On Network Correlated Data Gathering

Răzvan Cristescu*, Baltasar Beferull-Lozano* and Martin Vetterli*†

*Laboratory for Audio-Visual Communications (LCAV)
Swiss Federal Institute of Technology (EPFL), Lausanne CH-1015, Switzerland
†Department of Electrical Engineering and Computer Science
University of California at Berkeley, Berkeley CA 94720, USA
Email: {Razvan.Cristescu, Baltasar.Beferull, Martin.Vetterli}@epfl.ch

Abstract—We consider the problem of correlated data gathering by a network with a sink node and a tree communication structure, where the goal is to minimize the total transmission cost of transporting the information collected by the nodes, to the sink node. Two coding strategies are analyzed: a Slepian-Wolf model where optimal coding is complex and transmission optimization is simple, and a joint entropy coding model with explicit communication where coding is simple and transmission optimization is difficult. This problem requires a joint optimization of the rate allocation at the nodes and of the transmission structure. For the Slepian-Wolf setting, we derive a closed form solution and an efficient distributed approximation algorithm with a good performance. For the explicit communication case, we prove that building an optimal data gathering tree is NP-complete and we propose various distributed approximation algorithms.
KEYWORDS: Graph Theory, Combinatorics, Information Theory, Mathematical programming/optimization, Simulations.

I. INTRODUCTION

A. The problem and its motivation

Consider a number of distributed data sources with a certain correlation structure among the sources. A number of links connects 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, and we refer to this set up as the *correlated data gathering tree on a graph* problem. The goal is to achieve the data gathering while minimizing a cost functional (e.g. power on links or distance accumulated), possibly under constraints (e.g. some maximal capacities on links). This is of course an instance of a network flow problem, but with an original twist: because the data is correlated, standard solutions may not be optimal, leading to original rate allocation problems and original tree building problems, depending on the source coding model that will be used.

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. Nodes are supplied amounts of measured data which need to be transmitted to end sites, called sinks or base stations. Intermediate nodes can be also used as relays in addition to measuring data. They aggregate their own data
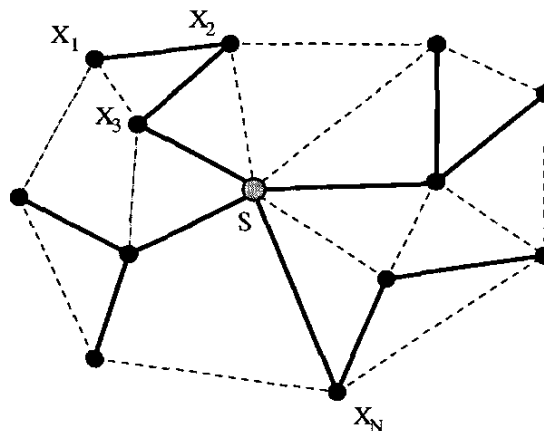
Fig. 1. An example of a network. Nodes transmit their data to the base stations. A rate supply R_i is allocated to each node X_i. In solid lines, a chosen tree structure is shown. In dashed lines, the other possible links are shown.

with the data received from other nodes. A very important task in this scenario is to find a transmission structure on the network graph that minimizes a cost of interest (e.g. flow cost [data size] × [link weight], distance, etc.). In the case of a single sink, the optimal structure is usually a tree. This leads to the question of how to construct efficient data gathering trees.

When the data measured at nodes are statistically independent, the problem separates: the rate allocation becomes trivial, as each node codes its data independently; then next, well developed algorithms can be used to solve various network problems (minimum and shortest path spanning tree, minimum cost flow etc.).

However, in many situations, data at nodes are *not* independent. Due to the correlation present, it is expected that approaches that take into account this correlation, will outperform traditional approaches, for various cost functions (metrics). Moreover, jointly exploiting the data structure and optimizing the transmission topology (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.

A practical instance can be found in sensor networks [1], [2], [3]: a number of sensors acquire measurements which are typically correlated to each other, and these measurements are sent to a base station for evaluation or decision purposes. Collecting images from various sources into a common repository on the internet is another example of correlated data gathering.

In particular, let us consider the case of a network of sensors taking measurements from the environment. Let $X = (X_1, \ldots, X_N)$ be the vector formed by the random variables measured at the nodes $1, \ldots, N$. We assume that the random variables are continuous and that there is a quantizer in each sensor (with the same resolution for all sensors). A rate allocation $(R_1, \ldots, R_N)$ (bits) has to be assigned at the nodes so that the quantized measured information samples are described losslessly[1]. That information has to be transmitted through the links of the network to the designated base station (see Fig. 1). We assume contention is solved by the MAC layer. We abstract the wireless case as a graph with point-to-point links, mainly because the full wireless multi-point case is an open problem, and practical schemes transform the problem into a graph with nearest neighbor connectivity. A meaningful metric to minimize is the energy consumption, which is essentially given by the sum of products [data size] × [link weight], for all the links used in the transmission. Here the weight of the link between two nodes is a function of the distance d of the two nodes (e.g. $kd^{-\alpha}$ or $k\exp(\alpha d)$, with k, α constants that depend on the medium properties).

We will consider two complementary approaches that may 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 with random binning arguments, that is using Slepian-Wolf coding [4], [5], [6], [7]). With this approach, finding a good routing structure turns out to be easy, because routing and coding are separated (decoupled); however data coding becomes complex and global network knowledge is needed for an optimal solution.

In the second approach, 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. In this case data coding is easy and relies only on locally available data as side information. However, now optimizing the routing structure becomes complex.

Summarizing, the main tradeoff between these two settings is

- If nodes are assumed to know the correlation structure (or equivalently, the dependence on the distance), then they can employ Slepian-Wolf coding. In this case, source coding is complex while the transmission structure can be found with classical polynomial time network flow algorithms.

[1]The use of a distributed lossy coding setting (distributed quantization) in our problem is a subject of our future research.

- If the correlation structure is not known a-priori, then side information is needed to reduce the entropy. In other words, the correlation structure is learned (explicitly) in a distributed manner through explicit communication. This leads to a simple source coding, but the transmission structure optimization is hard.

In this work, we show how the rate allocation is dependent on the chosen setting. In the Slepian-Wolf case, optimal coding allocates most of the load to nodes close to the sink and small rates to nodes at the extremity of the network. In the explicit communication model, data compression is done only when side data is available, and thus large rates are allocated to nodes far from the sink, and much smaller rates at nodes close to the sink. It is therefore expected that a combined third approach will provide not only a complexity tradeoff between the source coding and the transmission structure strategy, but will also distribute more evenly the rate load throughout the network. One possible approach is to consider a combination of Slepian-Wolf coding for clusters of nodes, and explicit communication among clusters, and this is a topic for further research.

B. Related work

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

In [8], 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 [9] introduces the PEGASIS algorithm, that uses the energy × delay metric over the routing tree; their algorithms find chains of nodes instead of clusters.

