Network Correlated Data Gathering With Explicit Communication: NP-Completeness and Algorithms

Răzvan Cristescu, *Member, IEEE*, Baltasar Beferull-Lozano, *Member, IEEE*, Martin Vetterli, *Fellow, IEEE*, and Roger Wattenhofer

Abstract—We consider the problem of correlated data gathering by a network with a sink node and a tree-based communication structure, where the goal is to minimize the total transmission cost of transporting the information collected by the nodes, to the sink node. For source coding of correlated data, we consider a joint entropy-based coding model with explicit communication where coding is simple and the transmission structure optimization is difficult. We first formulate the optimization problem definition in the general case and then we study further a network setting where the entropy conditioning at nodes does not depend on the amount of side information, but only on its availability. We prove that even in this simple case, the optimization problem is NP-hard. We propose some efficient, scalable, and distributed heuristic approximation algorithms for solving this problem and show by numerical simulations that the total transmission cost can be significantly improved over direct transmission or the shortest path tree. We also present an approximation algorithm that provides a tree transmission structure with total cost within a constant factor from the optimal.

Index Terms—Conditional coding, correlated data gathering, distributed algorithms, NP-completeness, routing tree, sensor networks, traveling salesman.

I. INTRODUCTION

A. Correlated Data Gathering

Consider a number of distributed data sources with a certain correlation structure and which are located at the nodes of a network. A practical example of such a situation is the case of sensor networks that measure environmental data [1], [23], [25].
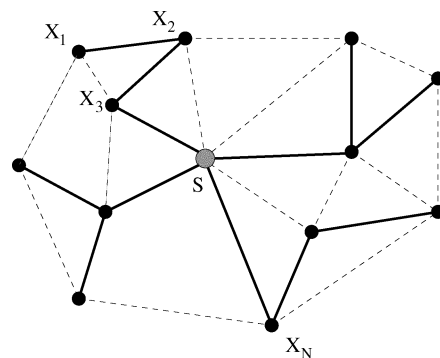
Fig. 1. In this example, data from nodes $X_1, X_2, \ldots, X_N$ need to arrive at sink S. A rate supply R_i is allocated to each node X_i. In thick solid lines, a chosen tree transmission structure is shown. In thin dashed lines, the other possible links are shown.

Collecting images from various sources into a common repository on the internet is another example of correlated data gathering. A number of links connect sources to each other, establishing a graph where sources are nodes and links are edges. The task is to send all the data to a particular node of the graph that is called *sink*. In the practical case of sensor networks, this node is denoted as base station. A typical transmission structure that is found in practice is the tree, that is, the data are sent from the nodes to the sink, using a tree which has the sink as a root. Since such structures are widely used in networks and lead to computationally efficient communication algorithms while requiring a minimum communication overhead, we will restrict our analysis to tree transmission structures. The goal is to gather all data at the sink using this tree (subgraph of the original graph), while minimizing a cost functional (e.g., total flow cost). We refer to this problem as the *correlated data gathering problem*. This problem can be viewed as an instance of a network flow problem, but with an original twist: because the data is correlated, standard solutions may not be optimal, which leads to an original problem that combines the joint optimization of rate allocation and tree building.

An example is shown in Fig. 1, where we have N nodes with sources $X_1, \ldots, X_N$, a sink S, and a number of edges that connect the sources. Intermediate nodes can be also used as relays in addition to measuring data. They aggregate their own data with the data received from other nodes, and at the same time, due to the correlation, the intermediate nodes can reduce the necessary rate to code their data. A very important task in this scenario is to find a tree transmission structure on the network graph that minimizes a cost of interest (e.g., flow cost [function(rate)] · [path

weight], total distance, etc.). This leads to the question of how to construct efficient data gathering trees.

When the data measured at nodes are statistically independent, the problem becomes separable: because of the statistical independence, the choice of transmission structure does not affect the rate at each node. Namely, first, each node simply encodes its own data independently; then, well developed algorithms can be used to solve various network problems involving costs related to only the link weights (minimum and shortest path spanning tree).

However, in many situations, such as in typical sensor networks, data at nodes are *not* independent. Thus, due to the correlation that is present, it is expected that coding approaches that take this correlation into account (e.g., conditional coding), will outperform traditional approaches, for various cost functions of interest. Moreover, jointly exploiting the data structure and optimizing the transmission structure in the network, can provide substantial further improvements. Therefore, it is worth studying the interaction between the correlation of the data measured at nodes and the transmission structure that is used to transport these data to the sink.

An important practical instance of this type of problem can be found in sensor networks [1], [22], [23]: a number of sensors acquire measurements from the environment (e.g., temperature) which are typically correlated to each other, and these measurements are sent to a base station for decision or control purposes. Let $\mathbf{X} = (X_1, \ldots, X_N)$ be the vector formed by the random variables measured at the nodes $1, \ldots, N$. The samples taken at nodes are spatially correlated. We assume that the random variables are continuous and that there is a quantizer in each sensor (with the same resolution for each sensor). A rate allocation $(R_1, \ldots, R_N)$ (each R_i is expressed in bits) has to be assigned at the nodes so that the *quantized* measured information samples are described losslessly, so that they can be fully reconstructed at the sink. That information has to be transmitted through the links of the network to the designated base station. We abstract the communication structure to a connectivity graph with point-to-point links given by the edges of the graph (see Fig. 1, where the edges are determined by either the transmission range of nodes, or by the k-nearest neighborhood). In other words, instead of considering the full wireless multi-point case, we assume a simplified communication model with a medium access control (MAC) protocol, which makes sure that there are no collisions or interferences at a node. A meaningful cost function to minimize is the energy consumption, which is essentially given by the sum of products [function(rate)] $\cdot$ [link weight], for all the links and node rates used in the transmission. Here, the weight of the link between two nodes is a function of the distance d between the two nodes of the link (e.g., kd^ν or $k\exp(\nu d)$, with k, ν constants that depend on the transmission medium properties).

There are two complementary approaches that can be used in this problem. The first approach is to allow nodes to use joint coding of correlated data without explicit communication (this is possible by using random binning coding strategies, namely, using Slepian–Wolf coding [6], [24], [28]). With this approach, finding the optimal transmission structure turns out to be simple, because the joint problem of optimizing the transmission structure and the source coding becomes decoupled and can be solved in a separable manner; however data coding becomes complex and global knowledge of the network structure and the correlation structure is needed for an optimal solution.[1] This approach has been treated in [7] and [8], where in addition, scenarios including more than one sink are studied.

In the second approach, considered in this paper, nodes can exploit the data correlation only by receiving explicit side information from other nodes (for example, when other nodes use a node as relay, their data is locally available at that relaying node). Thus, the correlation structure is exploited through communication and joint aggregate coding/decoding locally at each node. We call this approach the *explicit communication* approach. In this case, data coding can be performed in a simple way and relies only on locally available data as side information. However, optimizing the transmission structure becomes complex, as we show in this paper. Notice that in the explicit communication case it is not necessary to know the correlation structure a-priori. This is because the correlation structure is learned *explicitly* in a distributed manner through the explicit communication itself. This leads to a simple source coding, but the transmission structure optimization is hard.

B. Related Work

The problem of data gathering has been considered in several related works in the context of sensor networks. Let us briefly review some of the algorithms proposed so far.

In [14], the authors introduce the cluster-based LEACH algorithm. In their model, the cluster head nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the base station. The work in [19] introduces the PEGASIS algorithm, that uses the [energy] × [delay] metric over the routing tree; their algorithms find chains of nodes instead of clusters. However, none of these works exploits the correlation present in the data.

In [15], data gathering is done using directed diffusion. Sensors measure events, creating gradients of information in their respective neighborhoods, while the base station requests data by broadcasting interests, meaning events relevant for the base station. The best paths of information flow on which interests fit gradients are reinforced. In order to reduce communication costs, data is aggregated on the way on aggregation trees.

Similar work where the interplay of data compression and routing is studied can be found in [11], [17], and [27]. In [11], the authors address the problem of data gathering and compression at relay nodes by using the theory of concave costs applied to single source aggregation. The authors develop an elegant algorithm that finds good trees that simultaneously minimize several concave cost functions of interest. The related problem of single-source uncapacitated minimum concave-cost network flow has been extensively studied (e.g., [12], [29]). The main

---

[1]We prove in [8] that, when Slepian–Wolf coding is used, the optimal transmission structure is the shortest path tree (SPT). Further, the optimal rate allocation requires ordering of the nodes on the SPT: the optimal rate allocation at each node is obtained by conditioning the information at that node on all nodes closer to the sink on the SPT. This requires knowledge at each node of the SPT ordering, and thus global network knowledge.

difference with our work is that in our case, due to the correlation structure and coding model we consider, the amount of aggregated information sent down the tree to the next hop from a particular node depends on the structure of the subtree whose parent is that node, whereas in [11] that amount only depends on the number of nodes in the subtree, and not on the particular links chosen to build that subtree.

In [21], an empirical data correlation model is used for a set of experimentally obtained data, and the authors propose cluster-based tree structures shown to have a good performance depending on the correlation level. The correlation function is derived as an approximation of the conditional entropy, and the cost function is the sum of bits transmitted by the network. In our work, we additionally consider the inter-node distances in the cost function, and we provide a thorough complexity analysis of routing involving coding based on explicit communication. It is interesting to note that the transmission structures for routing driven compression found in [21] are similar to the ones we derive later in this paper (aggregating near the sources versus progressing toward the sink).

In [9], a circular-coverage correlation model on a grid is used, where correlation is modeled as a parameter proportional to the area covered by a sensor. The authors provide randomized shortest-path aggregation trees with constant-ratio approximations. We consider a different correlation model, where coding is entropy-based and relies on nearest neighbor conditioning; moreover, we allow for alternative routing structures other than shortest paths.

