\left\|x(1)-x^*\right\|^2/\epsilon+\left\|\lambda(1)-\lambda^*\right\|^2/\gamma}{2\tau}+\frac{\epsilon G_1^2+\gamma G_2^2}{2}. \tag{41}$$

where $\bar{\lambda}(\tau)=\frac{1}{\tau}\sum_{i=1}^{\tau}\lambda(i)$. So,

$$\tilde{L}(x^*,\bar{\lambda}(\infty)) \geq \tilde{L}\left(x^*,\lambda^*\right)-\frac{\epsilon G_1^2+\gamma G_2^2}{2}. \tag{42}$$

Now, consider

$$\begin{aligned}
\frac{1}{\gamma}\left\|\lambda(\tau+1)\right\|^2 \leq&\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+2\lambda^T(\tau)\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\\
=&\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+2\left(\lambda^T(\tau)h(x(\tau))-g(y(\tau))\right)\\
&+2\left(g(y(\tau))+\lambda^T(\tau)l(y(\tau))\right)\\
\leq&\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+2\left(\lambda^T(\tau)h(x(\tau))-g(y(\tau))\right)\\
&+2\left(g(y^*)+\lambda^T(\tau)l(y^*)\right),
\end{aligned} \tag{43}$$

where the last inequality follows from the fact that $y(\tau)$ is the minimizer for the problem $\min_y(g(y)+\lambda^T(\tau)l(y))$. By adding (35) and (43) together, we get

$$\begin{aligned}
&\frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau+1)\right\|^2\\
\leq&\frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2\\
&-2\left(f(x(\tau))+g(y(\tau))-f(x^*)-g(y^*)-\lambda^T(\tau)\left(h(x^*)+l(y^*)\right)\right).
\end{aligned} \tag{44}$$

Note that, by the optimality condition, we have $\lambda^{*T}\left(h(x^*)+l(y^*)\right)=0$; and furthermore, if any constraint $\{h(x^*)+l(y^*)\}_j<0$, the corresponding dual optimum $\{\lambda\}_j=0$. It follows that $(\lambda(\tau)-\lambda^*)^T\left(h(x^*)+l(y^*)\right)\leq 0$, i.e., $\lambda^T(\tau)\left(h(x^*)+l(y^*)\right)\leq 0$. Thus,

$$\begin{aligned}&\frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau+1)\right\|^2\\ \leq&\frac{1}{\epsilon}\left\|x(\tau)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2+\epsilon\left\|\nabla_x\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(\tau),\lambda(\tau)\right)\right\|^2\\ &-2\left(f(x(\tau))+g(y(\tau))-f(x^*)-g(y^*)\right).\end{aligned}\tag{45}$$

Applying the above inequality recursively, we get

$$\begin{aligned}&\frac{1}{\epsilon}\left\|x(\tau+1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(\tau)\right\|^2\\ \leq&\frac{1}{\epsilon}\left\|x(1)-x^*\right\|^2+\frac{1}{\gamma}\left\|\lambda(1)\right\|^2+\sum_{i=1}^{\tau}\left(\epsilon\left\|\nabla_x\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2+\gamma\left\|\nabla_\lambda\tilde{L}\left(x(i),\lambda(i)\right)\right\|^2\right)\\ &-2\sum_{i=1}^{\tau}\left(f(x(i))+g(y(i))-f(x^*)-g(y^*)\right).\end{aligned}\tag{46}$$

