Concurrent and Distributed Projection through Local Interference for Wireless Sensor Networks

Xabier Insausti[1], Pedro M. Crespo[1], Baltasar Beferull[2], and Javier Del Ser[3]

[1] CEIT and TECNUN (University of Navarra). 20018 Donostia-San Sebastián, Spain
{xinsausti,pcrespo}@ceit.es
[2] University of Valencia. 46980, Paterna (Valencia), Spain
baltasar.beferull@uv.es
[3] TECNALIA Research and Innovation. 48170 Bilbao, Spain
javier.delser@tecnalia.com

Abstract. In this paper we use a gossip algorithm to obtain the projection of the observed signal into a subspace of lower dimension. Gossip algorithms allow distributed, fast and efficient computations on a Wireless Sensor Network and they can be properly modified to evaluate the sought projection. By combining computation coding with gossip algorithms we proposed a novel strategy that leads to important saving on convergence time as well as exponentially decreasing energy consumption, as the size of the network increases.

Keywords: Wireless Sensor Networks, Computational Codes, Signal Subspace Projection, Neighborhood Gossip.

1 Introduction

The fast spreading of wireless sensor networks has recently encouraged researchers to design and develop fast and efficient algorithms for such networks. The most common approach for computation and information exchange in wireless sensor networks has been done by using a class of decentralized algorithms known as *Randomized Gossip Algorithms* [1]. Wireless sensor networks are characterized for having very particular properties, of which we can highlight the limited computation power and energy resources. Besides, such networks usually do not have a centralized entity that synchronizes communication and therefore the knowledge that nodes have about the topology of the entire network is very limited.

In order to be more precise, consider a network composed of N sensors distributed randomly (uniformly) within the unit area circle. Sensors are assigned a limited transmission power P_T per source symbol. Therefore, they can communicate reliably with a certain number of neighbors within their coverage area, which will be the set of sensors located to a distance less than d (that depends on P_T). Let $\mathcal{N}_d(i) \subset \{1, \ldots, N\}$ denote the local neighborhood of node i, i.e., the set of nodes within distance d of node i.

J. Del Ser et al. (Eds.): MOBILIGHT 2011, LNICST 81, pp. 109–119, 2012.

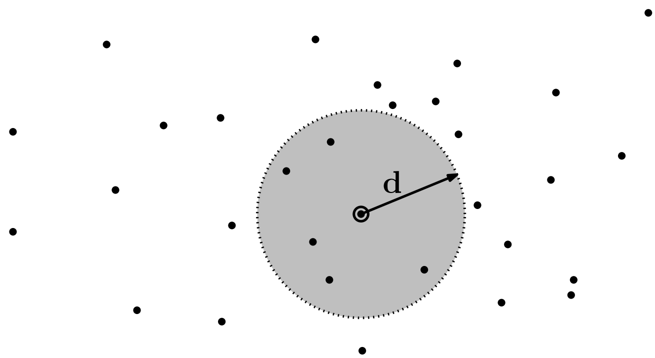

Fig. 1. Sensor network with N nodes. The gray circle is the local neighborhood of the active node, which is the set of sensors within distance d of the active node.

Randomized gossip algorithms have been mainly used for computing the average in an arbitrarily connected network of nodes as in figure 1 in a decentralized fashion. These algorithms are proven to be fast, efficient and to have a low computational cost and their operating basics are as follows: In the t^{th} time slot, a sensor awakens at random and randomly chooses another sensor within its neighborhood, setting their values equal to the average of their current values. As $t \rightarrow \infty$ the entire network will converge to the desired average. For a proof and for further details refer to the work of Boyd et al. [1].

In a recent work, Nazer et al. [2][3] have used a new coding technique, known as *computation coding* [4], which together with a modification of a randomized gossip algorithm, leads to both time and energy savings in computing the average. Nazer's algorithm, called *neighborhood gossip*, is based on the following modifications of Boyd's algorithm: A node wakes up randomly in the t^{th} gossip round and it requires all of the nodes within its neighborhood to transmit their values using a computation code that is designed so that the central node receives only the average of the values. The central node uses the received information and its own value to compute the average of the entire neighborhood, broadcasting the updated value to all nodes in its neighborhood. That way, the entire neighborhood gets the average in a single gossip round.

