Connected Dominating Set Problem and its Application to Wireless Sensor Networks

Razieh Asgarnezhad
Department of Computer Engineering
Arak Branch, Islamic Azad University
Arak, Iran
raziehasgarnezhad@yahoo.com

Javad Akbari Torkestani
Department of Computer Engineering
Islamic Azad University, Arak Branch
Arak, Iran
j-akbari@iau-arak.ac.ir

Abstract— **In Wireless Sensor Networks, all nodes are energy constrained. There are no predefined and no fixed infrastructures in networks. A Connected Dominating Set can be created by different algorithms to organize nodes in a better way. A Connected Dominating Set can be shown as a backbone. A backbone is a subset of nodes that are able to perform especial tasks and serve nodes which are not in the backbone. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a Wireless Sensor Network. For example, Connected Dominating Set nodes can perform efficient routing and broadcasting in networks. This paper tries to survey and classify different Connected Dominating Set formation algorithms. We compare their performances with each other.**

Keywords- wireless sensor network; maximal independent set; connected dominating set.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have attracted recent research attention due to wide range of network communications applications they support. In WSNs, all nodes are energy constrained. They include a number of wireless nodes and they can be divided into three parts: data collection, based-station and data management center. Also, there is no fixed or predefined infrastructure in these networks. A kind of broadcasting in sensor networks is normally flooding-based, where each node retransmits the broadcasting message that it receives. But it raises energy consumption because packet retransmission is needed when interference occurs. Also, it will has broadcast storm problem. [2][10]

The extensive research performed in the past of decades in WSNs. Among the topics that have received especially attention, clustering formation and interconnection, also referred as *backbone formation*. Backbone will remove unnecessary transmission links through shutting down some of redundant nodes. Although backbone will still guarantee network connectivity in order to deliver data efficiently in a WSN. In virtual backbones based WSNs, some nodes are chosen as dominator node (backbone node) in the backbone construction process.

A backbone is a subset of nodes that are able to perform especial tasks and it serve nodes which are not in the backbone. Therefore, the backbone construction depends on the task to be carried. In WSNs, a backbone could be the set of active sensors while the rest of the sensors are sleeping. The backbone of a network is normally required to be connected, so that the backbone nodes are able to communicate to perform especial tasks. For example, to connect backbone nodes in ad hoc networks can perform efficient routing and broadcasting. A Connected Dominating Set (CDS) can be showed as a backbone. Backbones improved the routing procedure. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a WSN. [15]

The nodes in CDS are called dominator (backbone node), other nodes are called dominatee (non-backbone node). With the help of CDS, routing is easier and can adapt quickly to network topology changes. To reduce the traffic during communication, it is desirable to construct a minimum CDS (MCDS). Constructing a MCDS is proved a NP-hard problem and in recent years many algorithms of constructing an approximate MCDS have been proposed. [12][15]

We classify different CDS formation algorithms in these networks in Section 2. In Section 3, we present some examples of this classification. We compare performance of these with each other in Section 4. In Section 5, we conclude the paper.

II. CLASSIFICATION OF CDS FORMATION ALGORITHMS

We will present a new classification of CDS formation algorithms. From varied aspects, we can be classified into blew different types.

A. UDG and DGB

The CDS construction algorithms can classified into two types: Unit Disk Graph (UDG) based algorithms and Disk Graphs with Bidirectional (DGB) links. In UDG and DGB, the link between any pair of nodes is bidirectional. The nodes transmission ranges in UDG are the same, but in DGB are different. Even in UDG and DGB, MCDS is proved as a NP-hard problem. In Figure 1, we show a UDG of CDS virtual backbone. [2][13][15][17]

 ISBN: 978-1-61208-161-8

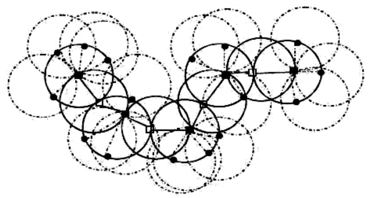

Figure 1. A sample for UDG of CDS virtual backbone [15]