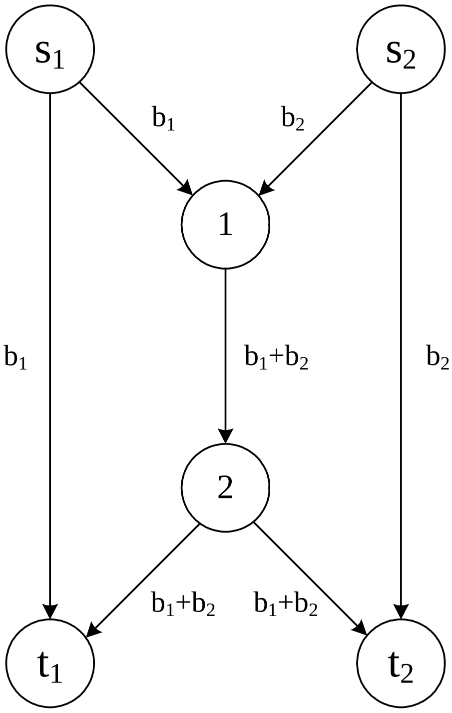
Following similar procedure and using the Jensen's inequality (for convex functions) as in the derivation of inequality (42), we get

$$f(\bar{x}(\tau))+g(\bar{y}(\tau))-f(x^*)-g(y^*)\leq\frac{\left\|x(1)-x^*\right\|^2/\epsilon+\left\|\lambda(1)\right\|^2/\gamma}{2\tau}+\frac{\epsilon G_1^2+\gamma G_2^2}{2},\tag{47}$$

where $\bar{x}(\tau)=\frac{1}{\tau}\sum_{i=1}^{\tau}x(i)$ and $\bar{y}(\tau)=\frac{1}{\tau}\sum_{i=1}^{\tau}y(i)$. So,

$$f(\bar{x}(\infty))+g(\bar{y}(\infty))\leq f(x^*)+g(y^*)+\frac{\epsilon G_1^2+\gamma G_2^2}{2}.\tag{48}$$

Remark: Inequalities (42) and (48) and their proof are very general. They apply to any primal-dual subgradient algorithms for convex optimization, and provide a general result on the performance of the primal-dual subgradient algorithm. They are a nice addition to the similar result on the dual subgradient algorithm that is presented in, e.g., [10].



(a)

Fig. 1. The butterfly network.