Some examples of network flow with joint coding of correlated sources under capacity constraints on the transmission links and Slepian–Wolf constraints on the rates are studied in [3], where trees are shown to perform suboptimally if splittable flows are allowed. Since practical use of Slepian–Wolf codes is still difficult in the case of large sensor networks, in this paper we use a model with explicit communication, where nodes use relayed side information to compress their data. Note that for the explicit communication model, when splitting packets the amount of overhead increases importantly with the network size, since nodes need to keep track from where the different subpackets are originating, in order to construct the conditional histograms. Additionally, to the best of our knowledge, there is no proper full solution available yet to the problem of wireless data gathering with capacities on the links.

C. Main Contributions

We first provide a formal definition of the problem of cost efficient data gathering with explicit communication in a sensor network. Namely, we study the case where joint coding of correlated data by the network is performed explicitly, that is, the reduction in rate by entropy coding due to the correlation is possible at a node only when side information is explicitly available (as relayed data from another node). We consider a simplified version of our general problem setting that results in an original flow optimization problem on a graph. We show some network examples where a joint treatment of rate allocation and transmission structure optimization provides important improvements over the shortest path tree. However, we prove that this optimization problem is NP-hard, by a nontrivial reduction

from the min-set cover problem. Then, we propose a set of distributed heuristic approximation algorithms that provide good solutions for this problem. We show by simulations how a combination of the shortest path tree and traveling salesman paths approximates well the solution given by simulated annealing, that is expected to provide results close to the optimum. Moreover, we present an approximation algorithm that provides tree transmission structure solutions within a constant from the optimal solution, for any possible graph instance (worst case). We compare the various scenarios through numerical simulations to show how our algorithms provide important improvements in terms of total costs, as compared to the shortest path tree.

D. Outline of the Paper

In Section II, we define the problem studied in this paper. In Section III, we present a scenario that uses simplified assumptions for our problem setting, and we prove that even in this case, the corresponding optimization problem is NP-hard. In Section IV, we propose a set of heuristic approximation algorithms that provide good average improvements over direct transmission or the shortest path tree. In Section V, we present an algorithm that generates a spanning tree with cost within a constant bound from the optimal solution. Then, in Section VI, we compare our proposed algorithms by numerical simulations. We provide our conclusions in Section VII.

II. PROBLEM FORMULATION

We consider the problem of data gathering with a single sink, to which all the data has to be sent. Let $G = (V, E)$ be a weighted graph with $|V| = N + 1$. We denote by S the particular $(N + 1)$th node called sink. Except the sink, every node in the graph generates a source. Each edge $e = (i, j) \in E$ has a weight w_e. Since the data are correlated, depending on the chosen transmission structure, each node i has to transmit a certain rate R_i through the network to the sink. Let $f(x_e, w_e)$ be an arbitrary cost function of the total rate (flow) x_e going through a particular edge with weight w_e. Then the general *minimum cost data gathering tree* problem is defined as follows. Find the spanning tree T of the graph G that minimizes the cost function

$$c_T = \sum_{e \in T} f(x_e, w_e) \tag{1}$$

with the additional constraint that the incoming flow at the sink provides enough information to reconstruct data measured at all nodes.

We restrict our discussion to functions $f(\cdot, \cdot)$ which are separable as the product of a function that depends only on the rate and another function that depends only on the link weights of the transmission structure.[2] Without loss of generality, we assume $f(x, w) = x \cdot w$.

Note that in a data gathering scenario based on explicit communication, it is not practical to allow splittable flows, the reason being that it is hard to implement a protocol where

---

[2]This corresponds to many practical settings (e.g., the [rate] · [path weight] cost function measures the transmission cost in wired networks, and the [exp(rate)] · [path weight] measures the battery consumption in wireless networks, where the [path weight] term is a function of the inter-node distances along a path).

intermediate nodes are able to disseminate from where pieces of information originate; such knowledge is crucial since data are correlated and the compression is based on entropy[3] coding, by conditioning on available relayed information.[4] Moreover, a fully wireless scenario requires solving the general problem with capacitated flows, and this requires considering of additional issues like wireless interference; to the best of our knowledge this is still an open problem. Thus, in this paper, we consider unsplittable and uncapacitated flows. Under these assumptions, the optimal transmission structure is a tree. Namely, in order that the sink has full information about all nodes, each node needs to either provide its full information, or it needs such side information from another node, to perform data conditioning. As a result, at least an incoming link to each nonleaf node in the transmission tree has to contain full side information; thus, adding links to a tree only increases the total cost of communication.

With these assumptions, the expression (1) to be minimized can be rewritten as

$$c_T = \sum_{i \in V} R_i d_T(i, S) \qquad (2)$$

where $d_T(i, S)$ is the total weight of the path connecting node i to S on the T tree. R_i is the incoming rate at node i that models the rate in bits obtained by coding either (a) the full information at that node, if there is no side information available, or (b) the information at that node conditioned by side information available from forwarded nodes.

The important new feature that makes this problem different from classical network flow theory is the following: by changing the transmission structure, since we change the inter-node distances, both the set of rates $\{R_i\}_{i=1}^N$, which depends on the inter-node correlation, and the path weights $\{d_T(i)\}_{i=1}^N$ are affected. Thus, the optimization of the set of rates and the path weights has to be done jointly, and it cannot be decoupled. We call this new problem the *minimum cost correlated data gathering tree* problem.

We now particularize the optimization problem (2) to the explicit communication-based coding setting. In classical network transport theory, the amount of supply (rate in our case) at a node is fixed and independent of the communication links that are chosen to transport the various supplies. In particular, the supply provided by the ith node is independent of the nodes that are connected to the ith node through the chosen edges. In our problem formulation, an important novelty is that the supply at a given node *depends* both on the incoming flow from other nodes that use that node as relay, and *also* on the transmission structure that is used for these nodes. This novel feature is not captured by generalized network flow settings [4], where supplies at nodes depend only on the incoming flow, but *not* on the transmission structure used to transport that flow.

Consider again the example in Fig. 1, where nodes have to communicate their correlated data to one sink. To reduce the complexity of local coding, we assume that each relay node forward received packets without decoding/re-coding received information and they only perform compression by conditional entropy coding of its own measured data, given the received data from the nodes that are using it as intermediate relay node. Denote by $H(X|Y)$ the conditional entropy of a random variable X given that the random variable Y is known. If we consider node X_3, then the rate it has to supply depends on whether (a) neither X_1 nor X_2 use it as a relay, and then X_3 uses a rate $H(X_3)$, or (b) X_1 and/or X_2 use it as a relay, and then X_3 uses a rate $H(X_3|X_1$ and/or $X_2)$. It is clear that in either of these two cases, the optimal transmission structure might not be the shortest path tree. We show in Section III how the joint dependence of rates and path weights on the transmission structure actually makes our optimization problem NP-hard.

III. COMPLEXITY ANALYSIS AND APPROXIMATION ALGORITHMS

For the sake of simplicity and clarity in our arguments, and without loss of generality in the complexity analysis, we use in this section a simplified model for the data correlation, which allows a clearer analysis of complexity, and for which we develop efficient heuristic approximation algorithms. As we show in this work, this model still completely preserves the original complexity of the optimization problem. Namely, in our model, data at each node are entropy coded with $H(X_i) = R$ bits if no side information is available from other nodes, but only $H(X_i|X_{j_1}, \ldots, X_{j_k}) = r \leq R$ bits, $\forall k$, are needed if the node i has side information available coming from at least another node, which uses node i as relay. Thus, our simplification is that r is constant and does not depend on the number of nodes on which conditioning is done. For instance, this naturally approximates well the case when the correlation function between two nodes decreases with distance.[5] Namely, in this case conditioning on the closest neighbor results in the most important entropy reduction as compared to conditioning on more than one node.

Note that this simple coding strategy is easy to implement in a practical scenario. Sensor nodes have limited information processing capabilities, and prior knowledge of their correlation structure is assumed unknown. Suppose a transmission protocol needs to be implemented in the nodes, to perform information reduction by using conditional entropy. Thus, one possible simplification is to consider fixed block size appendices (in our setting, of size r) to the forwarded packet, that accommodate the average expected entropy reduction, regardless of the position of the node. The entropy reduction can be performed by simple look-up tables once the correlation is learned.

For the rest of this paper, we denote by $\rho = 1 - r/R$ the correlation coefficient.

---

[3]Denote by $H(X)$ the entropy of a discrete random variable X. The entropy is a measure of uncertainty of a random variable [6]: $H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x)$, where $\mathcal{X}$ is the discrete alphabet of X.

[4]However, this is not needed in the Slepian–Wolf scenario, since nodes need not explicitly communicate to coordinate for coding with joint entropy rather than the sum of individual entropies [7].

[5]For instance, the *valid* correlation models $K_{ij} = \exp(-\alpha d_{ij}^\beta)$, $\beta \in \{1, 2\}$ are widely used in spatial statistics. In these models, correlation decreases rapidly with distance.
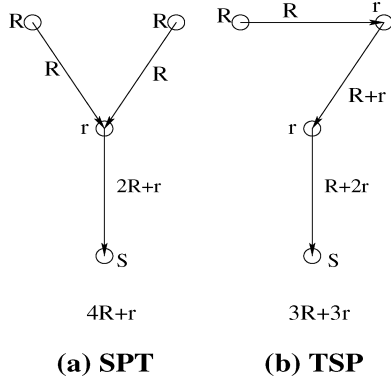
(a) SPT   **(b) TSP**

Fig. 2. All edges have length 1. The TSP (b) outperforms the SPT (a) if $R > 2r$.
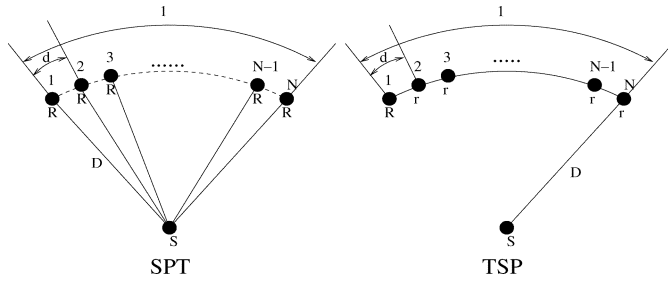

Fig. 3. SPT versus TSP.

