Received: January 31, 2020. Revised: March 5, 2020.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

The Novel Stick Clustering Algorithm and its Applications for Ad-hoc Networks

Abdulrahman Sabra[1]* **Mohamed Taha**[2] **Dina S. M. Hassan**[3]

[1]*Computer Science Department, Modern Academy, Egypt*
[2]*Faculty of Computers and Artificial Intelligence, Benha University, Egypt*
[3]*Faculty of Computers and Information, Princess Nourah Bint Abdulrahman University, Saud Arabia*
* Corresponding author's Email: abdosabra80@gmail.com

Abstract: Clustering is considered a critical process to make a fast and reliable connection establishment of ad-hoc networks. In this paper, a new clustering algorithm is proposed, namely Stick Clustering Algorithm (SCA). The proposed SCA is used to overcome the low connectivity, low energy and delay to enhance the performance of large-scale ad-hoc networks. The simulation results showed the efficiency and the reliability of the proposed clustering algorithm against many greatest challenges such as high intensity of nodes under small network area. Also, the results demonstrated that the proposed SCA is very competitive algorithm compared with RCA, CDS, Z-Degree and ZID. The proposed SCA obtained 3.4306 approximation ratio. Also, the efficiency of the SCA is of O(n) for both message complexities and time.

Keywords: Flying ad-hoc networks, Vehicular ad-hoc networks, Mobile ad-hoc networks, Connected dominating set, Balanced energy efficient virtual backbone construction algorithm.

1. Introduction

Ad hoc networks are a set of nodes connected without any infrastructure based on the transmission range of each node in the network. Ad-hoc networks include wireless ad-hoc networks, mobile Ad-hoc networks, vehicular ad-hoc networks (VANETs), and flying ad-hoc networks (FANETs). Ad-Hoc networks considered one of the important technologies in daily life. For example, Vehicular ad hoc networks can be exploited to minimize the number of accidents by providing information about the road. VANETs can be used to alert vehicles breaking regardless of vision clarity. Moreover, they can help in conveying injures as fast as possible, as mentioned in [1, 2].

The clustering of ad-hoc networks is one of the most important topics that aim to enhance the efficiency of ad-hoc networks. Clustering is the process of selecting a set of nodes called cluster heads, each one is responsible in part of the network called cluster. Clustering considered one of the best solutions for ad-hoc networks challenges, for example, increasing network size will lead to many problems such as overhead, delay, and low energy rate of nodes, as mentioned in [3]. These problems may cause a lot of issues during implementing routing protocols and channel allocation. For this reason, particular nodes are selected according to different factors such as node degree, load balancing, and energy of each node. Then routing protocols are implemented through these nodes, as mentioned [4–6] and apply channel allocation of the cluster head node as mentioned [7].

In this paper, a new clustering algorithm is proposed to overcome the main problems associated with ad hoc networks. These problems are low connectivity and delay of node. The new novel stick clustering algorithm (SCA) clustering algorithm consists of three phases: stick formation phase, member joining phase, and connector selection phase. The proposed algorithm handles the problem related to low connectivity through the criteria of selecting any two nodes which represent the clusters heads is based on the highest number of neighbours without repetition. For the delay issue, the proposed SCA broadcast the packet to only nearest cluster head, where each cluster head has information related to their neighbour clusters head. The experimental results showed that our proposed clustering algorithm has the best CDS size compared with previously proposed algorithms. Also, another way to validate the efficiency of the proposed algorithm, a mathematical proof has been discussed.

In the stick formation phase, every stick consists of two nodes called stick nodes (clusters head); the stick is formed if the two nodes have the highest number of neighbors with non-repetition. In other words, the two nodes have a minimum intersection area as long as possible. In the member joining phase, all one-hop neighboring of stick nodes join stick as a member. Then stick nodes choose winners from members to form new sticks. In the connector phase, getaway nodes are selected from the stick members. The selected getaway nodes fall on one-hop of two or more sticks of different areas.

The main contribution of the paper can be summarized as follows:

1. A new clustering algorithm called as stick Clustering Algorithm (SCA) is proposed.
2. The proposed SCA is applied for ad hoc networks.
3. The performance of the proposed SCA is compared with other well-known clustering algorithms in the same environment;
4. Different network nodes density with different transmissions ranges are considered to prove the reliability of the proposed SCA algorithm.
5. Several evaluation measurements are employed to assess the performance of the proposed SCA clustering algorithm.

The rest of the paper is organized as follows; Section (2) is briefly discussed the previous studies, Section (3) discusses the proposed clustering algorithm in details. In Section (4), introduces and discuss the performance analysis of the proposed algorithm. The experimental results are discussed in section (5). In section (6), a detailed discussion of the strength and the weakness of the proposed SCA. Finally, the conclusions and future work are introduced in Section (7).

2. Related work

There are many clustering algorithms previously proposed in literature.

In [8], the authors classified clustering techniques like the following backbone optimization, load balancing and energy efficiency, multi novel, and vehicular ad hoc networks. Backbone optimization is a subset of connected nodes that forms the backbone of the network. Connected dominated set (CDS) has two nodes types, independent nodes, and connectors or getaway nodes. Finding the minimum of CDS considered as NP-hard problem (nondeterministic polynomial time) problem [9-10]. In [11], the authors proved that the size of the Maximum Independent Set (MIS) $\leq$ 4MCDS + 1.In [12] the MIS $\leq$ 5.146MCDS + 7.228.In [13], the MIS $\leq$ 3.8 MCDS + 1.2. while the best relation in [14], the (MIS) $\leq$ 3.4306 MCDS + 4.8185.The approximation ratio used to define the size of CDS over MIS. In [15], the authors proved that the upper bound of the approximation ratio has small probabilities of being large, while in [15], they create the rule-k algorithm that using node ID to assign the CDS. In [16], the authors have evolved the rule-k algorithm, which uses the nodes' degrees to assign the CDS. In the Zone clustering algorithm [17], the authors use various factors such that nodes ID, energy level, and Degree to select dominator.

