The Fuzzy-Based Cluster Head Election Algorithm for Equal Cluster Size in Wireless Sensor Networks

Pichatorn Eak-Une
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
kinisanken@hotmail.com

Chotipat Pornavalai
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
chotipat@it.kmitl.ac.th

***Abstract*—In Wireless Sensor Network (WSNs), clustering technique is widely used to balance energy usages consumed by the sensors. In each round of operation, a number of sensors are chosen to be candidate cluster head (CCHs) with a fixed and predefined probability value. CCHs then compete among themselves to become Cluster Head (CH) based on some criteria such as its remaining residual energy. CH role is rotated among sensors within the network field to balance their residual energy. However, area near to the corner and edge of the network field usually has less number of sensors to be CHs role than sensors located around center of the field. This will create energy holes problem on the area where sensors that needed to be CHs more often than others. Another problem is almost existing cluster head competition methods cannot precisely control the size of clusters in the networks. Depend on the spatial correlation of sensor nodes, this may impact the quality of data aggregation performed by the CHs. In this paper, we propose fuzzy based cluster head election algorithm (called FuzzCHE) to control and maintain cluster size while balance residual energy of sensors and extend the network lifetime. With FuzzCHE, each sensor can dynamically adjust probability that each sensor becomes CCH in each round by fuzzy logic. Comparing with existing cluster head election algorithm, results from simulations show that FuzzCHE can give more precise control of the average operated cluster size in the network to the deployed size required by the application.**

***Index Terms*—fuzzy algorithm; clustering; cluster size control, wireless sensor networks;**

I. Introduction

Wireless Sensor Network (WSN) is a network that consists of sensors which is capable of collecting, processing, and transmitting data via radio waves. Number of sensors ranges from ten to thousands may be placed on the network field. Each sensor periodically observes and collects environmental data such as temperature, humidity, brightness or wildlife movement. Then it sends the data back to the sink or base station (BS) for further processing and analyzing. One of the most important restrictions in WSN is on the limited power resources. Nowadays, the main power source for sensor is normally from battery. Sensor can operate until it depleted all its available energy. Therefore, extending the network lifetime is one of the major and critical issues in WSNs.

Taking into account of the important factors required by the specific needs such as the duration of the network operation, or the coverage and so on, creating a cluster demonstrated the better performance such as extend network lifetime, and reduce power consumption [1]. By selecting some sensor nodes in the network field to be cluster heads (CHs) to form clusters, each sensor node then chooses one of the CHs and send data to its CH, instead of the sink. Each CH then aggregates data received from its sensor members (also called cluster members or CMs), and then forwards to the sink or to another CH in the multi-hop approach. Data aggregation takes small amount of power consumption but can reduce the amount of data from CMs that must be sent to the sink. However, CH will use more energy than CMs in order to form the cluster, receive data from CMs, aggregate data and send to the sink either directly or multi-hops. Therefore, CH role should be rotated among member nodes.

Basic mechanism of CH election can be summarized as follows. (1) Some sensors elected themselves to be candidate cluster head (CCHs) with a fixed and predefined probability value (called T-value). Only CCHs will enter Cluster Head (CH) competition phase. (2) Each CCH advertises itself via broadcast to other CCHs within its competition radius along with other local information such as its residual energy. This competition radius (also called competition range) is usually set to be the deployed cluster size which is required by the applications running on this network. Each CCH can then elect itself to be a cluster head (CH) based on advertisements received from other CCHs. Typically, each CCH that has the most residual energy within its competition range will be elected as a CH role to form the cluster. Other CCHs that are not elected as the CH will change their roles back to normal role or Cluster Member (CM) role. (3) Each CH broadcasts within another predefined range to announce its existent to other CMs. Each CM then selects to associate with one of the CHs, i.e., the nearest CH. This CH election method is widely used, yet simple and distributed. But there are some critical issues such as the optimal T-value is unknown, and the cluster size in the network is not precisely controlled. Aggregation of data on large cluster size is likely to be less accurate than on small cluster size due to the lower spatial correlation on the location of sensors.

