

Virtual Cluster Model in Clustered Wireless Sensor Network Using Cuckoo Inspired Metaheuristic Algorithm

May Kamil Al-Azzawi^{1,2}, Juan Luo¹ and Renfa Li¹

¹ College of Computer Science and Electronic Engineering, Hunan University
Changsha, Hunan, 410082, China

² Al-Mansour University College, Department of Computer Science and
Informatics

Baghdad, 69005, Iraq

juanluo@hnu.edu.cn¹, may_kamil_mahdi@yahoo.com²

Abstract

Energy efficient clustering is well known optimization problem which has been studied widely to extend lifetime of wireless sensor networks (WSNs). Grid computing, enable different entities to share computational, memory, and data resources using a flexible and secure framework. These resources enable users to run computational jobs that exceed the capabilities of available sensor at any single location.

The algorithm presented here is to create a virtual cluster within established clusters in wireless sensor network to share individual sensor resources, which permits the wireless sensor network to gain extra cluster resources to run resource consuming applications with private resource control policies. According to the power issue and the unpredictable wireless network characteristics, it is possible that applications running on the sensor nodes might fail due to sensor energy termination, hence, using Cuckoo Search Optimization (CSO) algorithm is necessary as inspired nature technique to improve the availability of sensor nodes by electing the optimum sensor node from each cluster according to precise constrains applied by network specifications.

The motivation for this virtual cluster is to create a virtual cluster environment with heterogeneous resources from clusters with homogeneous nodes when a critical event take place and need a service might requires several sensors resources which make it necessary to arrange these resources to achieve the required service.

Keywords: Grid computing, Distributed systems, Wireless sensor network, Cuckoo search optimization, Energy efficiency

1. Introduction

With the growth of wireless communication and device technology, sensors with sensing, computation and communication capability are widely used in various fields. Wireless Sensor Network (WSN) consists of hundreds to thousands of sensor nodes [1-2], which collect the information on the surroundings such as temperature, light, humidity, sound, vibration, pressure and send it to the cluster head (CH), that forward it to the Base station (Sink).

WSN is a network of tiny battery powered sensor nodes of limited on-board processing, storage and radio capability. Sensor nodes are usually deployed in a random fashion, and collect the context information and perform the given mission through the cooperation with other nodes. Sensor devices in a wireless sensor network are resource constrained since they have limited sensing capability, processing power, and communication bandwidth. In current WSN, the store, analyze, and process of sensory data by the ordinary nodes constrain and affect the application of WSN. Wireless sensor

networks (WSNs) have been deployed in various environments for applications such as environmental and habitat monitoring, healthcare monitoring of patients, weather monitoring and forecasting, military and homeland security surveillance, tracking of goods and manufacturing processes. These applications of WSN have diversity in their priorities that ranging from ordinary events to high priority events. There are three criteria to recognize a grid that is provided by Foster [3]. A grid

- Coordinates resources not under centralized control.
- Uses standard, open, general-purpose protocols and interfaces.
- Delivers nontrivial qualities of service.

In this paper we propose to use grid computing concepts to create a model of virtual cluster of a clustered wireless sensor network to deal with high priority events that causes overwhelming use of system resources and impacts network performance. We utilize the cuckoo search (CS) optimization algorithm to select the optimum sensors to join the sensor grid cluster which is considered as a primary key to generate the virtual cluster that lasting for a short time (milliseconds) to accomplish a high resource utilization application before the termination of the virtual cluster.

The rest of the paper organized as follows Section 2 show the related work. In Section 3 we discuss the grid computing fundamentals. Section 4 shows the proposed virtual cluster formation algorithm. Discrete cuckoo searching optimization algorithm will be discussed in Section 5. Section 6 shows the simulation results. The paper conclusion presents in Section 7.

2. Related Work

In [4], Chen *et al.* address the tracking efficiency of wireless sensor grids by devising a grid comprised of multiple sensor types cooperating in a hierarchy to improve overall detection and tracking. In [5] the individual sensors, sometimes called “motes” in the grid and take their measurements and then possibly perform some processing before passing the information on to a central computer.

In [6], Miller and Vaidya contribute their own work involving the use of one or more busy-tones to selectively wake appropriate sleeping nodes in a sensor network. Gui and Mohapatra in [7] provide a study combining theoretical analysis and simulated evaluations of the trade-offs between energy conservation and surveillance quality in a sensor network target tracking application. Their study is limited to an event-driven application model applicable to the tracking of intruders, animals or perhaps enemy troops. In [8] Hsin and Liua study similarly concerned with power conservation in sensor grids in a surveillance application, and examine the use of alternating wake and sleep cycles in sensors accompanied by the addition or redundancy of sensor deployment. Another related study of power conservation in sensor grids [9] suggests that a protocol in which groups of adjacent sensors coordinating with other adjacent groups with respect to event detection can allow each group to increase power efficiency through sleep cycling.