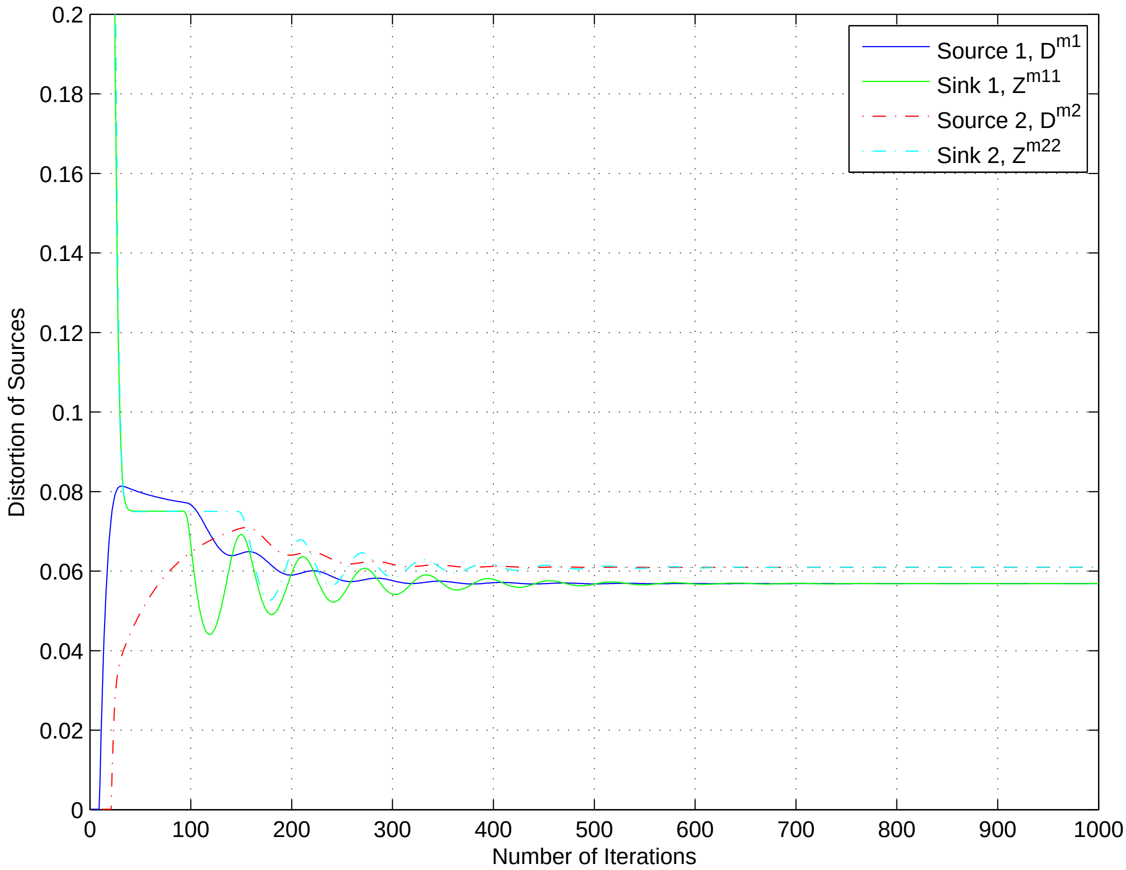


Fig. 2. The evolution of source distortions D^{m1}, D^{m2} and sink virtual distortions Z^{m11}, Z^{m22} versus the number of iterations with stepsizes $\epsilon = 1$ and $\gamma = 0.01$ for the butterfly network in Fig. 1.