The work which is most related to the problem we consider in this paper is the one involving the concept of directed diffusion and data aggregation [10]. In that model, sensors measure events, creating gradients of information in their respective neighborhoods. The base station requests data by broadcasting interests. When interests fit gradients, paths of information flow are formed and then, the best paths are reinforced. In order to reduce communication costs, data is aggregated on the way. The goal is to find a good aggregation tree which gets the data from some of the nodes to the base station. Similar work can be found in [11] and [12]. In [12], 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 maximize several concave cost functions of interest. Their model is significantly different from ours in the sense that their setting assumes information sources supply a constant amount of information. Moreover, their model does not take into account possible collaborations among nodes. In our case, due to the correlation structure, the amount of aggregated information sent down the tree to the base station from a particular node depends on the structure of the subtree whose

parent is that node. In addition, we consider the case of collaboration between nodes through Slepian-Wolf coding.

C. Main contributions

In our work, the case where Slepian-Wolf coding is employed is fully solved. We prove that in the case where a *global* Slepian-Wolf coding is performed, the optimal choice of the transmission structure is completely separated from the rate allocation optimization; however the optimal rate allocation does depend in general on the distances between nodes. We find the optimal rate allocation in closed form and propose a distributed approximation algorithm which provides a rate allocation that is very close to the optimal one. Then, we study a simplified coding case where Slepian-Wolf is performed locally only on clusters of nodes, thus avoiding the use of global Slepian-Wolf coding. However, we conjecture that in this case separating optimally the nodes into clusters becomes NP-complete.

Second, we study the case when joint coding of correlated data is performed explicitly, that is, the reduction in rate by entropy coding due to the correlation is possible only when side information is explicitly available (as relayed data). We prove that this makes the problem NP-complete and propose distributed approximation algorithms with a good performance.

The paper is structured as follows: In Section II, we state the formal definition of the optimization problem we address in this work. Section III is concerned with the treatment of the Slepian-Wolf case. We consider in Section IV a particular case of the mixed approach, where Slepian-Wolf coding on clusters is used. In Section V we treat the explicit communication case. We illustrate our ideas using an example based on an ergodic Gaussian random field as model of correlation. Finally, we conclude and present directions of future work in Section VI.

II. PROBLEM FORMULATION

Let $G = (V, E)$ be a weighted graph with $|V| = N+1$. We denote by S the particular $(N+1)$th node called the sink node. Each edge $e = (i, j) \in E$ has a weight w_e. Every node i has to transmit a rate (or supply) R_i through the network to the sink. The rate R_i might be negative because of the rate reduction by joint entropy coding at each node. Let $f(x, w)$ be an arbitrary cost function of the flow x through a particular edge and the edge weight w. Then the *minimum cost data gathering tree* problem is defined as follows. Find the spanning tree ST of the graph G and the rate allocations R_i's that minimize the cost function:

$$\sum_{e \in ST} f(x_e, w_e), \qquad (1)$$

under constraints

$$x_{i \to e} - \sum_{e \to i} x_e = R_i, i = 1, \ldots, N,$$

$$R_S = \sum_{i=1}^{N} R_i,$$

where we denote by $e \to i$ the set of edges entering node i, and by $i \to e$ the edge from node i to its parent in the tree. We restrict our discussion to *separable* functions f, and without loss of generality we assume $f(x, w) = x \times w$. This corresponds to many practical settings (e.g. many cases of sensor networks). Then the expression (1) to be minimized can be rewritten as:

$$\sum_{i \in V} R_i d_{ST}(i, S) \qquad (2)$$

where $d_{ST}(i, S)$ is the total weight of the path connecting node i to S on the ST tree.

There is an important generalization that makes the problem we consider in this work different from classical network flow theory: the rate R_i at node i *depends* on (a) the incoming flow $\sum_{e \to i} x_e$ entering that node, *and* (b) on the particular structure chosen to transmit information from the nodes that supply the incoming flow, that is, the links that form the subtree rooted at node i. In general, R_i is a decreasing function of the amount of incoming flow because joint coding always reduces information rates. This implies that the rate at each node depends in general on the particular transmission tree structure used, hence the rate allocation and the tree construction must be addressed together, rather than separately. Because of this important generalization, we call this the *minimum cost network correlated data gathering tree* problem.

We now particularize the optimization problem (2) to the two coding settings we introduced in Section I-A.

A. Slepian-Wolf coding

Consider (see Fig. 1) the case of two random sources X_1 and X_2 that are correlated. Intuitively each of the sources can code their data at a rate equal to at least their corresponding entropies, $R_1 = H(X_1)$, $R_2 = H(X_2)$, respectively. If they are able to communicate, then they could coordinate their coding and use together a total rate of $R_1 + R_2 = H(X_1, X_2)$. Slepian and Wolf [4] showed that two correlated sources can be coded with a total rate equal to the joint entropy $H(X_1, X_2)$ even if they are *not* able to communicate with each other, as long as their individual rates are at least equal to the conditional entropies, $H(X_1|X_2)$ and $H(X_2|X_1)$ respectively; this easily generalizes to the N-dimensional case.

Assume now that in the example given by Fig. 1, the nodes know in advance the correlation structure (which we assume depends only on the distance among the nodes for a given medium). Then nodes $1, 2, 3, \ldots, N$, can code their data jointly, without communicating with each other, with a rate $H(X_1, X_2, \ldots, X_N)$, as long as their individual rates obey constraints related to the different conditional entropies [4], [5]. We show in Section III that, when optimizing (2), the shortest path tree is optimal for any rate allocation and thus the whole optimization problem can be separated into a spanning tree optimization and the rate allocation optimization. Then the minimization problem in (2) becomes a problem of rate allocation that can be posed as a linear programming (LP) problem under linear constraints. In general, if the

2573

cost function is separable but nonlinear, then (2) becomes a non-linear programming problem that can be solved using Lagrange multipliers. For the sake of clarity, we will restrict ourselves in this paper to separable linear functions.

B. Explicit communication

In classical network transport theory, the amount of supply 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 links. The novelty of our problem formulation stems from the fact that we consider a class of problems where the supply at a given node *depends* on the incoming flow from other nodes that use that node as relay, and also on the transmission structure for these nodes.

Consider again the example in Fig. 1, where nodes have to communicate their correlated data to one base station. If we consider node X_3, then the quantity it has to supply depends on whether

1) neither node X_1 nor node X_2 use node X_3 as relay, or
2) node X_1 uses node X_2 as relay, and node X_2 transmits its aggregate further to node X_3 (this case is shown in solid line in Fig. 1), or
3) both node X_1 and node X_2 use node X_3 as relay.

In the first case, no side information is available at node X_3 from other nodes. Node X_3 sends its entire amount of data on a path to the sink.

In the second case, node X_3 does have side information available from node X_2. The information at these two nodes is not independent. Therefore node X_3 can reduce correspondingly the amount of data it sends further. It jointly codes its data with the data from node X_2 and sends the resulting coded data further.

In the third case, the amount of side information available is even larger at node X_3, since two nodes use it as relay. Thus, the data amount reduction at node X_3 is larger as well because the joint coding involves more sources.

This network setting occurs frequently, for example in sensor networks measuring correlated data.

It is clear that in either of these three cases the optimal transmission structure might not be the shortest path tree. We show in Section V how this joint treatment of source coding and transmission structure optimization actually makes the problem NP-complete.