In this paper, we propose fuzzy based cluster head election

978-1-5090-2033-1/16/$31.00

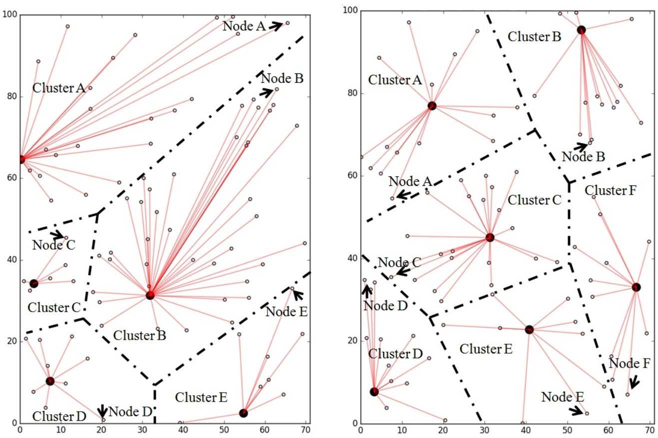
algorithm (called FuzzCHE) for controlling cluster size and balancing energy. The proposed FuzzCHE is very simple, local, low overhead, and distributed. In each round, during cluster formation, the residual energy and distance between each CM and CH is exchanged. The current average residual energy of sensors and the current operated cluster size in this cluster are then calculated and used by fuzzy engine in each sensor. The fuzzy output value is used to adjust T-value of that sensor for the next operation round. Therefore, sensor can learn and adjust its T-value in each round depending on status of other nearby sensors in the cluster. Simulation results show that FuzzCHE outperforms CH election algorithm used in UCR [2] and EC [3]. The FuzzCHE can give precise control of operated cluster size. The average operated cluster size in the network is very near to the deployed cluster size required by applications.

Reminder of this paper is organized as follows. We describe related work in section II. Network model, topology, and problem statement are presented in section III. In section IV, we present our FuzzCHE algorithm. Performance evaluation of FuzzCHE compared with EC solution is given in section V. Finally we conclude our paper in section VI.

II. Related Works

Regarding how the cluster being created, there are two categories namely uniform clustering and non-uniform clustering. In uniform clustering approach, all clusters in the field are expected to be created the same way, such as the same cluster size. In this paper, cluster size is defined as the distance from CH to the farthest CM in that cluster. Fig. 1(a) shows the area where there are 5 clusters with greatly varied in cluster sizes, but Fig. 1(b) all the cluster sizes are nearly the same. Node A-F in Fig.1 are the farthest CMs in cluster A-F respectively. Heinzelman et. al. proposed LEACH protocol [4] to use probability function to rotate the CH role where each sensor needs to be CH once for a number of predefined rounds. While it is fair and simple, the operated cluster sizes are greatly varied. Therefore, energy balancing with LEACH has limitation. The work in Nam et. al. [5] proposed an algorithm to create equal cluster size, where cluster size is defined as the number of nodes in the cluster. However, residual energy is not used in the algorithm. Therefore, it cannot be used to balance energy usage.

Enam et. al. [6] proposed DUCA clustering algorithm to create clusters by dividing network field into virtual grids of equal size. The number of CHs in the network field is expected to be very close to the number of grids in the field. DEEC algorithm [7] was proposed to use Voronoi diagram to divide the network field to equal cluster size. Then it determines the first kind and second kind of cluster heads to dynamically rotate the cluster head. Though DUCA and DEEC can create cluster with equal size, these algorithms are centralized algorithm and require all sensors know their locations by GPS and need to coordinate with the sink. Therefore, this approach is costly and has high overhead. Lin et. al. proposed FSC clustering protocol by partitioning circular network field into equal sized and fan-shaped clusters [8]. All sensors in FSC are required to know their locations by GPS. Therefore, it has the same limitation as in DUCA and DEEC.



(a) Unequal Cluster Size (b) Equal Cluster Size

Fig. 1: Unequal cluster size and equal cluster size in the network field.

