A Multi-agent Approach to Energy-Aware Wireless Sensor Networks Organization

M. del Carmen Delgado-Roman and Carles Sierra

Artificial Intelligence Research Institute (IIIA),
Spanish Scientific Research Council (CSIC), Universitat Autònoma de Barcelona,
Bellatera E08193, Barcelona, Spain
{delgado,sierra}@iiia.csic.es

Abstract. Wireless Sensor Networks when deployed in inaccessible or remote areas require sensing and communication algorithms that minimise energy consumption. This is needed to reduce battery replacement costs. At the same time, the information transmitted to the sink has to be good enough in order to make timely decisions on the environmental hazards being monitored. Sensor algorithms have to thus balance quality of information with energy consumption. We introduce in this paper an algorithm that uses multiagent co-ordination technology to organize the sensors in coalitions that share the burden of sensing and communicating. We provide experimental evidence of a good balance between information quality and energy consumption on a simulated river pollution phenomenon.

Keywords: Wireless Sensor Networks, Sensor Coalitions, Resourse Saving Strategies.

1 Introduction

Wireless Sensor Networks (WSNs) are networks composed of battery-operated sensing nodes that are often deployed in remote and hostile environments. The cost of replacing or recharging their batteries can become astronomical and, quite often, this is the reason that hinders their actual deployment. Sometimes, the replacement of the batteries is simply not possible. As the phenomena to monitor usually show large local variability, these networks have to be formed by a large quantity of sensing nodes. Hence, the cost of each node has to be necessarily low and therefore the battery has to be of limited capacity. AI techniques and in particular Multiagent systems (MAS) techniques can help in reducing energy consumption and thus enlarge the life span of these networks.

From a MAS perspective, agents co-operate within a group in order to share resources or reach shared goals that cannot be achieved individually. A whole range of different coalition formation (CF) mechanisms have been proposed for different kinds of applications and network nodes. In this paper we propose a new CF algorithm for homogeneous nodes in a sensor network that allows to extend the useful life time of the network by avoiding redundant sensing and

C.I. Chesñevar et al. (Eds.): AT 2013, LNAI 8068, pp. 32–47, 2013.

transmission. This CF algorithm is based on the nodes' state and the conditions of the environment. There is no intervention of any central authority and the algorithm is fully distributed and embedded in the node's behaviour. Saving energy is achieved by allowing nodes in a coalition to delegate their sensing tasks to a distinguished member of the coalition, while restricting the maximum information loss, so that the initial purpose of the system: faithfully monitoring the environment, is not missed. Although this CF algorithm has been originally conceived for networks of homogeneous nodes, it could also work for heterogeneous networks as long as the elements can communicate among them as the algorithm is based on individual knowledge and agents' state. Nonetheless, the performance of the algorithm would probably present higher variability due to the particular characteristics of each of the nodes in the system.

Negotiation among nodes [1] or the individual adoption of a global policy [2] have been considered to introduce CF techniques into distributed WSN. The evolution of these coalitions have been studied based on different aspects of the coalition, such as its members characteristics or the resulting performance [3–5]. Differently from us, none of these approaches took into account the energy consumption nor the cost derived from the rewiring policies.

Saving energy has been the focus of some clustering algorithms proposed for WSNs, such as LEACH [6], EEHC [7] and HEED [8]. These algorithms divide the sensor network into a set of non-overlapping clusters, each with a cluster head in charge of sending the data collected within the group to the sink. Although these approaches are distributed our approach fundamentally differs from them in the way the cluster head is chosen. For this decision, we take into account the characteristics of the node, its state and the model of the node maintained by its neighbours, instead of doing it randomly or just based on the node's available energy, as previous works do. A recent centralised approach to this clustering problem is presented in [9]. In this case, the sink determines the cluster heads and runs an algorithm to reduce the amount of transmitted data. In contrast, in our approach, nodes make autonomous decisions, thus reducing the coordination communication costs to and from the sink.

Different approaches not based on coalitions have also been proposed in order to extend WSNs' lifetime. The work of [10] focuses on the individual nodes' sampling regime, while the work of [11] reduces the energy consumption of the system by improving the sensor node hardware and software design. However, none of these works use coordination among the nodes in the system.

The rest of the paper is organized as follows. Section 2 introduces our algorithm. The environment simulation model that we have used to test it is described in Section 3. Section 4 presents the experimental results obtained and finally, conclusions and future work are discussed in Section 5.

2 Algorithm Description

The standard behaviour of a sensor in a WSN consists of sampling the environment according to a pre-established frequency and transmitting the data to a server. This basic behaviour wastes energy when the environment does not change, and misses information when it changes quickly. The objective of the *Coalition Oriented Sensing Algorithm* (COSA) proposed here is to radically improve this situation as we explain next.