Finally, the two approaches of Slepian-Wolf coding and explicit communication can be combined in a third mixed approach, where some clusters of nodes code data jointly with a Slepian-Wolf procedure, while the rest of the nodes use explicit communication. For instance, since nodes that are close to the base station relay data from the rest of the network, they will have an important amount of side information available and thus, they do not need to employ Slepian-Wolf coding, reducing the complexity in terms of coding.

III. SLEPIAN-WOLF CODING

A. Slepian-Wolf constraints

With Slepian-Wolf coding, sources *do not* need to communicate explicitly in order to code their data at a rate equal to the joint entropy. By using random binning based codes, it can be shown that a rate allocation $(R_1, \ldots, R_N)$ can be achieved if and only if it satisfies

$$\sum_{i \in \mathbf{Y}} R_i \geq H(\mathbf{Y}|\mathbf{Y}^C) \qquad (3)$$

for any of the $2^N - 1$ sub-vectors[2] $\mathbf{Y} \subseteq \mathbf{X}$ (see Fig. 8 for the rate region of two sources).

Proposition 1: – Separation of source coding and tree building:

When there is a single sink and Slepian-Wolf coding is used, the shortest path tree (*SPT*) is optimal, in terms of minimizing (2), for any rate allocation.

Proof: Once the rate allocation is *fixed*, the best way to transport any amount of data from any node to the sink is to use the shortest path. Minimizing the sum of costs in (2) becomes equivalent to minimizing the part of cost corresponding to each node in part. Since the shortest path tree is a superposition of all individual shortest paths, then it is optimal for any rate allocation that does not depend on the transmission structure, which is the case here. ∎

Therefore, global Slepian-Wolf coding separates in fact rate allocation from how the transmission structure is chosen. Correlated data is encoded at nodes with a rate given by the joint entropy and without explicit internode communication. However, the distances among nodes are needed implicitly in expressing the constraints (conditional entropies) in (3), as the correlation model is usually distance dependent.

In Section III-B, we derive the closed form solution for minimizing (2) under the constraints (3). This is however not enough from a practical point of view, for example for deriving a distributed solution. Global Slepian-Wolf coding involves a large complexity. Because of this, in Section III-C we present an approximation algorithm that provides a rate allocation close to optimum, for a large class of networks in which the correlation between the nodes decreases with the distance. Moreover, our algorithm is fully distributed and relies only on local information. In Section III-D we particularize our solution for the case of a Gaussian random field, and show numerical simulations.

B. Solution of the LP problem

We have seen that for any *fixed* rate allocation R_i, *SPT* is the optimal gathering tree. We now present the algorithm for finding the optimal rate allocation:

Algorithm 1: Optimal Slepian-Wolf rate allocation:

• Find the *SPT* of the initial connected graph G.

---

[2]Note that $\mathbf{Y}$ denotes both the vector of random variables, and the node set where those random variables are measured.

- Solve the constrained LP:

$$(R_1^*, \ldots, R_N^*) = \arg\min_{\{R_i\}} \sum_i R_i d_{SPT}(i, S),$$

under constraints: (4)

$$\sum_{i \in \mathbf{Y}} R_i \geq H(\mathbf{Y}|\mathbf{Y}^C), (\forall)\mathbf{Y} \subseteq \mathbf{X}$$

where $d_{SPT}(i, S)$ is the total length of the path in the *SPT* from node i to S, and $(R_1^*, \ldots, R_N^*)$ is the optimal rate allocation.

Note that in order to express the rate constraints, centralized knowledge of the correlation structure among *all* nodes in the network is needed.

Suppose without loss of generality that nodes are numbered in increasing order of their distance to the sink on the *SPT*: $(X_1, X_2, \ldots, X_N)$ with $d_{SPT}(X_1, S) \leq d_{SPT}(X_2, S) \leq \cdots \leq d_{SPT}(X_N, S)$. That is, node X_1 is the closest node to the sink and node X_N is the furthest one from the sink on the *SPT*.

Theorem 1 (LP solution): The solution of the optimization problem in (4) is:

$$\begin{aligned} R_1^* &= H(X_1), \\ R_2^* &= H(X_2|X_1), \\ \cdots & \cdots \cdots \\ R_N^* &= H(X_N|X_{N-1}, X_{N-2}, \ldots, X_1). \end{aligned}$$ (5)

In words, the closest node to the sink is coded with a rate equal to its unconditioned entropy. Each of the other nodes is coded with a rate equal to its respective entropy conditioned on all other nodes which are closer to the sink than itself.

The proof is shown in Appendix A.

C. Approximation algorithm

In the previous section, we have found the optimal solution of the linear programming rate assignment under Slepian-Wolf constraints. We consider now the problem of designing a distributed algorithm. Even if we can provide the solution in a closed form as (5), the nodes still need local knowledge of the overall structure of the network (distances between nodes and distances to the sink). This local knowledge is needed for:

1) Ordering the distances on the *SPT* from the nodes to the sink: each node needs its index in the ordered sequence of nodes so as to determine on which other nodes to condition when computing its rate assignment. For instance, it may happen that the distance on the graph between nodes X_N and X_{N-1} is large. Thus, closeness in the ordering does not mean necessarily closeness in distance.

2) Computation of the rate assignment:

$$\begin{aligned} R_i &= H(X_i|X_{i-1}, \ldots, X_1) \\ &= H(X_1, \ldots, X_i) - H(X_1, \ldots, X_{i-1}) \end{aligned}$$

Note that *all* distances among nodes $(X_1, \ldots, X_i)$ are needed locally at node X_i for computing this rate assignment.

Such global knowledge might not be available. Thus, we propose a fully distributed approximation algorithm, which avoids the need for a node to have global knowledge of the network, and which provides solutions very close to the optimum.

Suppose each node i has complete information (distances between nodes and distances to the sink) only about a local vicinity $\mathcal{N}(i)$ formed by its closest neighbors. This information can be computed by running for example a distributed algorithm for finding the *SPT* (e.g. Bellman-Ford). The approximation algorithm we propose is based on the observation that nodes that are outside this neighborhood count very little, in terms of rate, in the local entropy conditioning, under the assumption that the correlation decreases with the distance between nodes, which is a natural assumption.

Algorithm 2: Approximated Slepian-Wolf coding:
- Find the *SPT*.
- For each node i:
  - Find in the neighborhood $\mathcal{N}(i)$ the set C_i of nodes that are closer to the sink, on the *SPT*, than node i.
  - Transmit at rate $R_i^\dagger = H(X_i|C_i)$.

This means that data are coded locally at the node with a rate equal to the conditional entropy, where the conditioning is performed *only* on the subset formed by the neighbor nodes which are closer to the sink than the respective node.

The proposed algorithm needs only local information, so it is completely distributed. Still, it will give a solution very close to the optimum since the neglected conditioning is small in terms of rate for a correlation function that is sufficiently decaying with distance (see next subsection and Fig. 2).

The Slepian-Wolf setting, as well as the closed-form solution and approximation algorithm, can be easily extended to the case of multiple sinks.

