



A Minimum Spanning Tree based Clustering Algorithm for Cloud based Large Scale Sensor Networks

Zuleyha Akusta Dagdeviren

^{1*} Ege University, International Computer Institute, Izmir, Turkey (ORCID: 0000-0001-9365-326X), zuleyhaakusta@gmail.com

(3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications June 11-13, 2021)

(DOI: 10.31590/ejosat.960421)

ATIF/REFERENCE: Akusta Dagdeviren, Z. (2021). A Minimum Spanning Tree based Clustering Algorithm for Cloud based Large Scale Sensor Networks. *European Journal of Science and Technology*, (26), 415-420.

Abstract

Wireless sensor networks (WSNs) can be composed of huge numbers of nodes collecting data from the environment. WSNs are crucial communication layer technologies of Internet of Things. The obtained data by the WSNs can grow exponentially, hence utilizing big data analysis techniques and cloud computing technologies are of utmost importance. WSNs can be used in various applications such as habitat monitoring, military surveillance, smart agriculture, miner safety and healthcare applications. Sensor nodes are generally battery-powered, so conserving the residual energy of nodes is very important to prolong the lifetime of the applications. WSNs do not own a fixed infrastructure, hence messages of the applications transmitted in an ad hoc manner to the sink node. Since the transmission range of sensor nodes are limited, multi-hop communication is used. Clustering is a very important method for supporting multi-hop routing in WSNs. Data aggregation, time synchronization and load balancing are some of the well-known operations that benefit from clustering. Selecting efficient communication paths and distribution of nodes evenly to partitions in clustering operation lead to boost the network lifetime. In this paper, we propose a minimum spanning tree based clustering and backbone formation algorithm (MICUB) for WSNs. The proposed algorithm inputs node coordinates, transmission range, sensing area dimensions and partition numbers and outputs clustering and backbone information. MICUB algorithm first forms a minimum spanning tree backbone and divides the networking area into equal partitions where each partition is a cluster. In this manner, efficient links are selected for backbone formation and the clusters are constructed evenly. The intra-cluster links are constructed by again executing a minimum spanning tree algorithm inside the clusters. We measure the coefficient of variations of the proposed MICUB algorithm and its counterparts to obtain the clustering quality. These results show us that our proposed algorithm performs very well against node counts and degrees.

Keywords: Wireless Sensor Networks, Clustering, Minimum Spanning Tree, Internet of Things, Cloud Computing, Big Data.

Bulut Tabanlı Büyük Ölçekli Sensör Ağları için En Küçük Kapsayan Ağaç Tabanlı Kümeleme Algoritması

Öz

Kablosuz sensör ağları (KSA'ları), ortamdan veri toplayan çok sayıda düğümden oluşabilir. KSA'lar, Nesnelerin İnterneti'nin önemli iletişim katmanı teknolojilerindedir. KSA'lar tarafından elde edilen veriler katlanarak büyüyebilir, bu sebepten büyük veri analiz teknikleri ve bulut bilişim teknolojilerini kullanmak son derece önemlidir. KSA'lar, habitat izleme, askeri gözetim, akıllı tarım, madenci güvenliği ve sağlık uygulamaları gibi çeşitli uygulamalarda kullanılabilir. Sensör düğümleri genellikle pilden güç alır, bu nedenle düğümlerin var olan enerjisini korumak, uygulamaların ömrünü uzatmak için çok önemlidir. KSA'lar sabit bir altyapıya sahip değildir, bu nedenle uygulama mesajları tasarsız bir şekilde çıkış (sink) düğüme iletilir. Algılayıcı düğümlerin iletim aralığı sınırlı olduğu için çok zıplamalı (multi-hop) iletişim kullanılır. Kümeleme, KSA'larda çok zıplamalı yönlendirmeyi desteklemek için çok