The core of COSA lies in the establishment of coalitions among nodes through peer to peer negotiation. The resulting coalition structure depends at any time on the network topology, the state of the nodes and the environment. As WSNs are deployed in dynamic environments, coalitions' configuration in the system will change along time. The use and interpretation of the available information for a node is the key activity of COSA.

2.1 COSA

COSA modifies the standard node sampling behaviour by making the node autonomous, proactive and reactive. To achieve this behaviour, COSA relies on a simple negotiation protocol and two functions modelling graded relationships: *adherence* and *leadership*. The numerical degrees of these relationships determine the asynchronous dialogue in which nodes engage when negotiating, and at the same time the result of negotiations modify the numerical degrees of these relationships.

The basic idea of the algorithm is simple. When a node samples the environment, it sends the observed value to its neighbouring nodes. A node receiving a sample from a neighbour uses this information to evaluate the adequacy of forming a group. If this evaluation is positive, it tells the neighbour, who offers itself to work for the two of them (assuming the role of *leader* of the coalition). If they both agree one node becomes the leader while the other, called *dependant*, can sleep and stop its sampling and sink transmission tasks. Therefore, simple negotiations between neighbours situated one-hop distance away, lead the nodes to select their preferred role and build a coalition structure in a bottom-up fashion. Adopting the best system's organization translates into energy savings by avoiding unnecessary long-distance transmissions: those of the *dependant* nodes.

Algorithm 1 represents a simple view of the thread of action of a node implementing COSA as explained above.

Algorithm 1. COSA: Node basic behaviour

```
while energy > 0 do
    environment sampling;
    environment model update;
    relationship to neighbours update;
    social network update;
end
```

Given a set of sensing nodes A, called agents henceforth, the value of the functions $adh : A \times A \to \mathbb{R}$ and $lead : A \times 2^A \to \mathbb{R}$ changes along time depending, among other factors, on the value of the observed variable by the agents. In this work, we assume that the observed variable follows a Normal distribution, $\mathcal{N}$, as this is a common model for natural phenomena observations [12].

The *adherence* of an agent i to an agent j is a measure that indicates how much agent i intends to take part of a group led by agent j. Its definition takes into account the similarity between the values observed by the agents and, also, how *certain* is j's about its variable model (measured in terms of its entropy, H_j). The more similar the values the more adherence and the more certain j is about its monitored variable value the more adherence. These two multiplying factors can be identified in equation (1).

$$adh(a_i, a_j) = \frac{p(x_i, \mathcal{N}_j(\bar{x}_j, \sigma_j))}{p(\bar{x}_j, \mathcal{N}_j(\bar{x}_j, \sigma_j))} \cdot \left(1 - \frac{e^{H_j} - e^{H_{min}}}{e^{H_{max}} - e^{H_{min}}}\right) \tag{1}$$

The evaluation of the *leadership* attitude, unlike the *adherence*, does not only take into account the two negotiating agents. It also considers the relationships previously established by the node offering itself as a leader. $P(a_i)$ represents the set of nodes depending on a_i, together with a_i and the negotiating neighbour a_j. A good leader has to be a good representative of its neighbours and needs also enough energy to sense and communicate with the sink. Thus, the willingness of a_i to act as a leader of $P(a_i)$ depends on three factors that can be identified in equation (2). The first factor, *prestige*, is an average of the adherence level of the members of $P(a_i)$ towards a_i. The second factor, *capacity*, considers the available energy of the node to act as a leader. This value is a proportion of the current energy level of the node $E(a_i)$ minus the security energy level E_{sl} (energy needed to send the last disconnection message) over the maximum energy level of the battery E_{max}. Finally, the last factor, *representativeness*, indicates how well the potential leader's measurement fits as a representative of the potential group nodes' measurements. Thus, a_i characterises the set of values received together with its own value, that is, the set $\{x\}_{P(a_i)}$, with their mean and standard deviation, noted as $(\bar{x}_{P(a_i)}, \sigma_{P(a_i)})$. To encourage the formation of groups with very similar measurements, an exponential function establishes the divergence growing ratio. Those potential groups whose measurement distribution is very disperse are also penalized through the inclusion of the Pearson's coefficient $(CV_{P(a_i)})$ in the equation. Equation (2) presents the leadership capacity of an agent a_i for a potential group $P(a_i)$:

$$lead(a_i, P(a_i)) = \frac{\sum_{a_j \in P(a_i)} adh(a_j, a_i)}{|A|} \cdot \frac{E(a_i) - E_{sl}}{E_{max}} \cdot \frac{1}{e^{|x_i - \bar{x}_{P(a_i)}| CV_{P(a_i)}}} \tag{2}$$

