

# **An Efficient Clustering of Sensors using a Meta Heuristic Algorithm for IoT**

Varsha Deshpande  
Department of Computer Science,  
KLSGIT,  
Belagavi, India

## **ABSTRACT**

In recent years, with the increase in the devices like Smartphone's, Global Positioning System (GPS) monitoring devices, tablets, surveillance cameras etc...The numbers of devices that have been connected to the Internet have crossed the world population. The decrease in the cost of the devices, with increase in their capability was one of the major reasons for this change. With the increase in the devices connected to the network, there is enormous data that is produced by the devices on daily basis which has caused problem of selecting the devices (sensors) based on the data produced by them. The users have to face the problem of manually selecting the set of sensors required by them. To target this problem, the proposed work has implemented, the ant based clustering algorithm with some novel changes in order to provide the user with the optimal set of sensors. The algorithm intakes the whole sensor space and outputs the clusters from which the user can choose the one which is most optimal according to the user needs. The antclust algorithm strives to provide optimal solution and maintain its performance while fighting the dynamicity of the Internet. The proposed work shows better scalability and adaptability. Also has faster search of the sensors than the other works previously proposed.

## **General Terms**

Algorithm, Bio-inspired, Clustering, Sensor-Searching, Simulation

## **Keywords**

AntClust, Contiki, Cooja, Internet of Things (IoT)

## **1. INTRODUCTION**

Internet of Things (IoT) is continuously making its mark on the world. It can be used to describe any devices that can be connected via Internet. With IoT the world of web has turned into being more sense aware [1]. Sensors can be distributed based on the type of data it produces like temperature, pressure, vibration etc. The world is ready to face any situation proactively than being reactive. IoT portrays the increasing number of devices that are connected, beyond peer to peer covering a variety of services [2]. IoT is now facing a challenge of managing the huge data that is produced by these devices and maintaining the data in a proper format.

There are billions of devices connected and the devices need to be searched. The huge search space and the high velocity aggregation of data, creates a need for effective method in order to search the sensors and the data provided by them. The existing approaches do not deal with the dynamicity of the Internet. The whole motto of this work is to provide an efficient way for users to select sensors with minimal time consumption.

In the proposed work, the ant based clustering algorithm [3] is considered. The algorithm has some novel changes in order to

increase the performance of the system against dynamicity of the Internet of Things. The algorithm considers both, type of sensors and its characteristics in order to cluster the sensors. This approach is efficient because it can do both serial as well as parallel search of the sensors available in order to obtain the close to optimal solution. This nature of the ant based clustering algorithm, encounters the space problem more deeply and provides better solutions than the greedy algorithms which do only local search. It deals with both the distributed and dynamic nature of the IoT and shall have high performance.

## **2. RELATED WORK**

In fields like smart environment monitoring and IoT, searching the sensors is the key service. Depending on the type or characteristic, the sensors have been searched and selected previously. Pertinence can be resulted by checking of sensors' data or/and their properties. This part provides a brief view of the various works that have been put forward by various researches in the field of searching and selecting of the devices (sensors) in the IoT.

The early works focused on the data generated by the sensor and the clustering was based on it. The key thought in such works was to obtain the optimal subset of devices (sensors) that provide desired data at any instance of time.

For instance, Elahi et al. [4] proposed an approach considering the data generated by the devices. The approach built prediction models by studying the relations between sensors. The models created were used to predict which group of sensors would provide the desired value. The approach also provided the users with the ranked sensor. The models were built using the people-centric approach and the output of people centric sensors; depend on the behavior of the people which is not periodic. This was one of the challenges that the approach faced.

Another such approach created a real time search engine for the Internet called Dyser [5].The search engine allowed making predictions of the sensors registered in the network and to search the entities in real time and developed models to predict the state of the devices connected. After submission of the query by the users, the engine pulled the new data in order to identify its match with registered sensors. There were three Prediction models considered namely Aggregated Prediction Model (APM), Single Period Prediction Model (SPPM) and Multi Period Prediction Models (MPPM). The paper considered sensor gateways that created Prediction models and published sensor information on web. The indexer provided an index to each Sensor which is used by the reducer to select best set of sensors. Although the approach provided promising results, the approach was content-based which, was the major drawback.