A. Tradeoff Between Shortest Path Tree and Traveling Salesman Path

In the case of uncorrelated data, if the cost for transmitting over an edge is proportional (by a fixed constant) to the Euclidean length of that edge, then the problem is trivial and the optimal communication structure is the edge connecting the node to the sink. However, for an arbitrary weight function on the edge, transmitting via relays may be better than direct transmission (for example, if the edge weight is d^ν, $\nu > 1$). In the case of correlated data, as it is the case in sensor networks, things become even more interesting, even for very simple networks, because the rates $\{R_i\}_{i=1}^N$ are affected by the choice of the transmission structure.

The example in Fig. 2 shows that even in simple network cases, finding good correlated data gathering structures is not trivial at all. If the data were independent, the shortest path tree (SPT) would be optimal [see Fig. 2(a)]. However, we see that in this example, if $\rho > 1/2$, the SPT is no longer optimal, since its cost is larger than the one corresponding to the gathering tree in Fig. 2(b).

Fig. 3 shows one other simple network example, with N nodes equally spaced on an unit length arc circle at distance D from the sink. It is straightforward to show that $\lim_{N \to \infty}(c_{TSP}/c_{SPT}) = (1 - \rho)((1/2D) + 1)$, where c_{TSP}, c_{SPT} are the total flow costs of the two corresponding trees. Consider the case when the number of nodes is very large and the correlation coefficient is arbitrarily close to unity. This means that a path passing through all the nodes and ending at the sink (a traveling salesman path, TSP) can be arbitrarily

more cost efficient than the direct transmission which corresponds to the SPT in this case.

From these simple examples, it can be seen that the correlated data gathering problem with explicit communication is actually a hard optimization problem, in general. Formally, in terms of graph optimization, we can rewrite the minimization of (2) for the case of explicit communication as follows:

- **Given**: graph (V, E).
- **Find**: the spanning tree $T = \{L, I\}$ with L being the set of leaves, I being the set of nonterminal nodes, $L \cup I = V$, $L \cap I = \emptyset$.
- **such that**:

$$T = \arg \min_{\{L,I\}} \left(r \sum_{j \in I} d_T(j, S) + R \sum_{l \in L} d_T(l, S) \right)$$

where $d_T(i, S)$ is the total weight of the path on the T tree from node i to the sink S.

In terms of the correlation coefficient, $\rho = 1 - r/R$:

$$T = \arg \min_L \left((1 - \rho) \sum_{i \in V} d_T(i, S) + \rho \sum_{l \in L} d_T(l, S) \right). \quad (3)$$

Let us first look at the two extreme cases, that is $\rho \to 0$ and $\rho \to 1$. When $\rho \to 0$ (independent data), the optimal tree is the SPT, which is known to be solvable in polynomial time by, e.g., a distributed Bellman–Ford algorithm. At the other extreme, when $\rho \to 1$ (data maximally correlated), the optimal solution is a spanning tree for which the sum of paths from the leaves to the sink is minimum. For this, the core information is taken from the leaf nodes, and passing through all the in-tree nodes only adds an infinitesimally small amount of new information, since data is strongly correlated. It is straightforward to show that solving this problem is equivalent to solving the *multiple traveling salesman* optimization problem $(k - TSP)$ [18], which is known to be NP-hard.

To the best of our knowledge, (3) is an original spanning tree optimization problem on a graph. In Section III-B, we show that this problem is also NP-hard for the general case $0 < \rho \leq 1$. However, it is possible to design good approximation algorithms and we provide them in Section IV.

B. NP-Completeness

In order to prove the NP-hardness of the optimization problem given in (3), we show that the decision version of the problem is NP-complete. The decision version of our optimization problem is as follows.

Definition 1: Network data gathering tree cost decision problem:

- *Instance*: An undirected graph $G = (V, E)$ with weights w_e assigned to the edges $e \in E$, a positive integer M, and a particular node $S \in V$.
- *Question*: Does the graph admit a spanning tree T such that, when assigning supplies $R_i = R$ to the leaf nodes and $R_i = r < R$ to the in-tree nodes of T, the total cost of T given by (3) is at most M?

Theorem 1 (NP-Completeness): There is no polynomial time algorithm that solves the network data gathering tree cost problem, unless $P = NP$.

Proof: See Appendix A (we use a nontrivial reduction from the min-set cover problem).

Arbitrary Function of Rate: Note that the NP-completeness result holds for any monotonically increasing function of the rate, since an arbitrary function only modifies the values of R and r, but not the multiplicative separable form of the cost function.

In general, node i has information from all the nodes in the subtree $sbt(i)$ rooted at node i. Our simplified model is a particular case of this general entropy coding problem, where $H(X_i|\{X_j\}, j \in sbt(i))$ is approximated with $H(X_i|X_j)$, with j being a child of i. Then it can be shown easily that the NP-complexity of our simplified example extends also to this more general case by means of a trivial further reduction. Namely, since (3) is a particular version of the general problem (2), it follows by a trivial reduction that (2) is also NP-hard.

Corollary 1: Minimizing $\sum_{i \in V} R_i d_T(i, S)$, with R_i a monotonically increasing function of the sum of flows incoming at node i, is NP-hard.

The NP-hardness of the problem for a single sink generalizes by a straightforward reduction to the case of multiple sinks. However, the derivation of approximation algorithms for the multiple sinks case is significantly more difficult, due to the interactions between the approximated structures derived for each single sink in particular. This is because nodes that are leaves for a particular structure can be in-tree nodes for other structures, and thus their corresponding rate allocation cannot be uniquely determined.

C. The Dual Problem: NP-Completeness of Broadcast of Correlated Data

The problem formulation for correlated data tree broadcast is also essentially provided by the simplified problem (3), with the difference that now $r > R$, that is, the amount of forwarded data diminishes as it is broadcast from a source node to the extremities of the network. The outer (leaf) nodes can use the data from their parent nodes to fully reconstruct their own data. Thus, in general, relays only need to send further to their children an amount of data equal to the entropy of their corresponding children, conditioned on their own measured data.

Proposition 1: There is no polynomial time algorithm that solves the dual problem of correlated data broadcast (namely, $r > R$ in Theorem 1), unless $P = NP$.

Proof: See Appendix B.

IV. HEURISTIC APPROXIMATION ALGORITHMS

In this section, we introduce a set of approximation algorithms for solving problem (3).

A. Shortest Path Tree

SPT is computed by using the distributed Bellman–Ford algorithm [4] for simultaneously determining the shortest paths from all nodes to the sink. If the data is independent, this is the optimum solution, but it is far from optimal if there are high correlations.

B. Greedy Algorithm

We start from an initial subtree composed only of the sink node. Then, we add successively, to the existing subtree, the node whose addition results in the minimum cost increment. As expected, given the relationships between the problem in this paper and the TSP problem, greedy algorithms perform suboptimally, in the same way as the greedy approximation algorithm for TSP provides a quite suboptimal solution. The reason is that far nodes are being left out, so they need to connect to the sink via a path with large weight.

C. Simulated Annealing

We propose now a computationally heavy method which is known to provide results that are close to optimal for combinatorial problems involving a large number of variables, similar to the problem considered in this paper (e.g., TSP). This method was inspired by the *fitness landscape* concept used in evolutionary biology, physics of disordered systems, and combinatorial optimization [26]. The fitness landscape formulation of our problem is as follows: (a) the configuration space is the set of all spanning trees (completely defined by the parent relationship); (b) the move set is: one node changes its parent; (c) the fitness function is $g(T) = R \sum_{l \in L} d_T(l, S) + r \sum_{j \in I} d_T(j, S)$. Our goal is to minimize the fitness over the set of spanning trees.

A very general heuristic optimization method is simulated annealing (SA) [13]. It gives very good results when applied to another NP-hard combinatorial problem in graphs, the traveling salesman problem (TSP) [18], [26].

Algorithm 1 *Simulated annealing:*
- *Take a monotonically decreasing cooling schedule $h[k]$, $k = 1, \ldots, K$. Initialize parent nodes with $par(i) = S$, $i = 1, \ldots, N$. Denote by $\mathcal{N}(i)$ the set of one-hop neighbors of i.*
- **While** $k < K$
  - *$k = k + 1$, $l = g(T)$;*
  - *choose $i, j \in \mathcal{N}(i)$ at random, such that deleting edge $(i, par(i))$ and adding edge (i, j) to the tree, does not form a cycle; let T' be the newly generated spanning tree and let $l' = g(T')$ be its corresponding fitness.*
  - *make the change $par(i) \leftarrow j$, and assign $T \leftarrow T'$ with probability*

$$p = \begin{cases} 1, & \text{if } l' \leq l \\ \exp\left(-\frac{l'-l}{h[k]}\right), & \text{if } l' > l. \end{cases}$$

- **Endwhile**.

For $\rho = 0$ (SPT), our experiments show that it does provide the exact solution, and convergence is easy to achieve. When ρ is close to 1, the generated landscape is not smooth any longer, so convergence is difficult to obtain in a reasonable number of iterations. We obtained good results (iteration steps

versus ruggedness) with the Lundy and Mees schedule [20]: $h[k] = (h[k-1]/(1 + ((h[0] - h[K])/Kh[0]h[K])))$.

Simulated annealing is usually hard to implement in a decentralized manner, and is computationally expensive. It does, however, provide a good benchmark close to optimal against which other heuristic algorithms can be tested.

D. Balanced SPT/TSP Tree

We propose a heuristic approximation algorithm consisting of a combination of SPT and $k - TSP$, inspired from the solutions obtained using simulated annealing. The solution provided by this algorithm consists of an SPT structure around the sink that has a certain radius and a set of TSP paths starting from each of the leaves of the SPT [see Fig. 12(b) for an example]. Depending on the amount of correlation, that is the value of ρ, a certain radius for the SPT is more appropriate. We briefly describe the intuition why there is such a value for this radius. Since the leaf nodes contribute most to the cost ($R > r$), then in order to minimize the cost, the flows of R data coming from the leaves of the tree have to travel short paths to the sink (the SPT effect), but in the same time through as many nodes as possible, to reduce the total number of leaves (the TSP effect). On the other hand, when the correlation is large (r is small), the effect of transporting flows of r data through the tree is negligible, so it is essential to have as many in-tree nodes as possible, thus the TSP effect is more important; when the correlation is small (r is large), it is more important that the data from in-tree nodes reach the sink on shortest paths, and thus the SPT effect becomes more pronounced.

