International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Optimizing Virtual Backbone in Wireless Sensor Networks: A Novel Approach to Minimal Connected Dominating Set Construction

[1]Manabendranath Kar, [2]Prof Dr Indrajit De, [3]Pooja, [4]Dr. Sandeep Kumar, [5]Prof. (Dr.) Ambuj Kumar Agarwal

Submitted: 09/10/2023 **Revised**: 29/11/2023 **Accepted**: 10/12/2023

Abstract: This paper introduces an innovative algorithm for constructing a Minimal Connected Dominating Set (MCDS) in wireless sensor networks. The MCDS is a crucial component for efficient broadcasting and activity scheduling in these networks, serving as a virtual backbone. Our algorithm uniquely addresses this challenge by optimizing the process of virtual backbone formation. The methodology involves a multi-phase approach, beginning with the initialization of the network nodes, followed by a selection phase where dominator nodes are identified based on a novel weighting criterion. Subsequently, in the connection phase, connector nodes are chosen to ensure network connectivity. The algorithm also includes an optional maintenance phase for adapting to dynamic network changes. Our results, obtained through comprehensive simulations, demonstrate the algorithm's effectiveness in reducing the size of the MCDS and improving the time efficiency compared to existing methods. This research contributes significantly to the field of wireless sensor networks by providing a more efficient mechanism for constructing a virtual backbone.

Index Terms: *Wireless Sensor Network, Unit Disk Graph, Neighborhood of a Vertex, Maximal Independent Set, Minimum Connected Dominating Set.*

I. Introduction

Wireless sensor networks consist of a number of wireless nodes or sensors. They are easy to deploy to an application field and the cost are relatively low by the continuing improvements in embedded sensor, VLSI, and wireless radio technologies. These benefits make wireless networks useful for a wide range of applications in many areas, such as battlefield surveillance, disaster rescue, environmental monitoring, health applications, conferences and traffic control. A wireless sensor network is composed of a set of battery powered sensors. These sensors can communicate with one another through wireless links if they are within their transmission range, otherwise they can communicate via other sensors between them. However, the shortcomings of the sensors also limit network performance. Wireless nodes have limited energy, which greatly affect network lifetime. Activity scheduling and broadcasting are the two main problems in a sensor network. In activity scheduling some of the nodes let their radio antenna off to save energy and in broadcasting the host needs to send message to all other nodes in the network. But both are highly energy consuming. The minimal amount of energy must be used for this energy-intensive operation in order to conserve the nodes' limited resources. We must build a virtual backbone for our wireless network in order to address these issues [2][6][7][8].A virtual backbone in a Wireless Sensor Network (WSN) reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption and thus increases network operational life. Very limited number of sensors are involved in constructing the virtual backbone that can minimize the number of hops required to reach the sink, assuming that all nodes have equal transmission range. In the wireless domain, this backbone is a minimum connected dominating set (MCDS). It creates a G since every element is related to every other element. The nodes in C are referred to as dominators, while the nodes that are one hop away from C are referred to as dominatees. The backbone is selected to have the minimum CDS in order to reduce the number of hops. MCDS [5][6][7][8] is a CDS that has the least cardinality of all the CDSs in a graph, and it is adjacent to every node. After reaching a dominator, data is sent to the sink via the MCDS in order to ensure least hop communication. Unit Disk Graph requires the CDS to be determined since the nodes have the same transmission range (UDG).Here an algorithm has been adapted from [6] to construct MCDS for achieving power efficient activity scheduling and broadcasting in a WSN. This algorithm involves three phases, constructing Dominating Set or Maximal Independent Set in phase-I, determining CDS in phase-II, forming MCDS in phase-III.

Assistant Professor, Dream institute of Technology, Kolkata[1]
Professor, IEM-UEM Kolkata[2]
Professor, Faculty of engineering and technology, Sharda University Uzbekistan[3]
Professor Department of Computer Science and Engineering, Sharda School of Engineering And Technology, Sharda University, Greater Noida[4,5]
Corresponding Author Email id ambuj4u@gmail.com

This is how the remainder of the paper is structured. In part II, the works relevant to the current investigation are discussed. The suggested method is given in section III. Section IV presents the findings of the simulation. In part V, the paper comes to an end.

II.Related Work

The building of a Minimal Connected Dominating Set (MCDS) in an undirected and a unit disc graph is acknowledged as an NP-complete issue in the context of wireless sensor networks. NP-complete problems are a class of decision problems for which there is no known effective solution strategy. Previous papers [1][2] brought attention to this difficulty and laid the groundwork for further research initiatives.

A pioneering step was taken by Alzoubi et al. [3], who developed the first distributed algorithm that guaranteed a constant approximation factor for Connected Dominating Set (CDS) construction. This algorithm was built upon the concept of a Maximum Independent Set (MIS) in unit disk graphs (UD Graphs). Following this, another distributed algorithm was introduced [4], which also employed MIS for CDS construction in UD Graphs. This method's substantial reliance on lengthy message exchanges and transmissions—which are required to build a spanning tree as a prelude to the CDS—was a major problem.