* Sorumlu Yazar: zuleyhaakusta@gmail.com

önemli bir yöntemdir. Veri toplama, zaman senkronizasyonu ve yük dengeleme, kümelemeden yararlanılan iyi bilinen işlemlerden bazılarıdır. Kümeleme işleminde verimli iletişim yollarının seçilmesi ve düğümlerin bölümlere eşit olarak dağıtılması, ağ ömrünün artmasına neden olur. Bu makalede, KSA'lar için en küçük kapsayan ağaç tabanlı kümeleme ve omurga oluşturma algoritması (MICUB) öneriyoruz. Önerilen MICUB algoritması, düğüm koordinatlarını, iletim aralığını, algılama alanı boyutlarını ve bölüm numaralarını girer, kümeleme ve omurga bilgilerini çıkarır. MICUB algoritması ilk olarak en küçük kapsayan ağaç omurgasını oluşturur ve ağ alanını her bölümün bir küme olduğu eşit parçalara böler. Bu şekilde, omurga oluşumu için verimli bağlantılar seçilir ve kümeler dengeli bir şekilde oluşturulur. Küme içi bağlantılar, yine kümeler içinde bir en küçük kapsayan ağaç algoritması yürütülerek oluşturulur. Kümeleme kalitesini elde etmek için önerilen MICUB algoritmasının ve benzerlerinin varyasyon katsayısını ölçülmektedir. Bu sonuçlar, önerilen algoritmamızın düğüm sayılarına ve derecelerine karşı çok iyi performans gösterdiğini göstermektedir.

Anahtar Kelimeler Kablosuz Sensör Ağlar, Kümeleme, En Küçük Kapsayan Ağaç, Nesnelerin İnterneti, Bulut Bilişim, Büyük Veri.

1. Introduction

Wireless sensor networks (WSNs) are large scale networks and crucial technologies for Internet of Things (IoT) in terms of gathering big data from the environment (Harb et al., 2017) (Kim et al., 2019). Thanks to evolving cloud computing and WSN technologies, huge amount of IoT data can be stored and processed to obtain business intelligence. In WSNs, generally, ordinary nodes are battery powered, so energy conservation is of paramount importance. Since WSNs do not own a communication infrastructure, the messages are transmitted in an ad hoc way. A sink node is associated for the data collector and gateway roles. Hence, the collected data from the ordinary nodes is relayed to the cloud through the sink node. Generally, the transmission range of ordinary sensor nodes are bounded, so engineering a multi-hop routing protocol is a mandatory method for relaying the sensed data to the sink node for most of the WSN setups.

Clustering is a fundamental method to provide multi-hop routing operation in WSNs. Clustering provides an infrastructure for data aggregation, load balancing, security operations, time synchronization, etc. (Wang et al., 2018) (Palaniswami et al., 2020) (Tripathi et al., 2021). By selecting efficient communication paths for data transmission, clustering aims to prolong the WSN lifetime. Besides, one important objective of clustering is to decrease the number of delivered messages needed by the upper layers (Liu et al., 2020) (Vaiyapuri et al., 2021). Moreover, considering hardware/software problems, environmental effects and security attacks, periodic clustering in WSNs provides fault tolerance. Nodes are categorized as cluster members and cluster heads in this method where cluster heads and cluster members can be considered as servers and clients, respectively.

Graphs are significant structures for modelling WSNs and spanning tree based clustering is a fundamental graph-theoretic technique in WSNs. An undirected graph can be represented with $G(V, E)$ where V and E are set of vertices (nodes) and edges (links), respectively. A graph $G_S(V_S, E_S)$ is a spanning subgraph of $G(V, E)$ if $V_S = V$. A spanning tree (ST) of a graph is an undirected connected acyclic spanning subgraph. Intuitively, a minimum spanning tree (MST) for a graph is a subgraph that has the minimum edge weight for maintaining connectivity [30]. In Fig. 1, various graph-theoretic structures are given where node ids (given as letters) are written near to each node, physical communication links are depicted with dashed lines and selected links are drawn with the solid lines. Weights of edges are given near to each edges. Green labeled nodes are ordinary sensor nodes and the red labeled nodes are the sinks. In Fig. 1.a, selected edges do not constitute a tree since a cycle exists between nodes B, C and D. The chosen edges in Fig 1.b form a

tree, but since nodes H and F are not included, the tree is not spanning. An example spanning tree is given in Fig 1.c whereas the total weight of selected links is not minimum. Finally, an