In [18], the Connected Dominated Sets-Bounded Diameters-Distributed (CDS-BD-D) has two phases. The first phase applies a distributed depth-first search algorithm. In the second phase, it selects dominators and connectors. In [11], the WAF is proposed and has two phases. The MIS is constructed in the first phase, while the dominating tree constructed in the second phase. The same methodology of DA is used in WWY [19]. But in phase two of the WWY connector is selected more economically. In [12], the RCA clustering algorithm consists of three phases ring-formation phase, members-joining phase, and CDS-nodes selection phase. In the first phase, every three nodes with the highest number of neighbors with non-repetition formed a ring and called ring nodes. A member joining phase all one-hop neighbors that cannot form a ring be a member of the formed ring. Rings represent the CDS.

In [20], a new algorithm based on ant colony optimization for the minimum connected dominating set problem is proposed, the proposed algorithm reduces variable neighborhood search. The pheromone value changed according to the best solution on each iteration and the best so far solution. In [21], two algorithms are proposed for solving the MCDS problem. The first one is a hybrid algorithm based on genetic algorithm to local search to get higher performance. The second one is called Simulated Annealing used to enhance a stochastic local search and ability to escape from local solutions. In [22] a balanced energy efficient virtual backbone algorithm is proposed based on the genetic algorithm. Calculating fitness function on each iteration based on a set of factors such as energy efficiency, reliability, data rate, and coverage of neighbors. In [23], the proposed algorithm consists of two phases, in first phase the dominating set is constructed based on link and degree.in the second phase, the connected tree algorithm is used to connect all nodes in the network.

3. The proposed algorithm

This paper presents a new clustering algorithm called as Stick Clustering Algorithm (SCA). The target of the proposed SCA algorithm is to find the minimum number of nodes to construct a maximum independent set by creating sticks where each stick consists of two nodes called stick nodes. Nodes that have the lowest priority and cannot form a stick can join the network as a stick member. nodes in the SCA can be an ordinary node, probable stick node, stick member node, getaway node and winner node, but some of these statuses are temporal. At the end of the execution, nodes will be one of three: a stick node, a stick member, or a connector.

The network represented as a Unit Disk Graph (UDG) IEEE 802.11b. All nodes have the same transmission range. There is a unique ID for each node. Each node knows the IDs of its one-hop neighbouring nodes. The proposed SCA algorithm consists of three phases: stick formation phase, member joining phase, and connector phase.

3.1 Stick formation phase

Each node in the network starts as an ordinary node. To form a stick, two nodes are required. The first one tries to create a stick while the second is one of the first node one-hop neighboring. Each node can form a stick broadcast valid-stick message to all one-hop neighbouring suggested forming a stick. The broadcasted message (valid-stick) contains the partner ID, and the total number of neighbors stripped out from the shared neighbors between them. For example, if node 1 has four neighbors 3, 4, 5, and 6, and node 2 has three neighbors 5, 6, and 7. Then the stick-degree is 3 (3,4,7). The broadcasted message also contains the IDs of all one-hop neighboring of the broadcasted node arranged in ascending order. The broadcasted node initializes timer T. All neighboring nodes respond according to their orders in the broadcasted message to avoid collision. If the broadcasted node does not receive all reply-message during time T, it will rebroadcast valid-stick message contains the IDs of unreceived nodes.

When neighbors received a valid-stick message, they compare the total number of neighbors of the probable stick node with their suggested solutions. If the total number of neighbors of a probable stick node is higher than their solution. Then, they broadcasted reply-message with acceptance. Otherwise, the node that has a better-suggested solution broadcasts a valid-stick message containing its suggested solution and waiting for the accept message from its one-hop neighboring. When a probable stick node received the reply-message with acceptance from all neighbors. It broadcasts the correct-formed-stick message to inform its neighbors about its election as a stick-node (CH) with its partner node.

In this case, if two sticks have the same degree. Then, sum the number of neighbors for every two partners, and the highest degree is selected. If the tie still stands, then select the probable stick-node that has the lowest ID to break the tie.

3.2 Member joining phase

The ordinary nodes still ordinary nodes until they received correct-formed-stick messages from stick-nodes. Upon receiving the broadcasted message, the first step is repeated in the same manner for all ordinary nodes. This is done by removing elected stick nodes from the list of each ordinary nodes. Ordinary nodes start selecting a new partner from its neighbours list based on the highest number of neighbors stripped out from the shared neighbors with the selected node. The ordinary nodes join the stick and change its status from ordinary node to stick-member node. If the ordinary node has two or more stick in its one hop, then ordinary node will select the stick that has the highest degree as a stick leader. If the tie still stands, the stick node that has the lowest ID is selected. Then a stick-member message is broadcasted from ordinary nodes to join the stick. The message declares the id of the selected stick accompanied by ordinary nodes solution (the selected partner and degree).

The stick nodes wait for the stick-member message from their neighbors. Upon receive the broadcasted message from all nodes, the stick node selects the best solutions (new winners) from its received messages to form the new sticks. The stick node may select one or more winners from its members, but the new winners of the same stick node must not be a neighbors. A valid-winner message is broadcasted containing the new winners arranged in ascending order. The stick nodes initialize temporary time T(w). If the winner nodes do not respond by Winner-Accept message during this time, the stick node will rebroadcast a valid-winner message containing unreceived winners ID.

The winner nodes respond by a broadcasting winner-accept message containing the partner ID. When the partner received the winner-accept message, it checks its status.