B. *MIS Based and Non-MIS Based*

Independent set (IS) of a graph G is a subset of vertices so that no two vertices are adjacent in the subset. Maximal Independent set (MIS) is an IS, so that it is not a subset of any other IS. Note that in an undirected graph an MIS is also a Dominating Set (DS). The MIS based algorithms have two kinds of realization. The optimal nodes selection is based on some criterions such as node degree, rest energy of node, and node id, etc. [8] [15] [18] [19]

C. *Centralized Algorithms and Decentralized Algorithms*

Algorithms that construct a CDS can be divided into two types: centralized algorithms and decentralized algorithm. The centralized algorithms in general result in a smaller CDS with a better performance ratio than that of decentralized algorithm. The decentralized algorithms also can be divided into two types: distributed algorithms and localized algorithms. In distributed algorithms, the decision process is decentralized. But, in the localized algorithms, the decision process is not only distributed also requires only a constant number of communication rounds. Most of the distributed algorithms find a MIS then, connect this set. [3][8][9][18][19]

D. *Pruning-Based Algorithms*

Some algorithms use pruning rules to reduce the redundant nodes of backbone. In often these algorithms, all nodes of network considered to be backbone nodes for creating CDS. Then they pruned its redundant nodes to can create MCDS. [2][4][[5-6] [15] [19]

III. SOME EXAMPLES OF THIS CLASSIFICATION

We will present some examples of this classification and explain their approaches.

A completely localized algorithm was proposed to construct CDS in general graphs. At first, all vertices are unmarked. Then, they exchange their open neighborhood information with their one-hop neighbors. Thus, each node knows all of its 2-hop neighbors. The marking process applies the following simple rule: any vertex having two unconnected neighbors so that they are marked as a dominator. At last, the set of marked vertices form a CDS, but it had a lot of redundant nodes. There are two pruning principles so that they are provided to post-process the DS, according to the neighborhood subset coverage. Also, when two of its connected neighbors in S with higher ids can cover all of *u*'s neighbors then node *u* will be deleted from S. This pruning idea is expressed to the following general rule [11]. According to this rule, if there is *k* connected neighbors with higher ids in S so that can cover all *u*' s neighbors then, a node *u* can be removed from *S*. [19]

Guha et al. [9] proposed two CDS construction approach. The algorithm1 begins through marking all vertices white. At first, the algorithm selects the node with the maximal number of white neighbors. The selected vertex is marked black and its neighbors are marked gray. The algorithm iteratively seeks the gray nodes and their white neighbors, and it selects the gray node or the pair of nodes, whichever has the maximal number of white neighbors. The selected node or the selected pair of nodes is marked black, and also their white neighbors marked gray. The algorithm terminates, when all of the vertices are marked gray or black. All the black nodes form a CDS. This algorithm results in a CDS of size at most $2(1+H(\Delta)).|OPT|$, where *H* is the harmonic function, and *OPT* refers to a MCDS.

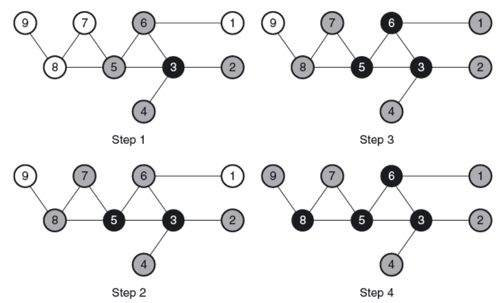

Figure 2. An example of Guha and Khuller's algorithm 1 (above) [9]

The algorithm2 also begins through coloring all nodes white. A *piece* is defined to be either a connected black component, or a white node. The algorithm includes two phases. The first phase iteratively selects a node that yield the maximum reduction of the number of pieces. A node is marked black and its white neighbors are marked gray, when it is selected. At last, the first phase terminates when no white node left. Therefore, there exists at most $|OPT|$ number of connected black components. The second phase constructs a Steiner Tree to connect all the black nodes through coloring chains of two gray and black nodes. The size of the resulting CDS formed via all black nodes is at most $(3+ln(\Delta))|OPT|$.[9]

Das et al. [8] proposed the distributed implementations of the two greedy algorithms. The first algorithm grows one node with maximum degree to be form a CDS. A node must know the degree of all nodes in the graph. Each step selects either a one- or two-edged path from the current CDS. Then the nodes in the CDS must know the number of unmarked neighbors for all nodes one and two hops from the CDS. This algorithm produces a CDS with approximation ratio of $2H(\Delta)$ in $O(|C|(\Delta+|C|))$ time, using the $O(n|C|)$ messages, where the harmonic function, *n* is the total number of vertices, and *C* represents the final CDS.