The inherent complexity of distributed algorithms in this context was further underscored by the revelation that is the lower bound on message complexity for such algorithms. Despite this, strides were made in [5] to develop a distributed algorithm aimed at constructing smaller-sized CDS for UD Graphs. This method utilized a Steiner tree to connect MIS nodes. To get to an MCDS, iteratively evaluating a CDS and then recursively eliminating nodes was required. But the details of the CDS's original construction were still unclear.

An innovative approach was proposed in [6] to find MCDS using dominating sets in UD Graphs. This algorithm was implemented in three distinct phases: initially identifying dominating sets, followed by the identification and connection of connectors through a Steiner tree, and finally pruning the CDS obtained in the second phase to derive an MCDS. Additionally, this algorithm incorporated a mechanism to adapt to continuous topological changes caused by node deactivation, typically due to battery exhaustion. This adaptation was facilitated by a local repair algorithm that reconstructs the MCDS, creating a power-aware solution using only neighborhood information.

More advancements were achieved in [7], when a method based on the progressive construction of a dominant set was suggested. The pre-existence of a workable CDS of the underlying network graph was assumed by this technique. Recursively eliminating vertices from this collection, it made sure that every deletion prevented the network from disconnecting. Until every node in the current set was either eliminated or verified as a member of the final dominant set, the process was repeated. Once more, pruning was used to convert a CDS into an MCDS.

In a different vein, an algorithm was introduced in [8] for finding MCDS for UD Graphs based on the computation of the convex hull of sensor nodes. The complexity of this algorithm was marking a significant computational consideration in the broader context of MCDS construction in wireless sensor networks.

These various approaches, while diverse in their methodologies, collectively contribute to the evolving landscape of MCDS construction in wireless sensor networks, each addressing unique aspects of the challenge and paving the way for future innovations in the field.

III. Proposed Algorithm

In this work an algorithm is proposed for MCDS construction by using dominating set. The proposed algorithm follows the algorithm presented in [6] for MCDS construction where a small modification in algorithm improve upon significantly the number of nodes present in the MCDS. The algorithm is originally presented in 3 phases. In phase-I, maximal independent set (MIS) or dominating set is constructed. Nodes in this set called dominators. In phase-II, set of connectors are found among the non-MIS nodes which are connected to maximum number of MIS nodes to get connected dominating set (CDS). In order to create an ideal, minimum MCDS, certain nodes in the CDS are decreased during the last step of pruning. Every node is labelled as a black, white, or brown node throughout the MCDS determination process. A black node indicates a dominator that must be present in the CDS, a white node indicates a dominatee, and a brown node indicates a connection that must be present in the CDS. The algorithm is described phase by phase in the following sections.

A. Phase I. Dominating Set Construction

This phase of the algorithm performs in the following steps.

1. Every node in the graph G is allocated a random integer, say id (V, E).

2. Select a node arbitrarily from vertex set V. It forms the first Independent Set.

3. Delete it’s adjacent node(s).

4. Do steps 2-3 until you get a Maximal Independent Set which is a Dominating Set also.

5. Color all the nodes in the MIS as black.

7. Those black nodes are dominators.

*Explanation Phase I: An arbitrary integer is assigned as the id to each node in the graph G (V, E). At first, the colour white is allocated to each node. Initially, we choose a node at random from vertex set V.*According to definition it is our first independent set. Now delete it's adjacent node(s). Repeat previous steps until you get a maximal independent set which is a dominating set also. Color all the nodes in the MIS as black. Main goal of our algorithm is to get a minimal connected dominating set, we have to focus in getting minimal MIS. After this phase dominators (black nodes) are obtained. An illustration of the aforementioned algorithm is shown in Figure 1 below. Every node in the original graph is regarded as a white node. In Figure 1, this graph is displayed (a).

From the vertex set V of our original graph G, pick a node at random. Let's say that this node is 4. Now remove its neighbouring nodes 2, 6, and 9. Node 4 should be coloured black. It is the initial independent set, as stated in the definition. In Figure 1, this graph is displayed (b).

In the next step suppose node 5 is selected and delete it's adjacent nodes 1, 7. Color the node 5 as black. This is the second independent set. This graph is shown in Figure 1(c).

Select node 8 as the next node and delete it's adjacent nodes 10, 11. Color the node 8 as black. This is the third independent set and corresponding graph is shown in Figure 1(d).

In the next step node 12 is selected and delete it's adjacent nodes 13, 14, 15. Color the node 12 as black. This is the fourth independent set and the graph is shown in Figure 1(e).

Now remaining two nodes 3, 16 are our fifth and sixth independent set. Color those as black. In this way after first phase we are able to get a maximal independent set or a dominating set {3, 4, 5, 8, 12, and 16}. These nodes are called dominators and the graph contains the isolated vertices which are dominators & the maximal independent set. This graph is shown in Figure 1(f).