COSA is designed with a set of parameters that constraints the agent's actions. These parameters are: $\langle d_{max}, \sigma_{min}, \sigma_{max} \rangle$. The first one puts a limit on the maximum difference between agents' samples to allow establishing an adherence relationship ($\|x_j - x_i\| \leq d_{max}\sigma_j$). This maximum difference is proportional to the neighbour's σ_j to take into account the shape of its distribution. $(\sigma_{min}, \sigma_{max})$ demand a level of certainty to the neighbour's model of the environment. σ_{min} corresponds to very precise models whether σ_{max} represents wider distributions (correspondingly, entropy values of H_{min} and H_{max}). Evaluating adherence values to a neighbour whose variable's model is over this range implies admitting

Algorithm 2. Message Processing

Data: me: focus node; a_j: generic neighbour; a_l: potential leader; a_r: potential dependant on me; a_p: dependant node on me; a_L: leader node of me; $D(me)$: set of dependant nodes on me

```
case rcvd(inform(a_j, me, meas, t))
    updateNeighbourInfo();
    adherence2NeighbourEvaluation();
    updateOwnMaxAdherence();
    if changesOnOwnMaxAdherence then
        inform(me, a_l, maxAdh, t);
    end
end
case rcvd(inform(a_j, me, maxAdh, t))
    inform(me, a_r, lead);
    updateNeighbourInfo();
    adherence2NeighbourEvaluation();
    updateOwnMaxAdherence();
    if changesOnOwnMaxAdherence then
        inform(me, a_l, maxAdh, t);
    end
end
case rcvd(inform(a_l, me, lead))
    if checkAgainstOwnLead then
        firmAdherence(me, a_l);
    end
end
case rcvd(firmAdherence(a_r, me))
    if checkAgainstOwnLead then
        ackAdherence(me, a_r);
        updateOwnLeadValue();
        updateDependentGroup();
    end
end
case rcvd(ackAdherence(a_l, me))
    if !leader ∧ a_l! = a_L then
        withdraw(me, a_L);
    end
    if leader ∧ D(me)! = ∅ then
        while D(me)! = ∅ do
            break(me, a_p);
        end
    end
    updateRoleState(dependant);
    sleep(t);
end
case rcvd(break(a_L, me))
    updateRoleState(leader);
end
case rcvd(withdraw(a_p, me))
    D(me) ← D(me)\a_p;
    updateRoleState(leader);
end
```

dissimilar agents' values in a coalition. This would imply larger errors at the sink but larger and more stable coalitions (i.e. more energy savings).

Agents implementing COSA exchange information via performatives and using a classical alternating negotiation protocol. Figure 1 shows a simple example of a negotiation between two agents that finishes in a group establishment. The set of performatives considered is: *inform*, that indicates the transmission of data; *firmAdherence*, that expresses the desire of the sending agent to adhere to the addressee agent; *ackAdherence*, that is an acknowledgement to a previously received *firmAdherence* message; *break*, that allows a leader agent to break a leadership relationship and, finally, *withdraw*, which is the message sent by a dependant agent to break a leadership relationship. All these performatives are used in Algorithm 2. The meaning of each procedure is rather self-explanatory.

When a message is received, its processing implies the update of the internal model of the agent about its leadership value and about the model of the agent sending the message. Depending on these updates, new messages can be

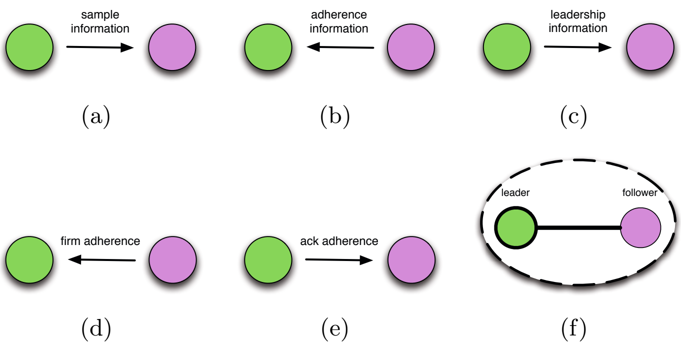


Fig. 1. Negotiation protocol stages

transmitted back to the sender or to other agents in the network. The code allows for the intermingling of dialogues with different neighbours.

2.2 COSA Strategies

We propose here two strategies that are used by COSA (in isolation or combined) and that produce different CF behaviour. This change in behaviour influences the balance between the energy consumption and the overall observation error of a WSN. The strategies are:

– *Sampling Frequency.* This strategy alters the sampling frequency of the leader agents based on the number of dependants. Specifically, the sampling frequency doubles its value in a group with 4 or more components. The aim of this simple adaptive sampling is to increase the reaction capacity of the network, as agents will detect changes in the environment sooner.
– *Coherence.* This strategy checks whether the leadership of an agent is still coherent with the last sampled values of the current members of the coalition. The leader agent works for its group members. While they are asleep, the leader samples the environment and updates its variable's model. However, if the current model differs from the model that the leader had when a member of the coalition joined in, it is unclear whether the node in sleeping mode would still be willing to stay within the coalition. Thus, when the difference is significant enough, over a threshold V_T, the leader proactively wakes up the agent so that it can sample and decide again which coalition to join. Drifts in the sensed values of a leader makes this agent wake its dependant nodes as the coalition raison d'être (similarity of sampled values) may be at stake. This behaviour makes the system react quicker to changes in the environment.