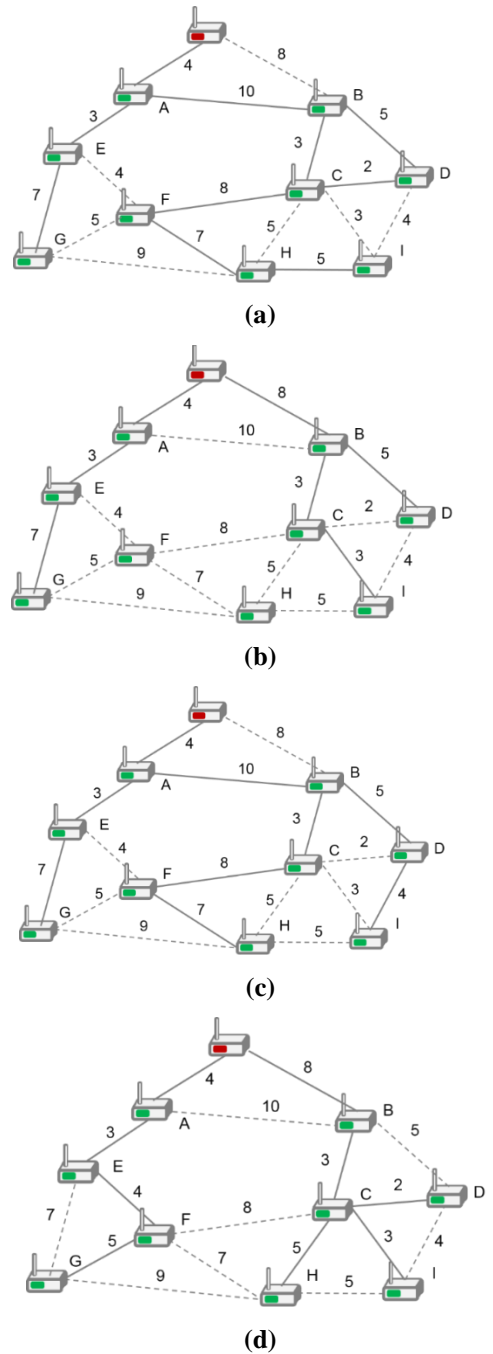


Figure 1. a) Non-Tree Example b) Non-Spanning Tree Example c) Non-Minimum Spanning Tree Example d) Minimum Spanning Tree Example