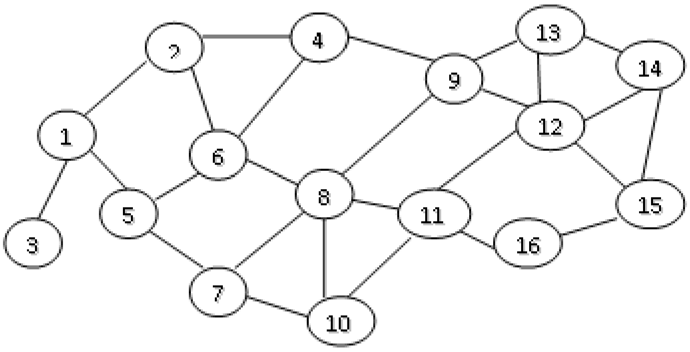

Fig 1(a): Initial graph

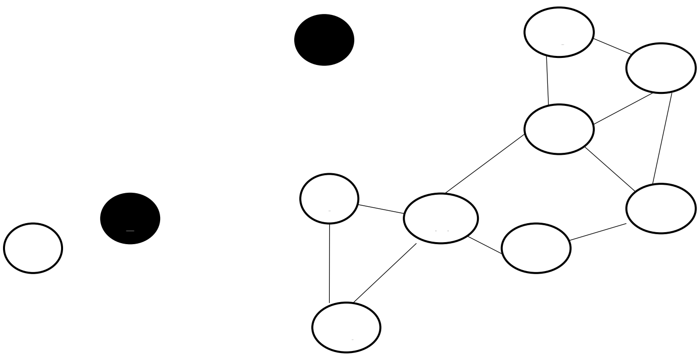

Fig 1(c): Node 5 selected as second IS. Color it black and delete its adjacent nodes 1, 7

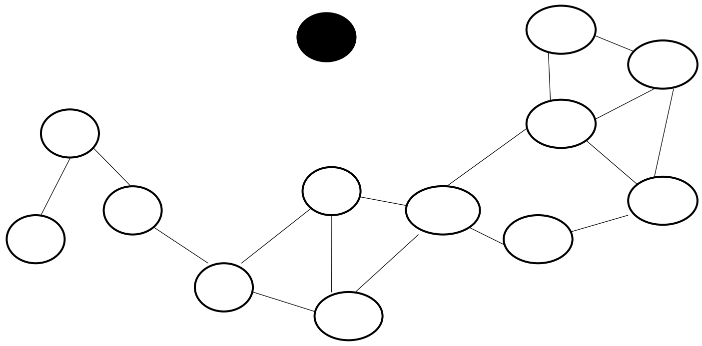

Fig 1(b): Node 4 selected as first IS. Color it black and delete its adjacent nodes 2, 6, 9

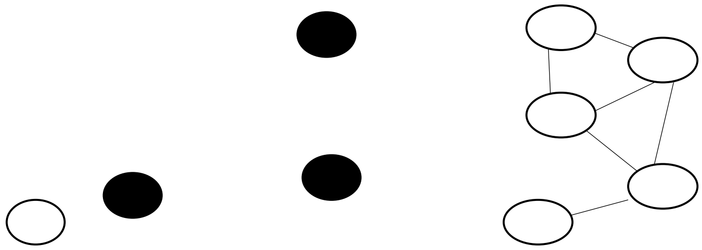

Fig 1(d): Node 8 selected as third IS. Color it black and delete its adjacent nodes 10, 11

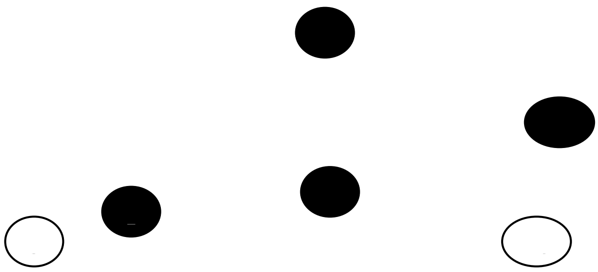

Fig 1(e): Node 12 selected as first IS. Color it black and delete its adjacent nodes 13, 14, 15

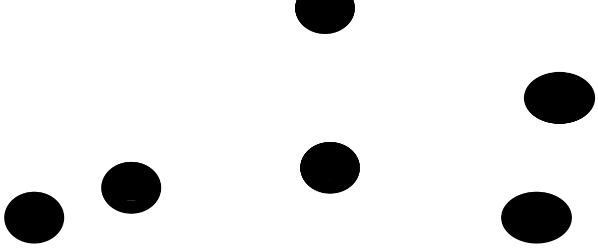

Fig 1(f): Remaining two ISs 3, 16 are our fifth and sixth IS. Color it black. Thus, we get a MIS/DS= {3, 4, 5, 8, 12, and 16}. These nodes are dominators.

Phase II. Determining Connectors

The algorithm's second phase includes figuring out the connections' phase. During this stage, a set of connectors is identified so that every node in the dominant or maximum independent sets is linked. The following are the steps in this phase:

1. Select such non-(dominating set or maximal independent set) nodes which are connected to the maximum number of dominating set node
2. Identify it as connectors between our dominating set nodes.
3. If there are two or more non-dS nodes connected with same and maximum number of DS nodes then choose a Non-DS nodes with smallest id
4. Do the step 2 until all the nodes in the DS get connected.
5. Color connecting nodes brown.
6. After this phase a Connected Dominating Set (CDS) will be invoked.