Algorithm 2 SPT/TSP balanced tree:
- *Build the SPT for the nodes that are in a radius $q(\rho)$ from the sink. Denote this SPT by T. The optimal choice for the radius $q(\rho)$ decreases with the increase of the correlation coefficient ρ. Let V_T denote the nodes in T. Let $V_{TS} = V \setminus V_{SP}$.*
- **While $V_{TS} \neq \emptyset$**
  - *Denote by L the set of leaves of T.*
  - *$\{i_0, l_0\} = \arg\min_{\{i \in L, l \in V_{TS}\}} (d(l, i) + d_T(i, S))$*
  - *$T = T \cup (i_0, l_0)$, $V_T = V_T \cup \{i_0\}$, $V_{TS} = V_{TS} \setminus \{i_0\}$.*

The second part of Algorithm 2 is actually a suboptimal nearest neighbor approximation of the $k - TSP$, which is easily implementable in a distributed manner.

Square Grid Network Graph: Optimal Radius for the SPT/TSP Algorithm: Since the TSP problem is NP-complete, it is difficult to provide an analytical study of the dependence of the optimal SPT radius on the correlation structure for a general connectivity graph. Therefore, for analysis, we restrict our attention to a square grid graph and study in detail the structure of our SPT/TSP algorithm in this case. Namely, we study the dependence of the optimal radius on the correlation coefficient ρ.

Consider a square grid network with $(2n+1) \times (2n+1)$ nodes (see Fig. 4). The SPT is built on the square area of $(2m+1) \times (2m+1)$ nodes around the sink. Note that the SPT subtree has $8m$ leaves. For the rest of the graph, equal length TSP paths are built. Namely, for each leaf of the SPT subtree, a TSP rooted
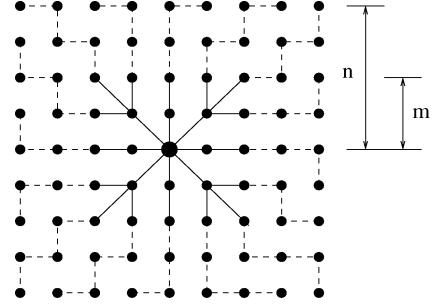

Fig. 4. Square grid network: the SPT (solid lines) is built on the nodes in the $m \times m$ sub-grid around the sink (larger black dot). The rest of the nodes are spanned by TSPs (dashed lines) rooted in the leaves of the SPT.
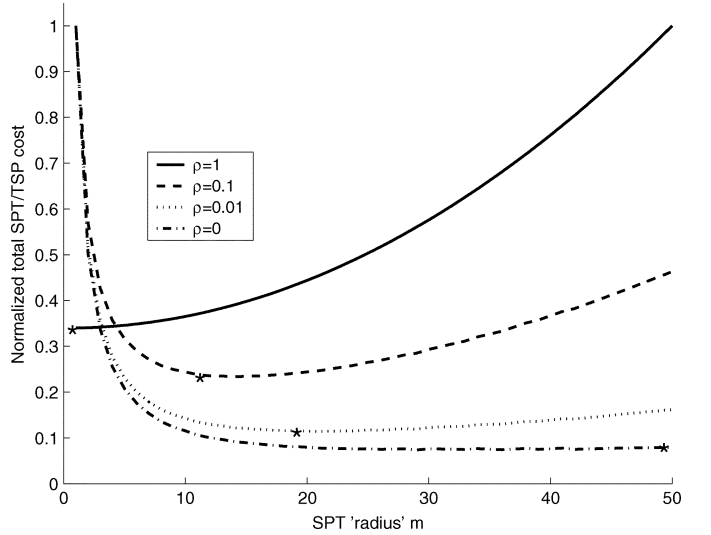

Fig. 5. Square grid network: normalized cost of the SPT/TSP tree for a grid network of size $N = 101 \times 101$ nodes ($n = 50$) and several values of the correlation coefficient ρ. Note how the optimum value of the radius m increases from 0 to n as ρ decreases from 1 (high correlation) to 0 (no correlation).

at that leaf node is constructed, which spans $\texttt{floor}\,((2n+1)^2 - (2m+1)^2/(8m))$ of the nodes left outside the SPT subtree.

We plot in Fig. 5 the total cost of the SPT/TSP tree as a function of the correlation coefficient ρ. As expected, the optimal SPT radius m decreases with the increase of the correlation coefficient ρ.

Next, we compute analytically the optimal "radius" m/n of the SPT subtree as a function of the correlation coefficient $\rho = 1 - r/R$. After some computations, we obtain that the optimal m is a root of the following polynomial: $P(Z) = 3rZ^4 + (8r - 16R)Z^3 + (-r - 4rn + 4R - 4rn^2)Z^2 + (rn^4 + 2rn^3 + rn^2)$. This polynomial has four roots, but by solving it numerically, we find that only one of them is in the interval $[0, n]$. We plot this solution for the optimal radius in Fig. 6. The discontinuity at $\rho = 0$ is due to the properties of the very particular regular grid structure that is analyzed. A particular interesting abrupt phenomenon is observed asymptotically: when n is sufficiently large, there is an optimal normalized radius for the SPT, which does not depend on the correlation coefficient ρ.

E. Leaves Deletion Approximation

This algorithm is a simplified version of the TSP/SPT algorithm. Namely, this algorithm constructs first the global SPT,
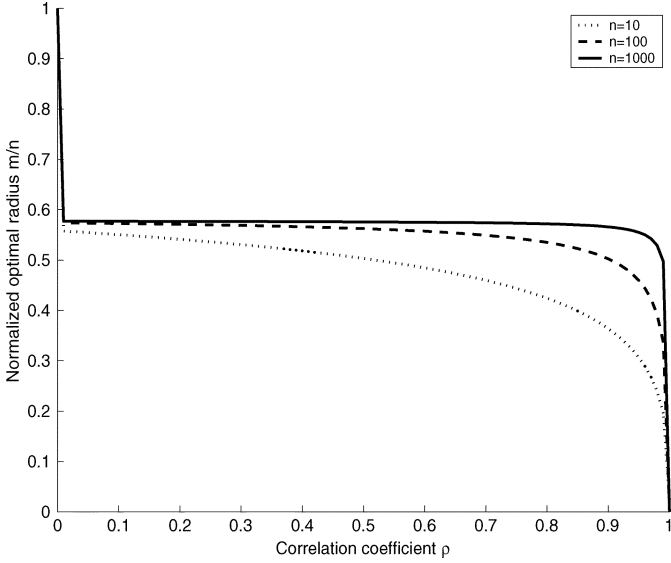
Fig. 6. Square grid network: optimal radius of the SPT (normalized with respect to the radius of the square grid), as a function of the correlation coefficient ρ, for various sizes $N = (2n + 1)^2$ of the network.

and then uses one-hop TSP paths from the outer nodes of the SPT. It is based on the observation that good cost improvements may be obtained mainly by making the leaf nodes change their parent node to some other leaf node in their neighborhood. This operation is done only if it reduces the total cost of the whole tree.

Algorithm 3 *Leaves deletion algorithm (LD):*
- *Initialize $T \leftarrow SPT$. Each node i maintains its parent, number of children, and total distance $d_T(i, S)$ on the current spanning tree to the sink. Let $par(i)$ denote the parent node of node i.*
- **While** *there is a decrease in cost:*
  - **For each** *leaf node i: Find the leaf node $j \in \mathcal{N}(i)$ that maximizes $R(d_T(i, S) + d_T(j, S)) - (R(d_{i,j} + d_T(j, S)) + r d_T(j, S)) - A(i)$, where $A(i)$ is an adjustment term indicating the cost lost by transforming single parent nodes into leaves. If the maximizing quantity is positive, then assign $par(i) \leftarrow j$ and update the corresponding distances on the tree to the sink, and number of children, for all the three nodes involved $\{i, former\ par(i), j\}$.*
- **Endwhile**.

This algorithm involves a small number of iterations after SPT is computed, and is fully distributed. Note that a known good approximation for the geometric TSP is to start from the minimum spanning tree (MST) and eliminate the leaves by successively passing the traveling salesman path through them. Here, we can see that a simplified similar procedure provides good results in our case as well, which confirms the link between our problem and the TSP.

V. STRICT APPROXIMATION ALGORITHMS

In this section, we present a strict approximation algorithm, which is an algorithm that guarantees a solution for which the
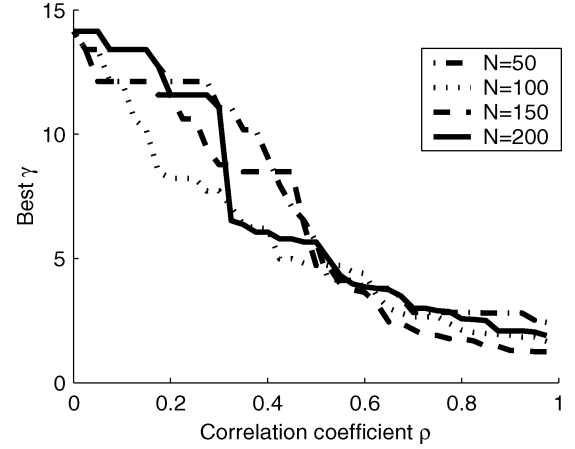

Fig. 7. Best choice of the parameter γ, as a function of the correlation coefficient ρ. The average has been done over 20 random network instances for each pair (ρ, N).

cost is only a constant factor higher than the cost of an optimal solution. We start this section by giving a lower bound on the cost of an optimal solution.

Lemma 1 (Lower Bound): The cost of the optimal solution c_{opt} is bounded from below by $c_{opt} \geq \max(r \cdot c_{SPT}, R \cdot c_{MST})$, where c_{SPT} is the sum of the costs of all the shortest paths to the sink, and c_{MST} is the cost of the minimum spanning tree of all the nodes, including the sink.