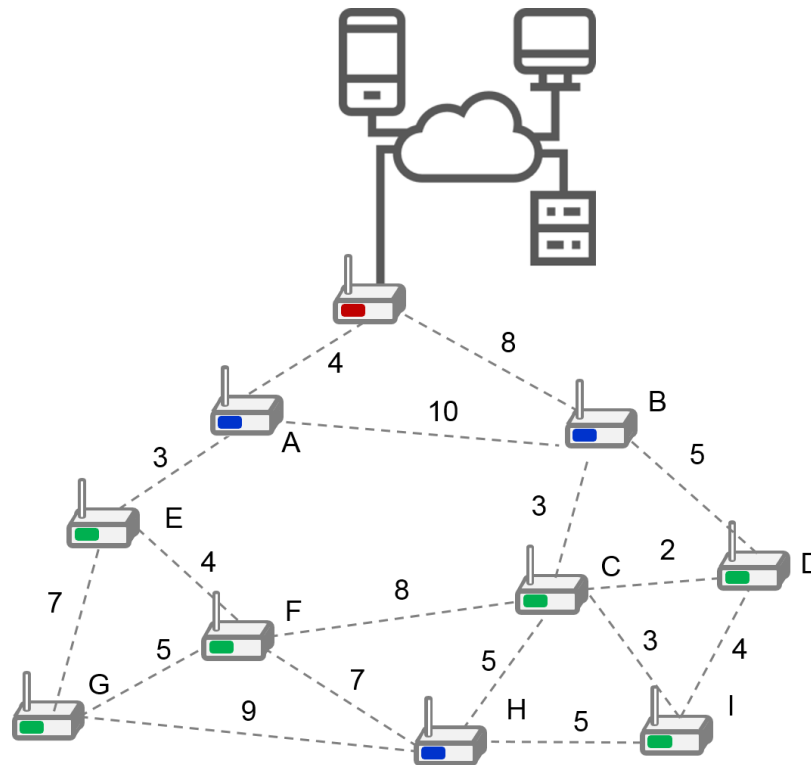


Figure 1. The Network Model

example minimum spanning tree structure is given in Fig. 1.d where the total weight of minimum spanning tree is 37. Various distributed algorithms are proposed to construct minimum spanning tree for computer networks (Gallagher et al., 1983), (Awerbuch, 1987), (Lien,1988), (Ahuja and Zhu, 1989), (Banerjee and Khuller, 2000), (Dai and Wu, 2004), (Dagdeviren and Erciyes, 2006), (Nanuvala, 2006). Although these distributed algorithms construct spanning tree backbone, they may lack balanced clusters.

In this paper, we propose a minimum spanning tree based clustering and backbone formation algorithm (MICUB) for large scale sensor networks. Our algorithm takes node coordinates, transmission range, partition numbers and sensing area dimensions as inputs and produce clustering and backbone information. The algorithm first constructs an MST for backbone formation, then divides the sensing area into equal partitions. Each partition corresponds a cluster, thus if the nodes are uniformly and randomly distributed then the expected sizes of the clusters are equal. The intra-cluster connections of the clusters are constructed again by executing an MST algorithm inside each cluster. We provide the simulations of the algorithms and show that the proposed algorithm outperforms its competitors.

2. Material and Method

2.1. Network Model

Fig. 2 displays our network model in general. As aforementioned, green labeled nodes are ordinary nodes, red labeled node is the sink node and the blue labeled nodes are the cluster head nodes. In our network model, sensor nodes know their positions either by using a GPS receiver or by executing a localization approach. Weights given in Fig. 2 are distances between nodes. These distances can be calculated from the

positions are adjacent nodes. If nodes are mobile, the positions of the nodes should be updated according to the mobility levels of nodes. The position data of the nodes is transferred to the cloud system through the sink node. Since WSNs can be very large scale, this position data is stored and processed by a big data analyzer residing in the cloud system. End users are connected to the cloud system to retrieve meaningful data.

2.2. Proposed Algorithm

We propose an algorithm for MST construction between cluster heads rooted at the sink, as well as formation of clusters with evenly distributed nodes. The steps of our MICUB algorithm is shown in Alg. 1. Firstly, a weighted graph is constructed from the positions of the nodes. Then an MST backbone of all nodes is constructed by calling a central MST algorithm, namely MST_Alg. Following that, the area is divided into $X \times Y$ grids, in which X and Y are predefined by the user. Each grid is a cluster, thus $X \times Y$ gives the total number of clusters. If the nodes are uniformly distributed across the sensing area, the number of expected nodes in each cluster is $N / (X \times Y)$ where N is the total number of nodes. After dividing into grids for cluster formation, nodes are connected through a MST for intra-cluster communication. The cluster head is the node with the maximum number of MST neighbors. We use Kruskal's algorithm (Kruskal, 1956) as MST_Alg() for construction of the MST.