Both strategies increase the sensing and thus the energy consumption with respect to the basic COSA operation while still satisfying the main objective: consuming energy only when that consumption keeps the error of the monitored variable under limits. The computational effort implied by these strategies is negligible although certainly the number of messages exchanged and the number of sampling actions taken by leaders and coalition members grow. Despite of this, these strategies increase the robustness of the algorithm as they allow a quicker detection of nodes' measurements deviation. These strategies do not mend nodes' malfunction, but they do reduce the effect that this can cause on the system's global error.

3 Simulation Model

To test our algorithm we have used the simulation environment RepastSNS [13]. It is an event-based simulator that, although being quite general, results specially appropriate for the study of WSNs from a MAS perspective. One of the main advantages of this environment is that it provides a scalable and extensible infrastructure to build up networks of basic WSN components. Therefore, it allows different application domains to be tested over it.

We consider a WSN deployed along a river, whose sensor nodes sample the presence of *hydrocarbons* in the environment. The set of simulation elements include: a river and its water flow, pollutant releases, a set of sensing nodes and a sink node. The river is a rectangular section 50 km long by 2 km wide represented as a discrete grid. The river flows according to the following expression: $River(x, y) = (1 - \rho)River(x, y) + \rho(\alpha(River(x - 1, y - 1)) + \beta(River(x, y - 1)) + \gamma(River(x + 1, y - 1)))$, where ρ is the sedimentation factor and α, β, γ determine the horizontal diffusion of pollutants.

The pollutant phenomenon considered in the simulation appears as an intensity-oscillating stain near the sink. Its spewing pace follows a sine function and lasts for the whole simulation time. The pollutant spreads along the river due to the river flow and can, therefore, be sensed by different nodes of the WSN. Our sensor nodes satisfy the communication and processing properties of actual Waspmote nodes [14]. In particular, we simulate Waspmote's energy consumption. Finally, the sink node represents the central monitoring station that receives the information sent from the sensing nodes. It is situated at the upstream extreme of the network.

To test the behaviour of the algorithm we assess its performance in two different scenarios. In the first scenario, the network configuration consists of 50 nodes that are evenly-spaced as a chain along the river. In this case, the network covers the whole river. The second scenario shows a completely different configuration. It consists of 30 nodes situated at the end of the river. Nodes are deployed in a grid distribution of three nodes per row. The same horizontal distance between each pair of nodes is also the distance between rows. This configuration shows two important characteristics that differentiates it from the first scenario. First, the energy cost of transmitting information to the sink increases, as all nodes are at a considerable distance to the sink. Second, as nodes are situated near each other, groups of a high number of nodes can be formed.

To completely define the experimental setup considered, Table 1 presents the values assigned to the COSA algorithm parameters.

Table 1. Parameter values

Parameter	Value
Sampling frequency	10min
d_{max}	1.75
σ_{min}	0.0005
σ_{max}	6
Node sleep time	1day
V_T	0.95

4 Experimental Results

To show the properties of COSA, we performed a set of experiments in the scenarios described in Section 3. Our goal is to study the performance of a WSN in terms of energy usage and accuracy of the data reported to the sink. We will compare the behaviour of COSA using different strategies with the performance resulting from a Random sampling schedule. According to a Random sampling schedule, every agent takes a sample from the environment at a random time instant within the sampling frequency and then, it transmits the observed value directly to the sink.

To evaluate the **energy consumption** of the system, we compute the median of the nodes' energy values at every time instant. This measure gives information on the number of nodes that are still alive in the network (with $E(a_i) > 0$) and also about their battery discharge pattern.

The quality of information is assessed using two measures: error and entropy. The **error** is measured as the difference between the sink's current known observed value for each alive node and the actual pollution value —as we are simulating, we know the exact value at any instant. The error measures how wrong the sink is with respect to the actual situation in the river. It is computed as:

$$e^t = \sum_{i \in N^t} \|xs_i^t - xp_i^t\| \tag{3}$$

where $N^t \subseteq A$ represents the set of nodes that have not depleted their batteries at time t; xs_i^t, the value known by the sink for the pollutant level at node i at time t and xp_i^t is the actual level of the pollutant at time t and node i.