On the other hand, for non-uniform clustering, each cluster may be created differently, such as different cluster sizes. EECS [9] was proposed unequal cluster size clustering where clusters far away from sink has smaller size than clusters near sink. CHs then aggregate and send data back to sink in single hop manner. UCR [2] was also proposed to use unequal size cluster but clusters near sink are smaller than clusters far away from sink. Unlike EECS, UCR transmits information from each cluster to sink with multi-hops. Wei et. al. proposed EC algorithm [3] where it partitions network field into smaller regions. The clusters in the same region are expected to have the same deployed cluster size, but different in different regions. The deployed size of clusters in each region can be calculated by EC algorithm to balance energy usage. Similar to UCR, sizes of clusters in regions near sink (inner regions) are smaller than clusters in far away regions (outer regions) in order to save intra-cluster energy.

There are many works using fuzzy logic to select proper cluster head in WSNs. For example, Kim et al. [10] proposed a fuzzy logic based election mechanism (CHEF) that uses two fuzzy inputs; residual energy and node closeness centrality with respect to immediate neighborhood. Lee et. al. proposed fuzzy-logic based clustering namely LEACH-ERE [11] with an extension to the energy prediction. Taheria et. al. proposed ECPF [12] an energy aware clustering protocol using fuzzy logic. These protocols can extend the network lifetime but the cluster size cannot be controlled. For non-uniform clustering, Bagci et. al. proposed EAUCF algorithm [13] that used Fuzzy Logic to adjust the cluster size according to residual energy and distance to the sink.

If clusters in the network field have different sizes, information from each sensor in a cluster will be aggregated by its CH with different granularities. Aggregation of data on large cluster size (as in cluster B in Fig. 1(a)) is likely to be less accurate than on small cluster size (as in cluster C in Fig. 1(a)) due to the lower spatial correlation on the location of sensors in cluster B. In practice, the size of the cluster at a specific location should be assigned by the application requirement, not outcome of the protocol. Therefore, some particular applications that required uniform granularity of information may be affected by unequal cluster sizes. In summary, to the best of our knowledge, there is no work on distributed yet simple algorithm that can control the size of cluster to the deployed value required applications while still be able to balance energy among sensors, and therefore extend the network lifetime.

III. Network Model and Problem Statement

A. Network model

There are N sensors or nodes that are uniform randomly placed to the network field with node density σ $(nodes/m^2)$. The network field is assumed to be rectangular. The length and width of the network field are A and W respectively. We assume that there is only one sink located on the left but outside the network field.

A sensor can operate either as a CH or CM. One of these two roles will be selected after finishing the CH election at the beginning of each operation round, called Data Collection Round or DCR. In each DCR, after clusters are formed, CMs then encapsulate data into packet and transmit to their CHs. Each CH aggregates all the packets from CMs within cluster to a single packet.

It is worth to mention that the size or radius of the cluster is very important factor. Normally the information observed in the environment from the sensors that are located near to each other has stronger spatial correlation than sensors that are located far away. Therefore, performing simple aggregation function such as averaging, minimum, maximum on data from CMs at CH can be affected if the size of cluster is too big than the expected or required cluster size by applications.

Given r_d the cluster size that application expected, also called deployed cluster size. And r_o is size of cluster that was created by cluster from the CH formation operation, also called operated cluster size. Larger cluster size r_o than expected r_d means that the aggregated packet performed by CH is summarized from packets received from members in larger area than application requirement. Therefore, the summarized information from larger area may not be as precise as in smaller area.

In this paper, we also assume that sink and all sensors are fixed in place after deployed. All sensors can estimate distance to other sensors and sink using received signal strength or other means. Each sensor is identified by a unique id. The operation phase is long enough and channel is perfect such that there is no packet loss or collision.

B. Problem statement

Our goal is to create a distributed algorithm that can efficiently control the average operated cluster size in each DCR and balance energy usages among sensor. Sensor that has lower residual energy compare to other sensors should decrease T-value to prevent sensor to enter CH competition and save its energy. On the other hand, a sensor that has energy higher than others should increase T-value to maintain cluster size in network. Therefore, T-value is a key to control amount of CCHs in CH competition, and should be adjustable by each node independently. So the problem is to find the best T-value for each sensor in the network such that the average operated cluster size (r_o) of all clusters is equal or near to the deployed cluster size (r_d) by the application.