In the second algorithm, they compute a DS and then select additional nodes to connect the set. According to the

 ISBN: 978-1-61208-161-8

DS in the first stage, an unmarked node compares its effective degree, with the effective degrees of all its neighbors in two-hop neighborhood. The greedy algorithm adds the node with maximum effective degree to the DS. When a DS is achieved, the first stage terminates. The second stage connects the components through a distributed minimum spanning tree algorithm. At last, the nodes in the resulting spanning tree compose a CDS. This algorithm has time complexity of $O((n+|C|)\Delta)$, and message complexity of $O(n|C|+m+n\log(n))$. It have the MCDS with a ratio of $2H(\Delta)+1$, where m is the cardinality of the edge set.

Akbari et al. [2] proposed an intelligent backbone formation algorithm according to distributed learning automata (DLA). The worst case running time and message complexity of the backbone formation algorithm has a $1/(1-\varepsilon)$ optimal size backbone. This was why that it was shown that through a proper choice of the learning rate of the algorithm, a trade-off between the running time and message complexity of algorithm with the backbone size can be made.

In implementation, a network of the learning automata isomorphic to the UDG was used. At first, it formed through equipping each host to a learning automaton. At each stage of this approach, the learning automata randomly choose one of their actions so that a solution can be found in the CDS problem. The created CDS is evaluated via the random environment, and the action probability vectors of the learning automata are updated depending on the response received from the environment. At last, in an iterative process, the learning automata converge to a common policy and it constructs a minimum size virtual backbone for us.

This algorithm used a pruning rule to avoid choosing the same dominators. In this rule point of view, it increases the convergence speed, and also, decreases the running time of the proposed algorithm. With comparing the results of proposed algorithm with the other of the best known CDS-based backbone formation algorithms, the results show that their algorithm always outperforms the others in terms of the backbone size, and also its message overhead is only a few more than the least cost algorithm. [2]

Alzoubi et al. [3] provided two versions of an algorithm to construct the DS for a wireless network. In these algorithms, they employ the distributed leader election algorithm [6] to construct a rooted spanning tree. A labeling strategy is used to divide the nodes in the tree to be either black or gray, according to their ranks. The rank of a node is the arranged pair of its level and its id. The labeling process begins from the root node and finishes at the leaves. At first, the node with the lowest rank marks itself black and broadcasts a *DOMINATOR* message. According to the following rules, the marking process continues:

- "If the first message that a node receives is a *DOMINATOR* message, it marks itself gray and broadcasts a *DOMINATEE* message."[3]
- "If a node received *DOMINATEE* messages from all its lower rank neighbors, it marks itself black and sends a dominator message."[3]

When it reaches the leaf nodes, the marking process finishes. Just now, the set of black nodes form an MIS. At last, in the final phase the nodes connect in the MIS to form a CDS through *INVITE* and *JOIN* messages. This algorithm has time complexity of $O(n)$, and message complexity of $O(n \log(n))$.

Rai et al. [15] proposed an algorithm for finding MCDS with using of DS. DSs are connected through using Steiner tree. The approximation algorithm includes of three stages. At first, the DS is determined through identifying the maximum degree nodes to discover the highest cover nodes. Then, it connects the nodes in the DS through a Steiner tree. At last, this tree prunes to form the MCDS. For local repair, rule *k* [11] is used to find the nodes so that can maintain the MCDS. This phase includes of the following steps:

- An arbitrary number say *id* is assigned to each node in the graph $G(V,E)$
- Each node is assigned white color
- The node u with maximum degree is taken from $G(V,E)$ and color as black, *i.e.* Dominator
- All the neighbor nodes of the node u are Colored
- Do step 3-4 till all the nodes in the graph $G(V, E)$ are colored either as black or gray.

Set of connectors B is found so that all the nodes in D connected. The set of D and B includes black nodes and also dark gray nodes, respectively. A node in B is connected through at most K. Set of dark gray nodes along with given D could be found via Steiner tree. Interconnecting all the nodes in D are through adding new nodes between them. Steiner nodes are nodes that are in the Steiner tree but not in set D. At last, constructed CDS will include of black and dark gray nodes.