The **information quality** is computed as the addition of the entropy value associated to every node in the network (computed according to Equation 4, [15]). This entropy value increases with the time spent since a node's last sampling action.

$$H_i(t) = \ln(\sigma_i(t) \cdot \sqrt{2\pi e}) \tag{4}$$

The information of the sink is thus considered as the addition of the entropy of the distributions of all nodes in the network, alive or dead. Being dead means that the sink stops receiving information from the node and thus its distribution should be that of complete ignorance —complete ignorance would equate to a flat distribution with a very large σ. We model this with a time decay function over $\sigma_i(t)$ as follows:

$$\sigma_i(t) = \begin{cases} \sigma_{bot} & \text{if } t = t_i \\ \sigma_{bot} + \frac{e^{t-t_i}}{e^{t_{max}}} \cdot (\sigma_{top} - \sigma_{bot}) & \text{if } t \neq t_i \end{cases}$$

where t_i is the time instant of the last value received from node i; σ_{bot} is the variance of the gaussian noise that the simulator adds to each sensor reading; and σ_{top}, in the order of $100\sigma_{bot}$, represents a very large variance that models maximum ignorance, i.e. a flat distribution. The parameter t_{max} is set to three times the sampling period. Receiving no information from a node for this amount of time would mean a node failure or a serious malfunction.

The following set of figures presents the results obtained for the evaluation of these measurements in the previously described scenarios. Figures 2 and 4 summarise the percentage of gain obtained by COSA with respect to Random sampling for scenarios I and II. We next explain how to interpret these numbers.

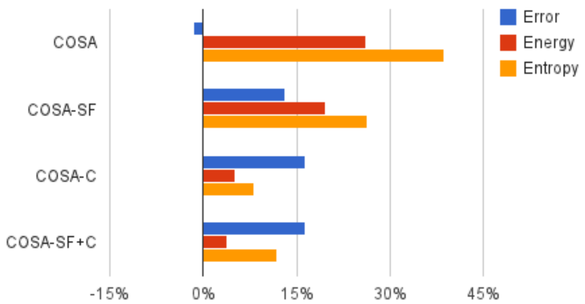


Fig. 2. COSA gains w.r.t. Random Sampling (Scenario I)

The vertical axis of Figure 2 identifies the particular instance of COSA: no strategies (COSA), sampling frequency strategy (COSA-SF), coherence strategy (COSA-C) and both strategies (COSA-SF+C). This figure corresponds to the performance evaluation of COSA for the first simulation scenario, in which a set of 50 nodes are deployed along the river course.

COSA shows the expected tradeoff between energy consumption and error. Figure 2, shows how COSA causes the sink to have slightly higher errors (1.6%) than the Random policy. However, this loss is compensated by a gain of 26% in terms of energy consumption and of 39% with regard to entropy.

The evaluation of the error gain is based on the mean error value registered at the sink. This gain is computed for a period of 100 days in which all the nodes in the network are alive and outside the bootstrapping phase. We select as midpoint of this period the time when the median of the random nodes' energy reaches 0.5 value (see Figure 3(b)). At this point, both sampling policies are in the same conditions and the comparison is therefore fairer. Otherwise, any of the COSA algorithms would be much better than Random because nodes live longer with them. The shape of the error curves can be seen in Figure 3(a).

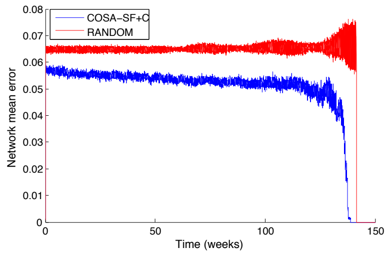
To compute the gains in terms of energy consumption and entropy, we select a reference point at which we evaluate the difference in performance between both policies. The reference point selected for the energy gain is the time when the median of the nodes' energy value reaches zero (Figure 3(b)). This timestamp is interesting as it represents the moment at which half of the nodes in the network have depleted their batteries. The reference point to evaluate the difference in entropy is set to the time when the overall entropy reaches zero (see Figure 3(c). The gain says how much time in percentage an algorithm needs to 'loose' information, i.e. to increase the entropy to reach zero. Any other point could have been equally interesting.

The first set of bars appearing in Figure 2 corresponds to the gain of COSA algorithm with regard to Random sampling in scenario I. It clearly shows that the adoption of COSA policy by the sensing nodes originates a little loss in the accuracy of the information but also an important increase of the WSN life-span. This life-span extension translates into a significant improvement of the quality of the information, as nodes live longer. The extension of the life-span of the network does not only represent a reduction of its battery replacement costs but also an improvement of the system's performance.