Explanation of Phase-II: This second phase of our algorithm is for finding the connectors between our Dominating Set nodes of first phase. Select Non-(Maximal Independent Set or Dominating Set) nodes which are connected to maximum number of Dominating Set nodes. Identify those as connectors between our Dominating Set nodes. If there are two or more Non-DS nodes connected with same and maximum number of DS nodes, and then choose a non-DS node with the smallest *id*. Color of the connecting nodes is shown as brown. After this phase a Connected Dominating Set (CDS) will be invoked. After finishing our first phase, now in our present graph there are three Non-DS nodes 6,9,11 which are connected to maximum number of i.e. three DS nodes. Choose a smallest-id node say 6. It is our first connector and colors it brown. This part of our connecting nodes method may be comprehended with the aid of Figure 2, which is shown below.

Node 9 is chosen as the second connector in the following stage since it is linked to the largest number of DS nodes—three. Apply a brown colour to it. In Figure 2, this graph is displayed (b).

Due to its ability to connect to the maximum number of DS nodes—three—Node 11 is chosen as the third connection. Apply a brown colour to it. In Figure 2, this graph is displayed (c).

In the next step it is observed that there are three Non-DS/MIS nodes 1,7,15 which are connected to second maximum number of (two) DS nodes. Choose node 1 as fourth connector as it has smallest id among the three. Color it as brown. Now all the dominating set nodes are connected. This graph is shown in Figure 2(d).

After this second phase Connected Dominating Set is invoked which is {1, 3, 4, 5, 6, 8, 9, 11, 12, 16}. Here black nodes are called dominator, white nodes are dominatees and brown nodes are connectors. This graph is shown in Figure 2(e).

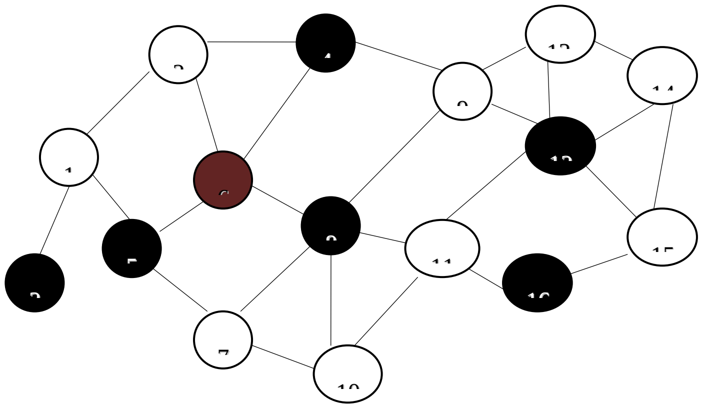

Fig 2(a): There are 3 Non-DS nodes 6,9,11 which are connected to 3 DS nodes. Choose node 6 as first connector and color it brown

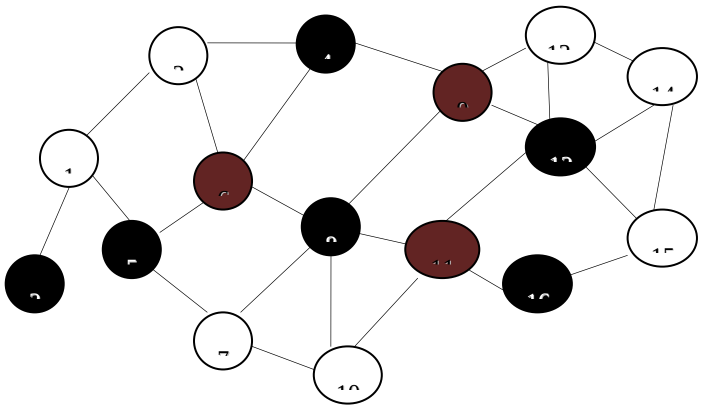

Fig 2(c): Next connector is node 11 which is connected to 3 DS nodes. Choose node 11 as third connector and color it brown

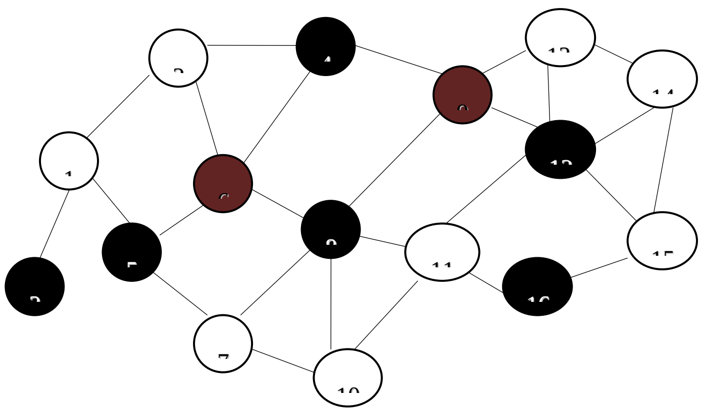

Fig 2(b): Next connector is node 9 which is connected to 3 DS nodes. Choose node 9 as second connector and color it brown