Example clustering operations of MICUB are given in Fig. 2 and 3. Inter-cluster and intra-cluster MST links are same for both figures. There are 4 clusters where $X=2$ and $Y=2$ in Fig. 2. Nodes K , E and D are cluster heads of clusters 1, 3 and 4, respectively. Sink node is the cluster head of cluster 2. Clusters 1, 2 and 4 consist of 4 nodes whereas cluster 3 includes 5 nodes. The network in Fig. 3 is partitioned into 6 clusters, nodes K , C , G , E and L are the cluster head of the clusters 1, 2, 4, 5 and 6. The sink node is the cluster head of cluster 2 same with the previous figure. Each cluster has 3 nodes in this partitioning scheme.

Algorithm 1. MICUB Algorithm

- 1: input: coordinates of nodes, partition numbers (X, Y), transmission range and sensing area dimensions.
- 2: construct weighted graph $G_w=(V, E)$ from nodes' coordinates.
- 3: $T_w = \text{MST_Alg}(G_w)$
- 4: divide G_w into $X \times Y$ grids.
- 5: for each grid G_i do
- 6: call $T_i = \text{MST_Alg}(G_i)$
- 7: ch_i is the cluster head of T_i . The node with the greatest degree in T_i is the cluster head. If there are more than one candidate, choose the node with the maximum energy.
- 8: orient the nodes of T_i to ch_i .
- 9: end for
- 10: output: T_w is the backbone, T_i s are the intra-cluster trees.

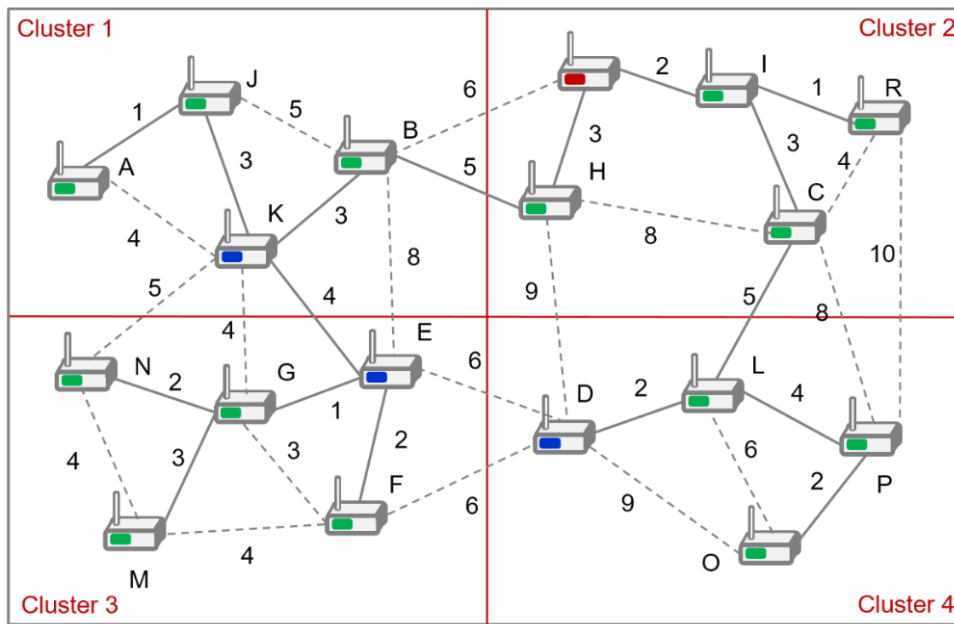


Figure 2. An Example WSN Clustered with MICUB (Cluster Count=4)