3. Grid Computing and Wireless Sensor Network

Grid computing is to gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating systems are.

Grid computing [2] is an important and developing computing initiative that involves the aggregation of network connected resources to form a large-scale, distributed system for coordinated problem solving and resource sharing. Recently, many research efforts are focusing on the integration of wireless sensor networks and grid computing in terms of compute grids and data grids.

A compute grid provides distributed computational resources to meet the computational requirements of applications, while a data grid provides continuous access to large amounts of distributed data and storage resources [10]. The sensors taking part in a sensor grid may be stationary once they have been deployed or they may be mobile. In the grid wireless sensor network participate detection, processing and communication while they share many behaviors shared with other ad hoc networks, they also have requirements unique to sensor networks [6]. The main concern with the design and development of wireless sensor network devices lies in the area of power efficiency. Because the sensors equipped with non-renewable batteries, their effective life is dictated by their initial power level and the efficiency with which it is used. Another factor affecting by power efficiency is the range of the sensor in detecting its target data. This range specify how dense the sensor grid must be in order to cover the grid area. Increasing of sensor range allows for a lower density of sensors but requires each sensor to expend more energy and determines what type of communication protocol should be used to ensure appropriate quality of service. These factors as well as application-specific challenges must be dealt with in order to deploy a given sensor network to achieve optimum coverage and maximum useable lifetime.

3.1 Grid Architecture

There are different types of grid architectures to fit different types of business problems. Some grids are designed to take advantage of extra processing resources, whereas some grid architectures are designed to support collaboration between various organizations.

3.1.1 Computational Grid

A computational grid provides distributed computational resources to meet the computational requirements of applications, by aggregating the processing power from a distributed collection of systems. This type of grid is primarily comprised of low-powered computers with minimal application logic awareness and minimal storage capacity. The main goal of computational grid is to expand abilities and maximize the computer utilization of existing resources through aggregation and sharing. The application may require computer capacity more than available to accommodate additional uses for a computational grid include mathematical equations, derivatives, pricing, and simulation (especially risk measurement) [10].

Computational grids can be recognized by these primary characteristics:

- Made up of clusters of clusters.
- Enables CPU searching for better utilize resources.
- Provides the computational power to process large-scale jobs.
- Satisfies the business requirement for instant access to resources on demand.

3.1.2 Data Grid: Computational grids are more suited for aggregating resources, while data grids focus on providing secure access to distributed data and storage resources. A data grid makes a group of available databases to function as a single virtual database. Through this single interface, the database provides a single query point, data modeling, and data consistency. Data grids also attach data, storage, and network resources located in distinct administrative domains, respect local and global policies governing how data can be used, schedule resources efficiently, and provide high speed and reliable access to data. Applications interested in data grids typically have advantages of expanding data-mining abilities while maximizing the utilization of an existing storage infrastructure investment, and to reduce the complexity of data management [11].

3.2 Grid Topologies

A grid topology can be categorized as:

- _ Intragrids
 - Single organizations
 - No partner integration
 - A single cluster
- _ Extragrids
 - Multiple organizations
 - Partner integration
 - Multiple clusters
- _ Intergrids
 - Many organizations
 - Multiple partners
 - Many multiple clusters

The simplest of the three topologies is the intragrid, which is comprised merely of a basic set of grid services within a single organization. The complexity of the grid design is proportionate to the number of organizations that the grid is designed to support, and the geographical parameters and constraints [12].

3.3 Design Concerns and Challenges

In this section, we discuss the important issues and challenges in the design of sensor grids. Most of these design issues and challenges arise due to the inherent limitations of sensor devices, such as limited processor performance, small storage capacity, limited battery power, and unreliable low-bandwidth wireless communication.

3.3.1. Network Connectivity and Protocols: The sensor nodes in sensor grids are connected via wireless ad hoc networks which are low-bandwidth, high-latency, and unreliable. The network connectivity of sensor nodes is dynamic in nature, and it might be intermittent and susceptible to faults due to noise and signal degradation caused by environmental factors. The sensor grid has to gracefully handle unexpected network disconnections or prolonged periods of disconnection. Wireless sensor networks are often based on proprietary protocols, especially for the MAC protocol [13]. It is not practical for sensor nodes to have multiple network interface capabilities. Thus, efficient techniques to interface sensor network protocols with grid networking protocols are necessary.