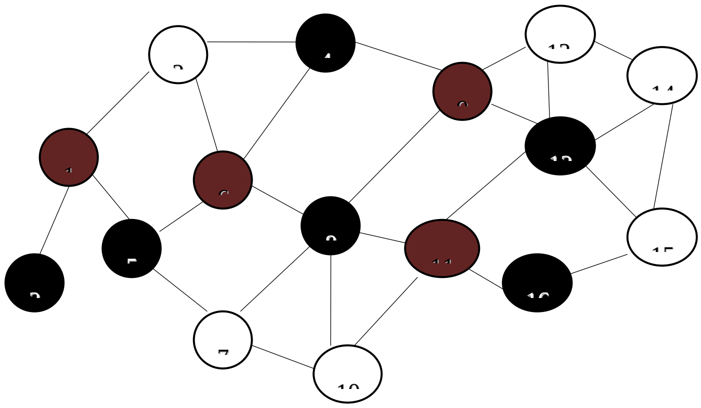

Fig 2(d): Next connectors are nodes 1,7,15 which are connected to 2 DS nodes. Choose node 1 as fourth connector and color it brown. Now all the DS nodes are connected

C. Phase III. Pruning

Our main goal is to get a Minimal Connected Dominating Set. So by pruning mechanism we can discard some nodes from our achieved CDS in previous phase to get a Minimal Connected Dominating Set.

1. Select one or more nodes in the CDS which has/have minimum degree keeping in mind that remaining nodes are connected.

2. Select one or more dominators which can be pruned, keeping the remaining CDS intact and the adjacent dominates can communicate through their dominators.

3. Do the steps 2-3 until you can get the smallest possible CDS in the given WSN graph.

Explanation of Pruning Phase: As our main goal is to get a Minimal Connected Dominating Set. So we apply pruning mechanism by which we can discard some nodes to get a Minimal Connected Dominating Set. We have to select one or more nodes in the Connected Dominating Set which has/have minimum degree keeping in mind that remaining nodes are connected. After this step is finished select one or more dominators which can be pruned, keeping the remaining CDS intact and the adjacent dominators can communicate through their dominators. After this phase we can get the smallest possible CDS in the given WSN graph. After getting a CDS in the graph presently we can observe that nodes 3 and 16 have minimum degree of 1 and 2, respectively among the all CDS nodes. These can be pruned. Now Minimal Connected Dominating Set= {1, 4, 5, 6, 8, 9, 11, 12}. This pruning phase of our algorithm can be understood with the help of Figure 3 below.

Minimum Connected Dominating Set Construction: In the next step further we can prune nodes 4 and 9. .Because although pruned, the remaining CDS is intact and the dominates can still communicate through the dominators. Now our final minimal Connected Dominating Set is= {1, 5, 6, 8, 11, and 12}. This is our final optimal Minimal CDS. The two graphs are shown below in Figure 3(a) and Figure 3(b).

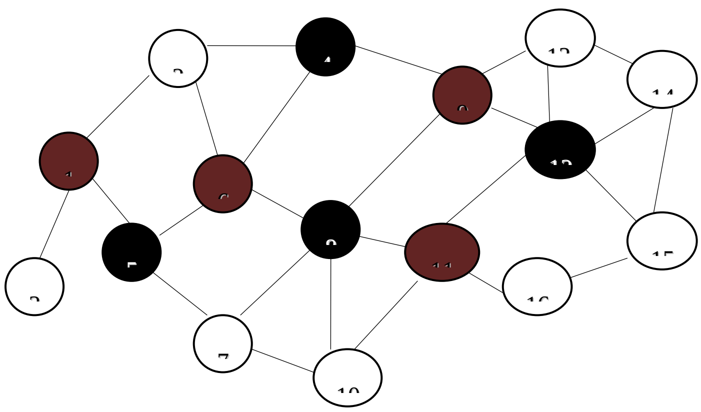

Fig 3(a): Nodes 3 and 16 have minimum degree of 1 and 2, respectively among the all CDS nodes. These can be pruned. Now minimal CDS= {1, 4, 5, 6, 8, 9, 11, 12}

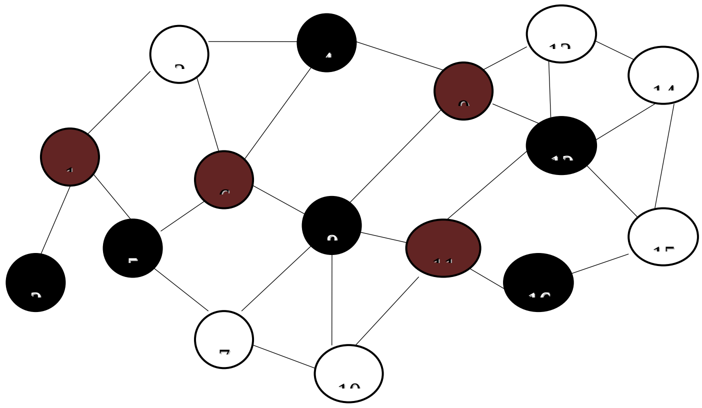

Fig 3(b): Further we can prune nodes 4 and 9. Because though pruned, the remaining CDS is intact and the dominatees can communicate through the dominators. Now the final Minimal CDS= {1, 5, 6, 8, 11, 12}

V. Result And Discussion

Simulation Framework

The proposed algorithm s written in C language and simulated with a machine having Intel Core i5 processor with 2.8GHz Clock speed and 2 GB RAM. The efficacy of the proposed algorithm is tested by simulation for network graphs with 5-1000 nodes. Number of nodes in resulting MCDS and time taken for each graph are compared with the existing methods.