This steps present in the following:

- Select a gray node which is connected to Maximum (K) number of black nodes, set Its color as dark gray
- Check whether the Dominating Set D
- if D gets connected stop
- else go to step *1* with $K-1$ number of Black nodes

Eventually, in the pruning phase, redundant nodes are deleted from the CDS to obtain the MCDS. These rules present in following steps:

- Select a minimum degree node u from F
- check if $N[u]$ is subset of $N[1]$ and $N[2]$ and ...$N[n]$ where i belongs to $F-\{u\}$
- if step 2 returns *true* then remove node u and go to step 1
- Otherwise do not remove node u and go to step 1

They also proposed a local repair algorithm to take care of node's deletion.

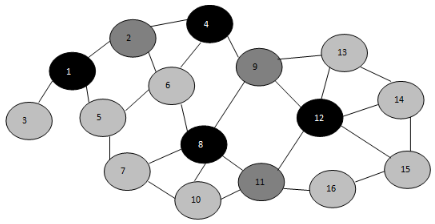

Figure 3. Show the final MCDS Backbone [15]

We have shown obtained solution of foresaid algorithm in above figure with a specific one.

Li et al. [13] proposed an algorithm for constructing CDS. They called it as Approximation Two Independent Sets based Algorithm (ATISA). The ATISA has three stages: (1) constructing a connected set (CS) (2) constructing a CDS (3) pruning the redundant dominators of CDS. ATISA constructs the CDS with the smallest size, compared with some well-known CDS construction algorithms. The message complexity of this algorithm is *O (n)*. The ATISA has two kinds of implementations: centralized and distributed. The centralized algorithm consists of three stages, which are CS construction stage, CDS construction stage, and pruning stage.

In the centralized algorithm, the initial node is selected randomly. Then the algorithm executed several rounds. When the first stage is ended, there are no black nodes generated in the network. The generated black node set is formed a CS. If a white node has black neighbors then, it will select the black neighbor with the minimum id as its dominator and it change its state into gray. If a white node only has the gray neighbors then, it will send an invite message to the gray neighbor with the minimum id and it change its state into gray. In the second stage, constructs a CDS and all the nodes are either black or gray. Finally, there is no white node left in the network. According to the third stage, if a black node with no children and if the neighbors of the black node are all adjacent to at least two black nodes, then the black node is put into connected set.

In the distributed implementation, all nodes are initialized white. After the first stage, there are white nodes, gray nodes, and black nodes. Then, in the second stage, there are black nodes, gray nodes and sometimes white nodes. White nodes can change their states into gray and also gray nodes can change their states into black. In the third stage, the redundant black nodes are deleted. [13]

Xie et al. [20] called their algorithm as Connected Dominating Set-Hierarchical Graph (CDS-HG). It is a distributed MCDS approximation algorithm. They showed that this algorithm generates smaller CDS sizes compared with the existing algorithms. Their algorithm includes of two phases. At first, in the first phase, rule1 (Essential Node Determination) is used. According to this rule, a set of dominators select for each hierarchical level so that all nodes in the next level are dominated by these dominators. A greedy strategy is used to select the dominators for creating a small initial DS. In the second phase, another rule (rule2) is used to remove the redundant dominators. This process repeated from the lowest level to the highest level of the hierarchical graph. According to The greedy strategy that created CDS is connected. Also, the size of CDS generated is at most $(\log n|opt|)$, where n is the number of nodes in the network and $|opt|$ is the cardinality of a minimum DS. The computation complexity of their algorithm is $o(n^2)$. Because a centralized CDS algorithm is impractical for WSNs, Thus, they implemented a distributed algorithm based on competition. It includes of three phases: creating the initial CDS through competition and reducing the CDS size through applying rule2 on all dominators. Respectively, the computation and message complexities of their algorithm are $o(\theta^2)$ and $o(\theta)$, where θ is the maximum number of child nodes in graph. [20]