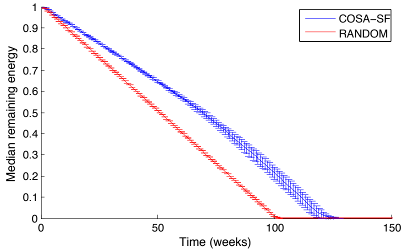
The results obtained when we use COSA with the *sampling frequency* strategy are slightly different. In this case, we get an important improvement in the error gain (reaching a value of almost 13.23%). This improvement comes at the cost of more moderate gains in terms of energy savings and entropy (correspondingly, values of 20% and 26%). Increasing the sampling frequency of the leaders allows them to better follow the changes in the environment caused by the sinusoidal pollutant, therefore committing less error. However, this extra effort in sampling and transmitting originates lower energy and entropy gains when compared to basic COSA.

The third set of bars represents the gains obtained when we compare Random and COSA with the *coherence* strategy. COSA *coherence* outperforms the *sampling frequency* strategy in terms of error gain (16.38%) but reports poorer results performance in terms of energy and entropy (with corresponding values of 5% and 8%). This result shows that this strategy, with the considered configuration parameters, is not the most convenient for a highly dynamic scenario, as scenario I. The cost of breaking groups and initiating negotiations reduces drastically the improvements in energy and entropy. Nonetheless, group dismantlement causes the nodes to sample the environment at the time this happens, what explains the global committed error reduction.

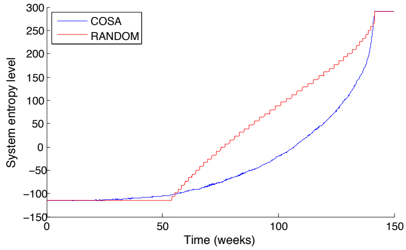
The results obtained for the combination of both strategies (COSA-SF+C) shows how the *coherence* strategy has a stronger impact on the combination than the *sampling frequency* strategy. In this case, the error gain is almost the same obtained as for COSA *coherence* alone. Error and entropy gains also present low values (4% and 12% correspondingly). Therefore, the adoption of the combined strategies does not seem convenient. COSA-SF+C and COSA-C give the best



(a) Information error: COSA-SF+C and Random.



(b) Median remaining energy: COSA-SF and Random.



(c) Overall entropy level: COSA and Random.

Fig. 3. Information error, Energy and Entropy performance in time for different COSA strategies and Random (Scenario I)

performance in terms of error but at the cost of an important reduction in energy and entropy gains. Hence, the characteristic trade-off of COSA renders its best results for scenario I for COSA-SF strategy.

Figures 3(a) to 3(c) shows the evolution in time of the evaluation variables (error, energy and entropy). These figures represent, for scenario I, the network performance for each individual variable during its whole lifetime.

Figure 3(a) represents the error at the sink when agents implement the Random policy and COSA-SF+C strategy. As the pollutant phenomenon has a periodical behaviour and is present in the scenario for the whole simulation time, the mean error value associated to the Random policy remains quite constant as long as all nodes are alive. The corresponding curve associated to COSA-SF+C shows more variability due to the group configuration and reconfiguration processes.

Figure 3(b) represents the median of the nodes' energy values per week for Random and COSA-SF strategy. It shows how half of the nodes deplete their batteries by week 101 when using the Random policy, whereas this situation is reached more than 25 weeks later for COSA-SF. Besides, COSA-SF allows the network to keep a higher level of global energy. Its corresponding curve also presents an increasing variance in time due to the influence of leaders' positions in the network —demanding different energy quantities when transmitting to the sink.

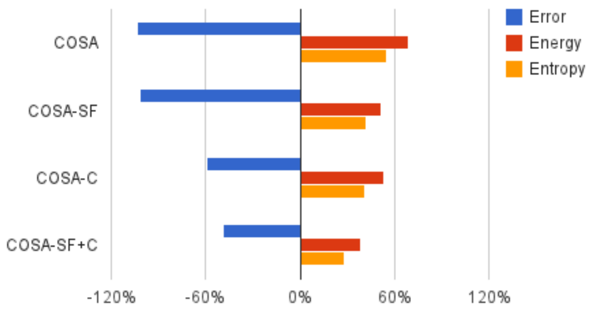


Fig. 4. COSA gains w.r.t. Random Sampling (Scenario II)

Figure 3(c) represents the overall entropy for scenario I, that is, the information quality. The Random approach causes the entropy to deteriorate almost at a constant pace since the first node's battery depletion. The level of entropy when using COSA is lower (i.e. better) during the whole system life-span. COSA results in an evenly distributed nodes' battery depletion, which allows the network to offer a fairly good representation of the whole environment, for most of its life time.