Unit disc graphs for the sensor networks have been built in the first section of the simulation. The procedure has been performed to identify the number of nodes in the resultant MCDS with an increasing number of nodes after the graph has been built.

Experimental Results

The findings are displayed in Figure 4, and a comparison with the approaches already in use [6] and [7] is made regarding the number of MCDS nodes for various sized sensor networks with a graph of nodes ranging from 20 to 1000. Figure 4 illustrates how much smaller the MCDS is in our approach than it is in [6] and [7]. Our algorithm's MCDS size is approximately half of [6] and one-sixth of [7]. Figure 5 compares how long it takes the method to run for every 20–1000 nodes in a network to create an MCDS. Compared to [6] and [7], it is seen that the time required by the suggested method grows evenly and in a well-distributed way with an increase in the number of nodes. Figure 5 illustrates how long the algorithm takes—roughly one-fourth of [7].

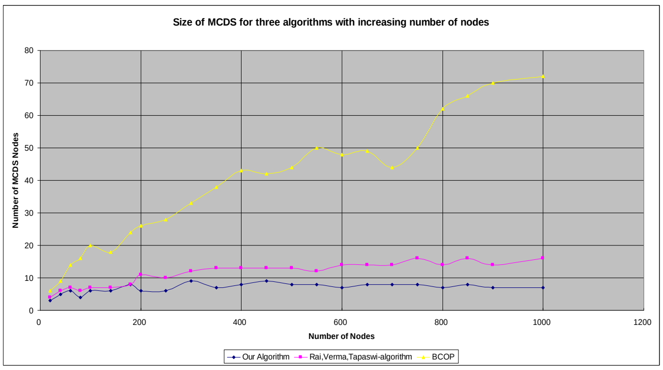

Fig 4: Entire MCDS Size for Three Algorithms with Increasing Node Count

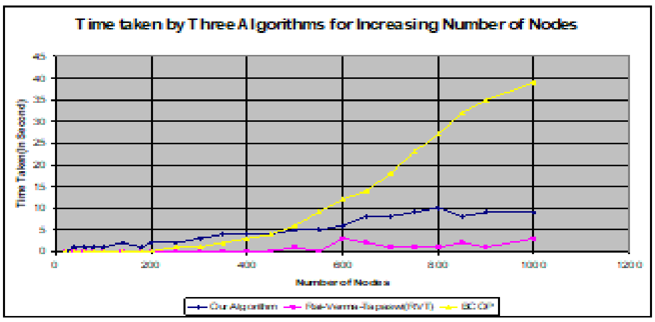

Fig 5: Three Algorithms with Increasing Number of Nodes: Time Taken

V. **Conclusion**

In conclusion, the proposed algorithm for constructing a Minimal Connected Dominating Set in wireless sensor networks marks a significant advancement in the field. By integrating a unique weighting mechanism and a systematic multi-phase approach, our algorithm efficiently reduces the size of the MCDS while ensuring robust network connectivity. The algorithm's performance, validated through extensive simulations, shows a notable improvement over existing methods in terms of both the size of the MCDS and the computational time. This efficiency is crucial for applications in wireless sensor networks where resource constraints are a primary concern. Furthermore, the inclusion of a maintenance phase allows for adaptability in dynamic network environments, making the algorithm practical for real-world scenarios. Our work not only enhances the understanding of virtual backbone formation in wireless sensor networks but also opens avenues for further research in optimizing network resource usage and improving overall network performance.

References

[1] Torres Martín, César, Acal, Christian, El Homrani, Mohammed, and Ángel C. Mingorance Estrada. "Impact on the Virtual Learning Environment Due to COVID-19." Sustainability 13, no. 2 (2021): 582. Accessed April 19, 2023. https://doi.org/10.3390/su13020582.

[2] Del Olmo-Muñoz, J., González-Calero, J.A., Diago, P.D. et al. Intelligent tutoring systems for word problem solving in COVID-19 days: could they have been (part of) the solution?. ZDM Mathematics Education 55, 35–48 (2023). https://doi.org/10.1007/s11858-022-01396-w

[3] Bozkurt, Aras, Karakaya, Kadir, Turk, Murat, Karakaya, Özlem, and Daniela Castellanos-Reyes. "The Impact of COVID-19 on Education: A Meta-Narrative Review." Techtrends 66, no. 5 (2022): 883-896. Accessed April 19, 2023. https://doi.org/10.1007/s11528-022-00759-0.