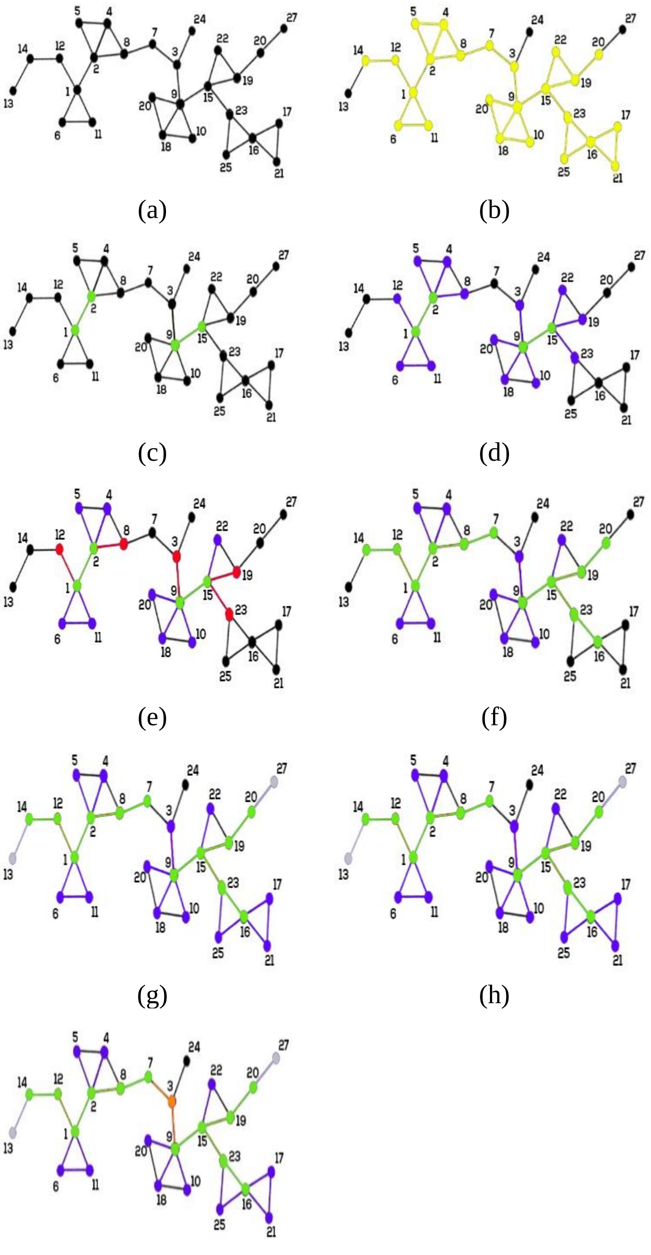
- If the partner is a stick member at any stick, then the partner responds with a reply-message with reject. When the winner node receives the reply message, it will select another partner in the same manner from its one-hop neighboring. The winner node broadcasts a winner-accept message containing the new ID of the selected partner. If the same situation still stands with all one-hop neighboring of the winner node. Then, the winner node converts its status from winner to stick member.
- If the partner still ordinary node, then it will broadcast a reply-message with acceptance. When the winner node receives a reply-message, it broadcasts a correct-formed-stick message. Then the winner node changes its status to stick node and repeats the second phase with its partner node.

3.3 Getaway phase

In this phase, the getaway/connector nodes are selected to connect different areas. Each member node receives two or more correct-formed-message from two or more different sticks/areas convert its status from stick member to probable getaway node. The probable getaway node broadcasts a message that suggests through it to be a getaway node accompanied by the list of the suggested stick nodes. When the message is received, the stick nodes broadcast a confirm or a reject message. The confirmation message will be broadcasted, if there is no connection with the suggested stick nodes. A reject message will be broadcasted if already there is a connection with the suggested stick nodes. If the probable getaway node receives at least one confirmation message, then it broadcasts a correct-formed-connector message advertise through it that has been elected as a gateway node. The probable getaway node initializes a timer T(c). If the stick nodes do not respond by connector-confirm message during this time, then the gateway node will rebroadcast the correct-formed-connector message contains unreceived stick nodes IDs.

3.4 Illustrative example

All nodes start in an ordinary state, as shown in Fig. 1 (a). For every node that has the number of neighbors greater than one, the total number of non-repeated neighbors with its one-hop neighboring is calculated. Hence nodes 24,13 and 27 are excluded from the calculations. Each node selects the best partner node with the highest number of non-repeated nodes, as shown in Table 1. In Fig. 1 (b) each node can form a stick broadcast a valid-stick message suggests forming a stick. In Fig. 1 (c), nodes 1 and 9 with their partner nodes win the computation because they are having the highest priority, as shown in Table 1. Then nodes 1, 2, 9, and 15 change their status to stick nodes. They start



(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i)

Figure. 1 (a) Nodes colored in black are in ordinary state, (b) yellow nodes are probable stick nodes and have valid stick, (c) green nodes are stick nodes, (d) blue nodes are stick member connected with a blue link with a stick node, (e) red nodes represent winner winner nodes that selected by every stick node, (f) winner nodes succeeded to form new sticks with their partner and change their colors to green, and (g) orange node is a gateway node between two different sticks from different areas

Table 1. The best partner selection of each node

ID	partner	Non- repeated nodes	Degree
1	2	(8,4,5,6,11)	5
2	1	(8,4,5,6,1)	5
3	9	(10,15,18,20,24)	5
4	8	(5,7)	2
5	2	(8)	1
6	1	(2,12)	2
7	3	(8,9,24)	3
8	7	(2,3,4)	3
9	15	(3,10,18,20,19,22,23)	7
10	9	(3,15,20)	3
11	1	(2)	1
12	1	(2,11,14)	3
13	14	(12)	1
14	12	(1,13)	2
15	9	(19,22,23,3,10,18,20)	7
16	23	(17,21,15)	3
17	16	(23)	1
18	9	(3,15)	2
19	15	(9,23)	2
20	9	(3,10,15)	3
21	16	(23)	1
22	15	(9,23)	2
23	15	(16,25,9,19,22)	5
24	3	(7,9)	2
25	23	15	1
26	19	27,22,15	3
27	26	19	1

to broadcast correct-formed-stick to their one-hop neighboring. All one-hop neighboring of the stick nodes update their neighboring list and select new partner after removing stick nodes from their neighboring list. Then All one-hop neighboring of stick nodes change their status to stick member as illustrated in Fig. 1 (d). Then every stick-member node broadcast a message to stick-nodes. The message contains their choices, degree, and the selected area. Nodes 6, 11, 5, 4, 19, 22, 20, 18, and 10 send their choice null and degree 0 because they do not have one node that is not one-hop neighboring to stick nodes.