IV. Fuzzy-Based Cluster Head Election Algorithm for Equal Cluster Size

The T-value is one of the important parameters for clustering election. It initializes number of sensors to act as CCHs in each round. Optimal T-value is different in different sensor density and deployed cluster size. Thus, T-value should be carefully chosen to guarantee the cluster heads quality and reduce message overhead [2]. For highly dense network, T-value should be decreased because there are too many potential CCHs in the network. Higher T-value than appropriate one will affect the performance of network because it will consume more energy during CH competition. The results from CH competition is likely to have more number of CHs than necessary and the operated cluster size will be smaller than the deployed cluster size. On the other hand, lower T-value will have the inverse effect.

In UCR and EC algorithms, all sensors use the same T-value and it was fixed and set to be 0.2 and 0.1 respectively. With our proposed FuzzCHE, the T-value is set in all sensors independently. In each round, the probability that a sensor will initialize itself as CCH depends on its own initial T-value. This initial T-value is set to be the same for all sensors before they are deployed in the network field. T-value of each sensor will slowly be changed each round so that the created clusters have the operated cluster size near to the deployed size.

In each DCR, sensors will participate in forming cluster, gathering and collecting data, aggregating, then send aggregated data back to the sink. In this paper, we have focused on cluster formation. Other parts in DCR will be described in short. Cluster formation also can be divided into 5 phases which are (1) CCH election phase, (2) CH competition phase, (3) Cluster discovery phase, (4) Cluster association phase and (5) CH confirmation phase. T-value adjustment is a function that is used in Cluster confirmation phase to adjust the current T-value to the new one for the next round. The pseudo code for T-value adjustment is shown in Algorithm 1.

Phase 1: In CCH election phase, sensors elected itself using its T-value. At the beginning of DCR, each sensor randoms a value from [0, 1] called $ProbValue$. If $ProbValue$ is less than or equal to its T-value, it promotes itself as CCH. It then calculates a delay value by inverse value of its residual energy (more residual energy means less delay), and then enters the next phase (CH competition phase). Other sensors that $ProbValue$ is more than its T-value take the CM role and skip CH competition phase by sleeping to save energy. They wake up again when Cluster discovery phase starts.

Phase 2: In CH competition phase, only elected CCHs in previous phase will enter this phase. When expiry of time delay as calculated in CCH election phase of a sensor occurred, that sensor sends CH advertisement packet to nearby CCHs within r_d distance. It then enters the CH role in the next phase. If CCH receives a CH advertisement packet before its delay time expired, it takes the CM role, and enters the sleep mode, and wake up again when Cluster discovery phase starts.

Phase 3: In Cluster discovery phase, CH advertises CH discovery packet with radius αr_d to announce itself to nearby CMs. CMs that listened CH discovery packets then select closest CH if there are more than one CHs nearby. The α value is a constant value used to amplify CH advertisement radius so that it is more than 99% confidence that CMs in network field will receive at least one CH advertisement packet [3]. Then both CH and CM enter Cluster association phase.

Phase 4: In Cluster association phase, each CM has to join or associate with CH that is closest to itself. CM sends a CH association packet which also contain the residual energy of that CM to selected CH. The CH then responses back with CH Ack packet. When CH receives all CH association packets, it calculates operated cluster size r_o by estimating distance between CH and its farthest CM in the cluster. CH also calculates average residual energy E_{avg} in this cluster from reported residual energy of its CMs and from itself. Then both CH and CM enters Cluster confirmation phase.

Phase 5: In Cluster Confirmation phase, each CH has to advertise allocated time slot to its CMs via CH Confirmation Timeslot packet. The operated cluster size (r_o) and average residual energy of all sensors in its cluster (E_{avg}) are also attached within this packet. The CH confirmation timeslot packet will be advertised with radius r_o distance. CMs will receive timeslot assignment for itself as well as r_o and E_{avg} for fuzzy calculation. CMs and CH use T-value adjustment function independently to adjust T-value for next round.