[4] Nayak, B., Bhattacharyya, S.S., Goswami, S. et al. Adoption of online education channel during the COVID-19 pandemic and associated economic lockdown: an empirical study from push–pull-mooring framework. J. Comput. Educ. 9, 1–23 (2022). https://doi.org/10.1007/s40692-021-00193-w

[5] Li, Xia, Huali Yang, Shengze Hu, Jing Geng, Keke Lin, and Yuhai Li. "Enhanced hybrid neural network for automated essay scoring." Expert Systems 39, no. 10 (2022): e13068.

[6] Tashu, Tsegaye Misikir, Chandresh Kumar Maurya, and Tomas Horvath. "Deep Learning Architecture for Automatic Essay Scoring." arXiv preprint arXiv:2206.08232 (2022).

[7] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).

[8] Soam, DMilind, and Sanjeev Thakur. "Sentiment Analysis Using Deep Learning: A Comparative Study." In 2022 Second International Conference on Computer Science, Engineering and Applications (ICCSEA), pp. 1-6. IEEE, 2022.

[9] Ali, Omar, Wiem Abdelbaki, Anup Shrestha, Ersin Elbasi, Mohammad Abdallah Ali Alryalat, and Yogesh K. Dwivedi. "A systematic literature review of artificial intelligence in the healthcare sector: Benefits, challenges, methodologies, and functionalities." Journal of Innovation & Knowledge 8, no. 1 (2023): 100333.

[10] Torres Martín, César, Acal, Christian, El Homrani, Mohammed, and Ángel C. Mingorance Estrada. "Impact on the Virtual Learning Environment Due to COVID-19." Sustainability 13, no. 2 (2021): 582. Accessed April 19, 2023. https://doi.org/10.3390/su13020582.

[11] Coman, Claudiu, Țîru, Laurențiu G., Stanciu, Carmen, and Maria C. Bularca. "Online Teaching and Learning in Higher Education during the Coronavirus Pandemic: Students' Perspective." Sustainability 12, no. 24 (2020): 10367. Accessed April 19, 2023. https://doi.org/10.3390/su122410367.

[12] Mutizwa, Melissa R., Ozdamli, Fezile, and Damla Karagozlu. "Smart Learning Environments during Pandemic." Trends in Higher Education 2, no. 1 (2023): 16-28. Accessed April 19, 2023. https://doi.org/10.3390/higheredu2010002.

[13] Amin, M. R., Islam, M. S., Islam, M. R., & Mahmud, M. (2021). Natural language processing techniques for hate speech detection and analysis: A comprehensive review. ACM Computing Surveys, 54(6), 1-39.

[14] Burstein, J., Kukich, K., Wolff, S., Lu, C., & Chodorow, M. (1998). Using NLP to Identify Misconceptions in Students' Writing. IEEE Intelligent Systems, 13(5), 46-54.

[15] Chen, Y., Cheng, W., & Huang, Y. (2016). Automated Feedback Assessment for Improving Students' Writing. IEEE Transactions on Learning Technologies, 9(3), 197-210.

[16] Kukich, K., J.& Becker, S.& R., & Mostow, J. (1993). An evaluation of an automated response evaluation system. Intelligent Tutoring Systems, 4, 4-12.

[17] Attali, Yigal, and Jill Burstein. "Automated essay scoring with e-rater® v.2." Journal of Technology, Learning, and Assessment 4, no. 3 (2006): 1-30.

[18] Qian, Y., Liu, J., Jia, J., & Xie, X. (2019). An Automated Feedback Assessment Model for Online Discussions Based on Multiple-Aspect Analysis. IEEE Access, 7, 93596-93608.

[19] Page, E. B. (1966). The use of the computer in analyzing student essays. Journal of Educational Measurement, 3(4), 211-220.

[20] Landauer, T. K., Laham, D., & Foltz, P. W. (2003). Automated essay scoring: A cross-disciplinary perspective. Lawrence Erlbaum Associates Publishers.

[21] Amado, S., & Dill, K. E. (1998). Automated essay grading: An application of n-gram analysis. In Proceedings of the 21st Annual Conference of the Cognitive Science Society (pp. 105-110).

[22] Burstein, J., Leacock, C., & Swartz, R. (2003). CRATER: using natural language processing to provide automated feedback on student writing. In Proceedings of the 8th International Conference on Intelligent Tutoring Systems (pp. 115-124). Springer.

[23] Alsariera, Y. A., Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A. A., & Ali, N. (2022). Assessment and evaluation of different machine learning algorithms for predicting student performance. Computational Intelligence and Neuroscience, 2022, 1–11.

[24] Zhang, Y., Li, B., Hu, Y., Zhang, Y., & Li, X. (2019). Automated feedback assessment of writing performance based on lexical and syntactic features. Journal of Educational Computing Research, 56(1), 144-164.

[25] Amin, A., Singh, P., & Arora, S. (2021). Automated feedback generation for programming assignments: A survey. Journal of Educational Computing Research, 59(6), 1489-1531.

[26] Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. "Automatic Dialogue Assessment with Sequence to Sequence Learning." In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 968-977. 2018.