Proof: Nodes in the network can either send their raw data directly to the sink, or use the raw data of other nodes to code their data, and then send their coded data to the sink. Let the nodes who send their data in the raw format be the set B. Let the nodes who code their data using the raw data of node u be the set C_u. The set B and the sets C_u for all $u \in V$ form a partition over all nodes, that is, $V = \{B, \cup_{u \in B} C_u\}$.

After deciding how the set of nodes will be partitioned, an optimal algorithm will use the shortest paths (SP) to deliver the raw data from nodes in B to the sink. Similarly, the encoded data of nodes in set C_u will travel along shortest paths (SP) to the sink. Nodes from C_u need to encode their data using the raw data of node u, u being a node in set B. On the other hand the sink needs to decode the encoded data of nodes from C_u; to do so, the sink needs the raw data of node u too. The optimal way to distribute the raw data of u is given by the MST between the nodes in the set C_u, node u itself, and the sink. Summing up, the cost of the optimal algorithm is therefore

$$c_{opt} = \sum_{u \in B} R \cdot |SP(u, sink)|$$
$$+ \sum_{u \in B} \left(R \cdot |MST(C_u, u, sink)| + \sum_{v \in C_u} r \cdot |SP(v, sink)| \right).$$

We can bound this equation in two ways from below. Since the sets B and C_u form a partition of all nodes V, and since $r \leq R$, each node must transmit its data to the sink on the shortest path, at least in the coded form. Therefore, the optimal
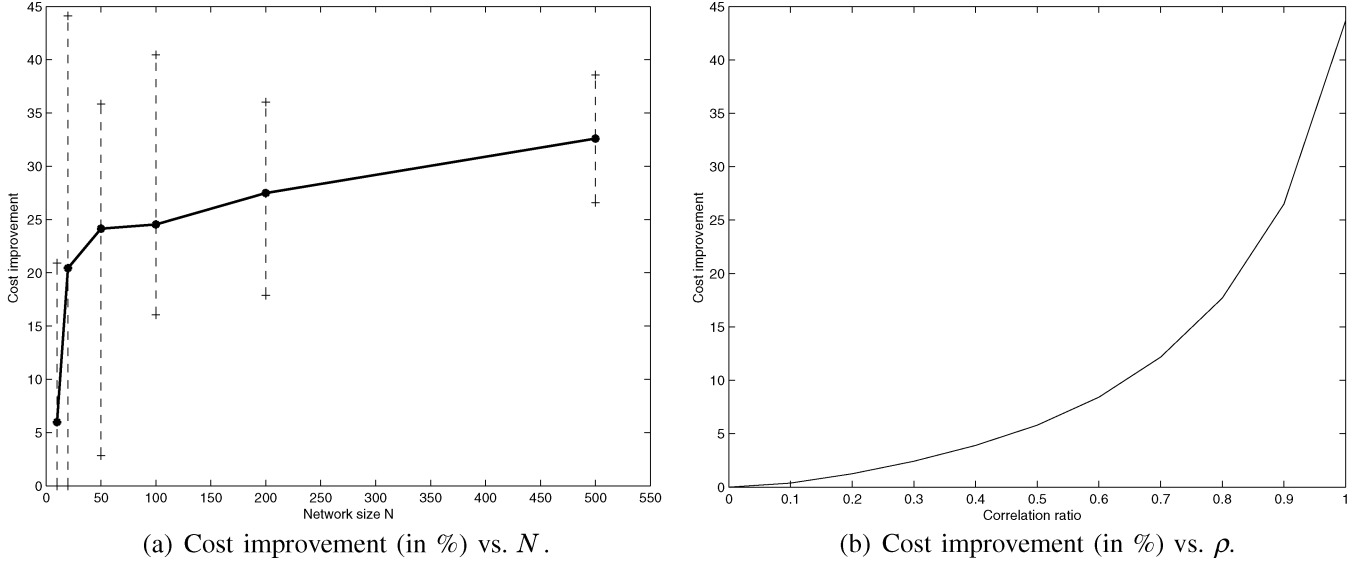
(a) Cost improvement (in %) vs. N.

(b) Cost improvement (in %) vs. ρ.

Fig. 8. Average total cost decrease $100 \cdot ((c_{SPT}/c_{LD}) - 1)$, in %, of leaves deletion (LD) over shortest path tree (SPT) for (a) $\rho = 0.9$ and $N = 10, 20, 50, 100, 200, 500$, and (b) $N = 200$ and $\rho = 0, 0.1, \dots, 1$.

cost contains at least the sum of the shortest paths (SPT) of the coded data:

$$c_{opt} = \sum_{u \in B} R \cdot |SP(u, sink)|$$

$$+ \sum_{u \in B} \left(R \cdot |MST(C_u, u, sink)| + \sum_{v \in C_u} r \cdot |SP(v, sink)| \right)$$

$$\geq \sum_{u \in B} r \cdot |SP(u, sink)| + \sum_{u \in B} \sum_{v \in C_u} r \cdot |SP(v, sink)|$$

$$= \sum_{u \in V} r \cdot |SP(u, sink)| = r \cdot c_{SPT}.$$

On the other hand, since B and C_u form a partition of all nodes V, the terms containing raw data (R) must include a spanning tree. Since the MST is the cheapest possible spanning tree, the cost of the optimal algorithm is also bounded from below by the cost of the MST, used to transmit the uncoded data. The lemma follows immediately. ∎

In the following, we present an approximation algorithm that is optimal up to a constant factor. The algorithm is based on the *shallow light tree* (SLT), a spanning tree that approximates both the MST and the shortest paths for a given node (e.g., the sink). The SLT was introduced in [2] and [5]. Given a graph $G(V, E)$ and a positive number γ, the SLT has two properties:

- its total cost is at most $1 + \sqrt{2}\gamma$ times the cost of the MST of the graph $G(V, E)$;
- the distance on the SLT between any node in V and the sink is at most $1 + \sqrt{2}/\gamma$ times the shortest path from that node to the sink.

For more details on the construction of the shallow light tree (SLT), refer to [16].

The algorithm is as follows. First, the SLT spanning tree is computed, the sink being the root of the SLT. Then the sink broadcasts its value R_{sink} to all its one-hop neighbor nodes in the SLT. When node v is receiving a value R_u from a neighbor u, node v encodes its locally measured data R_v using R_u, and transmits its encoded value r_v to the sink on the path given by

the SLT. Then node v broadcasts its value R_v to all its one-hop neighbors but u; in other words to all its children but not its parent in the SLT. We call this the SLT algorithm.

The sink has its own data R available locally (or it can use the R data of its first-hop neighbors), and thus can perform recursive decoding of the gathered data, based on the encoded r values that it receives from all the nodes.[6]

Theorem 2: The SLT algorithm is a $2(1 + \sqrt{2})$-approximation of (3).

Proof: The total cost of the SLT algorithm is given by

$$c_{SLT} = R \cdot |SLT| + \sum_{v \in V} r \cdot |Path_{SLT}(v, sink)|.$$

The first term follows from each node sending its raw data to all its children in the SLT. The second term corresponds to the sum of the paths in the SLT. Using the SLT properties we have $c_{SLT} \leq R \cdot (1 + \sqrt{2}\gamma)c_{MST} + r \cdot (1 + \sqrt{2}/\gamma)c_{SPT}$. We choose $\gamma = 1$, with $\alpha = R \cdot c_{MST}$ and $\beta = r \cdot c_{SPT}$. Then

$$c_{SLT} = (1 + \sqrt{2})(\alpha + \beta).$$

Dividing c_{SLT} by $c_{opt} = \max(\alpha, \beta)$ as derived in Lemma 1, the second factor of c_{SLT} will be upper bounded by 2, and the approximation ratio will consequently be $(1 + \sqrt{2}) \cdot 2 \approx 4.828$. This ratio becomes tight at $\alpha \approx \beta$; if $\alpha \gg \beta$ or $\beta \gg \alpha$ the approximation ratio of the SLT algorithm is better. ∎

Thus, SLT provides a worst-case bound for our problem. Fig. 7 shows the best choice of γ for the SLT, found experimentally, as a function of the correlation coefficient ρ. Note that when the data is independent $(\rho \to 0)$, the optimal choice of γ is large; that is, the SLT is close to the SPT. On the contrary, for high correlation $(\rho \to 1)$, a good SLT should be close to the MST (value of γ close to 1), and the MST is known to approximate the TSP within a constant. These results for the

---

[6]In a practical scenario, this corresponds to a powerful base station which broadcasts to the nodes a predicted message R, based either on previous reports from the nodes, or on the own measurement of the sink. The nodes only need to send back adjustments r, by conditioning on the side information (for example, in a temperature measuring scenario, the sink broadcasts its measurement and nodes only need to feedback the deviations from this measurement).
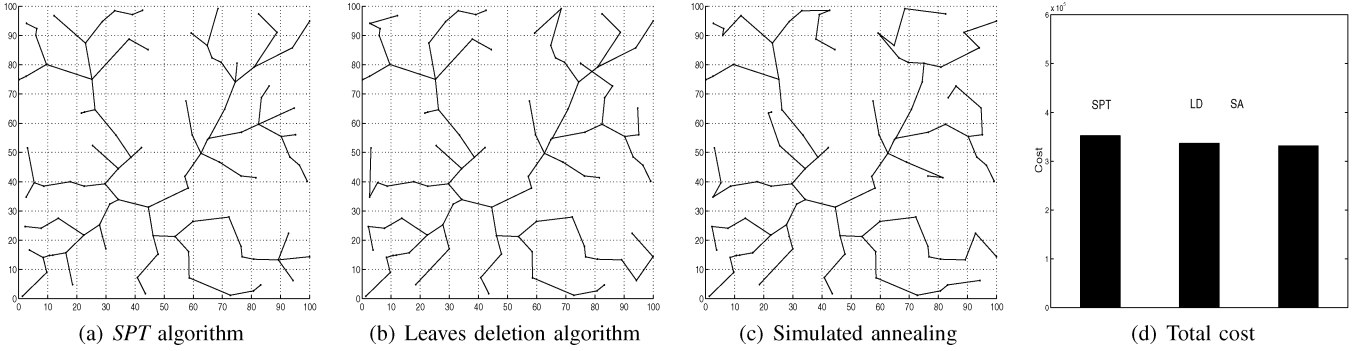
(a) *SPT* algorithm   (b) Leaves deletion algorithm   (c) Simulated annealing   (d) Total cost

Fig. 9.   Data gathering tree algorithms on a network instance: $N = 100$, $\rho = 0.5$: (a) shortest path tree (SPT), (b) leaves deletion (LD), (c) simulated annealing, (d) total flow cost. Costs for this instance: $SPT: \; 3.52\mathrm{e}+6$; $LD: \; 3.36\mathrm{e}+6$; SA: $3.31\mathrm{e}+5$.

