

# A Self-Organising Clustering Algorithm for Wireless Sensor Networks

I. Wokoma, L. Sacks, I. Marshall

† University College London

**Abstract:** This paper describes an application level, data-centric algorithm that creates clusters in a sensor network based on the changes of the signal being observed by the sensor nodes without any predetermination by the user. The algorithm was developed for the Self-Organising Collegiate Sensor Network (SECOAS) project and its design was influenced by biological examples of emergence in complex systems. Specifically, this paper refers to the processes Quorum Sensing (QS) and Local Activation and Lateral Inhibition (LALI). The former shows how bacterial cells get into clusters while the latter allows the sensors to recognise patterns in the environment that influences how the clusters are made. Testing the algorithm involved looking at the scalability and the tolerance of the algorithm with a simplified model of the signals to be monitored in SECOAS.

## 1. Introduction

Clustering is a good approach to managing sensor networks because it takes the emphasis away from individual sensor readings and places more importance on extracting data from the spatial areas covered by the sensor network. This gives the data more of a physical meaning in relation to the environment and uses the redundant operation of sensor nodes to provide accurate data. Given these benefits, a biologically-inspired clustering algorithm was developed to assist with the task of data aggregation in a sensor network deployed for environmental monitoring on the coast as a part of the Self-Organising Collegiate Sensor Network (SECOAS) project, the details of which can be found in [1]. One of the requirements of the project is the development of lightweight and flexible algorithms to be run on each resource-limited sensor node in the network that encourage them to collaborate to carry out complicated sensing tasks. This is similar to many situations found in the natural world, where there are examples of global emergent behaviour resulting from the actions of entities that operate on smaller scale local events [2]. Thus, this algorithm has been based on two processes used in biological systems, Quorum Sensing (QS) and Local Activation and Lateral Inhibition (LALI), and has been extended from the previous work detailed in [3] to incorporate more self-organization and to give more quantifiable results.

## 2. Implementation of Biological Concepts in Clustering

The foundation of the clustering algorithm is the process of QS, the biological process used in communities of bacterial cells with no global awareness to co-ordinate themselves for applications such as bioluminescence, where visible light is emitted from a living organism [4]. They organize themselves by transmitting and receiving signalling molecules called autoinducers to detect when there is a minimum population of cells. [3] shows how this has been incorporated into the design of the clustering algorithm to indicate to the sensor nodes when there are enough sensors to make a cluster that monitors the same change, or gradient, in the observed signal. The algorithm was subsequently changed to make the clusters formed more controllable and representative of the spatial characteristics of the signal by allowing the measurement of gradients between neighbouring sensors instead of between sensors and clusterheads that were potentially too far away for comparison. This new method of measuring gradients in conjunction with QS alone led to the requirement of an initial input from the user concerning the range of gradients allowed in a cluster, a parameter which allows the sensors nodes to decide on how the clusters should be made. This may be a difficult task for the user if nothing is known about the signal to be observed beforehand.

The sensor nodes need to decide on this parameter themselves despite having different views of what is going on the environment, i.e. sensor nodes in one area may find that they measure a small range of gradients with its' neighbours whereas sensor nodes in another area may measure a larger range of gradients. However, to prevent the clusters from overlapping the sensor nodes must have the same range of gradients allowed in a group e.g. if one cluster monitors the signal where its gradient varies between 10 and 20 signal units per unit distance, then neighbouring clusters should monitor either 0 and 10 signal units per unit distance or 20 and 30 signal units per unit distance. Finding a solution to this problem involved looking at biological phenomena demonstrating the emergence of ordered patterns given random initial conditions. Examples of this are ant cemetery construction [5], head formation in the organism hydra [6] and animal skin patterns [7]. A common thread with these self-organization examples is the use of the Local Activation, Lateral Inhibition (LALI) mechanism [8]. This is where an elevated local concentration of pattern-forming substance encourages more build up in one area but inhibits the build-up in another area some distance away. The initial distribution of the substance is unstable and any small increases in the substance above the average concentration will cause growth in local areas but its

effect will be down-regulated in areas further away. These fluctuations carry on until the self-enhancement is in equilibrium with the inhibition thus making the system stable. The only requirement is that the self-activation has a shorter range than the inhibition and that both are self-regulating [8].

The LALI mechanism was incorporated into the clustering algorithm by introducing two parameters activator, AC and inhibitor, IN. For example, if sensor node A has a gradient range of  $R_1$  and receives a message that sensor node B that is  $D$  distance units away has a gradient range of  $R_2$ . Sensor node A responds by comparing  $R_1$  to  $R_2$  and changing the inhibitor, IN, and activator, AC, according to the following rules:

- If  $R_1 > R_2$  and  $D = 1$  hop then increment IN
- If  $R_1 \neq R_2$  and  $D > 1$  hop, then increment AC

Once sensor node A has carried these steps out for all its nearest neighbours all the received messages, it stores the gradient ranges of the nearest neighbours. There are two gradient ranges from the neighbours that have the closest value to  $R_1$ , one which is higher,  $R_H$ , and one which is lower,  $R_L$ . These values can be used with the accumulated values of the inhibitor and activator to change  $R_1$  accordingly:

$$\text{difference} = AC - 0.1 * IN$$

- If difference  $< 0$ ,  $R_1 = R_L$
- If difference  $> 0$ ,  $R_1 = R_H$
- If difference = 0,  $R_1$  stays the same

The gradient ranges of the nodes continue to change until it is identical for the whole network.

### 3. Algorithm Details

The clustering algorithm is executed in a series of time steps called epochs; it takes 1 epoch for a sensor node to complete a cycle of the algorithm. The following sections explain the events that may occur during a cycle depending on the status of the sensor nodes.

#### Initialization

There are two types of packets sent and received by the nodes; the first type is a Node Synchronisation (NS) packet which the nodes start broadcasting at a high rate since at initialisation they are all ungrouped. These packets allow the nodes to carry out the following tasks:

- To build a table that stores information from their neighbours such as their identification number and the gradient of the observed signal between the node and each neighbour.
- To alter the range of gradients allowed in each group using the LALI mechanism described in section 2.

#### Clusterhead Proposal

Since the gradient range is continuously changing, they need to keep checking whether they are capable of becoming clusterheads. If the gradients giving the smallest difference in their tables lie within consecutive integer multiples of the current gradient range then they can form clusters with neighbours that have the same gradient range.

#### Cluster formation and maintenance

Potential clusterheads and grouped nodes alternate between sending NS packets and the second type of packet called the Group Synchronization (GS) packets. When GS packets are passed between nodes of the same cluster during intra-cluster communication the nodes broadcast less often and thus indicating that they are in a "quorum", the minimum number of sensors required to monitor a particular change in the environmental signal. GS packets also allow inter-cluster communication can take place between neighbouring nodes in differing clusters.

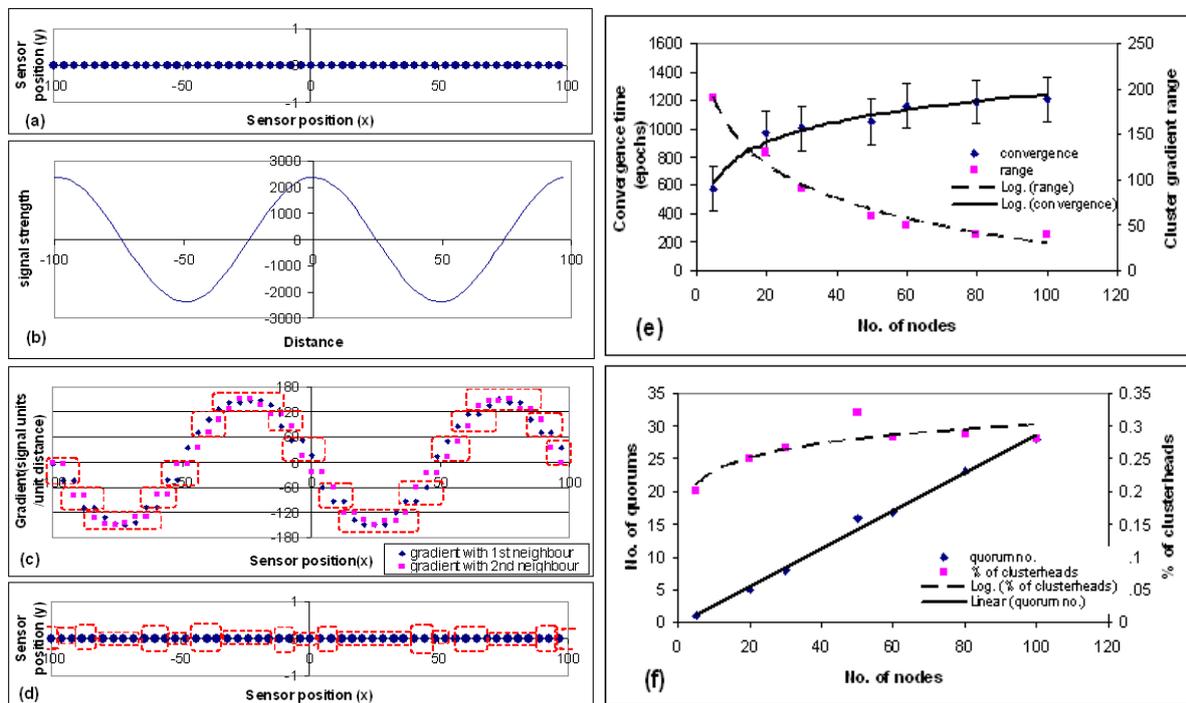
#### Reset

The following events cause a grouped node to reset and erase the group information:

- If none of the gradients measured between the node and each of its neighbour fits into the gradient boundaries of the group then the node may have to change clusters.
- If the node receives no contact from any other cluster member for a long period of time then the node must reset since the cluster seems to no longer exist.

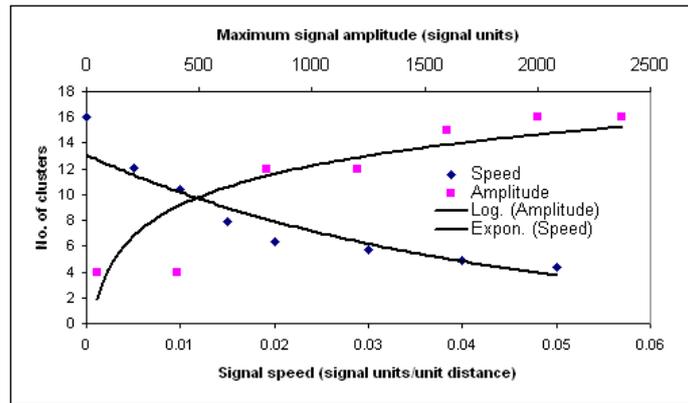
## 4. Tests

Simulations in Netlogo [9] were carried out using the clustering algorithm on a simplified model of the signal likely to be observed by the sensor nodes of an oceanographic sensor network. An appropriate selection was a signal varying sinusoidally over space and using the graphs in Figure 3(a) – (d), the groups formed by 50 evenly spaced sensors can be quantified. The scalability of the algorithm was tested by varying the number of nodes in the linear network in Figure 3(a) and measuring the convergence time, the total time in epochs taken to execute the algorithm and form the clusters shown in Figure 3(d). The graph in Figure 3(e) suggests that the convergence time increases logarithmically with the number of nodes, which may be due to the cluster gradient range. Figure 3(e) also shows that the range of gradients allowed in a cluster decreases logarithmically with the increasing number of nodes. The increase in density allows the network to dissect the signal on a finer scale into a larger number of smaller clusters. Figure 3(c) shows that the chosen range of gradients for each cluster is 60 signal units/unit distance, but for networks of different size the gradient range may have to become larger or smaller to make sure all the nodes are in a cluster. Figure 3(e) shows how the gradient range varies with the size of the network. Figure 3(f) shows the number of quorums formed and the percentage of clusterheads in the networks of varying size. Even though the number of quorums increases linearly with the number of nodes, the percentage of clusterheads sending information to the base station increases very gradually from 20% - 30% of the network.