In Fig. 1 (e), stick nodes, select the best nodes as winners according to their degrees, as shown in Table 2. Then stick-nodes broadcast valid-winner messages, as mentioned in phase 2. Then Winner nodes broadcast winner-accept messages to inform their partner about their selection. Node 7 receives two requests from winner nodes 3 and 8. Node 7 will respond to the first received request. Assume that the winner (node 7) accepts the message from node 8 before node 3. Node 7 will neglect the winner-accept message of node 3 and will broadcast reply-message containing the acceptance of being a partner to node 8. When node 3 listens to the broadcasted message, it will check its neighbor eligibility to select a new partner. However, there is no valid node to form a stick with node 3. Then node 3 will change its status from winner to stick member. In Fig. 1 (f), the stick nodes (12,14), (8,7), and (23,16) broadcast a correct-formed message. The previous steps will be repeated until all nodes determine its status as a stick node or stick member. Fig. 1 (g) shows the third phase in which node 3 receives two correct-formed-stick messages from two different areas. Hence node 3 broadcast a valid-connector message to stick nodes 7 and 15. When node 3 receiving at least one reply-message with acceptance from stick nodes. Then node 3 will broadcast the correct-connector-formed message and change its status from stick member node to a connector/gateway node.

Table 2. The winner nodes of each stick node

Stick nodes	Winners	Partners
1	12	14
2	8	7
9	3	7
15	19,23	26,16

Algorithm 1: sticks creation

```
foreach node N ∈ network and $C_N$>1
        Where $C_N$ is the total number of neighbors.
    Select highest priority node U.
        Probable-Stick-node(N)=true
        broadcast(valid-Stick(N))
foreach one hop 181eighbour I of N, then
    Upon receive (Valid stick(N)) do
        If priority(N) ≥ priority(I), then
            Probable-Stick-node(I)=false
            broadcast(reply-message(accept))
        Else
            Probable-Stick-node(I)=true
            broadcast (Valid-Stick(I))
Upon Receive (reply-message (I)) do
        RM + = 1. Where RM is the received messages.
        If RM = $C_N$
            RegionID=N
            broadcast(correct-formed-stick(N)).
            broadcast (correct-formed-stick (U))
```

Algorithm 2: advertise neighbors and winner selection(second phase)

```
for each one-hop node I of Stick Node N
    upon receive(correct-formed-stick(N)) do
        nodes update their neighbors list stripped out from stick-Nodes
```

```
            Select highest priority node.
            broadcast(stick-member(I))
Upon receive(Stick-member(I)) in StickNode N do
    RM += 1.
    If RM = $C_{(N)}$.
        Stick node select new winners.
        broadcast(valid-winner(N))
Upon winner node N receive (valid-winner(N)).
    Node N advertise partner I.
    If status(I)=ordinary node.
        broadcast(reply-message(accept)).
    Else
        broadcast(reply-message(reject))
If RM= $C_N$-1
    broadcast (correct-formed-stick (N))
```

Algorithm 3: Getaway selection

```
for each stick member N
    if Node N fall between more than one stick node, then
        Probable-connector(N)
        update list of $CN_{(N)}$ where $CN_{(N)}$ is a list of Stick Node Neighbors.
        broadcast(valid-connector($CN_{(N)}$))
Upon stick node I receive(valid-connector($CN_{(N)}$)) .
    If $CN_{(I)} \notin CN_{(N)}$.
        Broadcast(reply-message(accept))
    Else
        Broadcast(reply-message(reject))
Upon probable connector N receive(reply-message(accept))
    Connector(N)=true
    broadcast(correct-formed-connector(N)).
```

4. Performance analysis

4.1 SCA construct CDS

Theorem 1. The virtual path constructed using stick is a CDS.

Proof. At the end of the algorithm, after the third phase, nodes classified into three types: stick nodes, stick members, connectors.

- For any two adjacent nodes with the highest number of nodes are called stick nodes.
- Any node receives an advertising message from a stick node is called a stick member, if not selected in the second phase as a new winner.
- Any node selected as a new winner in the second phase, there is a possibility to be a stick node.
- Any node falls between two or more different areas; there is a possibility to be a connector.
- Then repeat the previous steps until all nodes in the network determine their status.

Hence, the CDS is constructed.

4.2 Stick approximation ratio

In [24], the authors prove that for any disk with a radius of two measuring units (meters, kilometers, millimeters, feet, inches, yards, miles) the maximum number of non-overlapping circles in one-unit diameter is 21, and the maximum number of non-overlapping circles/ independent nodes for a circular area of three units is 43.

In UDG, the distance between two nodes with non-overlapping circles is at least two hops. To connect two independent nodes, and in worst case two nodes are needed. Hence, the number of independent nodes equals the total number of connectors. Hence, the size of the CDS equal twice the number of independent nodes.

This section proves that the maximum number of sticks equals the maximum number of independent nodes. In SCA after executing the third phase, all sticks are included in the CDS. Each stick consists of two nodes. Hence, the CDS size produced by SCA equal the CDS size produced by the maximum number of independent nodes.

Authors proved the best known relation in [14] between the size MCDS and the size of MIS of connected UDG.

$(MIS) \leq 3.4306\ MCDS + 4.8185$.

Then the size produced by SCA equals $1(3.4306\ MCDS + 4.8185)$.

Theorem 2. The CDS produced by SCA has size at most 3.4306 MCDS + 4.8185.

Proof. Theorem 2 is correct if Lemma 1 is proper.

Lemma 1. The maximum number of sticks produced by the SCA algorithm equals the maximal number of independent nodes for any disk area.

Proof. Assume that the two adjacent nodes (a, b) together have the highest number of neighbors with non-repetition in other word nodes a and b have the minimal intersection area. Then node a and b form the first stick at the beginning of the simulation, as shown in Fig. 2. Then node (a) update its list and select new winners to form new sticks, as explained in section 3.2. New winners cannot be from the intersection area, and it should be farthest as long as possible to cover the largest number of nodes. Must be from the highlighted area as shown in Fig. 2.