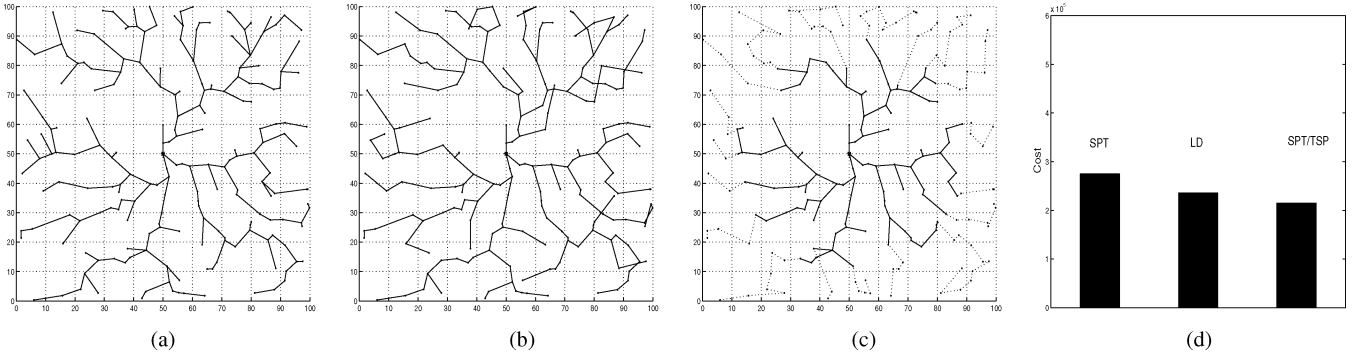(a)   (b)   (c)   (d)

Fig. 10.   Approximated gathering trees on a network instance: $N = 200$, $\rho = 0.2$: (a) shortest path tree (SPT), (b) leaves deletion (LD), (c) SPT/TSP algorithm, (d) total cost. Costs for this instance: $SPT: \; 2.74\mathrm{e}+5$; $LD: \; 2.36\mathrm{e}+5$; $SPT/TSP: \; 2.15\mathrm{e}+5$.

best choice of the parameter γ for the SLT approximation are as expected, following our discussion in Section III.

VI. NUMERICAL SIMULATIONS

Our simulations were done in MATLAB for a network of nodes randomly distributed on a 100×100 grid, with a value $\nu = 2$ for the power of the distance. We consider several sizes of the network, from $N = 10$ up to $N = 500$ nodes, and various values for the correlation coefficient ρ among the nodes, within the interval $\rho \in [0, 1]$. As mentioned before, the algorithm that is used for finding the SPT in a distributed manner is a distributed version of the Bellman–Ford algorithm, which runs in $\mathcal{O}(N|E|)$ steps. The actual speed of convergence depends on the degree of each node in the graph, which in turn depends on the range $\mathcal{N}(i)$ over which nodes search for neighbors. For the graph structures we consider, Bellman–Ford runs in an average of 50 steps for a network size of 500 nodes.

Our experiments show important average improvements of the LD algorithm over the SPT for nodes randomly distributed on a 100×100 grid (see Figs. 8–10). The computational load of LD is small, namely at most four iteration steps after the SPT are required for its implementation, while the algorithm is still distributed.

When comparing the various heuristic algorithms with the simulated annealing solution, which is expected to provide results close to optimal, we notice that our simple heuristic algorithms perform relatively well, while being completely distributed, scalable and efficient from a complexity point of view (see Fig. 9).
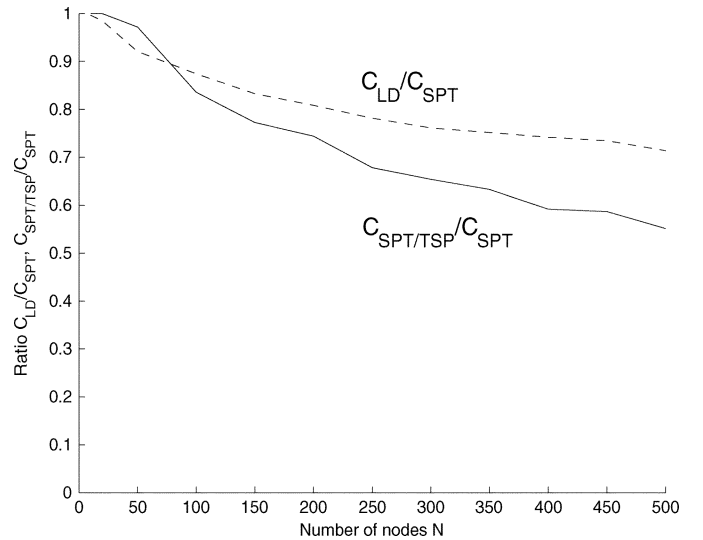

Fig. 11.   Average ratios of total costs between leaves deletion (LD) and SPT, and between balanced SPT/TSP and SPT: $\rho = 0.9$.

Solutions for typical network instances are shown in Figs. 9 and 10. We show in Fig. 10 some simulation results for the SPT/TSP algorithm. In Fig. 10(c), we plot the branches in the SPT subtree in solid lines, and the branches added in the step involving TSP paths are shown in dashed lines. For networks with $\rho = 0.2$ and $N = 200$, the improvements are of the order of 10% over the LD algorithm. Our extensive simulations show important improvements of the LD and the SPT/TSP algorithms over SPT, in terms of average performance over randomly generated networks (see Fig. 11).

For illustrative purposes, we show in Fig. 12 the SLT tree and the SPT/TSP tree for a network instance with $N = 100$.

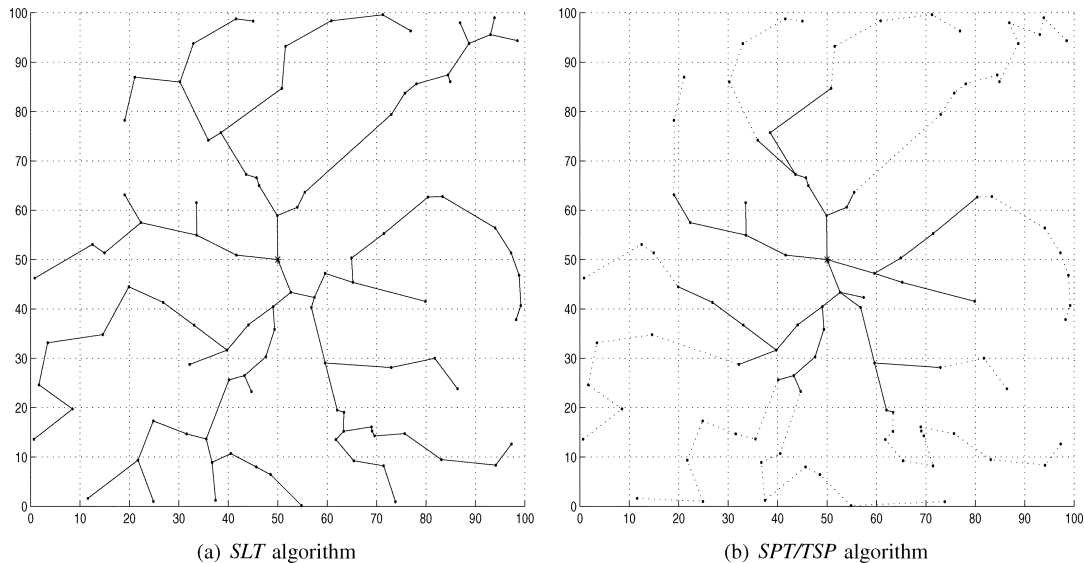(a) *SLT* algorithm   (b) *SPT/TSP* algorithm

Fig. 12. Approximated gathering trees on a network instance: $N = 100$, $\rho = 0.8$: (a) shallow light tree (SLT), (b) SPT/TSP algorithm. Costs for this instance: SLT: $1.79\mathrm{e} + 005$; SPT/TSP: $1.55\mathrm{e} + 5$.
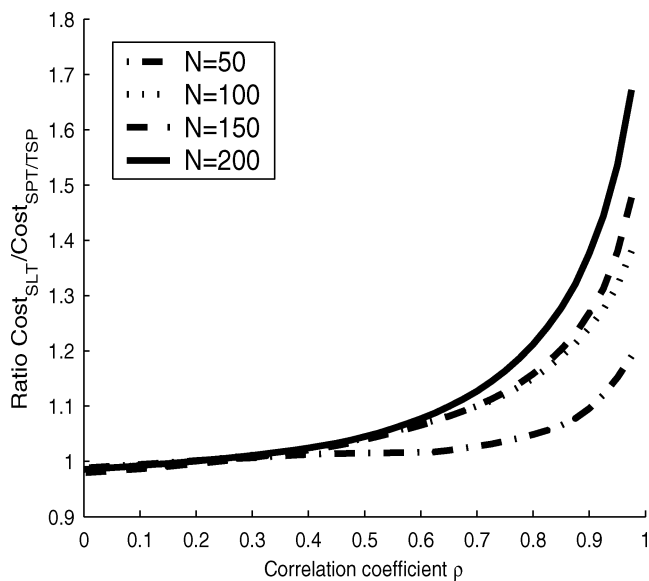

Fig. 13. Average ratio of the SLT cost versus the SPT/TSP cost. The average has been done over 20 random network instances for each pair (ρ, N), with values for γ as in Fig. 7.