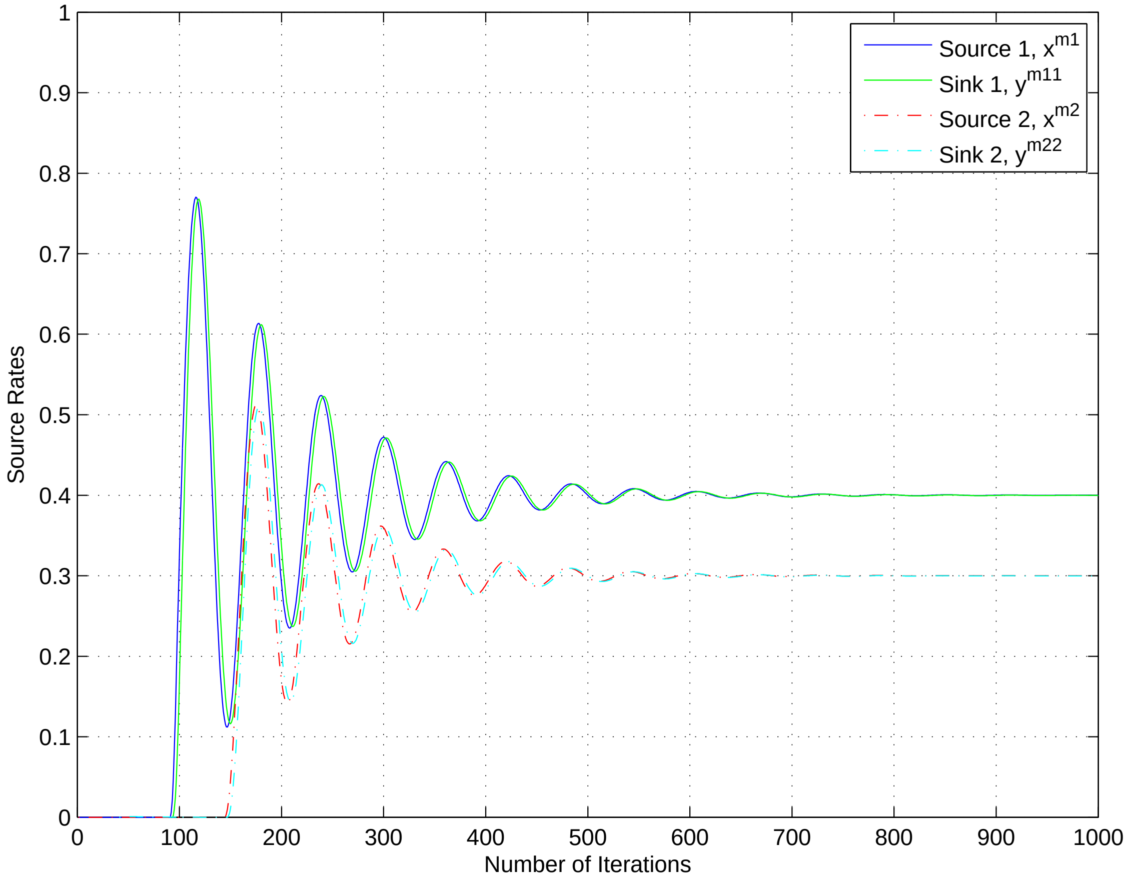


Fig. 3. The evolution of source rates x^{m1}, x^{m2} and sink virtual rates y^{m11}, y^{m22} versus the number of iterations with stepsizes $\epsilon = 1$ and $\gamma = 0.01$ for the butterfly network in Fig. 1.

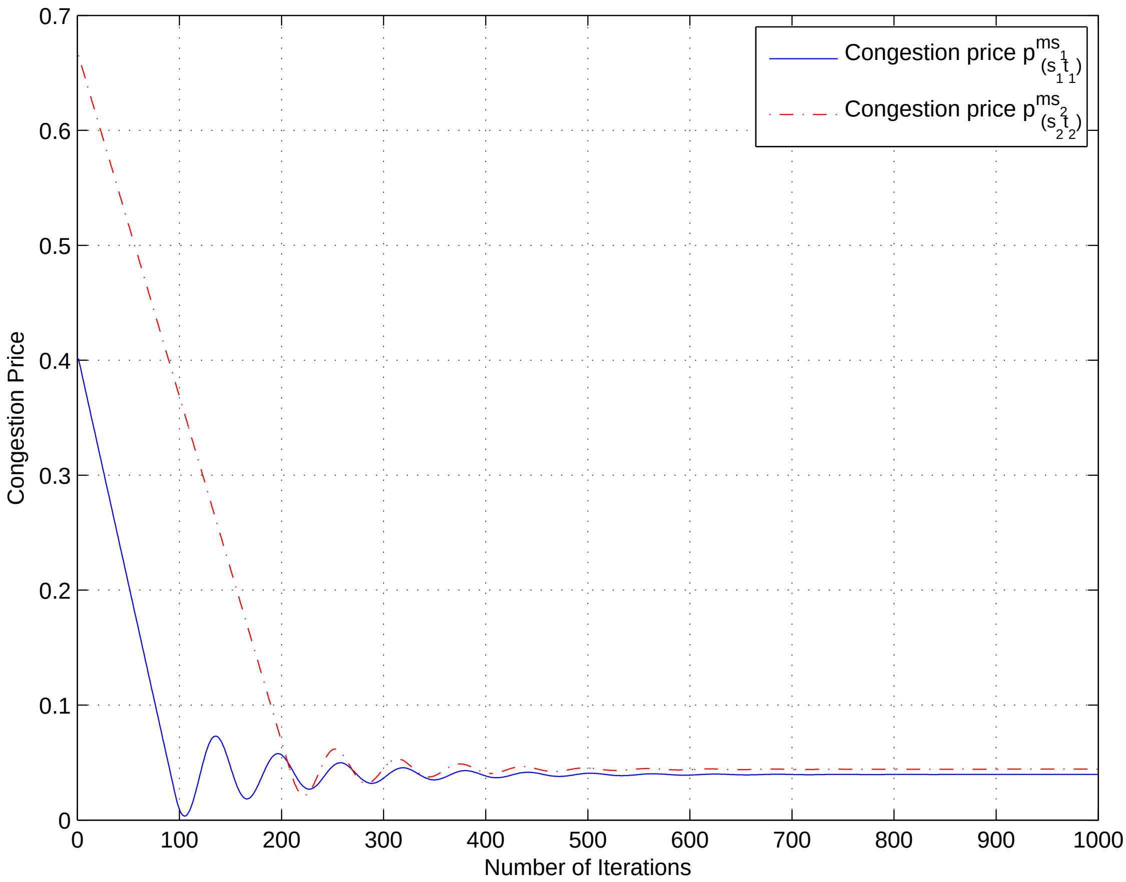


Fig. 4. The evolution of congestion prices versus the number of iterations with stepsizes $\epsilon = 1$ and $\gamma = 0.01$ for the butterfly network in Figure 1. Congestion price $p^{ms_1}_{(s_1t_1)}$ is for link $(s_1, 1)$, and congestion price $p^{ms_2}_{(s_2t_2)}$ is for link $(s_2, 1)$ in Fig. 1.