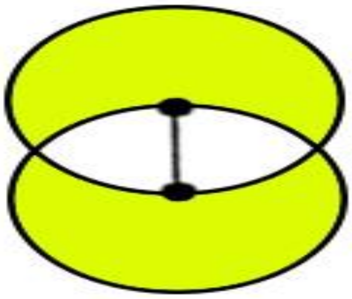


Figure. 2 The selected region of new winners

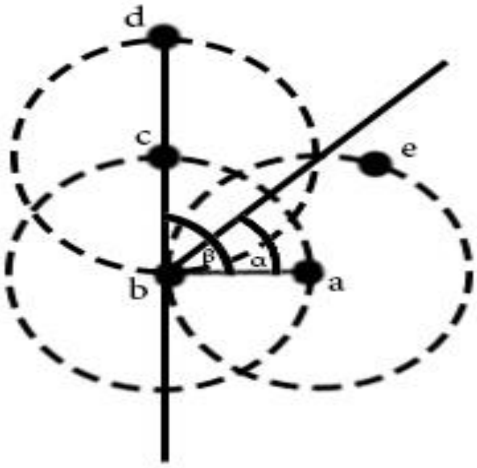


Figure. 3 The selection of new winners in the second phase

Assume that node b selects node C as a winner with its partner d to form a new stick, as shown in Fig. 3 <α̇ is 60_o hence in order for node c to be selected as a winner, it must be <B > <α̇, so node c selected as a winner. Then node c completes the progress as mentioned in section 2 and advertises its one-hop neighboring about its election as stick node with its partner. Same situation for winner node e that selected by stick node a.

Every stick node can select more than one winner in the same manner, but the selected winner must not be adjacent neighbors, which result in having seven sticks for stick node a and node b as shown in Fig. 4.

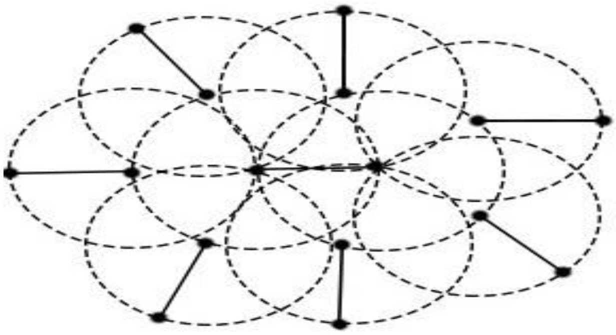


Figure. 4 Represent new winners and their partners to form new sticks

Proof of Lemma 1. By extending Preliminary1, for a disk with a radius of two or more units:

- As shown in Fig. 5, disks A, B have 4 πr and 8 πr circumferences, where r is a transmission range.
- Disk a covered by 7 sticks where disk b, which equals twice the circumference of A covered by 14, which means each increase in circumference needs seven sticks; hence the total number of sticks that needed to cover three units equal 43 sticks.

$$1 + \sum_{i=1}^{3} = 1 + 7 + 14 + 21 = 43 \text{ sticks}$$

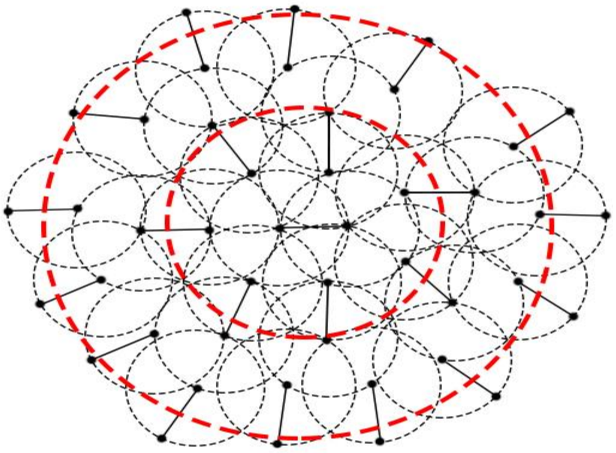


Figure. 5 The total number of sticks for a disk with two units

4.3 SCA complexity

This section evaluates the message complexity by calculating the maximum number of messages sent by any node in the network. In the first phase, all nodes that have a valid stick start to broadcast a valid-stick message suggests forming a stick. Then nodes that have the lowest priority (ordinary nodes) reply by an apply-message accept the suggested stick.

In the second phase, when the probable stick nodes receive an apply-message from all their one-hop neighbors, they change their status from probable stick nodes to the stick nodes. Stick nodes start to broadcast correct-formed-stick messages to advertise their one-hop neighboring that the new sticks have been formed and ask about the members. When one-hop neighboring nodes receive correct-formed-stick respond by stick-member message contains their choices as mentioned in second phase. Then stick nodes select new winners and broadcast valid-winner messages. When the winner receives a valid-winner message, respond by a winner-accept

Received: January 31, 2020. Revised: March 5, 2020.

message. When the partner of the selected winner receives a winner-accept message check its status if not join any stick area, then reply by apply-message with approval and start with its partner(winner) to repeat the second phase but as a stick node.

In the third phase, any stick member node received correct-formed-stick from two or more different areas broadcast a valid-connector message. When a stick node receives a valid-connector message checks its paths and then responds with reply-message contain its respond whether with approval or reject. When a stick-member (probable connector) node receive at least one apply-message message with approval. Then probable-connector broadcast a correct-formed-connector message that advertises all stick nodes that has been elected as a connector node and wait from stick-nodes a connector-confirm message to make sure that all adjacent stick nodes have been informed by-election.

Table 3 shows a comparison between the Zone, CDS-BDD, DA, WWY, RCA, and Stick algorithms for message and time complexities and approximation ratios. It shows that CDS-BD-D has the worst time complexity, while stick and RCA have message complexity better than message complexity of DA, WWY, and CDS-BD-D algorithms. As yet, stick gives the best approximation ratio.

Table 3. Performance analysis

Algorithm	Time complexity	Message complexity	Approximation ratio
WAF[11]	O(n)	O(n log n)	6.862
WWY[19]	O(n)	O(n log n)	6.075
Zone[17]	O(n)	O(n)	6.862
CDS-BD-D [18]	O($n^{1.6}$ + Diam)	O($n^{1.6}$ + E + Dn)	6.862
RCA[12]	O(n)	O(n)	5.146
Proposed	O(n)	O(n)	3.4306

5. Simulation and results

In order to evaluate the performance of the proposed SCA algorithm, several experiments have been conducted for all the compared algorithms SCA, RCA, CDS-BD-D, zone-min-ID, zone-max-ID and BEE-VBC in the same environment. Most of compared algorithms are chosen because have the lowest approximation ratios.