3.3.2. Power Management: The main challenge of overcoming the limited power resources available in sensor devices coupled with the difficulties in providing replacement power sources to these devices. In most current sensor devices the energy available to the device is provided by a battery installed during manufacture and this battery determines the useful lifetime of the device. As battery power runs out, the devices fail and must be replaced. The effect of such failures on the functionality of the sensor network is often reduced through the deployment of redundant sensors with some number held in a low-power idle state. As sensors fail, idle neighbor sensors are activated to take their place in order to maintain the sensor coverage of the network [14].

3.3.2. Scalability: Scalability is the ability to add sensor resources to a sensor grid to increase the capacity of sensor data collection, without substantial changes to its software architecture. The sensor grid architecture should allow multiple wireless sensor networks, possibly owned by different virtual organizations, to be easily integrated with compute

and data grid resources. This would enable an application to access sensor resources across increasing number of wireless sensor networks [12].

3.3.3. Scheduling: In wireless sensor networks, scheduling of sensor nodes is often performed to facilitate power management and sensor resource management. Researchers have developed algorithms to schedule the radio communication of active sensor nodes, and to turn off the radio links of idle nodes to conserve power. Similarly, for applications like target tracking, sensor management algorithms selectively turn off sensor nodes that are located far away from the target, while maximizing the coverage area of the sensors. Sensor grids are data-centric in nature. A scheduler is needed for the efficient scheduling of applications to use the sensor resources for collecting sensor data. A sensor grid has a set of sensor nodes spread across multiple wireless sensor networks. These sensor nodes may provide sensors that collect different types of data such as temperature, light, sound, humidity, vibration, etc. Also, the sensor nodes may be shared by multiple applications with differing requirements. The design of a sensor grid scheduler is influenced by some important differences between sensor jobs and computational jobs. Unlike computational jobs, sensor jobs are not multitasking in nature. A sensor node can execute only one sensor job at a time, and it cannot execute multiple sensor jobs via multitasking. While computational jobs automatically terminate upon completion, the durations of sensor jobs have to be explicitly specified. Sensor jobs are also more likely to require specific time slots for execution compared to general computational jobs [15].

3.3.4. Security: Wireless sensor networks are prone to security problems such as the compromising and tampering of sensor nodes, eavesdropping of sensor data and communication, and denial of service attacks. Techniques to address these problems include e sensor node authentication, efficient encryption of sensor data and communication, and secure MAC and routing protocol [16]. For sensor grids, it is necessary to ensure that the grid security techniques and the wireless sensor network security techniques are integrated seamlessly and efficiently [4].

3.3.5. Availability: Due to the power issue and the unpredictable wireless network characteristics, it is possible that applications running on the sensor nodes might fail. Thus, techniques to improve the availability of sensor nodes are necessary. Sensor grids should support job and service migration, so that a job can be migrated from a sensor node that is running out of power or has failing hardware to another node. If sufficient resources are available, services can be replicated so that the loss of a node will not result in service disruption. Finally, if unexpected interruptions occur, the system should be able to recover and restart the interrupted jobs [17].

3.3.6. Quality of Service: Quality of Service (QoS) is a key issue that determines whether a sensor grid can provide sensor resources on demand efficiently. Enforcing QoS in sensor grids is made complicated by the unpredictable wireless network characteristics and sensor power consumption. A service request might require several sensor resources. Thus, it might be necessary to make reservations of these resources to achieve the required QoS. Resource reservation is closely tied to the scheduling of sensor resources. Due to the highly dynamic sensor grid environment, any attempt at QoS provisioning should be adaptive in nature. It is necessary to consider the changes in resource availability, network topology, and network bandwidth and latency, so that the sensor grid can provide the best possible QoS to the application [15].

4. The Proposed Virtual Cluster Generation (Vir_Clu_Gen) Algorithm

The sensors deployed randomly and fully cover a geographic region such that every point in the region is within the detection range of at least one sensor node. A sequence of events take place that varies in their priorities from normal event to very critical event like earthquake and tsunami. In this paper we will give attention to a scenario with very critical event that need providing new resources by increasing processing power, data sharing and capabilities of wireless sensor devices.