Truong et al. [6] also came up with an approach that utilized the fuzzy sets for calculating a resemblance score for any two

sensors that were used to get a scored list of similar devices (sensors). A similarity score for one sensor with respect to other was calculated using the sensor range difference. The fuzzy sets were first constructed. The sensor range helped finding sensors with either similar properties or similar output. The more the belonging to two fuzzy sets, the more similar. Although this approach was also content-based, it can be used in some applications, for instance, monitoring whether sensors are functioning properly, identifying sensors that need maintenance, deploying new sensors, or defining new services based on particular sensors.

Even when data of data was provided, users had to know every level of detail such as database schemas or names of features that are particular to a device or operating. Therefore job of logically searching, relating and gathering sensor data was very burdensome. The ontology-based approach [7] consisted of revealing sensor observations in terms of reasoning enriched with linguistically similar metadata, providing data such as: which sensor recorded what, when, where, and in which conditions? The paper presented a framework whose key features were:(i) framework supported semantic-enriched query processing based on ontology information.(ii) The framework employed the semantic sensory network ontology, along with domain-specific ontology's.(iii) The framework enabled scalable search over distributed sensor data .

There are some other methods which had other drawbacks like For instance, the techniques used in Linked Sensor Middleware (LSM) [8] for sensor search and selection. To facilitate the combination of generated data with data from other resources, both sensor sources and generated data were being provided with acceptable descriptions, creating Linked Stream Data. One of the main characteristics of such data was its "live" nature. To target this matter they created Linked Stream Middleware (LSM), a firmware that brings together the "live" data and the web data. Though the middleware had potential it had limitations like sensor selection based on location and type. It was also difficult to store the large data produced. One direction of improvement was to create abstractions of live data which would reduce the need for large storage space.

There were other methods developed with LSM like GSN [9] and Microsoft SensorMap [10] was also created. GSN is a non-specific engine that aims to provide a middleware that is adjustable in order to target the issue of sensor data integration and distributed query processing. In GSN, all sensors are listed in a box from which the users need to select. In Microsoft SensorMap sensors are selected by users by using information like location, index terms and the type of sensors.

Perera et al. [11] proposed a method for sensor searching, selection, called CASSARAM, in order to select the sensors in an efficient way by considering their capabilities and functions. The sensor searching considered user priorities. This method involved collecting Point based and Proximity based needs from users and then plotting them in multidimensional space. The sensors were then ranked based on the index value. The index value was calculated using weighted Euclidean distance in multidimensional space technique. Perera et al. [11] did not consider the changes that occur in IoT. In other words, they ignored the performance of their system in dealing with the changes (e.g. joining new sensors) in the sensor network.

### 3. SYSTEM MODEL

#### 3.1 Existing System

LEACH [Low Energy Adaptive Clustering Hierarchy] is a Time Division Multiple Access [TDMA] based Medium Access Control protocol which is combined with the clustering problem in the WSN. In traditional LEACH, the cluster and cluster heads are formed in two phases namely the set up phase and the steady phase [12].

1) Set up phase: In Set up phase, each sensor (node) creates a random value between '0' and '1' and then compares it with the threshold value in order to select the cluster head (CH). If the random value is less than the threshold then that node is elected as CH. The node then informs all the other nodes by sending a message. The nodes decide to join a cluster by considering the energy of the message they receive. The nodes then send a request to CHs and the CHs then approve their membership after receiving all the requests [13].

2) Steady state phase: In Steady phase, the nodes that want to transmit messages send the data to the CHs. The CHs then combine the received data and send it to the base station [13].

The conventional LEACH has shortcomings like:

- 1) It elects the cluster heads based on the random value which is not very efficient as any node with a higher random value and lower energy is elected as cluster head which hinders the performance of the network.
- 2) It does not consider the distance between the base station and the cluster head.

The one that is used for comparison in this work is a modified version of LEACH protocol and considers the distance between the cluster head and the base station

Although the protocol considers distance between sensors, it does not provide near to optimal results as it does not consider the dynamicity of the internet.