The results obtained for the application of COSA to the second scenario proposed shows a quite different behaviour due to the specific characteristics of this scenario already discussed. The first thing we notice is that none of COSA strategies reaches a positive gain value for the error measurement. As it occurred for scenario I, the error gain obtains its worst value for the application of COSA policy and its best one for the combination of COSA and its two strategies (COSA-SF+C). The energy measurement shows the opposite behaviour giving the best result for COSA policy and the worst for COSA-SF+C. The relationship between the energy and the entropy measurement also changes in this scenario with regard to the first one. In this case, the entropy gain obtained for one of the COSA strategies is always lower than its corresponding energy gain. This is due to the specific network depletion pattern, as we will explain later.

In the scenario II composed of 30 nodes situated far from the sink, the error gain obtained for the network when it implements COSA strategy has a value of -103.3%. That is, a loss of 100% representing that the application of COSA doubles the error committed by the nodes with regard to the random sampling scheme. Therefore, favouring the formation of bigger groups in this scenario implies sending information to the sink on behalf of nodes that are poorly represented by their leaders. On the other hand, this high loss in error comes with high values for the energy and entropy gains. The same grouping phenomenon originates high energy savings that render an energy gain of 69.31% and an entropy gain of 55.31%.

The gain values obtained for the adoption of COSA with *sampling frequency* strategy shows a very little improvement in terms of the error gain and also, a little detriment of the energy and entropy gains. The error loss is still over 100%

(specifically 101.58%) while the energy gain reduces its value to 51.92% and the entropy to 41.98%.

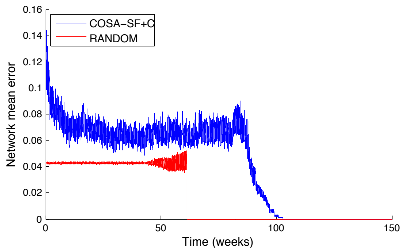
As occurred for the scenario I, COSA *coherence* strategy causes an important improvement in the error loss, almost dividing its value by 2. The error gain for this strategy and this scenario reaches a value of -59.77%. But, opposite to the what occurred with the energy and entropy gain, that also suffered an important decrease, in this case the corresponding energy gain is 53.94% and the entropy gain is 40.83%. This is due to a quick group reconfiguration process in which nodes find a new appropriate distribution.

Finally, the combination of both strategies results in the highest error gain (-49.14%) and the lowest energy and entropy gains (38.8% and 28.27% correspondingly). Once again, the trade-off between energy consumption and accuracy of the information reported to the sink appears for every COSA strategy. And, for this scenario II, we can observe how we are able to extend the network life-time over 60% with regard to the Random sampling strategy at the cost of admitting twice the error reported by the Random schedule.

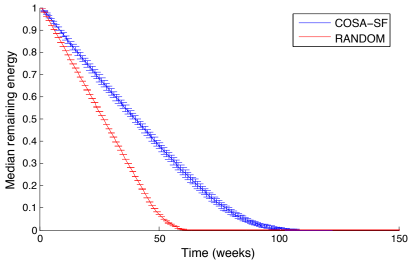
Figures 5(a) to 5(c) shows the performance of the network corresponding to the second scenario in terms of the error reported by the system, the remaining energy of the nodes and the overall entropy level of the network.

Figure 5(a) shows the network mean error per unit time for COSA-SF+C strategy and Random scheme. As in Figure 3(a), the reported error by the Random sampling policy shows a stable pattern. The specific situation of the nodes far from the sink and also far from the pollution source, together with the characteristic river flow, makes the pollution stain effects smoother, which explains the lower error committed by nodes adopting the random sampling policy. The application of COSA-SF+C strategy, after an initial phase, also returns a quite stable error pattern, although with higher variability around 0.065. The median of the remaining energy per node per unit time measures the available energy in the system. Figure 5(b) represents the value of this variable for COSA-SF and Random strategies. This figure shows clearly the improvement derived for the characteristic grouping scheme, as the temporal point when half of the nodes implementing COSA-SF have no energy is reached around 40 weeks later than for random nodes. The life-span of the network for scenario II is also lower than for scenario I. In this case, as all the nodes are situated at a considerable distance to the sink, the demands of energy for information transmission to this element are also higher. The tight grid distribution of the nodes for this scenario implies also low variance for the energy of COSA-SF (as whichever node is the leader, the energy needed to transmit to the sink is almost the same); and a steeper nodes' death pattern. This phenomenon can be observed in Figure 5(c).

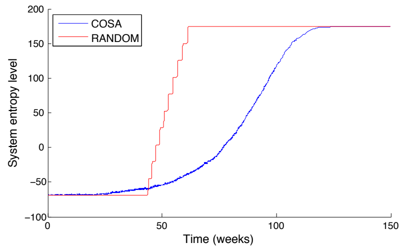
Figure 5(c) presents the overall entropy for scenario II. The strategies considered in this figure are COSA and Random. The Random approach shows the same behaviour pattern as for scenario I (see Figure 3(c)). However, the entropy value per week obtained for the application of COSA to the network in this scenario shows how the highest (worst) value of entropy is reached over 25 weeks later. That is, the extesion of the network life-span directly causes having information from the environment available for longer time. Another difference



(a) Information error: COSA-SF+C and Random.