A virtual backbone was proposed for Wireless Ad-hoc Sensor Networks. According to this algorithm, the sensor network is divided into clusters. This algorithm includes of two phases. First, they clustered sensor nodes through clustering algorithm and then implemented the CDS algorithm to intra clusters. They assume all vertices are unmarked. They exchange their open neighborhood information with their one-hop neighbors. With using two pruning rules are provided to post-process the DS. If there exists a node v with higher id so that the closed neighbor set of u is a subset of the closed neighbor set of v, node u can be taken out from the CDS. [4]

Acharya et al. [1] proposed Energy-Aware Virtual Backbone Tree (EVBT) that it is a distributed algorithm for constructing a backbone in WSN. It chooses only nodes with enough energy levels as the member of the virtual backbone. Also, it introduced a concept of threshold energy level for members of virtual backbone. According to it, only nodes with energy levels above a predefined threshold are included in the EVBT. They used an undirected graph to represent a WSN. Sensor node that does not belong to the backbone is termed as *leaf node*. Every node in the network has an EVBT node. They term this EVBT node as the dominator of the corresponding leaf node. They presumed each node v knows its *N(v)*. They check two types of vertices. A tree node is a *fixed vertex* so that cannot be removed from the EVBT. It means that this vertex will be a part of the final solution. If energy level of *Non-fixed* vertices is not above threshold energy level or its removal does not disjoin the resulting sub graph, then *Non-fixed* vertices will be removed. At each step of the algorithm, at least one vertex is either fixed, or removed. It is presumed that at first, all the nodes in the network form the EVBT. At last, these non-removed and fixed vertices form the EVBT. They presumed, the sink node is leader to starts execution of algorithm.

In this algorithm, every node in the network has one virtual backbone node, which it selects as its dominator. This dominator will be parent node for that node. Any node in the network will forward its packet to its dominator. In this way the packet eventually reaches the sink node. [1]

Hussain et al. [10] constructed a CDS-based backbone to support the operation of an energy efficient network. It focused on three key ideas in their design: (1) a realistic weight matrix, (2) an asymmetric communication link between pairs of nodes, and (3) a role switching technique to prolong the lifetime of the CDS backbone. This algorithm is distributed in nature, and does not require global information. Hence, it is deterministic.

Corresponding with the weight comparison among neighbors, some suitable nodes get selected as dominators. The set of dominators is a MIS. At first, those selected dominators, in conjunction with some Connector nodes (dominator2 nodes), then on form the dominating set of the network. On the other side, nodes that are not part of the dominating set remain as dominatees, and use neighboring dominators as next hops for data communication. This algorithm presumed that all nodes know 2-hop away neighborhood information. The weight matrix used in r-CDS algorithm is: $W_i(r_i, deg_i, id_i)$. Node i is more suitable to be a dominator than neighboring node j, if any of the following is true: [10]

deg(u)- The effective node degree of node u

r(u)- The number of 2-hop away neighbors

- $r(i) < r(j)$
- $r(i) = r(j)$ *and* $deg(i) > deg(j)$
- $r(i) = r(j)$ *and* $deg(i) = deg(j)$ *and* $id\ (i) < id\ (j)$

According to this algorithm, sensor nodes in the r-CDS algorithm can have three different colors: white, gray and black. At first, all nodes are white. In continue, all nodes change their color to either black or gray. Black nodes form network backbone, but gray nodes remain as dominatees. In their algorithm, nodes can broadcast the following messages: *BLACK*, *GRAY* and d(u) messages. After each node knows about its two hop away neighborhood, all nodes broadcast their r values. A node u can become dominator1, if it wins in the weight comparison. Then, node *u* turns black and broadcasts a *BLACK* message in the neighborhood. If a white node v receives *BLACK* message from its neighbor *u*, so *v* becomes gray and broadcasts *GRAY* message. This *GRAY* message includes the pair (*v'* s id, *u'* s id). If a black node w receives *GRAY* message from a gray node *v* and also the id of another black node u, and if *w* and *u* are not connected yet, then *v* becomes dominator2 node to connect *u* and *w*. In that case, after receiving a *BLACK* message from a node *w*, if a gray node *u* has already received a notification so that there is a 2-hop away black neighbor *v* sent through a neighbor *x* and *v* has not been connected to *w* yet, then both *u* and *x* become dominator2 nodes to connect node *v* and node *w*. [10]

An algorithm was provided to find MCDS in UDG. It is based on the computation of convex hulls of sensor nodes. Also, it describes an algorithm to find MCDS from a CDS. This CDS is found via algorithm described in [11]. They have to do following steps: [14]

- Select a minimum degree vertex u from the CDS.
- Calculate *CH(N[u])*.
- Calculate *CH(N[i]), i ϵ N(u)*.
- Check if *CH(N[u])* is contained in *UCH (N[i])* where *i ϵ N(u)*.
- If step 2 returns true then remove vertex u and go to 1).
- Otherwise do not remove vertex *u* and go to step 1.
- Algorithm terminates when all the nodes in *C* are considered and the node remains in *C* construct the MCDS.