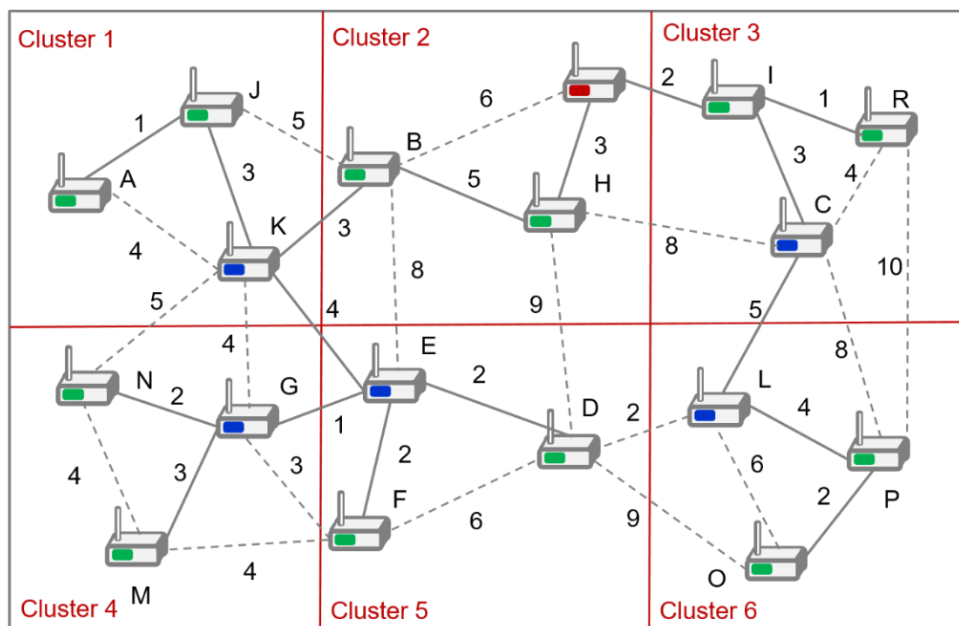


Figure 3. An Example WSN Clustered with MICUB (Cluster Count=6)