#### 3.2 Proposed System

Proposed approach is meta-heuristic in nature. In order to form clusters in which sensors (devices) that belong to similar types like weather, traffic, etc... are grouped together. In each cluster, sensors are clustered by using clustering algorithm that is inspired by ants and clustering occurs based on the sensor characteristics. The ant based algorithm, reduces the sensor search space and gradually increases the efficiency of searching process. Furthermore, there are novel changes in order to make sensor search process cost effective and propose an adaptive strategy to maintain the performance of system against dynamicity of IoT. The architecture of Ant-based clustering system is shown in Figure 1.

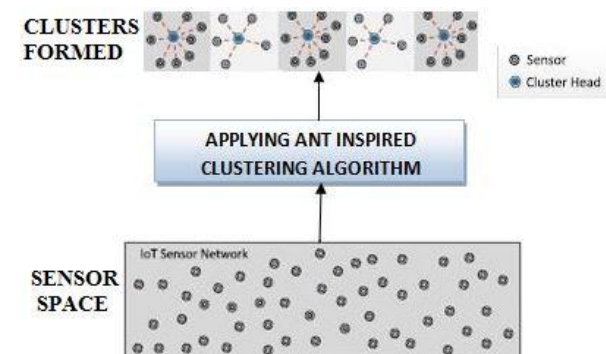
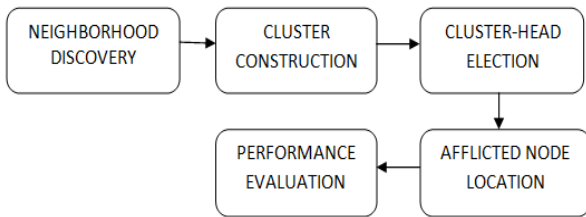


Figure 1: Architecture of Ant-based Clustering System

In specific, the development of the system is divided into five stages as explained below. The phases of the system development are depicted in Figure 2.



**Figure 2: Development Stages of Ant-based Clustering System**

### 3.2.1 Neighborhood Discovery

In the first stage, every sensor will for a certain time, send its unique identification (ID) also known as HELLO messages to other sensors for certain amount of time. The HELLO messages will be caught by other nodes (sensors) to create a list of neighboring nodes along with their respective identification (ID). No HELLO messages are forwarded to other nodes. When this stage ends, all the sensors will have the information of their 1-hop neighbors. This process of localization is valid for the nodes that are active during that instance of time. The sensors that become active later are not considered. The nodes are called neighbors if they are adjacent, i.e. within each other's frequency transmission range. The number of neighboring nodes to a node is called its degree.

### 3.2.2 Cluster Construction

The aim of this stage is to form clusters and there is some form of simple synchronization between the nodes, though there is no time-synchronization. In initialization phase, every ant agent is either marked as loaded or unloaded. Every loaded ant will drop the sensor near the Most Similar Sensor (MSS) in order to form the cluster. The unloaded ant will pick up the sensor from unvisited list of nodes without considering the probability of picking it up and then drop it near the MSS or empty space using its short term memory

In main clustering phase, the ant agent does consider what if the sensor cannot be dropped near MSS. If that is the scenario then it drops sensor anywhere randomly. Every ant agent is provided with a memory [14] in order to bias between the locations where the sensor can be dropped. Every sensor has a field of view that defines the number of sensors the ant can remember. It increases with the size of the cluster. Every cluster will be provided by a cluster number (CN) that is assigned to sensors depending on the cluster to which they belong.

### 3.2.3 Cluster-Head Election

The cluster head election in the ant based clustering algorithm is inspired by the MDS-Managed Print Services (P) [15]. The restraining of the nodes is done by the nodes merging its local maps to construct the global map that indicates the unique coordinates of all the nodes in the network. But only the cluster heads perform the MDS functions. This highly decreases the computational overhead as only few nodes are considered to be elected as cluster-head. But the main underlying factor behind selecting the cluster head is distance. The cluster-member that is at the centre of the cluster is more likely to be elected as cluster-head.