D. Example: Gaussian random field

A model frequently encountered in practice is the Gaussian random field. This also has the nice property that the dependence in data at different nodes is fully expressed by the covariance matrix, which makes it more suitable for analysis. Thus, we assume a *jointly Gaussian model* for the spatial data $\mathbf{X}$ measured at nodes, with an N-dimensional multivariate normal distribution $G_N(\mu, K)$:

$$f(\mathbf{X}) = \frac{1}{\sqrt{2\pi} \det(K)^{1/2}} e^{-\left(\frac{1}{2}(\mathbf{X}-\mu)^T K^{-1}(\mathbf{X}-\mu)\right)}$$

where K is the covariance matrix (positive definite) of $\mathbf{X}$, and μ the mean vector. The diagonal elements of K are the variances $K_{ii} = \sigma_i^2$. The rest of K_{ij} depend on the distance between the corresponding nodes (e.g. $K_{ij} = \sigma^2 \exp(-a d_{i,j}^2)$). Then, for any index combination $I = \{i_1, \ldots, i_k\} \in \{1, \ldots, N\}$, $k \leq N$, $\mathbf{W} = (X_{i_1}, \ldots, X_{i_k})$ is k-dimensional normal distributed. Its covariance matrix is the submatrix $K[I]$ selected from K, with rows and columns corresponding to $\{i_1, \ldots, i_k\}$.

We use here without loss of generality differential entropy instead of entropy, since we assume that data at all nodes
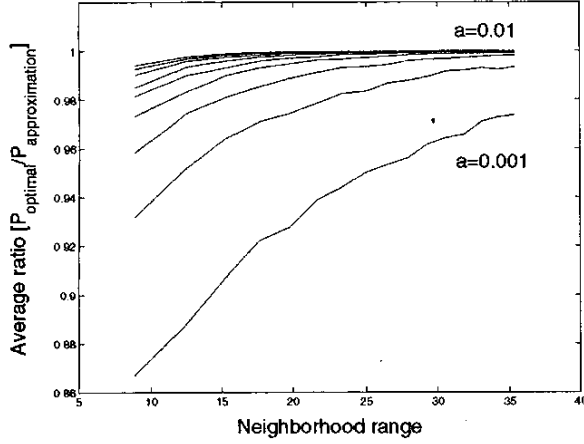
Fig. 2. Average value of the ratio between the optimal and the approximated solution, in terms of total cost, vs. the neighborhood range. The network instances have 50 nodes uniformly distributed on a square area of size 100 × 100, and the correlation exponent varies from $a = 0.001$ (high correlation) to $a = 0.01$ (low correlation).

is quantized with the same quantization step, and differential entropy differs from entropy by a constant for uniformly quantized variables [5]. The entropy of a k-dimensional multivariate normal distribution $G_k(\mu, K)$ is:

$$h(G_k(\mu, K)) = \frac{1}{2}\log(2\pi e)^k \det K.$$

The Slepian-Wolf constraints can readily be expressed as:

$$
\begin{aligned}
h(\mathbf{Y}|\mathbf{Y}^C) &= h(\mathbf{Y}, \mathbf{Y}^C) - h(\mathbf{Y}^C) \\
&= \frac{1}{2}\log\left((2\pi e)^{N-|\mathbf{Y}^C|}\frac{\det K}{\det K[\mathbf{Y}^C]}\right)
\end{aligned}
$$

where $K[\mathbf{Y}^C]$ is the selected matrix out of K, with indices corresponding to $\mathbf{Y}^C$ elements respectively.

This natural correlation model is useful for us because our approximation algorithm can be easily tested. Consider for example the case where the correlation decays exponentially with the distance. Then, the performance of our approximation algorithm will be close to optimal even for small neighborhoods $\mathcal{N}(i)$. Fig. 2 presents the average ratio between the approximated solution and the optimal one, for a correlation model $K_{ij} = \exp(-ad_{i,j}^2)$, for varying neighborhood range radius and varying correlation exponent a. The weight of an edge (i, j) is $w_{i,j} = d_{i,j}^2$ and the total cost is given by expression (2).

IV. CLUSTERED SLEPIAN-WOLF

Performing Slepian-Wolf coding of the data jointly *at all nodes* in the network is difficult as it involves a large number of bins and global knowledge of the network. In this section we consider the case where Slepian-Wolf coding is done locally, on clusters of nodes. This simplifies significantly the coding task, since only knowledge of the local network structure is needed. However, as we will see, the actual task of optimal clustering becomes also complex, thus, showing the inherent complexity that is present in our problem.

A. Disjoint clusters

Suppose first that a node is part of two clusters. It can use the same random codes, but it needs to use different binning for the two clusters since two independent Slepian-Wolf codings have to be performed. Then, it has to transmit the maximum rate allowable by its local cluster rate allocations. In order to solve the optimizations, the node needs to locally express the inequalities in (3). But this means it knows all the weights involved in these constraints, so clearly, the rate allocation can be improved by using the union of the two clusters. This is the reason why we consider only the case of *disjoint* Slepian-Wolf clusters for the rest of this section.
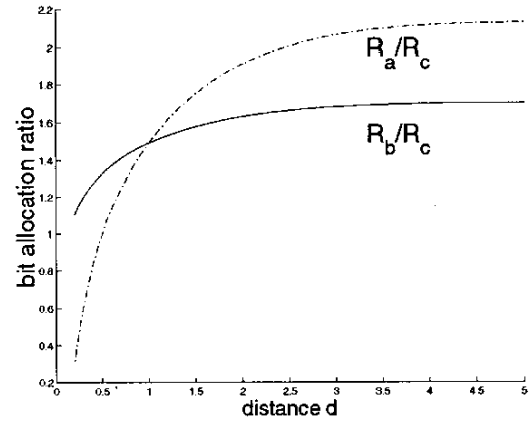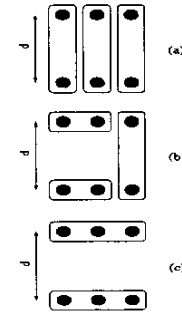
B. Gaussian random field

Fig. 3. Three possible clustering modes for a network of 6 jointly Gaussian nodes (up). The ratio between the optimal bit allocations between the examples in (a),(b) and (c) respectively, as function of the distance d (down).

For a Gaussian random field, the optimization problem can be posed in a closed form. We allow a maximum cluster size of B. Then the problem is to find an optimal clustering of V that minimizes the weighted sum of rates $\sum_i R_i d_{SPT}(i, S)$ (see Fig. 3, for $B = 2$). Consider first the particular case when the distances on the *SPT* are all unity (e.g. nodes placed on an arc of circle). We show that even in this simple case, the problem of optimal clustering becomes complex for $3 \leq B < N$. Denote by $I_1, \ldots, I_C$ the disjoint sets of indices such that $\bigcup I_i = \{1, \ldots, N\}$, with $C \leq N$ the (unknown) total number of clusters, and each cluster can be encoded with $H(\mathbf{X}_{I_i})$ bits using Slepian-Wolf in that cluster. The optimization problem

is then to minimize over $C, \{I_i\}_{i=1}^{C}$:

$$\sum_i h(G_{|I_i|}(0, K[I_i])) = \sum_i \frac{1}{2} \log(2\pi e)^{|I_i|} \det K[I_i]$$

So finally, we need to solve

$$\{I_i^*\} = \arg \min_{C, \{I_i\}_{i=1}^{C}} \left(\prod_{i=1}^{C} \det K[I_i] \right), \qquad (6)$$

with $|I_i^*| \leq B, i = 1, \ldots, C$.