We will work on a typical intragrid topology within a specific wireless sensor network providing a basic set of grid services. The single wireless sensor network made up of a number of homogenous sensors that share a common security domain, and share data internally on a private network. The proposed algorithm works when a critical event detected and a special process need to run. To provide advanced warning of potential future disasters, a virtual cluster formed from the established wireless sensor network clusters. The modeled cluster are built by selecting two sensor nodes from each cluster in the network, the first is the node with highest residual energy and relatively close to the event, and the second is the cluster's cluster head. This mechanism will add extra sensor resources to the sensors in the virtual cluster to increase the computational resource and capacity of these sensors without substantial changes to its software architecture. This mechanism should allow multiple sensors owned by different clusters to be easily integrated to run a process that need extra compute and data grid resources.

We proposed Virtual Cluster Generation (Vir_Clu_Gen) algorithm to select these sensors based on cuckoo search algorithm to select the optimum sensor from each cluster that will be used to create the virtual cluster.

In figure.1 a wireless sensor network consists of four clusters each with a special node as a cluster head (CH). When the critical event is detected by a sensor, in one of the network clusters (C_j), (Vir_Clu_Gen) search optimization algorithm is triggered to select the optimum sensor from each cluster (C_i) that is (relatively close to the event and with highest residual energy) coupled with the its cluster head (CH_i) to form the virtual cluster.

In the virtual cluster, the cluster head is the cluster head of the cluster where the critical event take place(CH_j), and the special process can initiate with affordable of extra resources to accommodate the overhead of the initiated process. When the process terminated the virtual cluster will vanish and each sensor will work on its original cluster.

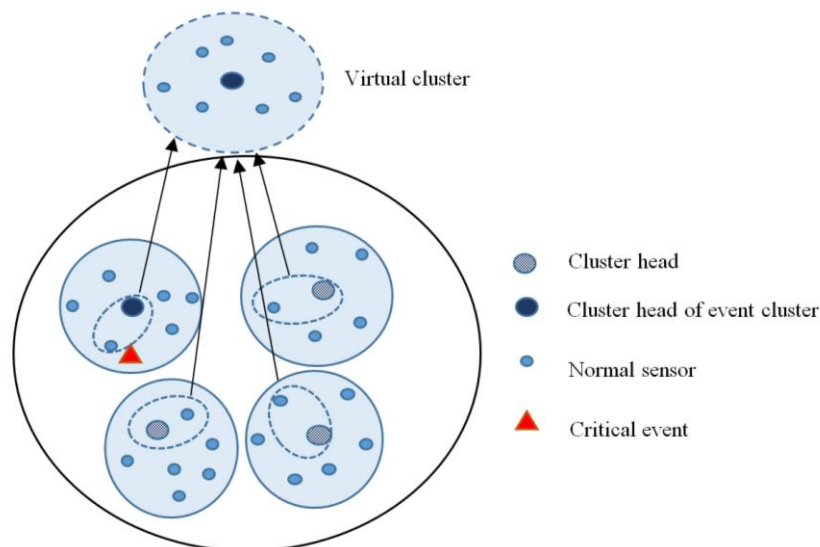


Figure 1. Virtual Cluster Formed when Critical Event is Detected in a Cluster of WSNs with Four Clusters

5. Discrete Cuckoo Searching Optimization

5.1 Cuckoo Searching (CS)

Cuckoo Searching (CS) [18-19] is inspired by the parasitism of some cuckoo bird species. These birds aggressively reproduce and then abandon their eggs in the nests of other host bird species. Some host birds behave aggressively and then throw away the alien eggs upon discovering an intrusion. Others simply leave their nests and build new nests elsewhere. Xin-She Yang and Suash Deb in 2009 uses the following representations: Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not so good solution in the nest. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions. CS is based on three idealized rules:

- 1) Each bird lays one egg at a time. The egg gets placed randomly among the host bird nests.
- 2) The nest with the highest fitness value will get carried over to the next generation.
- 3) The number of host bird nests is fixed. The probability of a host bird discovering an intrusion is set at a constant value of $pa \in [0, 1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

In generating of a new solution, the random-walk is best performed in using levy flights. The levy flight of cuckoo i is performed using

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{lévy}(\lambda) \quad (1)$$

Where $\alpha > 0$ is the step size which should be related to the scales of the problem of interests and can be selected by the user. In most cases, $\alpha = O(1)$. The above equation is essentially the stochastic equation for random walk. In general, the random walk via Lévy flight is more efficient in exploring the search space as its step length is much longer in the long run, and the product \oplus means entrywise multiplications [19-20]. The Lévy flight essentially provides a random walk while the random step length is drawn from Lévy distribution

$$\text{Levy} \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (2)$$

Which has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process with a power-law step-length distribution with a heavy tail. Based on [18-19], CS is very efficient in finding the global optima with high success rates.