In terms of total cost, as expected, from an average case point of view, the SPT/TSP algorithm performs better than the SLT algorithm (see Fig. 13). In these results, the value of the radius $q(\rho)$ for the SPT/TSP has been chosen as in Fig. 6, and for the SLT, the value of γ has been chosen as in Fig. 7. Note that for small values of the correlation coefficient $\rho \approx 0$, the two trees perform similarly, since both algorithms provide solutions close to the SPT, which is the optimal solution when there is no correlation in the data. When the correlation coefficient ρ approaches 1, the SLT provides a solution close to the MST, and thus, the ratio between the costs provided by the two algorithms shows how well the MST approximates the TSP. Namely, the MST provides a constant approximation for the TSP in the worst case, while, by design, the SPT/TSP algorithm searches for better approximations for the TSP.

VII. CONCLUSIONS

In this paper, we formulate the network correlated data gathering tree problem with coding by explicit communication. Namely, we address an optimization problem that considers transmission structure optimization in networks where connectivity is modeled as a graph. A transmission tree structure implies both a certain rate allocation at the nodes and a certain transmission cost per bit between connected nodes. We first proved that the problem is NP-hard even for scenarios with several simplifying assumptions. Further, we have found an algorithm that provides a constant approximation for our problem. We also propose approximation algorithms for the transmission structure that show significant gains over the shortest path tree. Moreover, our algorithms provide solutions close to the optimal, which is shown by comparing in simulations our approximation algorithms to a provably optimal but computationally heavy optimization method, namely, simulated annealing.

APPENDIX A
PROOF OF THEOREM 1

First, the decision version of our problem is in NP: a nondeterministic algorithm needs to guess the parent relationship (that is, specify the parent node for each of the nodes), and then find in polynomial time the nodes that are not parent nodes, assign to all nodes the number of bits corresponding to either leaf or in-tree node, and test that its total cost is less than the given value M.

Next, to prove the NP-hardness, we perform a reduction from the set cover problem [10], whose decision version is defined as follows:

Definition 2: Set cover.

- *Instance*: A collection C of subsets of a finite set P and an integer $0 < K \le |C|$, with $|C|$ the cardinality of C.
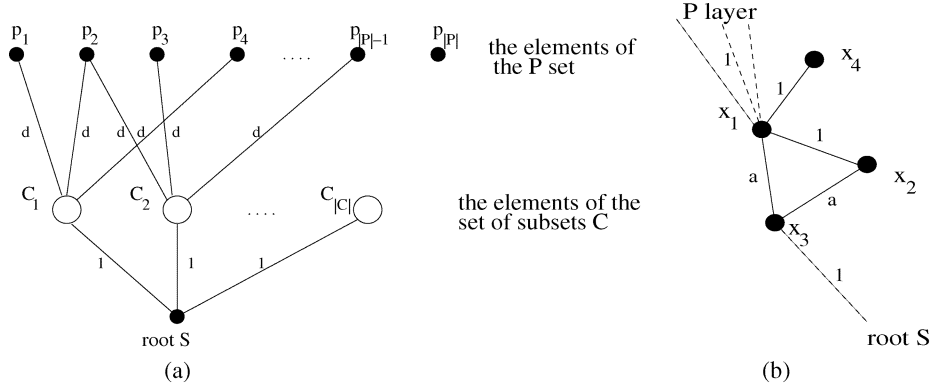
Fig. 14.   Reduction from the min-set cover problem. (a) A graph instance. (b) Inner structure of each subset C_i.

- *Question*: Does C contain a subset $C' \subseteq C$ with $|C'| \leq K$, such that every element of P belongs to at least one of the subsets in C' (this is called a set cover for P)?

For any instance of the set cover problem, we build an instance of our decision problem. Fig. 14(a) illustrates the construction of the graph instance for our problem. The resulting graph is formed of three layers: a sink node S, a layer corresponding to the subsets $C_i \in C$, and a layer corresponding to the elements $\{p_j\}$ of the set P. For each element $C_i \in C$ we build a structure formed by four nodes x_1, x_2, x_3, x_4, as in Fig. 14(b) (there are four different nodes for each subset C_i, but we drop the superscript C_i of the nodes x for the sake of simplicity). This structure originates from our toy example in Section III-A and has properties linked with the tradeoffs observed there. The node x_3 is linked to the sink S, node x_4 is connected only to node x_1, and x_1, x_2, x_3 are all interconnected. Furthermore, we connect each structure $C_i \in C$ (namely the node x_1 from that structure) to only the nodes in the P layer that correspond to elements contained in C_i (example: in the instance in Fig. 14(a), subset $C_1 = \{p_1, p_2, p_4\}$, $C_2 = \{p_2, p_3, p_{|P|-1}\}$ etc.) All the edges connecting the P layer to the C layer have a weight $d > 0$; for all C_i, the edges of type (x_1, x_3) and (x_2, x_3) have weight $a \geq 1$; the rest of the edges shown in Fig. 14(a) have all weight 1. All other edges are assumed of infinite weight and are not plotted. Without loss of generality, we consider that in-tree nodes use $r = 1$ bits for coding their data, while leaf nodes use $R > 1$ bits.

The goal is to find a spanning tree for this graph, for which the cost in (3) is at most M. We now show that if $M = |P|(d + a+1)R+K(2aR+3R+a+2)+(|C|-K)(aR+3R+2a+4)$, for the positive integer $K \leq |C|$, then finding a spanning tree with cost at most M is equivalent to finding a set cover of cardinality K or less for the set P. Notice that the construction of our graph instance from the set cover instance can be performed in polynomial time.

With a large enough value chosen for d (i.e., $d > |C|(2aR + 3R + a + 2)/R$), a tree with cost at most M will contain exactly $|P|$ links between the layers P and C. That means that no p_j node is used as relay, so all $p_j \in P$ are necessarily leaf nodes. If some p_j node was used as relay, then the cost of the tree would contain R bits passing through more than $|P|$ such links, which would result in a cost larger than M. This also implies that the only way the C_i structures can connect to the sink

S is via their corresponding x_3 node, so all x_3's must be in-tree nodes. Furthermore, all x_4's nodes need to connected to their corresponding x_1 node in order to belong to the tree, so necessarily all x_4's are leaf nodes and all x_1's nodes are in-tree nodes. The only degrees of freedom are the choices of two out of the three edges interconnecting the nodes x_1, x_2, x_3, for each structure C_i.

The key idea of our proof is that, for properly chosen values for d and a, finding a tree with cost at most M means connecting the nodes in layer P to *at most* K nodes of layer C. If the tree needs to connect the layer P to more than K nodes in layer C, then the cost of the tree will necessarily be higher than M. The intuition is that "detours" via the (x_1, x_2) edges are worthy from the point of view of cost reduction only if the flow that goes through node x_1 comes exclusively from node x_4 and no flow from the P layer goes through x_1. If some flow from the P layer joins as well, then the optimal path would use the edge (x_1, x_3) instead. In this latter case, we see now that for optimality, the edge (x_1, x_2) should not be used.

We choose a value of $a \geq 1$ such that $(a+2)/a < R < (a + 2)/(a - 1)$. Note that, for a given $R > 1$, it is always possible to choose a value for a that fulfills this condition.

With the given weights on the edges, if no p_j node is connected to a C_i structure, then since $R > (a + 2)/a$, the optimal pattern (pattern 1, see Fig. 15) for this structure contains the links (x_4, x_1), (x_1, x_2), (x_2, x_3), (x_3, S), with cost $(a + 3)R + (a + 2) + (a + 1) + 1$. The other possible structures contain either links (x_4, x_1), (x_1, x_3), (x_2, x_3), (x_3, S) (pattern 2) with cost $(a + 2)R + (a + 1)R + (a + 1) + 1$, or links (x_4, x_1), (x_1, x_3), (x_2, x_1), (x_3, S) (pattern 3) with cost $(a+2)R+(a+2)R+(a+1)+1$. They both are sub-optimal if $R > (a + 2)/a$ (since pattern 2 is always better than pattern 3, we will consider only pattern 2 for the rest of our proof).

However, when $m \geq 1$ nodes $\{p_j\}_{j=1}^m$ from the P layer connect to x_1, for any of C_i's, the pattern 1 is no longer optimal, because it has a cost $m(d+a+2)R+(a+3)R+(a+2)+(a+1)+1$. The alternative structure (pattern 2) has cost $m(d + a + 1)R + (a + 2)R + (a + 1)R + (a + 1) + 1$, which is more efficient if $m \geq 1$, and $R < (a + 2)/(a - 1)$. We notice that in an optimal tree the cost to transmit data from each p_j to the sink S is the same for all p_j's nodes (and equal to $(d + a + 1)R$). Therefore, the goal is to keep minimal the part of the total cost corresponding to the rest of the nodes (i.e., nodes in layer C).
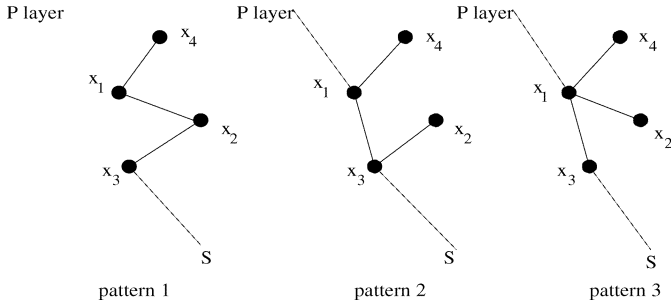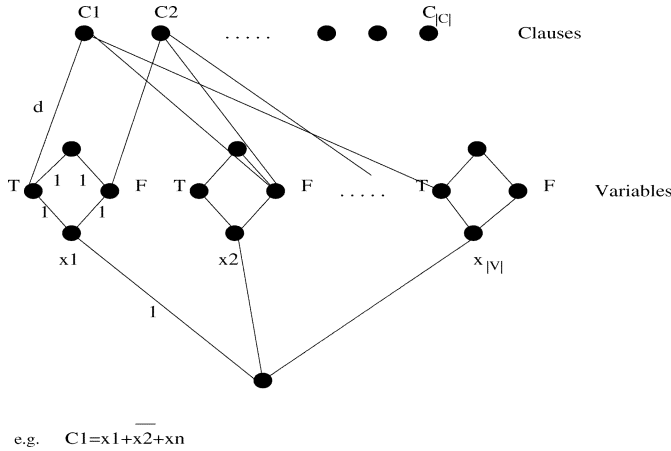
Fig. 15. The three possible gathering patterns for the substructure C_i.