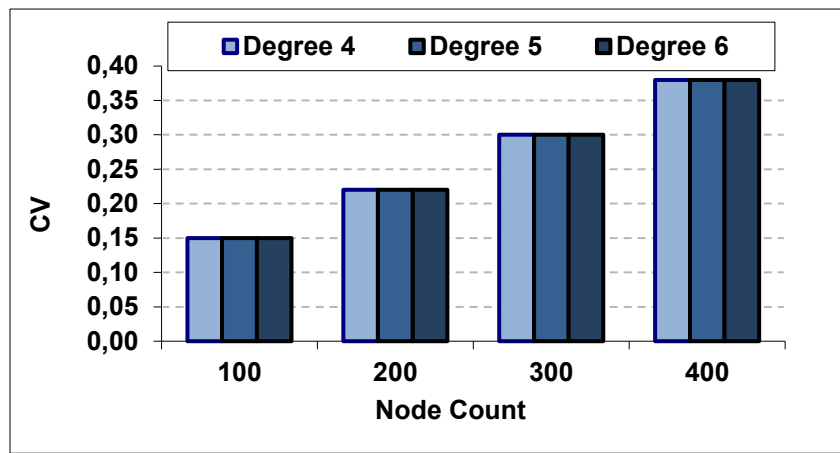


Figure 4. CV Values of MICUB against Node Count and Degree

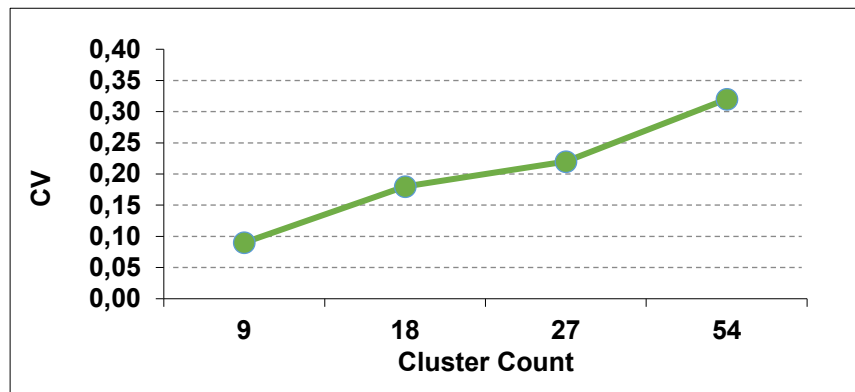


Figure 5. CV Values of MICUB against Cluster Count

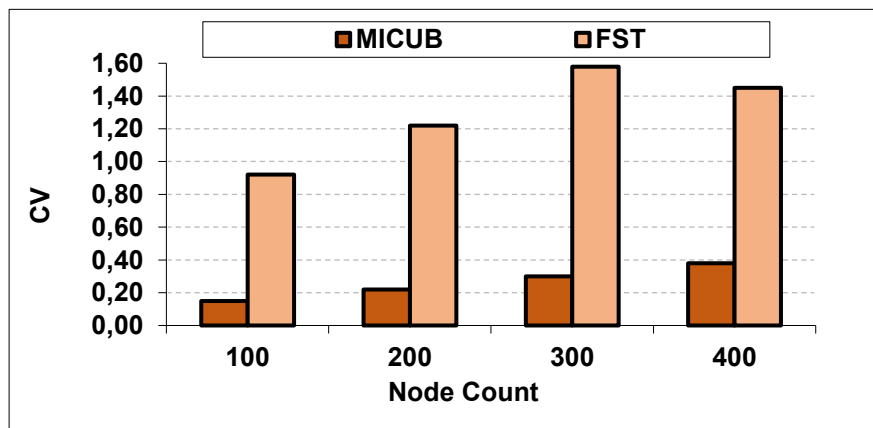


Figure 6. CV Values of MICUB and FST against Node Count

3. Results and Discussion

We simulate the proposed MICUB approach by varying node counts (from 100 to 400 nodes) and average degrees (from 4 to 6). We implement a flooding based distributed spanning tree algorithm (FST) to compare with the proposed algorithm. The sink node starts the FST algorithm by sending a probe message to its neighbors. When a node i receives a probe message in the first time, i sets its parent as the source of the probe message and it sends probe message to its neighbors. The cluster sizes of this algorithm can be adjusted by a *depth* parameter where *depth* is used to control the maximum cluster depth.

For ST algorithms, we use *coefficient of variation* (CV) to measure the clustering balance. The *coefficient of variation* is computed as *standard deviation / mean*. If $CV < 1$ then the distribution is considered low variance, else it is high variance. The clusters are more balanced for lower variance values. The balance of the clusters produced by MICUB is stable against the varying degrees as shown in Fig. 4. When the number of clusters of MICUB is increased, the CV values are increased as shown in Fig. 5. This means that MICUB divides the area more efficiently when the number of clusters is small. The comparison of the CV values of the clusters produced by MICUB and FST is shown in Fig. 6. MICUB produces more balanced clusters than FST and

CV values of MICUB increases with a smaller slope than FST as the number of nodes increases.

4. Conclusions and Recommendations

The size of the network and data collected by WSNs may rise exponentially. Thus, utilizing cloud computing and big data analysis techniques is of utmost importance to tackle with the emerging problems in IoT. Clustering and backbone formation are crucial operations to maintain an energy-efficient multi-hop communication structure for WSNs and to increase the efficiency of other services such as load balancing, time synchronization and load balancing. In this paper, we propose the MICUB algorithm that is based on minimum spanning tree construction, for WSNs.

The proposed algorithm MICUB inputs transmission range, node positions, sensing area dimensions and partition numbers and produce cluster and backbone information for the sensor nodes. The algorithm first constructs a minimum spanning tree backbone for inter-cluster communication. Then, the algorithm partitions the network into clusters, assigns a cluster head for each cluster and executes the minimum spanning tree algorithm to construct intra-cluster links. The coefficient of variation values of the proposed algorithm with its counterparts are measured to gain clustering quality. From the simulation results, we find out that MICUB outperforms its competitors against node counts and degrees.