Figure 2 illustrates the degree of membership function of residual energy E_{res}. The membership function of residual energy is divided into 4 sections, Low, MidLow, MidHigh and High. We consider that if the residual energy in sensor (E_{res}) is lower or higher than 10% from the average residual energy (in the current cluster), then it fully belongs to Low and High section respectively. If sensor node has higher residual energy than others, then it is rather safer to be CCH or CH than others by increasing its T-value for the next round.

Figure 3 illustrates the degree of membership function of operated cluster size r_o. The membership function of operated cluster size is divided into 4 sections, Small, MidSmall, MidLarge and Large. We consider that if a sensor knows that the current operated cluster size (r_o) in a cluster is smaller or larger than 10% from the deployed cluster size value, then it fully belongs to Small and Large section respectively. If r_o is greater than r_d, then it is reasonable for this sensor to increase its T-value for the next round. The chance of having more CCHs and CHs will be higher, thus smaller the r_o for

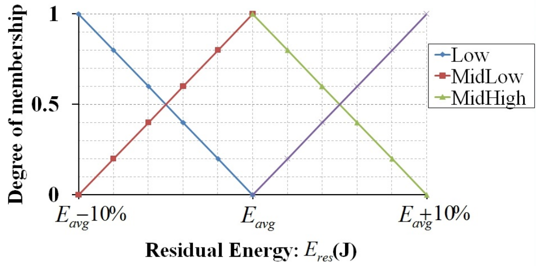


Fig. 2: Membership function of residual energy.

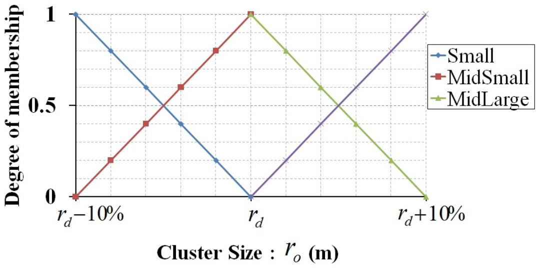


Fig. 3: Membership function of operated cluster size.

the next round. Please note again that the information about E_{avg} and r_o are obtained from CH confirmation packet

Since there are two fuzzy inputs (each with 4 fuzzy sections), we simply divide the rate of increasing or decreasing T-value for the next DCR into 16 T-value sections (TSs) as shown in Fig. 4. We also setup the fuzzy rules as shown in Table I. Rule number 1 can be described that it if E_{res} of a sensor is High (compare with E_{avg}) and its current r_o is Large (compare with r_d), then the rate of increasing or decreasing T-value for the DCR is in TS_0. TS_0 means that the rate to increase T-value should be maximum. On the other hand, TS_{15} means that the rate to decrease T-value should be maximum. In this paper, we limit the maximum change (increasing and decreasing) of T-value to δ%.

For defuzzification process, Center of Area (COA) is applied to create a crisp value. The fuzzy logic output value (variable F in Algorithm 1) is in the range from 0 to 1 We convert to adjust T-value from range [δ%, $-\delta$%]. For fuzzy

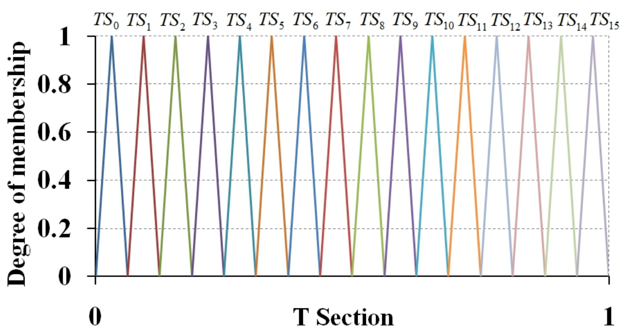


Fig. 4: Membership functions for T-value adjustment Sections.

output value 0, T-value calculate will be increased $\delta\%$ and 1 will be decreased by $\delta\%$ respectively. These process is repeated in every DCR.

TABLE I: Fuzzy rules for T-value adjustment.