In this section, the Stick algorithm has been analyzed using java simulator that used in [15], 200 X 200 m network dimension was selected; also, three transmission ranges were selected, and they are 20m, 50m and 100m, goal was to test the performance of the algorithm under small network area, different transmission ranges and a large number of nodes. The number of nodes starts from 50 to 600.and point of the curve is the average of ten runs, because of the different random distribution of nodes generated each time simulator run.

5.1 CDS size

Fig. 6 shows the total number of nodes that formed the CDS which generated by SCA, RCA, CDS-BD-D, zone-min-ID, zone-max-ID and BEE-VBC algorithms. In Fig. 6 (a), The number of nodes starts from 200 and increased 100 nodes in each step with a 20m transmission range. The Experimental results show that the SCA has the best CDS size while zone-min-ID has the highest number of nodes in the CDS. At low nodes population; CDS-BD-D algorithms have a few progresses than SCA, which have little fewer CDS size than SCA. In Fig. 6 (b), under 50m transmission range, SCA has the best CDS size followed by RCA while zone-min-ID has the highest number of nodes in the CDS. In Fig. 6 (c), under 100m transmission ranges and the number of nodes starts from 50 and increased 50 nodes in each step. RCA has the smallest CDS size followed by SCA while and BEE-VBC has the largest CDS size.

5.2 Retransmission volume

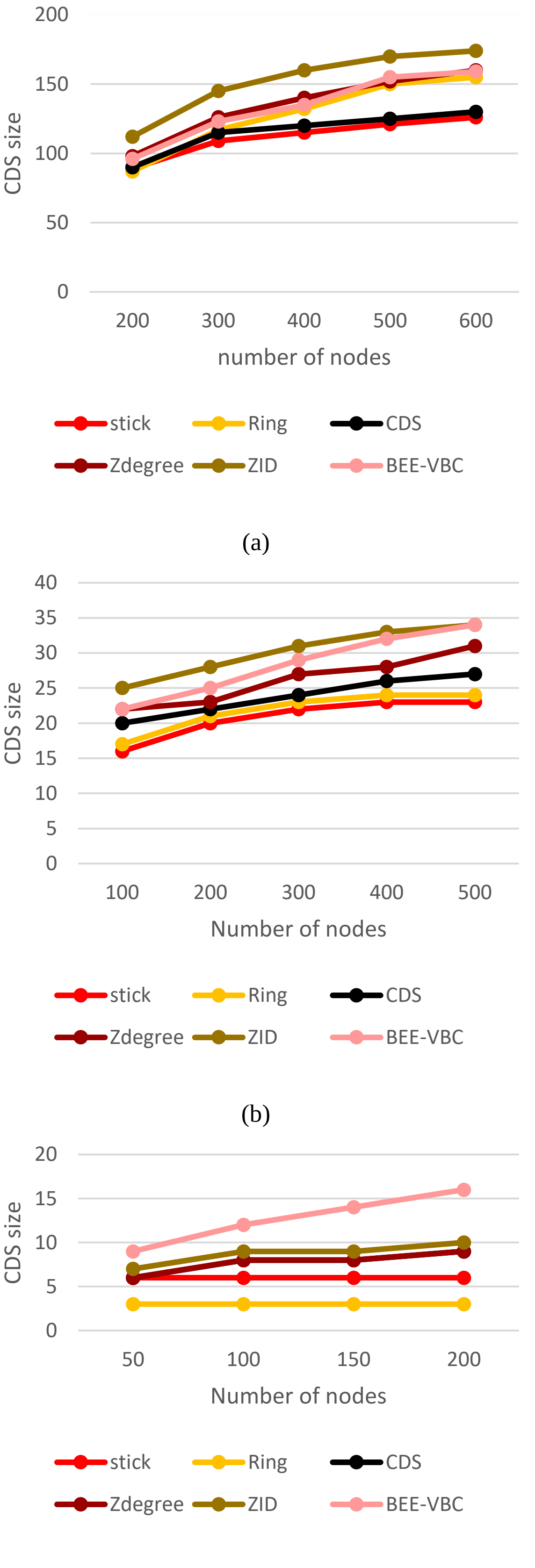
Fig. 7 shows the number of retransmissions between the compared algorithms. In Fig. 7 (a), under 20m transmission range. The RCA has the lowest number of retransmissions, followed by BEE-VBC and zone-max-degree while CDS-BD-D has the highest number of retransmissions. In Fig. 7 (b), under 50 m transmission range RCA has the lowest number of retransmissions followed by SCA, while CDS-BD-D has the highest number of retransmissions. In Fig. 7 (c), under 100 m transmission range zone-max-degree has the lowest number of retransmissions followed by RCA while CDS-BD-D and zone-min-ID have the highest number of retransmissions.

5.3 Completion time

Fig. 8 shows the completion time between the compared algorithms. In Fig. 8 (a), under 20m transmission range zone-min-ID, has the best completion time followed by RCA while CDS-BD-D has the worst completion time. In Fig. 8 (b), under 50m transmission range RCA has the best completion time followed by zone-min-ID while the CDS-BD-D has the worst completion time. In Fig. 8 (c), under 100m transmission range zone-max-

Received: January 31, 2020. Revised: March 5, 2020.

degree, has the best completion time followed by zone-min-ID and RCA while SCA and CDS-BD-D have the worst completion time.