In this work, we are interested in using the random gossip technique to solve a much more general problem than averaging. We are looking for the projection of the observation vector into a subspace. Barbarossa was the first to propose a decentralized technique to perform the projection minimizing convergence time in [5]. We will use computation coding in order to achieve both time and energy savings with respect to Barbarossa's technique.

Computing the projection of the observed signal into a subspace of a known dimension is very useful for reducing the measurement noise. In general, the observation of a single sensor may be unreliable due to noise or malfunctioning. But, if the environment we are monitoring is smooth, the projection into a subspace will improve the reliability of the observation thanks to the interaction among nodes. Mathematically speaking, being smooth means that the signal exhibits spatial correlation and therefore belongs to a subspace of dimension smaller than the number of nodes.

Consider for instance that the network composed of N sensors is measuring a signal that can be expressed in a Fourier basis of dimension r, $r < N$. Then, the projection of the observed signal (which lies in a space of dimension N) into a subspace of dimension r will lead to very important noise reduction. The goal of the current work is to compute the projection achieving both time and energy savings.

The remainder of the paper is organized as follows: Section 2 precisely defines the framework of the current problem. In section 3 a valid strategy for performing neighborhood gossip is concluded based on the general gossip algorithm proposed by Boyd et al. Section 4 introduces Computation Coding and states the conditions for the computation code to exist and to work properly in our framework. Section 5 compares the gain of the algorithm when it uses computation coding and finally, Section 6 concludes the paper.

2 Problem Statement

Given the random wireless sensor network introduced in section 1, we consider that sensors do not have any geographic information available, although they know the sensors within their local neighborhood. Sensors have an initial observation $\mathbf{z} \triangleq (z_1, \ldots, z_N)^T$, which is a version of the useful signal vector $\boldsymbol{\xi} \in \mathbb{R}^N$ corrupted by some additive noise vector $\mathbf{v} \in \mathbb{R}^N$, that is,

$$\mathbf{z} = \boldsymbol{\xi} + \mathbf{v} \tag{1}$$

where, z_i denotes the observation taken by sensor $i \in \{1, \ldots, N\}$. Furthermore, it will be assumed that the useful signal lies in some subspace of dimension $r < N$. That is, $\boldsymbol{\xi} = \mathbf{U}\mathbf{s}$ for some $\mathbf{s}$, where $\mathbf{U}$ is an $N \times r$ matrix representing a basis of the r-dimensional useful signal subspace.

The least square estimator of the $\boldsymbol{\xi}$ is $\widehat{\boldsymbol{\xi}}$, which is given by the orthogonal projection of the observed vector onto the subspace spanned by the columns of $\mathbf{U}$ as shown in Figure 2. Without any loss of generality, we consider that the columns of $\mathbf{U}$ are orthonormal and therefore, the projector can be written as

$$\widehat{\boldsymbol{\xi}} = (\widehat{\xi}_1, \ldots, \widehat{\xi}_N)^\top = \mathbf{U}\mathbf{U}^T\mathbf{z} \tag{2}$$

Now the goal is to compute $\widehat{\xi}_i$ at each node of the network $i \in \{1, \ldots, N\}$ in a decentralized fashion by performing neighborhood gossip using computation codes, so that both time and energy savings can be achieved. For that purpose, we seek to obtain the estimate $\widehat{\boldsymbol{\xi}}$ iteratively by using the following recursive dynamics

$$\widehat{\boldsymbol{\xi}}[t+1] = \mathbf{W}\widehat{\boldsymbol{\xi}}[t], \qquad t = 0, 1, \ldots \qquad \mathbf{W} \in \mathbb{R}^{N \times N} \tag{3}$$

where $\widehat{\boldsymbol{\xi}}[t]$ denotes the value of the estimate at iteration t, $\widehat{\boldsymbol{\xi}}[0] = \mathbf{z}$, and the matrix $\mathbf{W}$ rules the transmissions between sensors at every iteration or time

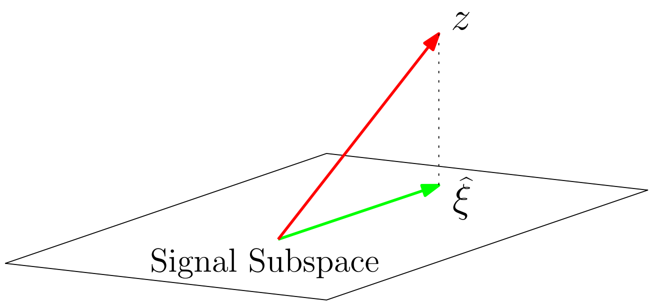

Fig. 2. Projection of the observed vector $\mathbf{z}$ into the target signal subspace. The goal is to have the network compute $\widehat{\boldsymbol{\xi}}$.