The cluster-members are not regarded as active. They will transfer messages, receive messages and reply to requests to and from the cluster-head. The aggregation of data will be

initialized by the cluster-head and the cluster-head will request all the other nodes within the cluster to send the data. The adjacent clusters are called 1-step neighbors. The cluster-heads communicate this information to other cluster-heads in order to keep track of 1-step neighbors as well as 2-step neighbors. In this work 2-step neighbors are considered as limit but, can be extended as Ant-Inspired algorithm can support it. The 2-step neighborhood provides a negotiable data size and message complexity.

### 3.2.4 Afflicted Node Election

The goal of ant based clustering algorithm is to study the unsupervised clustering of nodes. The algorithm is basically inspired by the ant behavior and the information exchanged between the ants. The ant remembers its way due to the chemical extracted by it along the path. Similarly the ant agents in the proposed algorithm use short term memory to identify the nodes visited. Once the cluster is formed, the cluster-head then communicates the information of the exact location of all nodes to every node in order to know their neighboring nodes. This also helps in knowing the nodes that affect or hinder other nodes.

### 3.2.5 Performance Evaluation

The simulation of the ant based clustering algorithm in this work is implemented using contiki tool. The implementation utilizes the Media Access Control (MAC) layer and physical layer from 802.11 as there are no specific implementations.

## 3.3 Ant-Inspired Meta-Heuristic Clustering Algorithm (AIMHCA)

Input: The sensors that form the sensor space.

Output: Clusters of sensors.

/\*Initialization\*/

While the sensors are unattended

if the ant is unloaded, then Pick-up the sensor  $sn_i$  and remove sensor  $sn_i$  from unattended list  
end if

else

if the ant is loaded with sensor  $sn_i$  and the ant finds a similar sensor then drop the sensor  $sn_i$  near the Most Similar Sensor(MSS) in the ant's memory and change the cluster number of  $sn_i$  to that of MSS

$sn_i$ \*Cluster Number ← MSS\*Cluster Number end if

if the ant is loaded with  $sn_i$  and does not find a similar sensor then drop  $sn_i$  in any empty space available and add a new cluster number to  $sn_i$

$sn_i$ \*Cluster Number ← New Cluster Number  
end if

end while

/\*Main Loop\*/

For each ant agent

if Ant agent  $A_v$  is unloaded, randomly select  $sn_i$  only if the probability of random is higher than that of  $sn_i$   
end if

if Ant agent  $A_v$  is loaded by  $sn_i$  then find the MSS in memory and drop it near it and update the cluster number of sensor

$sn_i$ \*Cluster Number ← MSS\*Cluster Number  
end if

if Agent  $A_v$  is loaded with  $sn_i$  then drop  $sn_i$  only when the drop probability of  $sn_i$  is greater than that of random

end if

```

if the number of unsuccessful tries is greater than  $\delta$  ,
    then update  $\alpha$ 
     $\alpha \leftarrow \alpha + \Delta\alpha$ 
end if
if The value of function  $f(i) > \theta$  and the field of view
     $S^2 < S_{max}^2$  then update
     $S^2 \rightarrow S^2 + n_s$ 
end if
end for
/* Maintaining the dynamicity*/
if the Number of changes are less than the threshold
    if a sensor joined the network then find most
    similar cluster head(MSCH) and add the sensor
    to that cluster and update the MSCH
    end if
    if the context properties of the sensor have
    changed then find MSCH based on the new
    properties and change the cluster of sensor and
    update both new and old cluster
    end if
    if a sensor left the network then remove the
    sensor from cluster and update its cluster head
    end if.
    Increment the Number of changes by 1
end if
if Number of changes exceeds threshold then set
    Number of changes to zero and rerun the
    AIMHCA
end if

```

The ant based algorithm [3] is inspired by the ant behavior. In the proposed algorithm, there are  $N_{sn}$  sensors considered and  $N_{an}$  ants in the network. The ants initially move around randomly picking up the sensors and then dropping them in the space near the (MSS).