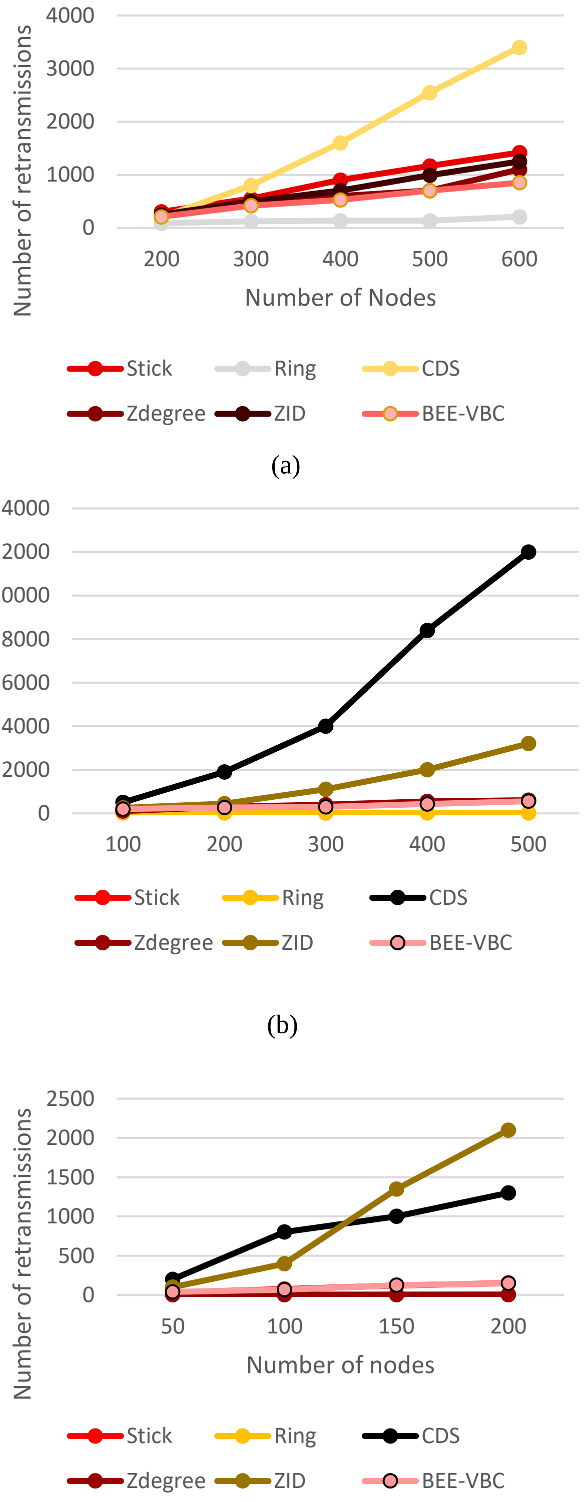


(a)

(b)

(c)

Figure. 6 The number of nodes in the CDS for 200m X 200m network, and the transmission ranges is: (a) 20m, (b) 50m, and (c)100m

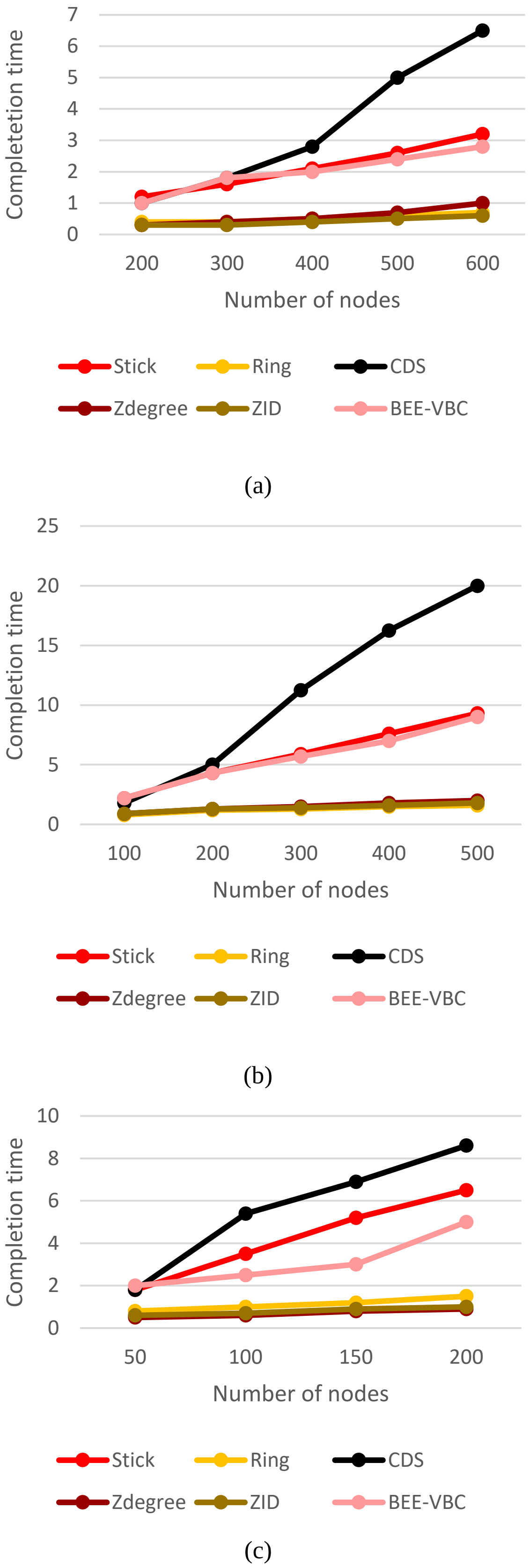


(a)

(b)

(c)

Figure. 7 Total retransmissions when network is 200m X 200m, and the transmission ranges is: (a) 20m, (b) 50m, and (c) 100m



(a)

(b)

(c)

Figure. 8 The Completion when network is 200m X 200m, and the transmission ranges is: (a) 20m, (b) 50m, and (c) 100m

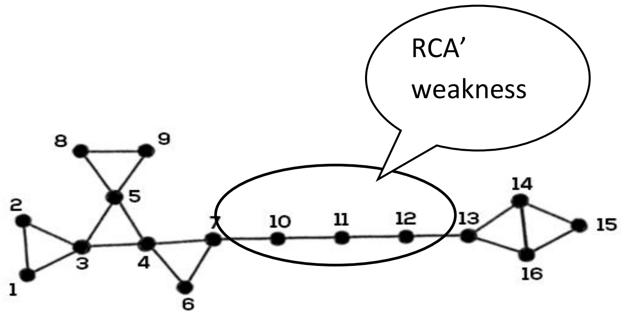


Figure. 9 Illustrative example2

6. Discussion

RCA failed to form a ring when some nodes fall in a straight line in part of the network, as shown in Fig. 9 Hence, some areas will be isolated from the rest of the network. SCA solves the problem because each stick consists of two nodes, so the SCA algorithm reaches any connected area in the network.