Rule No.	Residual Energy E_{res}	Cluster Size r_o	T section
1	High	Large	TS_0
2	High	MidLarge	TS_1
3	MidHigh	Large	TS_2
4	MidHigh	MidLarge	TS_3
5	MidLow	Large	TS_4
6	MidLow	MidLarge	TS_5
7	Low	Large	TS_6
8	Low	MidLarge	TS_7
9	High	MidSmall	TS_8
10	MidHigh	MidSmall	TS_9
11	MidLow	MidSmall	TS_{10}
12	Low	MidSmall	TS_{11}
13	High	Small	TS_{12}
14	MidHigh	Small	TS_{13}
15	MidLow	Small	TS_{14}
16	Low	Small	TS_{15}

Algorithm 1 T-Value Adjustment: Sensor

```
Require: Sensor.E_res, Sensor.E_avg, Sensor.r_o, Sensor.r_d
F ← calculate fuzzy(Sensor.E_res, Sensor.E_avg, Sensor.r_o, Sensor.r_d)
if F <= 0.5 then
    Sensor.T += δ × (0.5 − F)/0.5        //increase T-value
else
    Sensor.T -= δ × (F − 0.5)/0.5        //decrease T-value
end if
if Sensor.T > 1 then
    Sensor.T = 1
end if
if Sensor.T < Sensor.T_limit then
    Sensor.T = Sensor.T_limit
end if
```

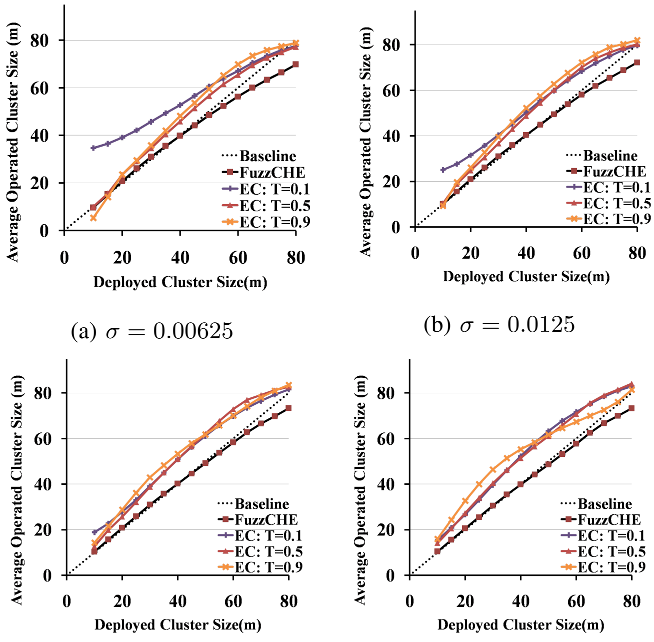
V. Simulation Results

The performance of FuzzCHE is evaluated by simulation. This simulation was implemented by C#. EC was shown to outperform other solutions such as UCR and HEED in term of SOP, energy balancing and throughput. The CH election algorithm of UCR can be considered as a special case of EC. Therefore, we mainly compare of the results of FuzzCHE with EC.

A. Simulation environments

We set up network field area (A x W) 71 $\times$ 100 m^2 that sensors are randomly deployed in the field. Base station or sink is placed in (-50, 50), outside the field. The results in each point as shown in Fig. 5 and 6 were averaged over 100 times (runs). For EC, T-value is fixed but we vary there T-values $\{0.1, 0.5, 0.9\}$ to find the best result for EC. For FuzzCHE, initial T-Value for FuzzCHE algorithm is 0.2 and be adapted in each DCR.

T_{limit} as shown in Algorithm 1 is chosen such that it is more than 99.9% confidence that it will be at least 1 CCH in each DCR. This value depends on sensor density value and network area. In this simulation, T_{limit} are $\{0.146, 0.075, 0.038, 0.02\}$ for sensor density $\{0.00625, 0.0125, 0.025, 0.05\}$ respectively. These values are obtained by monte-carlo simulation. In Algorithms 1, we define constant δ value to be 1% to determine the maximum and minimum changing of T-value can be increased or decreased in each DCR. In network simulation, we specify packet data size l_d and l_c. l_d is size of data packet that sensor gathered from field of interest, and the size of data aggregated packet processed by CH. l_c is control packet that is used for sensors communication. Any advertisement packet and other packet that are used during cluster formation, routing to base station will use the size of l_c. Other simulation parameters and values are described in Table II. We also use the simplified energy model which was used in many work such as in [2][4][11][12][13][14].