5.2 Nest Representation

A solution representation for the (Vir_Clu_Gen) is a permutation representation. In this case, nests represent by a square array of size = NoC, where NoC is number of clusters. For example a WSN with four clusters will have four nests, each nest with four sensors selected randomly from the cluster sensors list. When a critical event detected by sensor S_i in cluster C_j , all clusters will participate to find the optimum virtual cluster. From figure2 if a critical event occurs in C_2 , then *NEST 2* will be the active nest and will participate in the virtual cluster with sensor S_i and the cluster head of C_2 , while the cuckoo algorithm will work on clusters (1, 2, 3, and 4), to find the best sensors that will form the virtual cluster (S_{24} from C_1 , S_{48} from C_2 , S_{100} from C_3 , and S_{77} from C_4).

<i>Sensor ID</i>	S ₁	S ₂	S ₃	S ₄
<i>Nest No.</i>				
<i>NEST 1</i>	11	24	83	3
<i>NEST2</i>	10	22	108	48
<i>NEST 3</i>	100	29	44	26
<i>NEST 4</i>	36	77	87	45

Figure 2. Nest Representation when a Critical Event Take Place in C₂

5.3 Discrete Step Size and Updating Scheme

In continuous optimization problem [18], the step size is related to the scales of the problem of interests and can be selected by the user. In most cases, $\alpha = 1$. For (Vir_Clu_Gen) step size for cuckoo can be defined as:

$$\text{Step size} = \varepsilon \quad \text{where} \quad 1 \leq \varepsilon \leq L-1 \quad (3)$$

While L is the number of clusters in the WSN. The diversity of ε as a random number allow to move in the search space in variable step size to cover the search space of our problem that facilitate the convergence of the best solution rapidly.

In our approach, we use no Lévy flight (Lévy flight=1). The exploitation around the best solutions is performed by using a local random walk, and express uniform distribution which become a standard random walk:

$$x_i^{(t+1)} = x_i^{(t)} + \varepsilon \quad (4)$$

This random walk may become too aggressive which makes new solutions jump outside of the design domain and thus wasting evaluations. Now the actual random walks or flights:

$$x_i^{(t+1)} = (x_i^{(t)} + \varepsilon) \text{ modulo } NC_j \quad (5)$$

where NC_j is the number of nodes in the cluster (C_j).

5.4 Discrete Fitness Function

If S_j denotes the sensor that detect the critical event in Cluster (C_j), and S_i denotes a sensor in cluster C_i , then our maximize fitness function is given by:

$$F = \begin{cases} -1 * 10^{10} & \text{if } dist > Grid_threshold \\ S_i.Energy & \text{otherwise} \end{cases} \quad (6)$$

Where $dist$ is the distance (S_i, S_j), $Grid_threshold$ is the distance (S_j, CH_i).

Thus the fitness function is related to the distance and energy. On maximizing this function optimum sensor node is selected from each cluster to create the virtual cluster.

5.5 Virtual Cluster Generation (Vir_Clu_Gen) Algorithm

In this section we will present our virtual cluster generation (Vir_Clu_Gen) algorithm to generate a cluster with extra resources in wireless sensor network. The input parameter

is the set of all sensors in the WSN and the number of clusters while the output is a set of sensors that will collaborate to generate the virtual cluster.

Algorithm *Vir_Clu_Gen*

Input : $S1 = \{s_1, s_2, \dots, s_n\}$, n is the number of sensors in the network;

NoC number of clusters

Output: $S2$, set of sensors that will form the virtual cluster

1. For $j= 1$ to NoC do ;
2. Find S_j ; set of all sensors in cluster j
3. Generate a host nests x_i ($i=1, 2, 3, \dots, NoC$) ;
 - Repeat
 - Get a sensor randomly from S_j (say k);
 - Evaluate its fitness F_k using Equation (6);
 - Choose a sensor among x_i (say k_1) randomly;
 - If ($F_k > F_{k_1}$),
 - Substitute by the new sensor;
 - end
 - Replace a fraction (pa) of worse sensors by new sensors;
 - Keep the best solution (the sensor with maximum residual energy);
 - Until stop criterion met
 - $S2 = S2 + S_k$
 - $S2 = S2 + CH_j$; CH_j is the cluster head of the cluster j
 - endfor
4. End