The similarity between the sensors is calculated using the equation (1).

$$\text{Function } f(sn_i) = \left(0, \frac{1}{S^2} \sum \left(1 - \frac{\text{Euclidean distance}}{\alpha\mu}\right)\right) \quad (1)$$

In the equation (1) above  $S^2$  is the view of field [16] of any ant. The formula in turn depends on the Euclidean distance given by equation (2)

$$d(sn_i, sn_j) = \sqrt{\sum_{n=1}^{N_{Attr}} (sn_i \text{ kth attribute} - sn_j \text{ .kth attribute})^2} \quad (2)$$

Where,  $Attr_k$  is the kth attribute of the devices and the pickup and the drop probabilities [17] are computed as depicted in equation (3) and (4). The pickup and drop probabilities are compared with a random value in order to decide whether to pick up or drop the sensors

$$p_{pickup}(sn_i) = \left(\frac{k_p}{k_p + f(sn_i)}\right)^2 \quad (3)$$

$$p_{drop}(sn_i) = \left(\frac{f(sn_i)}{k_d + f(sn_i)}\right)^2 \quad (4)$$

In equation (3) and (4)  $k_p$  and  $k_d$  are thresholds of picking up and dropping of sensors.

The ant agent keeps track of the previous sensor dropped in a temporary memory as it needs to know where the next similar sensor needs to be dropped. The clusters will be provided with a number, that uniquely identifies it and the sensors with

highest similarity are grouped into one cluster and they all have same cluster number (CN).

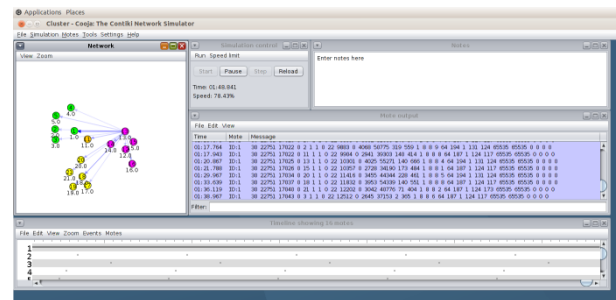
## 4. RESULTS

The experimentation results of the concept are acquired using contiki tool which has inbuilt cooja simulator. Contiki is an operating system that is mainly meant for networked systems and the systems that have memory limitations as of low power devices of the Internet of Things. The cooja simulator has various unique features like it has inbuilt feature to collect the data from motes (sensors or devices) and output the performance parameters. All that one has to do is to create the network that one desires. The simulation parameters are as depicted in Table 1.

**Table 1. Simulation Parameters**

Symbols	Parameters
$E_o$	Initial Energy (10 joules)
$E_l$	Receiving Energy
$E_t$	Transmitting Energy
N	Number of Sensors
$B_i$	Beacon Interval
P	Power

The sensor network is shown in Figure 3. The performance parameters of the project can be collected from the collect view option of “tools” icon. The node information option provides a lot of information like the number of hops, beacon interval, receive power, transmit power etc.



**Figure 3: Simulation Interface**

### 4.1 Residual Energy

The residual energy can be defined as the difference between the initial energy and that of the energy lost in either receiving or transmitting of packets.

$$E_r = E_o - (E_l + E_t) \quad (5)$$

In equation (5)  $E_o$  represents initial energy,  $E_l$  represents receiving energy,  $E_t$  represents transmitting energy and  $E_r$  represents residual energy.

The residual energy decreases with time because, the traffic increases i.e. the receiving and transmitting energy increases and the difference between the energies reduces, implying maximum utilization of the sensors. The obtained results are as shown in Figure 4. The x axis represents simulation time in hours and y axis represents residual energy in mill joules.

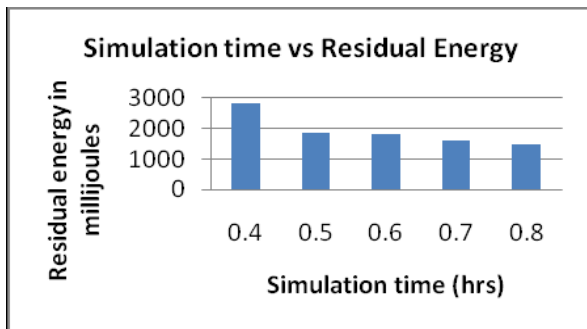


Figure 4: Graph of Simulation Time vs. Residual Energy

## 4.2 Power Consumption

The simulation of the algorithm shows the average power consumption of 25 nodes (motes), run for five time instances. The graph depicts that the power consumption of the cluster head is higher than the other nodes as it is the one that communicates with all other nodes. In Figure 5, x axis represents the nodes and the y axis represents the power in mw. In Figure 5, nodes 1, 3, 7, 13, 19 and 22 are the cluster heads.