(a) $\sigma = 0.00625$ (b) $\sigma = 0.0125$ (c) $\sigma = 0.025$ (d) $\sigma = 0.05$

Fig. 5: Average operated cluster size vs. deploy cluster size at different node densitys.

TABLE II: Network setup parameters.

Parameter	Value
Simulation times	100 times
Sensor Density (σ)	$\{0.00625, 0.0125, 0.025, 0.05\}$ $nodes/m^2$
Initial Energy	3 J
Deployed Cluster Size	[10, 80] meters, step by 5 meters
δ	0.01
α	$\sqrt{2\ln 10}$
l_d	4000 $bits$
l_c	200 $bits$

B. Average operated cluster size

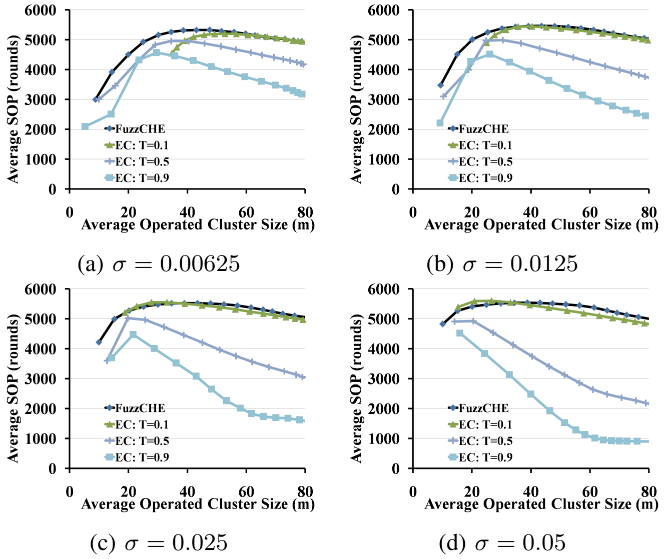
We compare the average operated cluster size with the deployed cluster size in Fig. 5. Baseline as shown in Fig. 5 is the ideal straight line where deployed cluster size equal to operated cluster size, which is 45 degree from origin point.

Any lines or points that are plotted above this baseline means average operated cluster size is larger than deployed cluster size. Lines or points lower this baseline has the opposite meaning.

The results clearly shows that FuzzCHE outperform EC. FuzzCHE can create and maintain operated cluster size very close to deployed cluster size, as its line is very near to the Baseline in all possible network density (Fig.5(a)-(d)). Only for larger deployed cluster size, FuzzCHE operated slightly smaller cluster size than deployed one. This is because the diagonal line of the network field is only 122 meters. We need only 1 CH located at the center of the network field with 61 meters or more deployed cluster size. Therefore, to maintain operated cluster size more than 60 meters is rather difficult because it needs only 1 or 2 CHs in the network field, where the operated cluster size is highly depended on the CH locations.

C. Average stable operation period

Stable Operation Period (SOP) is defined as the number of rounds before the first sensor depleted all its energy [3]. High SOP means network can stand stable for a long run. Fig. 6 compares the SOP obtained by FuzzCHE and EC solution at different average operated cluster size which is the result from Fig.5. It is again clear that FuzzCHE gives good SOP results for most of the scenarios. Though EC with T-value equal to 0.1 has similar SOP result in many scenarios, it has several limitations. For example, at network density equals to 0.0625, the minimum operated cluster size that EC can achieve is only 34 meters, where it is 10 meters for FuzzCHE. To make operated cluster size smaller, EC has two options, either to increase its T-value which the cost of lower SOP, or increase network density which increase the cost of sensors. By adjusting T-value appropriately at each sensor by FuzzCHE, each sensor can save and balance energy with other sensors. Therefore, FuzzCHE can both control the average operated cluster size, and extend the network lifetime.