1) Case $B = 1$: Trivial, data are coded separately.

2) Case $B = 2$: If $B = 2$, the problem (2) is equivalent to finding a *minimum weighted perfect matching*, and it can be solved in polynomial time with the Edmonds 'blossom' perfect matching algorithm [13]:

Algorithm 3: Rate allocation for $B = 2$:

- For each edge (i, j):
  - assign (R_i, R_j) under *local* constraints:
    * **if** $d_{SPT}(i, S) < d_{SPT}(j, S)$
    * **then** $(R_i, R_j) = (H(X_i), H(X_j|X_i))$,
    * **else** $(R_i, R_j) = (H(X_i|X_j), H(X_j))$
    (see Section III-B).
  - let $d'(i, j) = R_i d_{SPT}(i, S) + R_j d_{SPT}(j, S)$.
- Run Edmonds algorithm for $d'(i, j)$.

3) Case $B = N$: If $B = N$, the optimal cluster division is the cluster formed by the whole set of nodes $(1, \ldots, N)$, which follows from the fact that K is positive definite, and applying Fischer's inequality [14].

4) Case $3 \leq B < N$: We show how this becomes very complex, being a particular case of an NP-complete problem, minimum 3-dimensional (3-D) matching. We conjecture the optimization of (6) is also NP-complete.

Definition 1: Minimum 3-D matching.

- *Instance*: Three sets X, Y, and W and a cost function $c : X \times Y \times W \to N$.
- *Question*: An assignment A, i.e., a subset $A \subseteq X \times Y \times W$ such that every element of $X \cup Y \cup W$ belongs to exactly one triple in A.
- *Measure*: The cost of the assignment, i.e., $\sum_{(x,y,w) \in A} c(x, y, w)$.

The minimum 3-D matching problem is NP-complete, and this extends to the B-dimensional problem. The sets X, Y, W are replaced by V in the above problem definition, and the cost $c(\cdot)$ is replaced by the joint entropy of the corresponding nodes.

Suppose there exists a polynomial time algorithm to solve the B-dimensional problem, for $3 \leq B < N$. Then (2) could be solved in polynomial time similarly to the 2-dimensional matching case. Unfortunately, such a polynomial time algorithm is unlikely to exist.

V. EXPLICIT COMMUNICATION

We consider now the case where Slepian-Wolf coding is not used, either due to its complexity, or to the lack of global knowledge of the network. We address the second approach

introduced in Section I-A, where the reduction in entropy at a node is due to *explicit* communication. Thus, the reduction in data size at a node is due to the direct availability of side information locally at the node.

For the Slepian-Wolf case, we have seen that the problems of rate allocation and gathering tree construction separate, when optimizing the [data size] × [link weight] metric. This is not any longer true for the case of the explicit communication model. The reason is that with this model, the amount of data supplied at nodes *does depend* on the transmission structure.

For the sake of simplicity and clarity in our arguments, and without loss of generality, we use in this section a simple model for the data correlation, which allows a clearer analysis of complexity, and for which we develop good approximation algorithms. This model still completely preserves the original complexity of the 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 $H(X_i|X_j, \ldots) = r \leq R$ bits are needed if the node has side information available coming from at least another node, that uses node i as relay (thus, our simplification is that r does not depend on the number of conditional nodes). We denote by $\rho = 1 - r/R$ the correlation coefficient.

A. The SPT / TSP tradeoff

In the case of uncorrelated data, if the cost for transmitting over an edge was proportional (by a fixed constant) to the length of that edge, then the problem is trivial and the optimal communication structure is the straight line from the node to the base station. However, for an arbitrary weight function on the edge, transmitting via relays may be better than direct transmission. In the case of correlated data, things become even more interesting, even for very simple networks (Fig. 4). If data were independent, the shortest path tree (*SPT*) would be optimal. However, in Fig. 4(a), as soon as $\rho > 1/2$, the *SPT* is no longer optimal. In Fig. 4(b), the ratio of total costs is $\lim_{N \to \infty} \frac{p_{TSP}}{p_{SPT}} = (1 - \rho_N)\left(\frac{1}{2D} + 1\right)$. Since $\rho_N \to 1$ when $N \to \infty$, a traveling salesman path (*TSP*) is arbitrarily more cost efficient than direct transmission (*SPT*).

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, we can formulate the minimization of (2) for the case of explicit communication as follows:

- **Find:** the spanning tree $ST = \{T, L, F\}$ with:
  - T non-terminal nodes, L leaves, $T \cup L = V$.
  - **Such that:**

$$ST = \arg \min_{\{T, L\}} \left(R \sum_{l \in L} d_{ST}(l, S) + r \sum_{t \in T} d_{ST}(t, S) \right)$$

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

$$ST = \arg \min_{L} \left(\rho \sum_{l \in L} d_{ST}(l, S) + (1 - \rho) \sum_{i \in V} d_{ST}(i, S) \right) \tag{7}$$
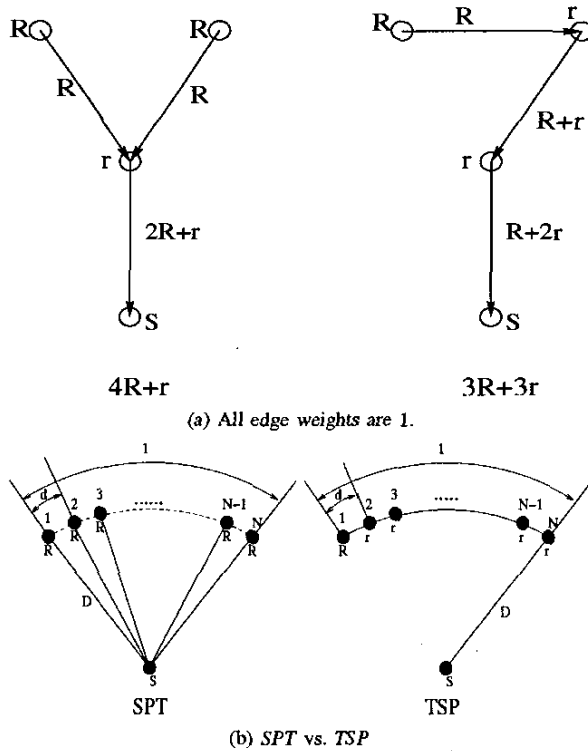
(a) All edge weights are 1.

(b) SPT vs. TSP

Fig. 4. Simple network examples.

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 base station is minimum. For this, the core information is taken from the leaf nodes, and passing through all the in-tree nodes only adds infinitesimally small new information, as data is strongly correlated. It is straightforward to show that solving this problem is equivalent to solving the *multiple traveling salesman* problem [15], which is known to be NP-hard.

To the best of our knowledge, (7) is an original spanning tree optimization problem on a graph. In Section V-B we show that this problem is NP-complete for the general case $0 < \rho \leq 1$ as well. However, we propose good approximation algorithms in Section V-C.