[27] Madnani, N., Dabre, R., & Mott, B. (2018). Automated writing evaluation for feedback on draft essays. Handbook of Automated Essay Evaluation, 121-140.

[28] Wilson, T., Wiebe, J., & Hoffman, P. (2017). Recognizing contextual polarity in phrase-level sentiment analysis. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 347-352).

[29] Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 1882-1891).

[30] Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies, 1480-1489.

[31] Jamil H. and Mou X., "Automated Feedback and Authentic Assessment for Online Computational Thinking Tutoring Systems," 2022 International Conference on Advanced Learning Technologies (ICALT), 2022, pp. 53-55, doi: 10.1109/ICALT55010.2022.00024.

[32] Lu, W., Vivekananda, GN & Shanthini, A. Supervision system of english online teaching based on machine learning. Prog Artif Intell (2022). https://doi.org/10.1007/s13748-021-00274-y

[33] Nayak S., Agarwal R. and Khatri S., "Review of Automated Assessment Tools for grading student SQL queries," 2022 International Conference on Computer Communication and Informatics (ICCCI), 2022, pp. 1-4, doi: 10.1109/ICCCI54379.2022.9740799.

[34] Zhang, Y., Liu, H., & Yang, P. (2021). An Automated Feedback Assessment Model for EFL Writing Based on Sentiment Analysis and Hierarchical Attention Network. IEEE Access, 9, 16571-16581. https://doi.org/10.1109/ACCESS.2021.3051497

[35] Wang, Y., Li, L., Li, H., Zhang, X., & Li, L. (2021). An Automated Feedback Assessment Model for Online Courses Based on Machine Learning and Rule-Based Techniques. IEEE Access, 9, 39712-39720. https://doi.org/10.1109/ACCESS.2021.3064022

[36] Burstein, J., Deane, P., & Chodorow, M. (2013). The e-rater® automated essay scoring system. Handbook of Automated Essay Evaluation: Current Applications and New Directions, 55-67.

[37] ttali, Y., & Powers, D. E. (2015). Hybrid Scoring Model (HSM): A new approach to automated text scoring combining syntactic, semantic, and discourse processing. Educational and Psychological Measurement, 75(2), 185-206.

[38] Rudner, L. M., & Liang, S. (2002). An overview of Bayesian adaptive testing. Journal of Educational and Behavioral Statistics, 27(4), 323-340.

[39] Wright, H., Kemp, J., & Varner, L. K. (2016). The Writing Mentor: A Comprehensive Writing Tool for the Middle Grades. Journal of Educational Technology Development and Exchange, 9(1), 1-14.

[40] Bai, Yunpeng, Linjie Li, Jingjing Xu, Xu Sun, Weiwei Liu, and Qi Su. "PARS-BERT: Pre-training of Parsimonious Language Model for Adapting BERT to Resource-poor Domains." arXiv preprint arXiv:2010.16024 (2020).

[41] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[42] Li, Pengfei, Yuxian Deng, and Shuai Wu. "Semantics-aware BERT for Language Understanding." arXiv preprint arXiv:2106.13139 (2021).

[43] Kumar Agarwal A, Angeline Ranjithamani D, Pavithra M, Velayudham A, Shunmugam A, et al., (2021) Machine Learning Technique for the Assembly-Based Image Classification System. J Nucl Ene Sci Power Generat Techno 10:9.

[44] R. G. Tiwari, A. Misra, V. Khullar, A. K. Agarwal, S. Gupta and A. P. Srivastava, "Identifying Microscopic Augmented Images using Pre-Trained Deep Convolutional Neural Networks," 2021 International Conference on Technological Advancements and Innovations (ICTAI), Tashkent, Uzbekistan, 2021, pp. 32-37, doi: 10.1109/ICTAI53825.2021.9673472.

[45] K. Agarwal, G. Khan, S. Qamar, and N. Lal, “Localization and correction of location information for nodes in UWSN-LCLI,” Adv. Eng. Softw., vol. 173, p. 103265, 2022, doi: 10.1016/j.advengsoft.2022.103265.

[46] Tarun Sharma, Ambuj Kumar Agarwal, Danish Ather and Ashendra Saxena, "SEARCH BASED SOFTWARE ENGINEERING IN REQUISITE PHASE OF SDLC: A SURVEY", Technical journal of LBSIMDS.

[47] K. . Agarwal, V. . Kiran, R. K. . Jindal, D. . Chaudhary, and R. G. . Tiwari, “Optimized Transfer Learning for Dog Breed Classification”, Int J Intell Syst Appl Eng, vol. 10, no. 1s, pp. 18–22, Oct. 2022.

[48] R. Sharma, H. Pandey, and A. K. Agarwal, “Exploiting artificial intelligence for combating COVID-19 : a review and appraisal,” vol. 12, no. 1, pp. 514–520, 2023, doi: 10.11591/eei.v12i1.4366.

[49] D. Srivastava, H. Pandey, and A. K. Agarwal, “Complex predictive analysis for health care : a comprehensive review,” vol. 12, no. 1, pp. 521–531, 2023, doi: 10.11591/eei.v12i1.4373.