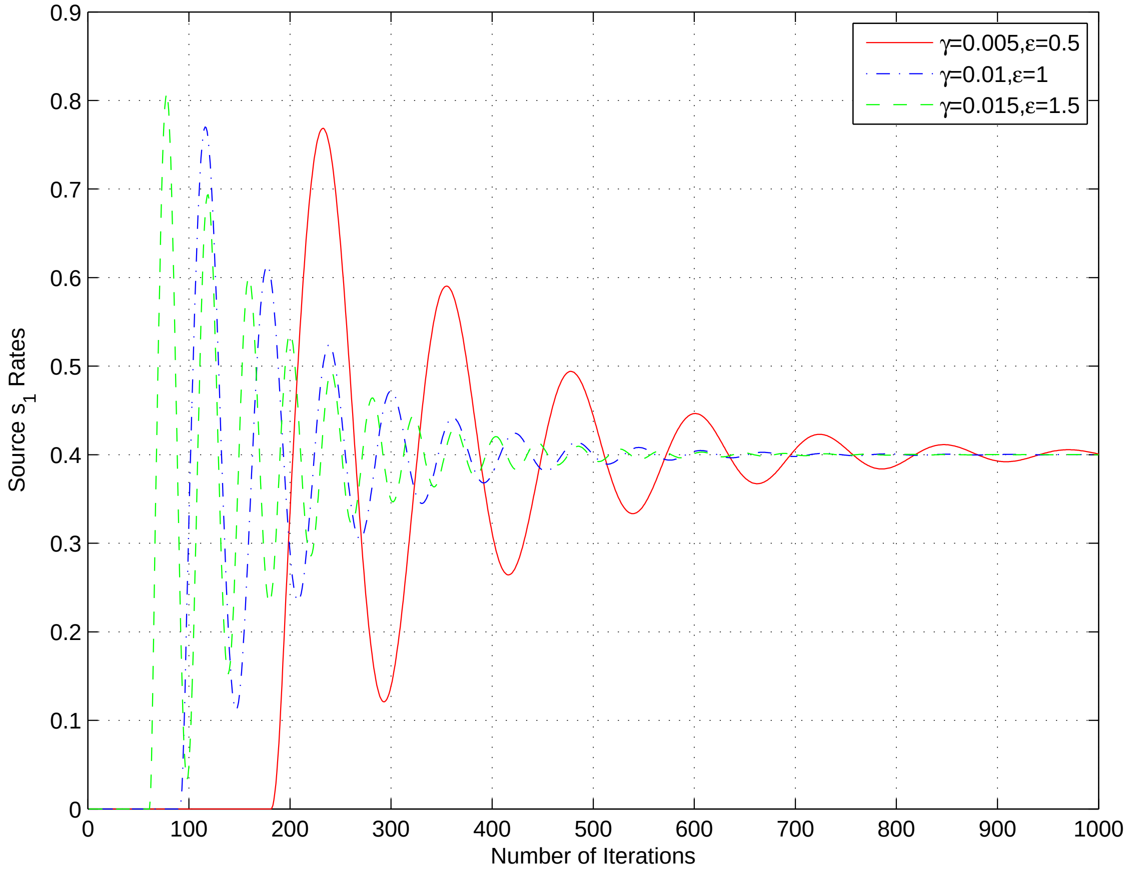


Fig. 5. The evolution of source s_1 rate versus the number of iterations with different ϵ and γ for the butterfly network in Fig. 1.