5.6 Energy Consumption Model

Power management is a major concern as sensor nodes do not have fixed power sources and rely on limited battery power. Sensor applications executing on these devices have to make tradeoffs between sensor operation and conserving battery life. The sensor nodes should provide adaptive power management facilities that can be accessed by the applications. From the sensor grid perspective, the availability of sensor nodes is not only dependent on their load, but also on their power consumption. Thus, the sensor grid's resource management component has to account for power consumption. According to energy model proposed in [1], when sending m bit data over a distance d , the total energy consumed by a node is given by:

$$E_{Tx}(m,d) = E_{elec} * m + \epsilon_{amp} * m \quad (9)$$

$$E_{Rx}(m) = E_{elec} * m \quad (10)$$

Where $E_{Tx}(m,d)$ is the energy consumption of transmitter, and $E_{Rx}(m)$ is the energy consumption of receiver. E_{elec} is the energy consumption of electronics per bit in the transmitter and receiver sensor nodes. ϵ_{amp} is the energy consumption of amplifier in transmitter sensor nodes, which can be calculated by:

$$\epsilon_{amp} = \begin{cases} \epsilon_{fs} * d^2, & \text{when } d < d_0 \\ \epsilon_{mp} * d^4, & \text{when } d \geq d_0 \end{cases} \quad (11)$$

Where d_0 is the threshold distance, ϵ_{fs} is the free space communication energy parameter, and ϵ_{mp} is the multipath communication energy parameter. The radio energy dissipation model is shown in Figure 3.

For a normal node the residual energy after sending m data bits to its cluster head is:

$$E_{res-N} = E_{ini} - E_{Tx}(m, d_{N,CH}) \quad (12)$$

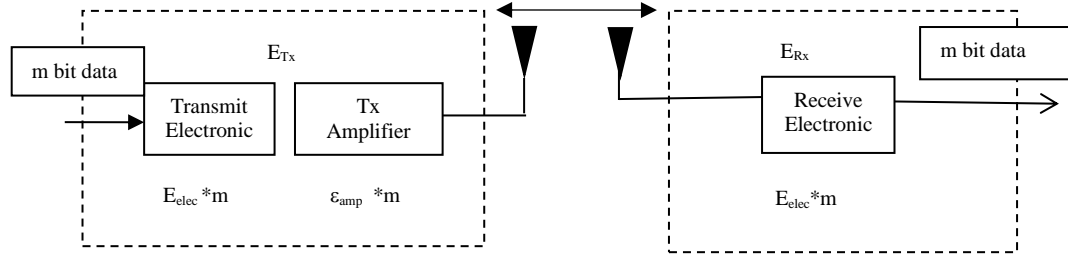


Figure 3. Radio Dissipation Energy Model

Where E_{ini} is the initial energy, and $d_{N,CH}$ is the distance from the normal node to its cluster head.

While in the classical data aggregation, for a cluster head node the residual energy is:

$$E_{res-C} = E_{ini} - E_{rx}(m) - E_{DA} - E_{Tx}(m, d_{C,BS}) \quad (13)$$

Where E_{DA} is energy for data aggregation, and $d_{C,BS}$ is the distance from cluster head node to the base station. While for our proposed system there is no data aggregation operation, so there is no energy consumption for data aggregation and the cluster head residual energy is:

$$E_{res-C} = E_{ini} - E_{rx}(m) - E_{Tx}(m, d_{C,BS}) \quad (14)$$

The proposed algorithm was simulated using Matlab, and the network model is based on the following assumptions:

1. N sensors are randomly dispersed within a field of interest in area $A=M*M$. The Base station is positioned in the center of the squared region.
2. Each sensor node knows its position using one of localization services.
3. The sensor nodes and the base station are stationary. The location of the base station is known by each node. Each sensor with enough energy can communicate with the base station directly.
4. All nodes are homogeneous, have the same capabilities.

The parameters assumptions for our simulation model are shown in Table 1.

Table 1. Simulation Parameters

Parameter	Value
Sensing area ($M*M$)	200*200 m ²
Number of nodes (N)	500
Initial energy of nodes (E_0)	0.3 joules
Electronics energy (E_{elec})	$5*10^{-8}$ joules
Position of BS	100,100
Amplification energy for free space model (ϵ_{fs})	10^{-11} joules
Amplification energy for multipath model (ϵ_{mp})	$13*10^{-17}$ joules
<i>Distance_threshold</i>	78 m

6. Simulation Results

We simulate detection of an emergency event in a network of 500 sensors deployed randomly in a field of $200 \times 200 \text{ m}^2$ and arranged in 9 clusters each with its cluster head labeled by circles, as shown in Figure 4.