References

- Ahuja, M. and Zhu, Y. (1989) A distributed algorithm for minimum weight spanning trees based on echo algorithms. *Proc. of the 9th Int. Conf. on Distributed Computing Systems*, 5-9 June, pp. 2-8.
- Awerbuch, B. (1987) Optimal distributed algorithms for minimum weight spanning tree, counting, leader election and related problems. *Proc. of the 19th Annual ACM Symp. on Theory of Computing*, New York, United States, pp. 230-240. ACM Press, New York.
- Banerjee, S. and Khuller, S. (2000) A clustering scheme for hierarchical routing in wireless networks. *Technical Report CS-TR-4103*. UMD, College Park.
- Chatterjee, M., Das, S. K., and Turgut, D. (2001) WCA: A weighted clustering algorithm for mobile ad hoc networks. *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 5, 193-204.
- Dagdeviren, O. and Erciyes, K. (2006) A distributed backbone formation algorithm for mobile ad hoc networks. *Proc. of the 4th Int. Symp. on Parallel and Distributed Processing and Applications*, Sorrento, Italy, 4-6 December, pp. 219-230. Springer-Verlag, Berlin.
- Dai, F. and Wu, J. (2004) An extended localized algorithm for connected dominating set formation in ad hoc wireless networks. *IEEE Trans. on Parallel and Dist. Systems*, 15(10), 908-920.
- Gallagher, R. G., Humblet, P. A., and Spira, P. M. (1983) A distributed algorithm for minimum-weight spanning trees. *ACM Trans. on Prog. Languages and Systems*, pp. 6677. ACM Press, New York.
- Harb, H., Makhoul, A., Idrees, A., Zahwe and O. and Taam, M. (2017) Wireless Sensor Networks: A Big Data Source in Internet of Things. *International Journal of Sensors, Wireless Communications and Control*.
- Kim, B.-S, Kim, K.-I., Shah, B., Chow, F. and Kim, K. H. (2019) Wireless Sensor Networks for Big Data Systems, *Sensors* 19, no. 7, 1565.
- Kruskal, J. B. (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. of the American Mathematical Society*, 7(1), 48-50.
- Lien, Y. N. (1988) A new node-join-tree distributed algorithm for minimum weight spanning trees. *Proc. of the 8th Int. Conf. on Distributed Computing System*, pp. 334-340. IEEE.
- Liu, X., Zhu, R., Anjum, A., Wang, J., Zhang, H. and Ma, M. (2020) Intelligent data fusion algorithm based on hybrid delay-aware adaptive clustering in wireless sensor networks, *Future Generation Computer Systems*, vol.104, pp. 1-14.
- Nanuvala, N. (2006) An enhanced algorithm to find dominating set nodes in ad hoc wireless networks. *MSc. Thesis*, Georgia State University.
- Palaniswami, M., Rao, A. S., Kumar, D., Rathore, P. and Rajasegarar, S., (2020) The Role of Visual Assessment of Clusters for Big Data Analysis: From Real-World Internet of Things, *IEEE Systems, Man, and Cybernetics Magazine*, vol. 6, no. 4, pp. 45-53.
- Tripathi, A. K., Sharma, K., Bala, M., Kumar, A., Menon, V. G. and Bashir, A. K. (2021) A Parallel Military-Dog-Based Algorithm for Clustering Big Data in Cognitive Industrial Internet of Things, *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2134-2142.
- Wang, Q., Guo, S., Hu, J. and Yang, Y., (2018) Spectral partitioning and fuzzy C-means based clustering algorithm for big data wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 54.
- Vaiyapuri, T., Parvathy, V.S., Manikandan, Krishnaraj, V. N., Gupta, D. and Shankar, K. (2021) A Novel Hybrid Optimization for Cluster-Based Routing Protocol in Information-Centric Wireless Sensor Networks for IoT Based Mobile Edge Computing. *Wireless Personal Communications*, <https://doi.org/10.1007/s11277-021-08088-w>.