(a) $\sigma = 0.00625$ (b) $\sigma = 0.0125$

(c) $\sigma = 0.025$ (d) $\sigma = 0.05$

Fig. 6: Average SOP vs. average operated cluster size at different node densitys.

VI. Conclusion

In this paper, we proposed fuzzy logic based cluster head election algorithm, called FuzzCHE, to control cluster size in the network. It is simple, distributed and has low overhead. With FuzzCHE, each sensor can adjust T-value which is the probability to become CCH by itself after receiving two fuzzy inputs which are current operated cluster size and average residual energy in the cluster from CH during cluster formation. This information is just only 2 parameters which can be easily attached during cluster formation, therefore it has low overhead. Results from simulation show that FuzzCHE outperforms EC in terms of the closeness between deployed cluster size and the operated cluster size, and longer network lifetime.

References

[1] S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Conputer and Communications. IEEE Societies (Volume:3).

[2] G. Chen, C. Li, M. Ye, J. Wu, "An unequal cluster-based routing stategy in wireless sensor networks", Wireless Networks, 2009, 15, 193-207.

[3] D. Wei, Y. Jin, S. Vural, K. Moessnet, R. Tafazolli, "An Energy-Efficient Clustering Solution for Wireless Sensor network", Wireless Communications, IEEE Transactions on, Vol.10, No.11, pp. 3973-3983, November 2011.

[4] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660-670, October 2002.

[5] C. Nam, Y. Ku, J. Yoon, D. Shin, "Cluster Head Selection for Equal Cluster Size in Wireless Sensor Networks" , New Trends in Information and Service Science, 2009. NISS '09. International Conference on, pp 618 - 623, June 30 2009-July 2 2009, Beijing.

[6] R. N. Enam, R. Qureshi, S. Misbahuddin, "A Uniform Clustering Mechanism for Wireless Sensor Networks" , International Journal of Distributed Sensor Networks, vol. 2014, Article ID 924012, 14 pages, 2014. doi:10.1155/2014/924012.

[7] D. Jia, H. Zhu, S. Zou, P. Hu, "Dynamic Cluster Head Selection Method for Wireless Sensor Network" , IEEE Sensors Journal (Volume:16 , Issue: 8), pp 2746-2754, 24 Dec 2015.

[8] H. Lin, L. Wang, R. Kong, "Energy Efficient Clustering Protocol for Large-Scale Sensor Networks" , IEEE Sensors Journal (Volume:15 , Issue: 12), pp 7150-7160, 21 Aug 2015.

[9] M. Ye, C. Li, G. Chen, J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks", Performance, Computing, and Communications Conference, 2005, IPCCC 2005, 24th IEEE International, pp. 535-540, 7-9 April 2005.

[10] J. Kim, S. Park, Y. Han, T. Chung, "CHEF: Cluster Head Election mechanism using Fuzzy logic in Wireless Sensor Networks" , ICACT 2008, pp 654-659, 17-20 Feb 2008, Gangwon-Do.

[11] J. Lee, W. Cheng, "Fuzzy-Logic-Based Clustering Approach for Wireless Sensor Networks Using Energy Predication" , IEEE Sensors Journal (Volume:12 , Issue: 9), pp 2891 - 2897, 13 June 2012.

[12] H. Taheria, P. Neamatollahia, O. Mohamed Younisb, S. Naghibzadehc, M. H. Yaghmaeed, "An energy-aware distributed clustering protocol in wireless sensor networks using fuzzy logic" Ad Hoc Networks, Volume 10, Issue 7, September 2012, pp 1469-1481.

[13] H. Bagci, A. Yazici, "An energy aware fuzzy approach to unequal clustering in wireless sensor networks" , Applied Soft Computing, Volume 13, Issue 4, April 2013, pp 17411749.

[14] P. Eak-Une, C. Pornnavalai, "Unequal Initial Energy Assignment in Wireless Sensor Networks", Computer Science and Software Engineering (JCSSE), 2015 12th International Joint Conference on, pp 212-217.