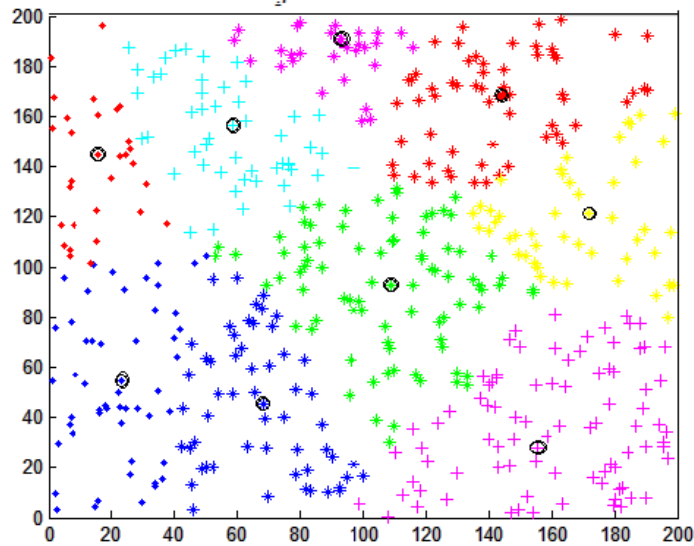


Figure 4. Wireless Sensor Network with Nine Clusters and its Cluster Heads

The critical event take place randomly in a cluster in the network, and detected by many sensors in the cluster, a single sensor will take the role to annotate to its cluster head that superior action happened and need loftier attention by initiating of special process that need extra computational and memory resources.

We employ the cuckoo search optimization algorithm to find extra computational resource from each cluster in the WSN by finding the optimum sensor (relatively close to the event and with higher residual energy). These sensors with its cluster heads will cooperate to construct the virtual cluster that is accommodates to run the required application. Figure 5 shows the WSN when the critical event occurs and the sensors are selected by cuckoo search algorithm to generate the virtual cluster.

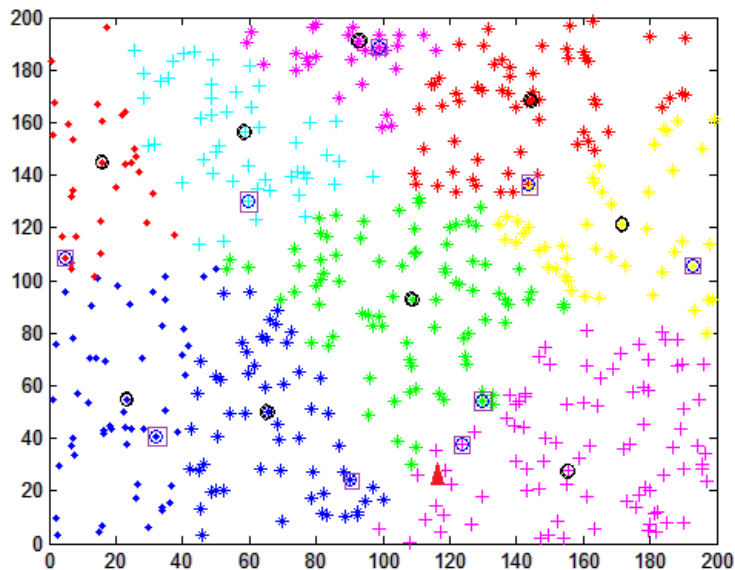




Figure 5 Virtual cluster consist of 18 sensors

7. Conclusion

Wireless sensor networks and grid computing are promising technologies that are being adopted in the industry, detecting volcanic eruptions, earthquakes and tsunamis, detecting and tracking people crossing borders illegally, potentially identifying and tracking enemy soldiers in times of war, measuring and recording changes in environmental conditions. In this paper, and under the event classification condition, we proposed creation of a virtual cluster composed of high residual energy sensor and cluster head that are selected optimally from each cluster using cuckoo search optimization (CSO) algorithm.

Computational grids, and data grids are involved with the sharing of processing power, memory, and data across multitude of sensors toward producing a resources that can be utilized to run resource consuming task that represent a response to a high priority event while it's difficult to run under the ordinary resources of the wireless sensor network cluster.

Acknowledgements

This work was partially supported by National Key Technology R\&D Program (2012BAD35B06); Program for New Century Excellent Talents in University (NCET-12-0164); National Natural Science Foundation of China (61370094); Natural Science Foundation of Hunan (13JJ1014).