(b) Median remaining energy: COSA-SF and Random.



(c) Overall entropy level: COSA and Random.

Fig. 5. Information error, Energy and Entropy performance in time for different COSA strategies and Random (Scenario II)

between this figure and its corresponding one for the first scenario refers to the slope of the curve, higher than in the previous case. This also reflects the depletion pattern of nodes' batteries in this case and justifies always obtaining lower entropy gains than energy gains for this second scenario, as shown in Figure 4.

5 Conclusions and Future Work

In this paper we have introduced the COSA algorithm and have given experimental evidence of its computing properties for a particular scenario. This algorithm uses Multiagent systems technology to make WSNs self-configurable at run time. That is, WSNs are able to self-organize in order to adapt the energy consumption to the changes of the environment while fulfilling their sampling objectives in terms of the quality of the reported information. COSA innovates by reaching this objective via a peer to peer negotiation protocol, that results in a global organization producing a network-wide benefit. To attain a good group configuration the algorithm relies on the node local information about its environment state and neighbouring nodes. This information together with the appropriate COSA parameter configuration leads to the formation of groups of nodes that act as a single entity, avoiding redundant sensing and transmissions efforts. The results obtained by the experimentation showed how a sensor network whose nodes implement COSA can adapt to different performance requirements reaching a balance between the energy consumed by the system and the quality of the information reported.

As future work, we plan to test the behaviour of COSA and the different strategies in scenarios showing different dynamic behaviours and different network topologies. From this study we expect to lay out the impact of the different COSA's parameters on the overall network performance. This will fully characterise the algorithm and will allow to establish guidelines on how to use it on different environment monitoring situations.

Acknowledgments. This work has been supported by the Agreement Technologies project (funded by CONSOLIDER CSD 2007-0022, INGENIO 2010).

References

1. Sims, M., Goldman, C.V., Lesser, V.: Self-organization through bottom-up coalition formation. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2003, pp. 867–874. ACM, New York (2003)
2. Ruairí, R.M., Keane, M.T.: The dynamic regions theory: Role based partitioning for sensor network optimization. In: Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (2007)
3. Gaston, M.E., desJardins, M.: Agent-organized networks for dynamic team formation. In: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2005, pp. 230–237. ACM, New York (2005)
4. Barton, L., Allan, V.H.: Methods for coalition formation in adaptation-based social networks. In: Klusch, M., Hindriks, K.V., Papazoglou, M.P., Sterling, L. (eds.) CIA 2007. LNCS (LNAI), vol. 4676, pp. 285–297. Springer, Heidelberg (2007)
5. Glinton, R., Scerri, P., Sycara, K.: Agent-based sensor coalition formation. In: 2008 11th International Conference on Information Fusion, pp. 1–7 (July 2008)
6. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Hawaii International Conference on System Sciences, HICSS 2000, vol. 8, pp. 8020–8029. IEEE Computer Society, Washington, DC (2000)
7. Bandyopadhyay, S., Coyle, E.J.: An energy efficient hierarchical clustering algorithm for wireless sensor networks. In: Proceedings of IEEE INFOCOM 2003, pp. 1713–1723 (April 2003)
8. Younis, O., Fahmy, S.: Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing 3, 366–379 (2004)
9. Cordina, M., Debono, C.J.: Maximizing the lifetime of wireless sensor networks through intelligent clustering and data reduction techniques. In: Proceedings of the 2009 IEEE Conference on Wireless Communications & Networking Conference, WCNC 2009, pp. 2508–2513. IEEE Press, Piscataway (2009)

10. Padhy, P., Dash, R.K., Martinez, K., Jennings, N.R.: A utility-based sensing and communication model for a glacial sensor network. In: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2006, pp. 1353–1360. ACM, New York (2006)
11. Dyo, V., Ellwood, S.A., Macdonald, D.W., Markham, A., Mascolo, C., Pásztor, B., Scellato, S., Trigoni, N., Wohlers, R., Yousef, K.: Evolution and sustainability of a wildlife monitoring sensor network. In: SenSys, pp. 127–140 (2010)
12. Manning, C.D., Schütze, H.: Foundations of statistical natural language processing. Massachusetts Institute of Technology (1999)
13. IIIA-CSIC: Repast sensor network simulation toolkit (2012), http://www.iiia.csic.es/~mpujol/RepastSNS/
14. libelium (2012), http://www.libelium.com/documentation/waspmote/waspmote-technical-guide-eng.pdf
15. Goldman, S.: Information Theory. Dover Phoenix Editions (2005)