instant t. There are multiple choices for $\mathbf{W}$ but we are particularly interested in a sparse one, such that (3) can be performed using neighborhood gossip techniques. Therefore, we set that $w_{ij} = [\mathbf{W}]_{ij} \neq 0$ only if $j \in \mathcal{N}_d(i)$ or $j = i$. In words, the only nonzero elements of $\mathbf{W}$ correspond to pairs of sensors that are within a distance d. We will therefore say that $\mathbf{W}$ satisfies the topology constraint.

It is shown in [5] that, if the transmission power P_T is high enough (and equal for all the sensors), a symmetric and sparse matrix $\mathbf{W}$ exists, which satisfies the topology constraint, minimizes the *convergence time* and makes the network converge to $\widehat{\boldsymbol{\xi}}$, i.e.,

$$\lim_{t\to\infty} \widehat{\boldsymbol{\xi}}[t] = \lim_{t\to\infty} \mathbf{W}^t \mathbf{z} = \mathbf{U}\mathbf{U}^T \mathbf{z} = \widehat{\boldsymbol{\xi}} \tag{4}$$

and therefore,

$$\lim_{t\to\infty} \mathbf{W}^t = \mathbf{U}\mathbf{U}^T \tag{5}$$

Necessary and sufficient conditions for (5) are given next [5]:

Proposition 1.- *Given the dynamical system in* (3) *and the projection matrix* $\boldsymbol{U}\boldsymbol{U}^T$, *the vector* $\widehat{\boldsymbol{\xi}} = \boldsymbol{U}\boldsymbol{U}^T \boldsymbol{z}$ *is globally asymptotically stable for any fixed* $\boldsymbol{z} \in \mathbb{R}^N$, *if and only if the following conditions are satisfied:*

$$\begin{aligned} \boldsymbol{W}\boldsymbol{U}\boldsymbol{U}^T &= \boldsymbol{U}\boldsymbol{U}^T \\ \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{W} &= \boldsymbol{U}\boldsymbol{U}^T \\ \rho(\boldsymbol{W} - \boldsymbol{U}\boldsymbol{U}^T) &< 1 \end{aligned} \tag{6}$$

where $\rho(\cdot)$ *denotes the spectral radius operator. Under these three conditions, the error vector* $\boldsymbol{e}[t] \triangleq \widehat{\boldsymbol{\xi}}[t] - \widehat{\boldsymbol{\xi}}$ *satisfies the following dynamics:*

$$\boldsymbol{e}[t+1] = (\boldsymbol{W} - \boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{e}[t], \qquad t = 0, 1, \ldots \tag{7}$$

Intuitively, the previous proposition can be understood as follows:

– There should be a communication path between any two nodes with at most $N-1$ hops.
– The number of neighbors of each node should not be smaller than the dimension of the signal subspace r for generic subspaces.

In one word, the radius of the neighborhoods d (and so the transmission power P_T) must be large enough to satisfy the convergence conditions.

Provided that the iteration in (3) minimizes the *convergence time* of the network, the goal now is to reduce the energy and time of each local iteration. This can be done using computation coding, which efficiently converts the wireless channel into a set of reliable equations between users [4].

In order to describe the algorithm required for the distributed projection we will rewrite (3) to clearly define the set of equations that should be transmitted within the local neighborhoods:

$$\widehat{\xi}_i[t+1] = \sum_{j=1}^{N} w_{ij}\widehat{\xi}_j[t], \qquad i = 1, 2, \ldots, N \qquad t = 0, 1, \ldots \tag{8}$$

Due to the sparsity of matrix $\mathbf{W}$ (given by the topology of the network and the transmission power) most of its terms will be zero. Actually, the only non-zero terms will be those referring to the local neighborhood and to the sensor itself, and therefore, the previous expression reduces to

$$\widehat{\xi}_i[t+1] = \sum_{j \in \mathcal{N}_d(i) \cup \{i\}}^{N} w_{ij}\widehat{\xi}_j[t], \qquad i = 1, 2, \ldots, N \tag{9}$$

Notice that in order to perform the distributed projection, the i^{th} sensor only needs to know the i^{th} column of $\mathbf{W}$, which in turn means that sensor i only needs to keep $|\mathcal{N}_d(i)| + 1$ values.

We now consider a wireless channel with finite bandwidth so that we can model it as a discrete-time channel. Assuming transmissions occur within the neighborhood, the received signal at time instant $t \in \mathbb{N}$ by node i is

$$\begin{aligned} y_i[t] &= \sum_{j \in \mathcal{N}_d(i)} h_{ij}[t]x_j[t] + n_i[t], \\ h_{ij}[t] &= d_{ij}^{-\alpha/2} e^{j\theta_{ij}[t]} \end{aligned} \tag{10}$$

where d_{ij} is the distance between nodes i and j, $\alpha \in \mathbb{R}_+$ is the power path loss coefficient, $\theta_{ij}[t]$ are the phases, and $x_j[t]$ is the signal transmitted by sensor j at time t. The channel noise samples $\{n_i[t]\}$ are realizations of i.i.d circular symmetric Gaussian random variables with variance σ^2.

As already mentioned, the main goal of our technique is to reduce power consumption. To that end, we propose the use of computation codes for reliable communication among nodes, strategy that will lead to important power gains.

Notice that in order to apply coding, one has to assume that each sensor has an i.i.d sequence of observation (vector) instead of just a single observation, as it has been assumed so far. The proposed strategy can be performed by using a general gossip algorithm as in [1] and it follows on a similar strategy as the one used in [2] and [3] for including computation codes[1].

3 Neighborhood Gossip Algorithm

The most important aspect when computing (9) in a distributed fashion is to try to maintain synchronized the number of iterations at the nodes, so that convergence is guaranteed using as few gossip rounds as possible. In other words, all the estimated quantities $\widehat{\xi}_j$, $j \in \mathcal{N}_d(i)$, available at the nodes inside $\mathcal{N}_d(i)$ should have been updated the same number of times t (i.e., all should be at iteration t), before computing the next iteration $t{+}1$ of $\widehat{\xi}_i$, at node i. This requires a substantial change in the general averaging gossip algorithm proposed by Boyd et al. in [1], and also in the neighborhood gossip algorithm with computation codes introduced by Nazer et al. in [2] and [3]. The reason is that the goal of these algorithms is to obtain a single parameter estimation at all the nodes of the network, mainly the average of the observations, and the convergence to the average does not require to keep the same number of updates for the nodes inside a neighborhood. Therefore, none of those algorithms can be directly applied in the current context.

Let us look at the gossip algorithm proposed in [2]. A node i wakes up randomly and requires all the nodes within its neighborhood $\mathcal{N}_d(i)$ to transmit their observations by using a computation code so that the decoded value at node i is the average of the observations of all the nodes belonging to $\mathcal{N}_d(i)$. Then, node i broadcasts the new computed average to all the nodes in $\mathcal{N}_d(i)$ so that these nodes get in a fast way the new average value in a single iteration or gossip round. This process is repeated until a good estimate of the average is obtained. If one tries to apply this gossip algorithm into our problem, it will fail for two reasons. First, the final broadcasting stage is clearly inapplicable in our problem since we are not looking for a single parameter estimator (average) at all the nodes, but rather for a set of N parameter estimators, one for each node of the network. Second, the system dynamics given in (3) would be violated (i.e., synchronization of iterations at the nodes would fail) whenever a node woke up more times than others within a neighborhood. To better understand this problem let us consider two sensors, s_1 and s_2 within a distance d. Assume that s_1 starts the t^{th} gossip iteration: node s_1 wakes up at random and requires s_2 and the rest of its neighborhood to send their observations. They will be sending their observation that comes from gossip iteration $t-1$. After receiving the messages, s_1 will update its current observation. Now consider that s_2 wakes up and requires their neighbors to send their observations too. All neighbors of node s_2 will be transmitting their last estimation except node s_1, which will be required

[1] However, here the goal is to compute the projection, whereas in [2] [3], the goal was to compute the average of the sensors.

not to send its latest update but the previous one (observation referring to iteration $t-1$). This fact requires that nodes keep track of previous observations for each neighbor and then Boyd's general gossip algorithm can be applied to our problem.

According to [5], the *convergence time* $\tau_0(\mathbf{W})$ of our strategy, defined as the number of gossip rounds required for the error (7) to decrease by a factor $1/e$ for the worst possible initial vector, given that all nodes have performed the same amount of gossip rounds, is given by

$$\tau_0(\mathbf{W}) \triangleq \frac{1}{\ln\left(\frac{1}{\rho(\mathbf{W}-\mathbf{U}\mathbf{U}^T)}\right)}. \quad (11)$$

The problem of finding the matrix $\mathbf{W}$ that minimizes the *convergence time* given a particular network topology can be converted into a convex problem [5] and solved by using classical *Semidefinite Programming* (SDP) tools [6].

4 Computation Coding

The most energy consuming stage is when all nodes in the local neighborhood transmit their corresponding observations to the central node. The key here is to realize that the central node does not need to know the observation of each neighbor. Rather, it only needs to know the weighted sum of the observations. Since the weighting process is already done in the source nodes, the central node only requires the sum of the incoming messages. This can be done very efficiently by using a code construction recently developed in [4] and known as computation coding.

First, we will assume that sensors know the channels in their local neighborhood from themselves to the central node. For node i, this is equivalent to knowing the channel coefficients $(d_{ij}, \theta_{ij}[t])$ in (10) for every $j \in \mathcal{N}_d(i)$. Exploiting this knowledge leads to a simplification of the multiple-access channel and it can be considered to be the following simple multiple-access channel:

$$y_i[t] = \sum_{j \in \mathcal{N}_d(i)} x_j[t] + n_i[t] \quad (12)$$

Now that the channel behaves as a simple adder, we know that we can efficiently and reliably compute sums by using computation coding to improve the performance of the transmissions as in [3]. All nodes in the neighborhood of i encode and send their values using identical linear codebooks. The transmitted codewords will be added on the channel and node i will receive the sum of the codewords. Since the codebook is linear, the sum of the codewords is also a codeword and it is actually the codeword corresponding to the desired sum. Node i has to simply add the received message to its observation weighted by the coefficient w_{ii} to get its desired new observation.

This said, we are ready to state the following theorem from [3], which establishes the required condition for the computation code to exist.

Proposition 2.- *Choose $\epsilon > 0$. Assume each node in a local neighborhood has a length L bounded real-valued weighted observation vector, that is, $\|s_{ij}[t]\|^2 \leq LP_T$ $\forall i, j, t$. For L large enough, there exists a coding scheme such that the receiving node i can make an estimate $\widehat{\xi}_i[t_i+1]$ of the sum $\xi_i[t_i+1]$ as in* (9)*that satisfies*[2]*:*

$$\mathcal{P}\left(\|\xi_i[t_i+1] - \widehat{\xi}_i[t_i+1]\|^2 \geq \frac{P_T}{L} 2^{-2B} \right) < \epsilon \quad \forall i, t_i \tag{13}$$

so long as:

$$T \log \left(\frac{1}{|\mathcal{N}_d(i)|} + \frac{P_T}{(\max_{j \in \mathcal{N}_d(i)} d_{ij})^{-\alpha/2} \sigma^2} \right) > B \tag{14}$$

for some choice of T channel uses per observation symbol and precision B bits.

See [4] for a proof.
The computation code is based on using the same lattice code by all sensors, which is chosen to simultaneously be a good channel and a good source code. The transmitters within a neighborhood quantize their weighted observation vector to the lattice and they transmit such quantization simultaneously. The receiver decodes the sum and makes an estimate of it. Next, the transmitters send their quantization errors using the same lattice code. This continues until the total number of channel uses T is exhausted and the desired precision B is reached.

5 Performance Comparisons

We now compare the performance of the proposed strategy when it uses the best possible separation scheme or when it uses a computation code. We will conclude that we can achieve an exponentially increasing power gain by using computation codes as the density of sensors in the network increases. To that end, we will make use of two theorems from [4], which compute the achievable distortion for sending a Gaussian sum over a Gaussian MAC. Notice that distortion here is again measured by the mean-squared error criterion,

$$D = \max_{i, t_i} \left(\frac{1}{L} \|\xi_i[t_i+1] - \widehat{\xi}_i[t_i+1]\|^2 \right). \tag{15}$$

Proposition 3.- *For T channel uses per observation symbol, $T \in \mathbb{Z}_+$, the following distortion is achievable for sending a Gaussian sum (of M transmitters) over a Gaussian MAC with noise variance σ^2 so long as $P_T > \frac{M-1}{M}\sigma^2$:*

$$D = M\sigma_S^2 \left(\frac{\sigma^2}{\sigma^2 + MP_T} \right) \left(\frac{M\sigma^2}{\sigma^2 + MP_T} \right)^{T-1}, \tag{16}$$

where σ_S^2 is the variance of the Gaussian sources.

On the other hand, we have an interesting result for separation-based schemes.

[2] Notice that $\xi_i[t_i+1]$ refers to the actual sum of the observations and $\widehat{\xi}_i[t_i+1]$ refers to the value that node i gets after T channel uses.

Proposition 4.- *The best achievable distortion for a separation-based scheme for sending a Gaussian sum (of M transmitters) over a Gaussian MAC with noise variance σ^2 is given by:*

$$D = M\sigma_S^2 \left(\frac{\sigma^2}{\sigma^2 + MP_T} \right)^{\frac{T}{M}}, \tag{17}$$

where σ_S^2 is the variance of the Gaussian sources and now the number of channel uses per observation symbol T might not be an integer. See [4] for the proofs.

Since computation coding is optimal for symmetric linear MACs [4], the distortion given in proposition 3 is achievable using computation codes. Although we cannot guarantee that the messages of the sensors follow a Gaussian distribution, we will consider it as the worst case result. Therefore, based on the previous propositions, we can compute the number of channel uses per source symbol that computation codes and the best separation-based scheme need to achieve the given distortion D. Consider that node i wakes up and requires all the nodes within its neighborhood $\mathcal{N}_d(i)$ to transmit their observations. The number of channel uses per symbol is then given by:

$$T_{COMP}(i) = \frac{\log\left(\frac{\sigma_S^2}{D}\right)}{\log\left(\frac{1}{|\mathcal{N}_d(i)|} + \frac{P_T}{\sigma^2}\right)}. \tag{18}$$

$$T_{SEP}(i) = \frac{|\mathcal{N}_d(i)| \log\left(\frac{D}{|\mathcal{N}_d(i)|\sigma_S^2}\right)}{\log\left(\frac{\sigma^2}{\sigma^2 + |\mathcal{N}_d(i)|P_T}\right)} \tag{19}$$

If we consider that all channel uses require the same energy and since we know that both schemes will converge in the same number of gossip rounds, the energy ratio will be given by the difference in the number of channel uses per source symbol.

Nevertheless, both quantities depend on the number of transmitters, i.e. the neighborhood of the current active node $\mathcal{N}_d(i)$, and that quantity is in general a random variable which we will model next.

Consider a circular region of unitary area (radius $1/\sqrt{\pi}$) and distribute N sensors uniformly on it, so that we have N sensors per unit square. The distribution of the distances of the nodes to the center of the circle R will be:

$$f_R(r) = 2\pi r \qquad 0 < r < \frac{1}{\sqrt{\pi}} \tag{20}$$

Now assume without loss of generality that our target node i is on the center of the circle and has a coverage area given by $d < 1/\sqrt{\pi}$. The probability of a sensor falling within the coverage area of sensor i is then given by integrating (20) between 0 and d, which yields $P_d = \pi d^2$. Therefore, the number of neighbors M of node i is given by the set of nodes that fall within its coverage area, which is distributed as

$$P_M(m) = \binom{N}{m} \left(\pi d^2\right)^m \left(1 - \pi d^2\right)^{N-m} \tag{21}$$

We only have to finally compute the average of the channel uses per symbol, which are given by:

$$\overline{T_{COMP}} = \sum_{m=1}^{N} \binom{N}{m} \frac{\log\left(\frac{\sigma_S^2}{D}\right)}{\log\left(\frac{1}{m} + \frac{P_T}{\sigma^2}\right)} \left(\pi d^2\right)^m \left(1 - \pi d^2\right)^{N-m} \tag{22}$$

$$\overline{T_{SEP}} = \sum_{m=1}^{N} \binom{N}{m} \frac{m \log\left(\frac{D}{m\sigma_S^2}\right)}{\log\left(\frac{\sigma^2}{\sigma^2 + mP_T}\right)} \left(\pi d^2\right)^m \left(1 - \pi d^2\right)^{N-m} \tag{23}$$

And therefore the average gain is given by

$$g = \frac{\overline{T_{COMP}}}{\overline{T_{SEP}}} = \frac{\sum_{m=1}^{N} \binom{N}{m} \frac{\log\left(\frac{\sigma_S^2}{D}\right)}{\log\left(\frac{1}{m} + \frac{P_T}{\sigma^2}\right)} \left(\pi d^2\right)^m \left(1 - \pi d^2\right)^{N-m}}{\sum_{m=1}^{N} \binom{N}{m} \frac{m \log\left(\frac{D}{m\sigma_S^2}\right)}{\log\left(\frac{\sigma^2}{\sigma^2 + mP_T}\right)} \left(\pi d^2\right)^m \left(1 - \pi d^2\right)^{N-m}} \tag{24}$$

so long as $P_T \geq \sigma^2$ and $d < 1/\sqrt{\pi}$.

Figure 3 shows the shape of the gain when the density of nodes increases. In general terms, the gain is very little dependent on the distortion but it grows exponentially as the density of sensors increases.

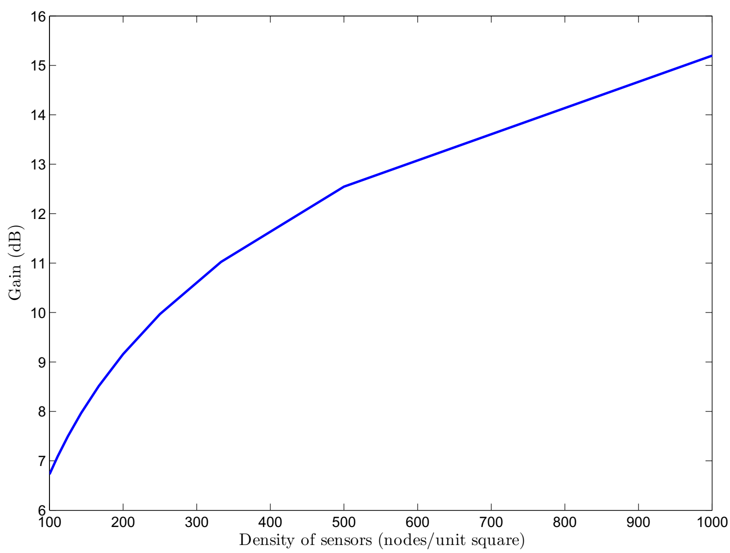

Fig. 3. Energy gain of the projection algorithm with computation codes with respect to the best possible separation-based scheme. $d = 0.2$, $\sigma^2 = 1$, $\sigma_S^2 = 1$, $P_T = 3$, $D = 10^{-5}$.

6 Conclusion

The main goal of the current paper has been to modify a general gossip algorithm by introducing computation coding in order to achieve both time and energy savings in a projection problem. The results that lead to the goal are summarized in the following list:

- The problem of finding the projection of the observed signal into a subspace can be solved in a distributed fashion by solving an SDP problem.
- A general gossip algorithm has been modified to allow the use of computation coding.
- Combining computation coding with a gossip algorithm to perform the projection leads to exponentially increasing power savings when the number of nodes increases.

Acknowledgments. This work was supported in part by the Spanish Ministry of Science and Innovation through the projects COSIMA (TEC2010-19545-C04-02) and COMONSENS (CSD2008-00010), as well as by the Basque Government through the Ph.D. grant 2009/3723.

References

1. Boyd, S., Ghost, A., Prabhakar, B., Shah, D.: Randomized Gossip Algorithms. IEEE Transactions on Information Theory 52, 2508–2530 (2006)
2. Nazer, B., Dimakis, A.G., Gastpar, M.: Local Interference Can Accelerate Gossip Algorithms. In: 46th Annual Allerton Conference, Monticello, IL (2008)
3. Nazer, B., Dimakis, A.G., Gastpar, M.: Neighborhood Gossip: Concurrent Averaging through Local Interference. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan (2009)
4. Nazer, B., Dimakis, A.G., Gastpar, M.: Computation over Multiple-Access Channels. IEEE Transactions on Information Theory 53, 3498–3516 (2007)
5. Barbarossa, S., Scutari, G., Battisti, T.: Distributed Signal Subspace Projection Algorithms with Maximum Convergence Rate for Sensor Networks with Topological Constraints. In: IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan (2009)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)