B. NP-completeness

In order to prove the NP-completeness of the optimization problem given in (7), we prove that the decision version problem is NP-hard. The decision version of our optimization problem is:

Definition 2: Network data gathering tree cost problem.

- *Instance:* A undirected graph $G = (V, E)$ with weights $d_{i,j}$ assigned to the edges $(i, j) \in E$, a positive integer M, and a particular node $S \in V$.

- *Question:* Does the graph admit a spanning tree ST such that, when assigning supplies $R_i = R$ to the leaf nodes and $R_i = r < R$ to the in-tree nodes in the spanning tree ST, the total cost of ST given by (7) is at most M?

Theorem 2 (NP-completeness): There is no polynomial time algorithm that solves the network data gathering tree cost problem, unless P=NP.

The proof, which is based on a non-trivial reduction from *set cover*, is shown in Appendix B.

Since this particular problem is NP-complete, then the general version of problem (2) is also NP-complete:

Corollary 1: Minimizing (2) with $R_i = f(\sum_{e \to i} x_e)$ for an arbitrary monotonic function $f(\cdot)$, is NP-complete.

Note also that, in general, node i has information from all the nodes in the subtree $st(i)$ rooted at node i. Our simplified model is a particular case of this general entropy coding problem, where $H(X_i|\mathbf{X}_{st(i)})$ is approximated with $H(X_i|X_j)$, with j being child of i. Then the NP-complexity of our simplified example extends also to this general case by means of a trivial further reduction.

C. Approximation algorithms

We propose and compare the following five algorithms:

1) Shortest path tree: SPT is computed by using the distributed Bellman-Ford algorithm for simultaneously determining the shortest paths from all nodes to the root. If the data is independent, this is the optimum solution, but it is far from optimal if there are high correlations.

2) Greedy algorithm: We start from an initial subtree composed only of the root node. Then, successively, we add, to the existing subtree, the node whose addition results in the minimum power increment.

3) Leaves deletion approximation: This algorithm 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 4: Leaves deletion algorithm:

- Initialize with the SPT. Each node maintains its parent, number of children, and total distance on the current spanning tree to the base station $d_{ST}(i, S)$. Let $par(i)$ denote the parent node of node i.
- **While** there is a cost improvement:
  - **For each** i leaf node (with no children): Find the leaf node $j \in \mathcal{N}(i)$ that maximizes $R(d_{ST}(i, S) + d_{ST}(j, S)) - (R(d_{i,j} + d_{ST}(j, S)) + rd_{ST}(j, S)) - I(i)$, where $I(i)$ is an adjustment term that indicates the cost lost by transforming single parent nodes into leaves. If the maximizing quantity is positive, $par(i) \to j$ and the corresponding distances on the tree and number of children are updated, for all the three nodes involved (i, former $par(i)$, and j).
- **Endwhile**.

This algorithm involves at most $3 - 4$ supplementary iterations after SPT is computed, and is fully distributed.

4) Balanced SPT / TSP tree: We propose an approximation based on a combination between the *SPT* and multiple traveling salesman paths (*TSP*). It builds the *SPT* subtree up to a radius $q(\rho)$ away from the root. It then builds paths for the remaining nodes starting from the leaves of the *SPT* subtree, in their respective subregions (see Fig. 6(b)).

Algorithm 5: SPT/TSP balanced tree:

- Build the *SPT* for the nodes that are in a radius $q(\rho)$ from the root. The radius $q(\rho)$ decreases with the increase of the correlation coefficient ρ.
- From the rest of the nodes, successively add the node i that minimizes $d(i,l) + d_{ST}(l,S)$, $l \in L$ being one of leaves of the current obtained subtree. This is actually a suboptimal nearest neighbor approximation to the multiple *TSP*, which is easily implementable.

This algorithm was inspired from the insights of our experiments using *simulated annealing*, a computationally hard method known to provide results close to optimal for large combinatorial problems similar to ours (e.g. *TSP*).

5) Simulated annealing: The fitness *landscape formulation* [16] 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 $g(ST) = R\sum_{l \in L} d_{ST}(l,S) + r\sum_{t \in T} d_{ST}(t,S)$. Our goal is to minimize the fitness over the spanning trees set.

Algorithm 6: Simulated annealing:

- Take a cooling schedule $T[k], k = 1, \ldots, K$.
- Initialize with $par(i) = S, i = 1, \ldots, N-1$.
- **While** $k < K$
  - $k++$, $l = g(ST)$;
  - 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 $l' = g(ST')$, the fitness of the new generated spanning tree ST'.
  - make the change $par(i) \rightarrow j, ST \rightarrow ST'$ with probability $p = \begin{cases} 1, & \text{if } l' \leq l \\ \exp(-\frac{l'-l}{T[k]}), & \text{if } l' > l \end{cases}$

- **Endwhile**

With a correctly chosen cooling schedule and if the algorithm runs for a sufficient number of steps, it can be proved to converge to the optimal spanning tree ST.

D. Numerical simulations

Our simulations were done in MATLAB for a network of up to $N = 500$ nodes randomly distributed on a 100×100 square grid. We use an approximated correlation model in which the correlation coefficient decreases with the number of nodes, that follows in a straightforward manner from the exponential correlation models introduced in Section III-D. We assume a uniform distribution of the nodes across space on a fixed area.

SPT was found with the distributed Bellman-Ford algorithm, that runs in $\mathcal{O}(N|E|)$ steps. Our extensive experiments show important improvements of the leaves deletion and the balanced *SPT/TSP* approximation algorithms over *SPT*, for randomly distributed nodes (Fig. 5, 6).
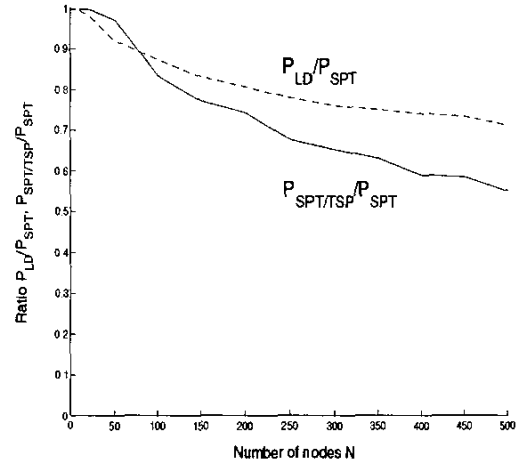

Fig. 5. Average ratios of total costs between leaves deletion (*LD*) and *SPT*, and between balanced *SPT/TSP* and *SPT*.
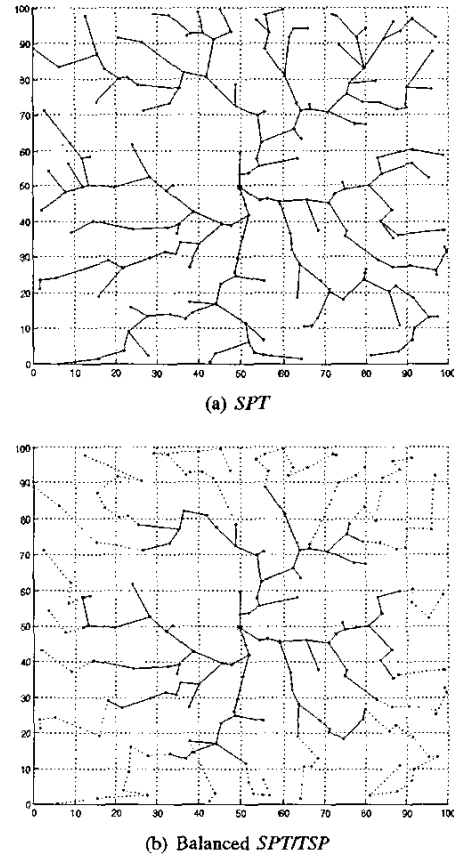