Problem of SCA at the beginning of the algorithm in the suggestion state because the number of valid sticks is very large but this problem doesn't exist in RCA because it need three nodes to form ring, hence the probability of three nodes (valid ring) with minimum intersection area to form ring less than the probability of two nodes to form stick, so note the number of retransmission and completion time in a Stick much larger than RCA algorithm.

7. Conclusion and future work

This paper produces a new powerful CDS clustering algorithm called Stick Clustering Algorithm (SCA). In which, every two nodes with the highest number of neighbors with non-repetition form stick. This paper proved that the maximum number of sticks equals the maximum number of independent nodes in the corresponding area. SCA produces a CDS with size at most 3.4306 MCDS + 4.8185.SCA has O(n) for message complexities and time. The Experimental result showed that SCA has the best CDS size, which has the minimum CDS size for 20m and 50m while it has the second-best CDS size for 100m.

According to the analysis of SCA performance, the number of the retransmissions of the SCA algorithm is very large; also, the completion time of SCA relatively large, hence future work includes creating a hybrid clustering algorithm between SCA and RCA. It starts with a ring in the first phase to reduce the number of retransmissions. In the second phase, each node in the ring chooses winners, which in turn are responsible for forming sticks, hence solve the weakness of RCA.

References

[1] L. Rivoirard, M. Wahl, P. Sondi, M. Berbineau, and D. Gruyer, “Chain-Branch-Leaf: A clustering scheme for vehicular networks using only V2V communications”, *Ad Hoc Networks*, Vol. 68, pp.70–84, 2018.

[2] M. Ren, L. Khoukhi, H. Labiod, J. Zhang, and V. Vèque, “A mobility-based scheme for dynamic clustering in vehicular ad-hoc networks (VANETs)”, *Veh. Commun.*, Vol. 9, pp. 233–241, 2017.

[3] R. Assareh, M. Sabaei, A. Khademzadeh, and M. Reshadi, “A Novel Many-Objective Clustering Algorithm in Mobile Ad Hoc Networks”, *Wirel. Pers. Commun.*, Vol. 97, No. 2, pp. 2971–2997, 2017.

[4] K. G. Preetha and A. Unnikrishnan, “Improving the routing performance of mobile ad hoc networks using domination set”, *Procedia Comput. Sci.*, Vol. 46, No. Icict 2014, pp. 1209–1215, 2015.

[5] K. G. Preetha and A. Unnikrishnan, “Enhanced domination set based routing in mobile ad hoc networks with reliable nodes”, *Comput. Electr. Eng.*, Vol. 64, pp. 1339–1351, 2017,

[6] K. H. Li and J. S. Leu, “Weakly connected dominating set-assisted ant-based routing protocol for wireless ad-hoc networks”, *Comput. Electr. Eng.*, Vol. 48, pp. 62–76, 2015.

[7] B. Karaoglu, W. Heinzelman, and S. Member, “Cooperative Load Balancing and Dynamic Channel Allocation for Cluster-Based Mobile Ad Hoc Networks”, *IEEE Transactons on Mobile Computing,*. Vol. 14, No. 5, pp. 951–963, 2015.

[8] P. H. J. C. J. Yu, “A survey of clustering schemes for mobile ad hoc net- works”, *Commun. Surv. Tutorials, IEEE*, Vol. 7, pp. 32–48, 2005.

[9] J. N. Al-Karaki, A. E. Kamal, and R. Ul-Mustafa, “On the optimal clustering in Mobile Ad hoc Networks”, In: *Proc. of IEEE Consum. Commun. Netw. Conf. CCNC*, pp. 71–76, 2004.

[10] T. N. Nguyen and D. T. Huynh, “Connected D-Hop dominating sets in mobile Ad Hoc networks”, In: *Proc. of 2006 4th Int. Symp. Model. Optim. Mobile, Ad Hoc Wirel. Networks, WiOpt*, 2006.

[11] P. J. Wan and K. M. Alzoubi, “Distributed construction of connected dominating set in wireless ad hoc networks”, In: *Proc. of INFOCOM 2002*, pp. 141–149, 2002.

[12] D. S. M. Hassan, H. M. A. Fahmy, and A. M. Bahaa-Eldin, “RCA: Efficient connected dominated clustering algorithm for mobile ad hoc networks”, *Comput. Networks*, Vol. 75, No. Part A, pp. 177–191, 2014.

[13] W. Wu, H. Du, X. Jia, Y. Li, and S. C. H. Huang, “Minimum connected dominating sets and maximal independent sets in unit disk graphs”, *Theor. Comput. Sci.*, Vol. 352, No. 1–3, pp. 1–7, 2006.

[14] M. Li, P. J. Wan, and F. Yao, “Tighter approximation bounds for minimum CDS in unit disk graphs”, *Algorithmica (New York)*, Vol. 61, No. 4, pp. 1000–1021, 2011.

[15] F. Dai and J. Wu, “An extended localized algorithm for connected dominating set formation in ad hoc wireless networks”, *IEEE Trans. Parallel Distrib. Syst.*, Vol. 15, No. 10, pp. 908–920, 2004.

[16] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating sets and neighbor elimination-based broadcasting\nalgorithms in wireless networks”, *IEEE Trans. Parallel Distrib. Syst.*, Vol. 13, No. 1, pp. 14–25, 2002.

[17] B. Han, “Zone-based virtual backbone formation in wireless ad hoc networks”, *Ad Hoc Networks*, Vol. 183–200, p. 7, 2009.

[18] D. Kim, F. Zou, D. Z. Du, Y. Wu, and Y. