Collective Sensor Configuration in Uncharted Environments

Norman Salazar and **Juan A. Rodriguez-Aguilar** and **Josep Ll. Arcos** [1]

Abstract.
Sensor networks (SN) are rapidly becoming the tool of choice for monitoring. Their versatility makes them useful in numerous and diverse application domains. However, most SN deployments assume that the area and events to monitor/control are well known/understood at design time. Thus, the sensors' configurations can be defined prior to their deployment. Nevertheless, when the purpose of an SN is to monitor the events of an *uncharted environment*, where the distribution and nature of events is uncertain, it is rather intricate to configure its sensors at design time. Instead, sensors should be able to self-configure at run time. In this paper, we propose a low-cost (in terms of energy and computation) *collective* approach that allows the sensors in an SN to collaboratively search for their most appropriate configurations only using their local knowledge. We empirically show that our approach can help sensors efficiently monitor environments where various dynamic events exist.

1 Introduction

As technology continuously improves, it is becoming apparent that sensor networks (SN) are a powerful and versatile tool [5]. They have been employed by numerous applications on domains of different characteristics. Nevertheless, many of these applications rely on static sensor configurations (i.e pre-configured at design time), which can be detrimental. It has been argued that in real-world deployments the complexity, diversity, and dynamicity of the sensing requirements is a major issue that cannot be tackled through static configurations ([2] [4]).

Moreover, the current literature assumes that the deployment environment has been well studied, and thus that the sensor designers and the sensors themselves can use the available domain knowledge for configuration purposes. Nonetheless, this may not always be the case. It has been argued that sensor networks can be particularly useful in remote or dangerous environments that have rarely been studied due to their inaccessibility [1]. Therefore, sensors need to be able to (re)configure and coordinate themselves in a decentralized manner according to the occurring events of these uncharted environments to maximize the amount of information perceived over time.

Furthermore, it has been noted that in large environments various distinct events are prone to occur at once. In other words, there is a spatial distribution of concurrent events. Hence, a sensor's configuration depends on the event(s) present on its geographic location (a sensor must be able to adopt as many configurations as events are possible). In these cases, it is likely that neighboring (close-by) sensors experience the same event(s), consequently making them require similar (the same) configurations. *Collective active sensing* [5], is a known type of approaches that take advantage of the fact that collectives of sensors may need similar configurations, by making them coordinate and cooperate towards a common goal (discovering the most useful configuration).

In this paper we propose a collective approach to monitor uncharted environments where only (at most) partial domain knowledge is available to the sensors in a SN. Thus, unlike in most current SN applications, we are not aware of the kind of events that may occur, nor of their possible locations in the environment. For our approach, we embed in each sensor a distributed algorithm that: i) has a low computational overhead and a low energy consumption; ii) employs diffusion search to collectively find/construct the most useful sensor configurations for to the occurring events; iii) promptly reconfigures the sensors in response to dynamicity of the events; and iv) works when the sensor cannot rely on the available domain knowledge (uncharted environment).

2 A Collective Approach

In situations where the sensors are deployed to an uncharted environment, it may be the case that the only available useful information is the one provided by the sensors' own feedback function. Under such circumstances, cooperation becomes necessary since sensors can improve their partial domain knowledge (regarding the configurations) by sharing their local experiences. Moreover, the number of possible configurations may be very large, thus it may be unfeasible for the sensors to individually search for their configurations. Hence, if multiple sensors search for the same configuration, they can save time and power by searching together. Once a sensor finds a good configuration, it can be promptly shared with its searching peers.

To that aim, we designed the *collective diffusion search* (CDS) as an algorithm based on the collective sharing of configurations amongst neighboring sensors. In what follows we describe the main components of CDS and their rationale.

Diffusion. It is an efficient (computation-wise) component in charge of sharing the configurations. In a sensor, diffusion consists in a broadcast (to its close-range neighbors) of its configuration. However, to reduce energy consumption (which may be a sensor's priority) a *probability of diffusion* can be employed to regulate a sensor's likelihood of broadcasting. Computationally, diffusion has a low overhead on the transmitting sensor side (sending a message without caring who will receive it). Nonetheless, receiving various broadcasts raises an issue, because a receiving sensor needs to decide what to do with these received configurations

Culling. Attaching in each broadcast the utility of a configuration effectively provides a receiving sensor with the means to decide how

[1] IIIA, Artificial Intelligence Research Institute, CSIC, Spanish National Research Council, Spain, email:{norman, jar, arcos}@iiia.csic.es. Research partially funded by projects TIN2009-14702-C02-01, CSD2007-0022.

Figure 1. a) Distribution of four different events in an environment.

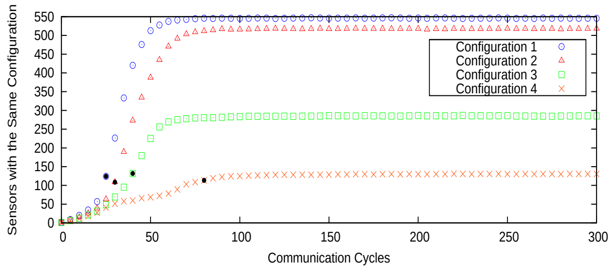

Figure 2. Results of convergence after the initial deployment. The black dots mark when the best configurations were found.

to deal with multiple incoming configurations. This new information allows each receiving sensor to implement a culling component to dismiss useless (received) configurations. For instance, we implement this through a filter that selects the best received configuration and only if it is better than the sensors own.

Intermixing. Diffusion and culling do not have searching capabilities, at most they will establish the best configuration known by any of the sensors (per event). Thus, some searching needs to be incorporated since its unlikely to expect that some sensor knows its most useful configuration a priori. A low-overhead search method, consists in intermixing (combining) two configurations (the selected through culling and the sensor's current one) to create a new one. This can be regarded as using someone else's experience without completely forgetting your own. Nevertheless, sensors cannot always depend on the usefulness of their neighbors configurations (e.g if all the neighboring sensors share a bad configuration).

Local improvement. Through this component each sensor is capable of searching for new configurations without depending on its neighbors. Local improvement can be accomplished (without expending much processing power) by introducing a random change a sensor's configuration with some *probability of improvement*. Various disciplines have shown this to be effective [3].

Altogether, in collective diffusion search each sensor continuously attempts to propagate its configuration while trying to improve it at the same time. The sensor receives some broadcasts which are then filtered through culling in an attempt to determine if there is a better configuration. In case there is, the sensor's configuration and the selected one are combined in an attempt to create a new (and ideally better) configuration. Afterwards, local improvement can be used to continue the search for the best configuration. Once this is over, the sensors configuration is used and its utility valuated through the feedback generated by the performed actions. Lastly (is a matter of perspective) the sensor wraps its configuration along with its utility into a message for broadcasting. An execution of this process shall hereafter be referred as a *communication cycle*.

3 Experimental Results

Our preliminary experiments were designed to test if through collective diffusion search randomly deployed sensors can find the configurations needed to monitor the events occurring in the environment. For this experiment we ran 50 discrete event simulations (each one up to 5000 ticks) over a 100 x 100 grid environment covered by four distinct events (figure 1 depicts with different colors the shape of each event). During a simulation a set of 1500 sensors is *randomly* deployed unto this environment. However, because we are evaluating their self-configuration capability in an uncharted environment none of the sensors is aware of the environment partitioning and thus each sensor starts with a random configuration. A sensor configuration is given by an ordered sequence of 5 actions selected from a pool of 20 possible actions (i.e $\|K\| = 20^5$).

The CDS parameters used for the experiments were: a broadcast range of 4 cells, a 20% diffusion probability per sensors at a given point in time, and a 0.0008 probability of local improvement.

Figure 3 shows that CDS is quite effective in finding the most useful configurations for most of the sensors. Observe that once such configurations are found (black dots in the figure) the sensors promptly adopt them. These configurations are found at $\sim 20, \sim 30, \sim 40$ and ~ 60 communication cycles for each of the four events respectively.

However, notice that depending on features of the event, some configurations require more time to be adopted by sensors. This appears to be related to the dimension of the area occupied by the event, and thus by the number of sensors that require the same configuration. *Event 4* is a particularly pronounced example of this effect (sensors localized in the region of this event take the longest to find the best configuration and thus to adopt it). From the algorithmic point of view, this is reasonable in a collective approach because fewer sensors are looking for the same configuration. Although there may be another factor to consider, the location of the event. Observe that event 4 is completely surrounded by the other events, which means that sensors in that area are constantly receiving conflicting configurations from their neighbors. Additionally, because of its small dimensions a broadcast originated in its frontiers may cover a significant area of the event. In other words, sensors far from the center of the event may receive conflicting configurations.

To conclude, collective diffusion search is a low overhead, but powerful distributed algorithm that when embedded in each sensor empowers them to dynamically find the most useful configuration for the events in their locations (even when only partial domain knowledge is available). Nonetheless, the dimensions and location of the events affect how promptly such configuration is found and adopted.

REFERENCES

[1] Kirk Martinez, Jane K. Hart, and Royan Ong, 'Environmental sensor networks', *Computer*, **37**, 50–56, (2004).

[2] Rónán Mac Ruairí and Mark T. Keane, 'An energy-efficient, multi-agent sensor network for detecting diffuse events', in *In Proceedings of IJCAI'07*, pp. 1390–1395, San Francisco, CA, USA, (2007). Morgan Kaufmann Publishers Inc.

[3] Maxi San San Miguel, Victor M. Eguiluz, Raul Toral, and Konstantin Klemm, 'Binary and multivariate stochastic models of consensus formation', *Computing in Science and Eng.*, **7**(6), 67–73, (2005).

[4] Mark Sims, Claudia Goldman, and Victor Lesser, 'Self-Organization through Bottom-up Coalition Formation', in *Proceedings of AAMAS 2003*, pp. 867–874, Melbourne, AUS, (July 2003). ACM Press.

[5] M. Vinyals, J. A. Rodriguez-aguilar, and J. Cerquides, 'A survey on sensor networks from a multi-agent perspective', *The Computer Journal*, (2009).