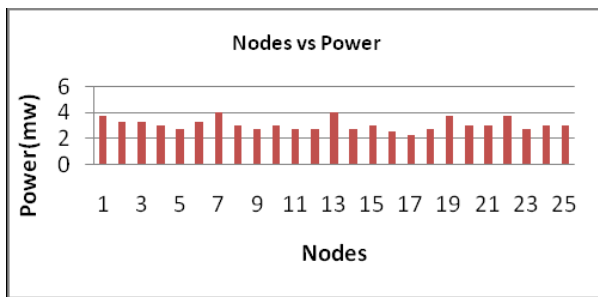


Figure 5: Graph of Power vs. Nodes

## 4.3 Comparison

### 4.3.1 Simulation Time vs. Residual Energy

The experiments were carried out by keeping the sensors constant to 25. Figure 6 shows the comparison of AIMHCA and LEACH with respect to residual energy. The graph depicts that AIMHCA outperforms LEACH by 3.94%. This is because AIMHCA considers the dynamicity of IoT which is not considered by LEACH.

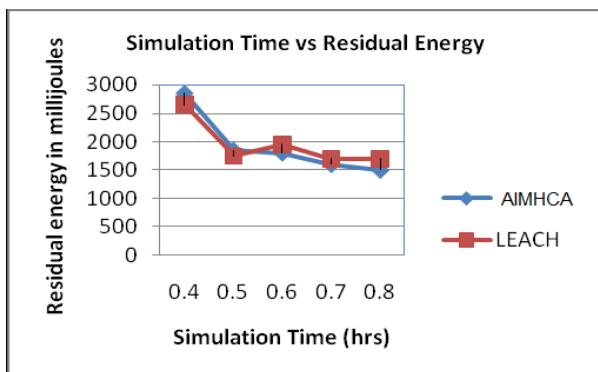


Figure 6: Comparison of Residual Energy.

### 4.3.2 Simulation Time vs. Throughput

Figure 7 shows the comparison of AIMHCA and LEACH with respect to throughput. The graph depicts that Throughput of proposed AIMHCA performs better than the LEACH by 11.32%. This is because Ant inspired algorithm considers the changes that occur in network and provides a way to handle it which is not considered in LEACH.

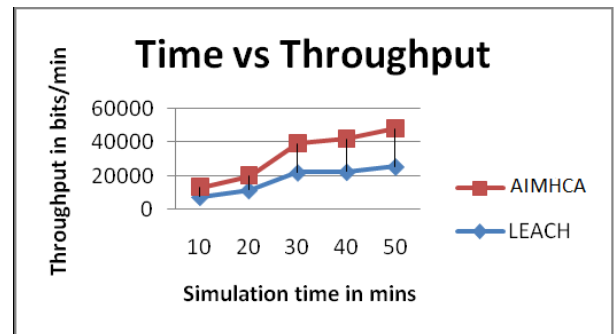


Figure 7: Comparison of Throughput

## 5. CONCLUSION

With the increasing population of users, the use of wireless devices that are connected to the internet, there is tremendous amount of data produced by them and it is becoming difficult to select a sensor by analyzing the data produced by it. The proposed work has put forth an approach to search and select the sensors by using an ant inspired algorithm called AIMHCA. The algorithm searches the sensor space and clusters the sensors in a form similar to that of ant behavior. The algorithm deploys the biased search approach as every ant agent is provided with a temporary memory in order to find the most similar sensor. The ant based clustering algorithm outperforms the traditional clustering algorithm. Comparison of ant based clustering algorithm with the existing system shows that the ant based algorithm is better in aspects of residual energy and throughput. This is because, the algorithm considers the changes that occur in the network and the algorithm is rerun only when it exceeds the threshold value hence causing higher utilization of network. The results prove that the algorithm is ascendable and meets the dynamicity of the internet world. In the future, the system can be tested by increasing the buffer size (memory) of the ants and also increasing the threshold value of acceptable changes in the network. The AIMHCA can be combined with other sensing approaches to observe the effects on sensitivity.