(a) *SPT*

(b) Balanced *SPT/TSP*

Fig. 6. Approximated gathering trees on a random network instance: $N = 200$, $\rho = 0.2$. Costs for this instance: *SPT*: 2.74e+5; *LD*: 2.36e+5; *SPT/TSP*: 2.15e+5.
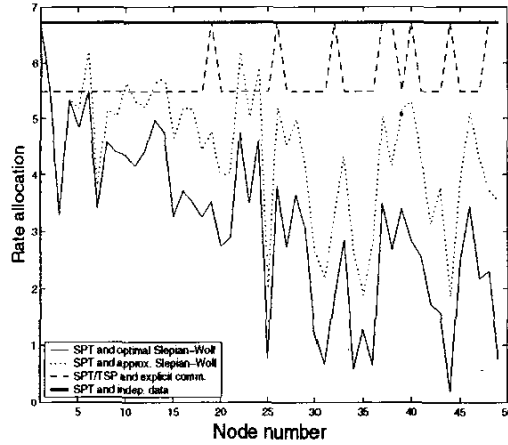
Fig. 7. Typical rate allocation for a network instance of 50 nodes, and correlation exponent $a = 0.0005$. On the x-axis, nodes are numbered in order as the distance from S increases, on the corresponding spanning tree. Total costs for this instance: SPT with optimal Slepian-Wolf: $0.84e + 5$; SPT with approximated Slepian-Wolf: $1.30e + 05$; SPT/TSP with explicit communication (with conditioning on first order children): $1.94e + 5$; SPT with independently coded data: $2.24e + 5$.

In Fig. 7 we show a comparison of the rate allocations with our different approaches for rate allocation, as a function of the distances from the nodes to the sink. For the explicit communication approach, we used a first-order correlation model, in which in-tree nodes condition only on the closest child, and the value of ρ is computed a-priori as an average. Our experiments have shown that either full conditioning on all children, or use of a distance dependent correlation coefficient between pairs of nodes, do not improve much on the total cost (2). The main reason for which the approximated Slepian-Wolf approach outperforms the explicit communication approach is that Slepian-Wolf approaches allocate much smaller values of rates for nodes far from the sink. However, this is achieved with the cost of increased network knowledge.

VI. CONCLUSIONS AND FUTURE WORK

We presented two complementary approaches for solving the complex problem of joint rate allocation and transmission structure optimization for network correlated data gathering, and good approximation algorithms for this task. We showed how the rate allocation depends on the chosen approach, and assessed the complexity of implementing each of these approaches. Directions of future work include the combination of the two approaches for further cost improvements, and the case of general traffic matrices.

REFERENCES

[1] G. Pottie and W. Kaiser, "Wireless integrated sensor networks," *Communications of the ACM*, no. 43, pp. 51–58, 2000.
[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–116, 2002.
[3] C. Perkins, *Ad Hoc Networking*. Adison-Wesley Pub Co, 2000.
[4] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Information Theory*, no. IT-19, pp. 471–480, 1973.
[5] T. Cover and J. Thomas, *Elements of Information Theory*, ser. Wiley Series in Telecommunications. John Wiley and Sons, Inc., 1991.
[6] S. Pradhan, "Distributed source coding using syndromes (discus)," Ph.D. dissertation, U.C. Berkeley, 2001.
[7] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," in *Proc. IEEE Data Compression Conference*, March 1999.
[8] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rd International Conference on System Sciences (HICSS '00)*, January 2000.
[9] S. Lindsey, C. S. Raghavendra, and K. Sivalingam, "Data gathering in sensor networks using the energy*delay metric," in *Proc. of IPDPS Workshop on Issues in Wireless Networks and Mobile Computing*, April 2001.
[10] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," *IEEE/ACM Trans. on Networking*, vol. 11, no. 1, February 2003.
[11] B. Krishnamachari, D. Estrin, and S. Wicker, "Modelling data centric routing in wireless sensor networks," in *USC Computer Engineering Technical Report CENG 02-14*, 2002.
[12] A. Goel and D. Estrin, "Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk," in *ACM-SIAM Symposium on Discrete Algorithms*, 2003.
[13] J. Edmonds, "Maximum matching and a polyhedron with 0,1-vertices," *J. Res. Nat. Bur. Standards 69B*, pp. 125–130, 1965.
[14] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge Univ. Press, 1991.
[15] E. Lawler, J. Lenstra, A. R. Kan, and D. Shmoys(ed.), *The traveling salesman problem*. Wiley, 1990.
[16] C. Reidys and P. Stadler, "Combinatorial landscapes," *SIAM Review*, vol. 44, pp. 3–54, 2002.
[17] M. Garey and D. Johnson, *Computers and intractability*. W.H. Freeman, 1979.

APPENDIX A
PROOF OF *Theorem 1*

Proof: First, we prove that (5) is indeed a feasible solution for (4), that is, it satisfies all the constraints given by (3). Consider any constraint from (3), for some subset $Y \in X$. Denote by $Q = |Y|$ the number of elements in Y. Order the indices of $X_i \in Y$ as $i_1, i_2, i_3, \ldots i_Q$, with i_1 closest and i_Q furthest from the sink on the SPT.

We rewrite the left-hand-side in terms of the solutions we give in the theorem:

$$\sum_{i \in Y} R_i = H(X_{i_Q}|X_{i_Q-1}, \ldots, X_1) +$$
$$H(X_{i_{Q-1}}|X_{i_{Q-1}-1}, \ldots, X_1) +$$
$$\cdots + \cdots + \ldots \qquad (8)$$
$$H(X_{i_1}|X_{i_1-1}, \ldots, X_1)$$

and expand the right-hand-side terms with the chain law for conditional entropies:

$$H(Y|Y^C) = H(X_{i_Q}|Y^C \cup \{Y - \{X_{i_Q}\}\}) +$$
$$H(X_{i_{Q-1}}|Y^C \cup \{Y - \{X_{i_Q}, X_{i_{Q-1}}\}\}) +$$
$$\cdots + \cdots + \ldots$$
$$H(X_{i_1}|Y^C \cup \{Y - \{X_{i_Q}, \ldots, X_{i_1}\}\})$$
$$= H(X_{i_Q}|X - \{X_{i_Q}\}) +$$
$$H(X_{i_{Q-1}}|X - \{X_{i_Q}, X_{i_{Q-1}}\}) +$$
$$\cdots + \cdots + \ldots \qquad (9)$$
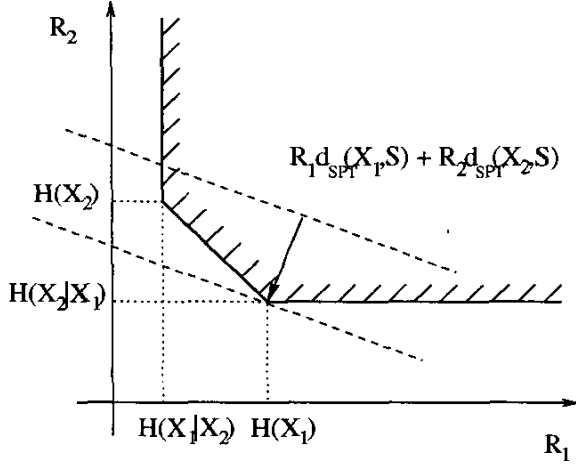$$H(X_{i_1}|X - \{X_{i_Q}, X_{i_{Q-1}}, \ldots, X_{i_1}\})$$