Stojmenovic et al. [16] According to the context of clustering and broadcasting, presented three synchronized distributed constructions of CDS. In all of these, the CDS includes of two kinds of nodes: the cluster-heads and the border-nodes. The cluster-heads form a MIS. If a node is not a cluster-head and there are at least two cluster-heads within its 2-hop neighborhood, then it is a border-node. The set of cluster-heads is extracted through three rankings such as: the id only, an ordered pair of degree and id, and an order pair of degree and location.

The selection of the cluster-heads is given via a synchronized distributed algorithm, which can be generalized to the following framework. Initially all nodes are colored white. In each stage of the synchronized distributed algorithm, all white nodes which have the lowest rank among all white neighbors are colored black. Then all white nodes adjacent to these black nodes are colored gray. Finally, the ranks of the remaining white nodes are updated. When all nodes are colored either black or gray, the algorithm ends. All black nodes form the cluster-heads. Algorithms have $O(n^2)$ message complexity and $\Omega(n)$ time complexity.

IV. Comparison of Some Algorithms

We have surveyed some well-known backbone formation algorithms in term of time and message complexity. Performance comparison of more algorithms is shown in below table. We can see that proposed algorithms in [3], [20] have the less time and massage complexity among other algorithms in this table.

Proposed algorithms in [9]-I, [9]-II result in a CDS of size at most $2(1+H\ (\Delta)).|OPT|$ and $(3+ln\ (\Delta)).|OPT|$, where H is the harmonic function, and OPT refers to a MCDS. Also [20] results in a CDS of size at most $(logn).|OPT|$.

TABLE I. PERFORMANCE COMPARISON

Ref.	Performance comparison		
	Approximation factor	*Time complexity*	*Message complexity*
[2]	-	$O(\Delta)$	$O(n\Delta^2)$
[3]	*8 opt +1*	$O(n)$	$O(nlog(n))$
[8]-I	$2H(\Delta) + 1$	$O((n+|C|)\ \Delta)$	$O((n|C|+m+nlog(n))$
[8]-II	$2H(\Delta)$	$O(|C|\ (\Delta+|C|))$	$O(n|C|)$
[16]	n	$\Omega(n)$	$O(n^2)$
[19]	$O(n)$	$O\ (\Delta^3)$	$\Theta(m)$
[20]-I	-	$O(n^2)$	-
[20]-II	-	$O(n^2)$	$O(n)$

(n and m are the number of vertices and edges respectively, *opt* is the size of MCDS, Δ is the maximum degree, $|C|$ is the size of the computed CDS.)

V. Conclusion and Future Works

The CDS have proven to be an effective construct within which to solve a variety of problems that arise in WSNs. In this paper, we classified CDS formation algorithms and a few instances of these classifications. Also, we have surveyed some well-known backbone formation algorithms in term of time and message complexity. Significant attention has been paid to CDS formation algorithms yielding a large number of publications. A backbone reduces the communication overhead, increases the bandwidth efficiency, decreases the overall energy consumption, and, at last, increases network effective lifetime in a WSN. The important issue that we can be reached is selection algorithm according to our use.

Acknowledgment

We would like to thank the reviewers who helped us to improve the quality of the current paper.

References

[1] T. Acharya, S. Chattopadhyay, and R. Roy, "Energy-Aware Virtual Backbone Tree for Efficient Routing in Wireless Sensor Networks," in Proc. of Int. Conf. on Networking and Services, (ICNS '07), IEEE, pp. 96-102, Athens, Greece, June 19, 2007.

[2] J. Akbari Torkestani, M. R. Meybodi, "An intelligent backbone formation algorithm for wireless ad networks based on distributed learning automata," Computer Networks 54, pp. 826–843, 2010.