References

- [1] A. Hac', "Wireless Sensor Network Design", John Wiley & Sons, United Kingdom, (2003).
- [2] I. Foster, C. Kesselman and S. Tuecke, "The anatomy of the grid, enabling scalable virtual organization", International Journal on Supercomputer Applications, vol. 15, no. 3, (2001).
- [3] I. Foster., "What is the Grid? A three point checklist", Argonne National Laboratory, fp.mcs.anl.gov/~foster/Articles/WhatsTheGrid.pdf, (2002).
- [4] W. Chen, J. Hou, and L. Sha. Dynamic clustering for acoustic target tracking in wireless sensor networks, *IEEE CS, CASS, ComSoc, IES, & SPS*, (2004) July.
- [5] K. Pister, J. Kahn, B. Boser, and S. Morris., "SMARTDUST—Autonomous sensing and communication in a cubic millimeter", <http://robotics.eecs.berkeley.edu/~pister/SmartDust>.
- [6] M. Miller, N. Vaidya. Power save mechanisms for multi-hop wireless networks. In Proceedings of the First International Conference on Broadband Networks (BROADNETS'04), (2004) October.
- [7] C. Gu and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks", In Proceedings of the 10th annual international conference on Mobile computing and networking, (2004) September.
- [8] C. Hsin and M. Liu, "Network coverage using low duty-cycled sensors: random & coordinated sleep Algorithms", In Proceedings of the third international symposium on Information processing in sensor networks, (2004) April.
- [10] G. Xing, C. Lu, R. Pless, and J. O'Sullivan, "CoGrid: an efficient coverage maintenance protocol for distributed sensor networks", In Proceedings of the third international symposium on Information processing in sensor networks, (2004) April.
- [11] I. Foster, C. Kesselman, J. M. Nick, and S. Tuecke, "The physiology of the grid, an open grid services architecture for distributed systems integration", Open Grid Service Infrastructure WG, Global Grid Forum, (2002) June.
- [12] A. Agarwal, D. Norman, A. Gupta, "Wireless grids: approaches, architectures and technical challenges", MIT Sloan School of Management, (2004) January.
- [13] B. Jacob, M. Brown, K. Fukui and N. Trivedi, "Introduction to grid Computing", IMB corporation, (2005).

- [14] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless sensor networks: A survey", *Computer Networks*, vol. 38, no. 4, (2002) March, pp. 393–422.
- [15] G. Wang, Y. Ji, D. Marinescu and D. Turgut, "A routing protocol for power constrained networks with asymmetric links", In *Proceedings of the 1st ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, (2004) October.
- [16] L. McKnight, J. Howison and S. Bradner, "Distributed resource sharing by mobile, nomadic, and fixed devices", *IEEE INTERNET COMPUTING*, (2004) August.
- [17] V. Welch, "Security for grid services", In *Proc.of the 12th IEEE Intl Symp on High Performance Distributed Computing (HPDC '03)*, (2003) June, pp. 48-57.
- [18] A. Gilbert, "New tech consortium to focus on sensor networks", <http://earthlink.com.com/New+tech+consortium+to+focus+on+sensor+networks/2100-10333-5471817.html>, Nov. 30, (2004).
- [19] X. S. Yang, "Nature-Inspired Metaheuristic Algorithms", 2nd Edition, Bristol, U.K: Luniver, (2010).
- [20] X. S. Yang and S. Deb, "Cuckoo Search via Levy Flight. In: *Nature & Biologically Inspired Computing (NaBIC)*", IEEE publications, USA, (2009), pp. 210-214.
- [21] X. S. Yang and S. Deb, "Engineering Optimization by Cuckoo Search. In: *Int. J. Mathematical Modelling and Numerical Optimization*", vol. 1, no. 4, (2010), pp. 330-343.

Authors



May Kamil Al-Azzawi, she received hers B.Sc. Degree in Computer Science from Baghdad University 1993, and Master's Degree in computer Sciences, University of Technology in 2001, She continued working in academic teaching in AL-Mansour University College for 16 years, She is now a Ph.D. degree student at Hunan University in P.R. China.



Luo Juan, Ph.D., associate professor, in 1997 graduated from the National Defense University, 2000 Masters graduated from Wuhan University, 2005 PhD graduate of Wuhan University. 2000 worked Wuhan Youkeyuan FHN companies, has served as director of software development and project manager. July 2005 joined Computer and Communication, Hunan University. Current research interests include middleware, wireless sensor networks, wireless embedded systems, cloud computing



Li Renfa, he was born in Hunan – China in 1957, he works as Professor and PhD supervisor of Hunan University, Senior member of China Computer Federation (CCF), his main research interests include Wireless Networks, artificial intelligence and optimization theory etc.