Fig. 16. The reduction from any instance of 3-SAT to an instance of our problem.

That means that to find a tree with cost less or equal to $|P|(d + a + 1)R + K(2aR + 3R + a + 2) + (|C| - K)(aR + 3R + 2a + 4)$ is equivalent to finding a set of K elements or less from the C layer to which all nodes in the set P connect. This is actually achieved by having at most K nodes of type x_1 used to connect to the p_j's nodes, which turns out to be equivalent to finding a set cover for the set P of size K or less, that is to solving the set cover problem.

Thus, our decision problem is NP-complete and our optimization problem NP-hard.

APPENDIX B
NP-COMPLETENESS OF CORRELATED DATA BROADCAST

We prove that the problem is NP with a reduction from 3-SAT. The reduction works as follows: for any 3-SAT instance, we build a three-layered network: sink, variables, clauses (Fig. 16). We link the sink to nodes corresponding to each of the variables, and add two nodes for each variable corresponding to the true and false possible values for the respective variable, and one more node for selecting *at least* one of the variables values. Then we add one more layer with one node for each clause, and link it to the corresponding true or false node, that is contained in that clause. We show that finding a minimum tree for this instance of the problem is equivalent to finding a satisfying assignment of the variables in the 3-SAT instance. We will do this by choosing such values for the edges so as to force the optimal tree to contain *one single branch*, corresponding to either the

true or false node, per variable (that is, all clauses are linked to at most one of the two nodes of any variable).

Assign weight 1 to all edges except the ones connecting the clauses to the variables, which have weight d chosen large enough so an optimal tree will not pass through more than $|C|$ such links. Then, a 3-SAT instance is satisfiable if and only if the corresponding graph admits a data gathering tree of size $|C| \cdot R(d + 2) + |V| \cdot 3R + |V| \cdot 2R + |V| \cdot 2r + |V| \cdot r$, where $|C|$ is the number of clauses and $|V|$ is the number of variables. If both T/F branches corresponding to the same variable need to be connected to the clause nodes, then one of the $2R$ terms is replaced with a $2r$ term so the tree is no longer optimal. As the construction of the tree corresponding to the 3-SAT instance is polynomial, then our problem is at least as hard as 3-SAT, and thus NP-complete. ∎

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–116, Aug. 2002.

[2] B. Awerbuch, A. Baratz, and D. Peleg, "Cost-sensitive analysis of communication protocols," in *Proc. 9th Annu. ACM Symp. Principles of Distributed Computing*, 1990, pp. 177–187.

[3] J. Barros, C. Peraki, and S. D. Servetto, "Efficient network architectures for sensor reachback," in *Proc. Int. Zurich Seminar on Communications*, Zurich, Switzerland, 2004, pp. 184–187.

[4] D. Bertsekas, *Network Optimization: Continuous and Discrete Models*. Belmont, MA: Athena Scientific, 1998.

[5] K. Bharat-Kumar and J. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. COM-31, no. 3, pp. 343–351, Mar. 1983.

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

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

[8] ——, "On network correlated data gathering," in *Proc. IEEE INFOCOM*, 2004, pp. 2571–2582.

[9] M. Enachescu, A. Goel, R. Govindan, and R. Motwani, "Scale free aggregation in sensor networks," in *Proc. 1st Int. Workshop on Algorithmic Aspects of Wireless Sensor Networks*, 2004, pp. 71–84.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability*. San Francisco, CA: W. H. Freeman, 1979.

[11] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk," in *ACM-SIAM Symp. Discrete Algorithms*, 2003, pp. 499–507.

[12] G. M. Guisewite and P. M. Pardalos, "Algorithms for the single-source uncapacitated minimum concave-cost network flow problem," *J. Global Optimization*, vol. 1, no. 3, pp. 245–265, 1991.

[13] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, no. 13, pp. 311–329, 1988.

[14] W. Rabiner Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. System Sciences (HICSS'00)*, Jan. 2000, pp. 3005–3014.

[15] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 2–16, Feb. 2003.

[16] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning and shortest path trees," in *Proc. 4th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1993, pp. 243–250.

[17] B. Krishnamachari, D. Estrin, and S. Wicker, "Modeling Data Centric Routing in Wireless Sensor Networks," Dept. Comput. Eng., Univ. Southern California, Los Angeles, CA, Tech. Rep. 02-14, 2002.

[18] E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, Eds., *The Traveling Salesman Problem*. New York: Wiley, 1990.

[19] S. Lindsey, C. S. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," in *Proc. IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, San Francisco, CA, Apr. 2001.

[20] M. Lundy and A. Mees, "Convergence of an annealing algorithm," *Math. Programming*, no. 34, pp. 111–124, 1986.

[21] S. Pattem, B. Krishnamachari, and R. Govindan, "The impact of spatial correlation on routing with compression in wireless sensor networks," in *Proc. Information Processing in Sensor Networks (IPSN)*, 2004, pp. 28–35.

[22] C. Perkins, *Ad Hoc Networking*. Reading, MA: Addison-Wesley, 2000.

[23] G. J. Pottie and W. J. Kaiser, "Wireless integrated sensor networks," *Commun. ACM*, vol. 5, no. 43, pp. 51–58, 2000.

[24] S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS): design and construction," in *Proc. IEEE Data Compression Conf.*, Mar. 1999, pp. 158–167.

[25] J. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, "Pico-Radio supports ad hoc ultra-low power wireless networking," *IEEE Computer*, vol. 33, no. 7, pp. 42–48, Jul. 2000.

[26] C. Reidys and P. Stadler, "Combinatorial landscapes," *SIAM Rev.*, vol. 44, pp. 3–54, 2002.

[27] A. Scaglione and S. D. Servetto, "On the interdependence of routing and data compression in multi-hop sensor networks," in *Proc. MobiCom*, Atlanta, GA, 2002.

[28] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.

[29] E. Tardos, "A strongly polynomial minimum cost circulation algorithm," *Combinatorica*, vol. 5, pp. 247–255, 1985.

Baltasar Beferull-Lozano (M'02) was born in Valencia, Spain, in 1972. He received the M.Sc. degree in physics from Universidad de Valencia, Valencia, Spain, in 1995 (First in Class Honors) and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Southern California, Los Angeles, in 1999 and 2002, respectively. His Ph.D. work was supported by a National Graduate Doctoral Fellowship from the Ministry of Education of Spain and a Research Assistantship from the Department of Electrical Engineering at the University of Southern California.

From January 1996 to August 1997, he was a Research Fellow Assistant at the Department of Electronics and Computer Science, Universidad de Valencia, and from September 1997 to September 2002, he was a Research Fellow Assistant in the Department of Electrical Engineering, the Integrated Media Systems Center (NSF Engineering Research Center). and the Signal and Image Processing Institute at the University of Southern California. He has also worked for AT&T Shannon Laboratories (formerly AT&T Bell Laboratories), Information Sciences Center, Florham Park, NJ. Since October 2002, he has been a Research Associate in the Department of Communication Systems at the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, and a Senior Researcher within the Swiss National Competence Center in Research on Mobile Information and Communication Systems (Swiss NSF Research Center). His research interests are in the general areas of signal and image processing, distributed signal processing and communications for sensor networks, information theory and communication theory.

Dr. Beferull-Lozano received several awards at the University of Southern California, including the Best Ph.D. Thesis Paper Award in April 2002 and the Outstanding Academic Achievement Award in April 1999 and April 2002. He has served as a member of the Technical Program Committees for several ACM and IEEE international conferences.

Martin Vetterli (F'95) received the Engineering degree from ETH Zurich (ETHZ), Switzerland, the M.S. degree from Stanford University, Stanford, CA, and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.

In 1986, he joined Columbia University, New York, first with the Center for Telecommunications Research and then with the Department of Electrical Engineering, where he was an Associate Professor of electrical engineering. In 1993, he joined the University of California at Berkeley, where he was a Full Professor until 1997. Since 1995, he has been a Professor at EPFL, where he headed the Communication Systems Division (1996/1997) and now heads the Audiovisual Communications Laboratory. From 2001 to 2004, he directed the National Competence Center in Research on mobile information and communication systems. He has also been a Vice-President for International Affairs at EPFL since October 2004. He has held visiting positions at ETHZ (1990) and Stanford (1998). His research interests include sampling, wavelets, multirate signal processing, computational complexity, signal processing for communications, digital video processing, and joint source/channel coding.

Dr. Vetterli has been a member of the Association for Computing Machinery (ACM) since 2000.

Răzvan Cristescu (M'04) was born in Ploiesti, Romania, in 1974. He received the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 2004, the Lic.Tech. from Helsinki University of Technology, Finland, in 2000, and the M.Sc. degree from the Polytechnic University of Bucharest, Romania, in 1998.

Since October 2004, he has been a Postdoctoral Fellow in the Center for Mathematics of Information at the California Institute of Technology, Pasadena. His research interests include information theory, network theory, computational complexity, signal processing for communications, and applications in sensor networks.

Roger Wattenhofer received the Ph.D. degree in computer science from ETH Zurich, Switzerland, in 1998.

From 1999 to 2001, he was in the USA, first at Brown University, Providence, RI, then at Microsoft Research, Redmond, WA. He is now a Professor in the Information Technology and Electrical Engineering Department, ETH Zurich, Switzerland. His research interests include a variety of algorithmic aspects in networking and distributed computing, in particular, peer-to-peer computing, ad hoc and sensor networks.