## 6. ACKNOWLEDGMENTS

I thank all the professors, friends and parents for their constant support and encouragement.

## 7. REFERENCES

- [1] J. Cooper, A. James, Challenges for database management in the internet of things, IETE Tech. Rev. 26 (5) (2009) 320–329.
- [2] J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S. Karnouskos, D. Boyle, From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Academic Press, 2014.
- [3] Mohammad Ebrahimi , Elaheh ShafieiBavani , Raymond K. Wonga, Simon Fong and Jinan Fiaidhi “An adaptive meta-heuristic search for the internet of things” accepted on December 2015. Springer.
- [4] B.M. Elahi, K. Romer, B. Ostermaier, M. Fahrmaier, W. Kellerer, Sensor ranking: A primitive for efficient content-based sensor search, in: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks, IEEE Computer Society, 2009, pp. 217–228.
- [5] B. Ostermaier, K. Romer, F. Mattern, M. Fahrmaier, W. Kellerer, A real-time search engine for the web of things,

- in: Internet of Things (IOT), 2010, IEEE, 2010, pp. 1–8.
- [6] C. Truong, K. Romer, K. Chen, Fuzzy-based sensor search in the web of things, in: 2012 3rd International Conference on the Internet of Things (IOT), IEEE, 2012, pp. 127–134
- [7] J.-P. Calbimonte, H. Jeung, O. Corcho, K. Aberer, Semantic sensor data search in a large-scale federated sensor network.
- [8] D. Le-Phuoc, H.N.M. Quoc, J.X. Parreira, M. Hauswirth, The linked sensor middleware-connecting the real world and the semantic web, in: Proceedings of the Semantic Web Challenge 152
- [9] K. Aberer, M. Hauswirth, A. Salehi, Infrastructure for data processing in largescale interconnected sensor networks, in: 2007 International Conference on Mobile Data Management, IEEE, 2007, pp. 198–205.
- [10] S. Nath, J. Liu, F. Zhao, Sensormap for wide-area sensor webs, *computer* 40 (7) (2007) 90–93.
- [11] C. Perera, A. Zaslavsky, C.H. Liu, M. Compton, P. Christen, D. Georgakopoulos, Sensor search techniques for sensing as a servicearchitecture for the internet of things, *IEEE Sens. J.* 14 (2) (2014) 406–420.
- [12] Monika, Sneha Chauhan, Nishi Yadav LEACH-I Algorithm for WSN 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015)
- [13] Rohini Sharma, Narendra Mishra and Dr. Sumit SrivastavaA proposed energy efficient distance based cluster head (DBCH) Algorithm: An Improvement over LEACH. Elsevier, *Procedia Computer Science* 57 ( 2015 ) 807 – 814
- [14] E.D. Lumer, B. Faieta, Diversity and adaptation in populations of clustering ants, in: Proceedings of the Third International Conference on Simulation1 of Adaptive Behavior: From Animals to Animats 3: From Animals to Animats 3, SAB94, MIT Press, Cambridge, MA,USA,1994,pp.501–508.URL <http://dl.acm.org/citation.cfm?id=189829.190043>
- [15] A.P. Sheth, and J.A. Larson, "Federated database systems and managing distributed, heterogeneous and autonomous databases", *ACM ComputingSurveys*, vol. 22, 3, 1990, pp. 183-226.
- [16] A.L. Vizine, L.N. de Castro, E. Hrusch, Towards improving clustering ants: an adaptive ant clustering algorithm, *Informatica* 29 (2) (2005).
- [17] J.-L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L.Chrétien, The dynamics of collective sorting robot-like ants and ant-like robots, in: Proceedings of the First International Conference on Simulation of Adaptive Behavior on From Animals to Animats, 1991, pp.356–363.