[3] K. M. Alzoubi, P. J. Wan, and O. Frieder, "New Distributed Algorithm Connected Dominating Set in Wireless Ad hoc Networks," Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), IEEE, Vol. 9, pp. 297-304, 7 January 2002.

[4] R. Azarderakhsh, A. H. Jahangir, and M. Keshtgary, "A New Virtual Backbone for Wireless Ad-hoc Sensor Network with Connected Dominating Set," Third Annual Conference on Wireless On-demand Network Systems and Services (WONS), pp. 191-195, 2006.

[5] S. Butenko, X. Cheng, and Carlos A. S Oliveira, and P. M. Pardalos, "A New Heuristic For The Minimum Connected Dominating Set Problem On Ad Hoc Wireless Networks," Recent Developments in Cooperative Control and Optimization, pp. 61-73, Kluwer Academic Publishers, 2004.

[6] I. Cidon, O. Mokryn, "Propagation and Leader Election in Multihop Broadcast Environment," Proc. 12th Int. symp. Disrt. Computing, pp. 104 – 119, Greece, September 1998.

[7] X. Cheng, M. Ding, and D. Chen, "An approximation algorithm for connected dominating set in ad hoc networks," Proc. of International Workshop on Theoretical Aspects of Wireless Ad Hoc, Sensor, and Peer-to-Peer Networks (TAWN), 2004.

[8] B. Das, V. Bharghavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," Proc. of IEEE Conf. Communications (ICC 97), Vol. 1, p. 376-380, Montreal, Canada , June 1997.

[9] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets,“ Algorithmica, 20(4), pp. 374-387, April 1998.

[10] S. Hussain, M. I. Shafique, and L. T. Yang, "Constructing a CDS-Based Network Backbone for Energy Efficiency in Industrial Wireless Sensor Network," In Proceedings of HPCC, pp.322-328, 2010.

[11] K. Islam, S. G. Akl, and H. Meiher, "A constant Factor Localized Algorithm for Computing Connected Dominating Sets in Wireless Sensor Networks," Proc of 14th IEEE International Conference on Parallel and Distributed Systems, (ICPADS), pp. 559-566, Melbourne, VIC, December 2008.

[12] B. Jeremy, D. Min, and T. Andrew and C. Xiuzhen, "Connected Dominating Set in Sensor Networks and MANETs," Handbook of Combinatorial Optimization, Springer, US, 2004.

[13] Z. Liu, B. Wang, and Q. Tang, "Approximation Two Independent Sets Based Connected Dominating Set Construction Algorithm for Wireless Sensor Networks," Inform. Technol. J., Vol. 9, Issue 5, pp. 864-876, 2010.

[14] G.N. Purohit, U. Sharma, "Constructing Minimum Connected Dominating Set: Algorithmic approach," International journal on applications of graph theory in wireless ad hoc networks and sensor networks (GRAPH-HOC) Vol. 2, No. 3, September 2010.

[15] M. Rai, Sh. Verma, and Sh. Tapaswi, "A Power Aware Minimum Connected Dominating Set for Wireless Sensor Networks," Journal of networks, Vol. 4, no. 6, August 2009.

[16] I. Stojmenovic, M. Seddigh, J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," IEEE Transaction on Parallel and Distributed Systems, Vol. 12, No. 12, pp. 14 – 25, December 2001.

[17] M.T. Thai, W. Feng, and L. Dan, and Z. Shiwei and D. Ding-Zhu, "Connected dominating sets in wireless networks with different transmission ranges," IEEE Trans. Mobile Comput. , Vol. 6, pp. 721-730, 2007.

[18] P.J. Wan, K.M. Alzoubi and O. Frieder, "Distributed construction of connected dominating set in wireless ad hoc networks," Proc. of IEEE Conf. Computer and Communications Societies, pp. 1597-1604, New York, June 23-27, 2002.

[19] J. Wu, H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," Proc. of ACMDIALM'1999, pp. 7–14, August 1999.

[20] R. Xie, D. Qi1, and Y. Li, and J. Z. Wang, "A novel distributed MCDS approximation algorithm for wireless sensor networks," Mobile & Wireless Communications, Vol. 9, Issue 3, pp. 427–437, March 2009.