**Figure 3(a)** A linear evenly spaced network of 50 sensors, **(b)** the static signal being monitored, **(c)** the gradient of the observed signal measured between sensors which is grouped every 60 signal units/unit distance, **(d)** 17 clusters of sensors are formed based on the gradient groups, **(e)** the convergence time and cluster gradient range versus the number of nodes, and **(f)** the no. of quorums and % of clusterheads in the network versus number of nodes.

The signals that the sensor nodes will be monitoring are likely to be dynamic and have varying amplitudes. The tolerance tests were carried out to find out how much of this variation in the observed signal the clustering algorithm can handle before the clustering becomes ineffective. Figure 3(b) and (c) shows that are 4 parts of the sine wave where the gradient is either positive or negative. When the sensor network forms only 4 quorums, the clusters only represent the separation between positive and negative gradients and show no other detail about the signal variation. It may be more optimal in this case to obtain individual data readings from all the sensor nodes than to use clustering because their readings differ too much. It is also expected that if the amplitude of the signal is very small and if the signal moves too fast, then 4 quorums will be created. Figure 5 shows these limits of the algorithm; the maximum amplitude of the signal has to be greater than 400 signal units to give accurate results. It also shows that when the speed of the signal was varied between 0 and 0.05 signal units/unit distance and the cluster algorithm ceased to be useful after 0.04 signal units/unit distance.



**Figure 4: The effect of signal speed and maximum signal amplitude on the no. of quorums**

## 5. Conclusion

The clustering algorithm discussed in this paper is focused on highlighting areas of change in the observed signal of a sensor network thus preventing the user from having to make the difficult decision of how many clusters are appropriate. It is a unique approach sensor network design because of the inspiration from relevant biological examples of emergence. The results show that the algorithm is scalable, flexible for dynamic signals and accurate as long as the signal variation is within the stated limits. Implementation of the algorithm in SECOAS trials due in August will enable the analysis of the energy efficiency and network lifetime when using the algorithm. Currently, the algorithm is being converted into a suitable format for kOS [1].

## References

- [1] L. Sacks et al, "The Development of a Robust, Autonomous Sensor Network Platform for Environmental Monitoring", Sensors & their Applications XII conference in Limerick, Sep. 2003.
- [2] S. Johnson, "Emergence: the connected lives of ants, brains, cities and software", Allen Lane The Penguin Press, London, 2001.
- [3] I. Wokoma, L. Sacks and I. Marshall, "Clustering in Sensor Networks using Quorum Sensing," in the London Communications Symposium, University College London, 8th-9th September, 2003
- [4] M. B. Miller and B. L. Bassler, "Quorum Sensing in Bacteria", Annual Review in Microbiology, pp.165-99, 2001.
- [5] Guy Theraulaz et al, "Spatial patterns in ant colonies", Proceeding of the National Academy of Sciences of the United States of America, vol. 99, no. 15, July 2002.
- [6] U. Technau et al, "Parameters of self-organization in Hydra aggregates", Proc Natl Acad Sci USA, October 2000, 97(22):12127-12131
- [7] Wilensky, U. (2003). NetLogo Fur model. <http://ccl.northwestern.edu/netlogo/models/Fur>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- [8] H. Meinhardt et al, "Pattern formation by local self-activation and lateral inhibition", BioEssays 22:753-760, 2000.
- [9] U. Wilensky, NetLogo 1999, <http://ccl.northwestern.edu/netlogo/>, Connected Learning and Computer-Based Modeling Centre, Northwestern University, Evan, IL.