Fig. 8. A simple example with two nodes. The total weights from sources X_1, X_2 to the sinks, are respectively $d_{SPT}(X_1, S), d_{SPT}(X_2, S)$, $d_{SPT}(X_1, S) < d_{SPT}(X_2, S)$, in this particular case. In order to achieve the minimization, the cost line $R_1 d_{SPT}(X_1, S) + R_2 d_{SPT}(X_2, S)$ has to be tangent to the most interior point of the Slepian-Wolf rate region, given by $(R_1, R_2) = (H(X_1), H(X_2|X_1))$.

Consider the terms on the right-hand-side in expressions (8) and (9). It is clear that for any $i_k \in \mathbf{Y}$, the term corresponding to X_{i_k} in (9) is at most equal to its counterpart in (8). This is because the set of nodes on which the entropy conditioning is done for each term in (8), is a subset of its counterpart in (9). Since the choice of $\mathbf{Y}$ was arbitrary, then any constraint in (3) is satisfied by the assignment (5).

On the other hand, note also that the rate allocation in (5) satisfies with equality the constraint on the total sum of rates:

$$\sum_{i \in \mathbf{X}} R_i \geq H(X_1, \dots, X_N). \qquad (10)$$

So we have proven that (5) is a valid rate allocation. We have to prove now that the assignment in (5) makes the expression to be minimized in (4) smaller than any other valid assignment. In Fig. 8 we illustrate with an example involving only two nodes how the cost function is indeed minimized with such a rate allocation. The assignment (5) corresponds in this case to the point $(R_1, R_2) = (H(X_1|X_2), H(X_2))$.

We prove this by recursion. Note first that the rate allocation to node N is minimal. That is, we cannot allocate to X_N less than $H(X_N|X_{N-1}, X_{N-2}, \dots, X_1)$ bits, due to the Slepian-Wolf constraint corresponding to $\mathbf{Y} = \{X_N\}$. Assume now that a solution that assigns $H(X_N|X_{N-1}, X_{N-2}, \dots, X_1)$ bits to X_N is not optimal, and X_N is assigned $H(X_N|X_{N-1}, \dots, X_1) + b$ bits. Due to (10), at most b bits in total can be extracted from the rates assigned to some of the other nodes. But since $d_{SPT}(X_N, S)$ is the largest coefficient in the optimization problem (4), it is straightforward to see that any such change in rate allocation increases the cost function in (4). Thus assigning $R_N = H(X_N|X_{N-1}, \dots, X_1)$ bits to X_N is indeed optimal.

Consider now the rate assigned to X_{N-1}. From the rate

constraint corresponding to $\mathbf{Y} = \{X_{N-1}, X_N\}$, it follows that:

$$\begin{aligned} R_N + R_{N-1} &\geq H(X_N, X_{N-1}|X_{N-2}, \dots, X_1) \\ &= H(X_N|X_{N-1}, X_{N-2}, \dots, X_1) + \\ &\quad + H(X_{N-1}|X_{N-2}, \dots, X_1) \end{aligned}$$

Since for optimality R_N must be given by $R_N = H(X_N|X_{N-1}, X_{N-2}, \dots, X_1)$, it follows that $R_{N-1} \geq H(X_{N-1}|X_{N-2}, \dots, X_1)$. Following a similar argument as for X_N, we can show in the same way that the optimal solution allocates $R_{N-1} = H(X_{N-1}|X_{N-2}, \dots, X_1)$. The rest of the proof follows similarly by considering in order the constraints corresponding to the subsets $\mathbf{Y} = \{X_i, X_{i+1}, \dots, X_N\}$, with $i = N - 2, N - 3, \dots 1$. $\blacksquare$

APPENDIX B
PROOF OF *Theorem* 2

Proof: 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 [17], whose decision version is defined as follows:

Definition 3: Set cover.

- *Instance:* A collection C of subsets of a finite set P and an integer $0 < K \leq |C|$, with $|C|$ the cardinality of C.
- *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. 9 illustrates the construction of the graph instance for our problem. The resulting graph is formed of three layers: a root 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 4 nodes x_1, x_2, x_3, x_4, as in Fig. 10 (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). The node x_3 is linked to the root 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. 9, 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. 9 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.
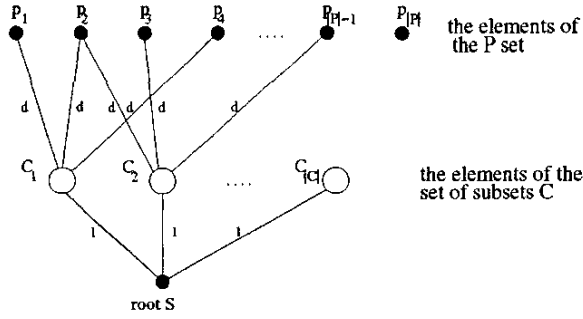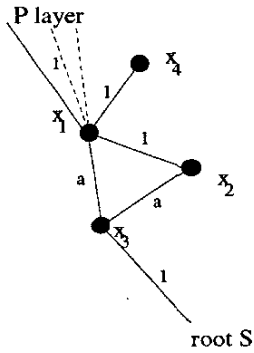
Fig. 9. A graph instance.

Fig. 10. Inner structure of each subset C_i.

Fig. 11. The three possible gathering patterns for the substructure C_i.

The goal is to find a spanning tree for this graph, for which the cost in (7) 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 root 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 be 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
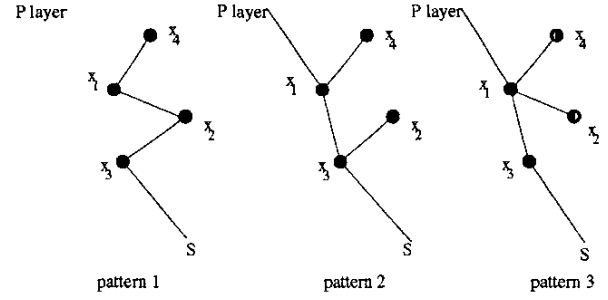
higher than M. The intuition is that 'detours' via the (x_1, x_2) edges are worthy from the point of view of cost reduction only if the flow that goes through node x_1 comes exclusively from node x_4 and no flow from the P layer goes through x_1. If some flow from the P layer joins as well, then the 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. 11) 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 root 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).

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-hard and our optimization problem NP-complete. ∎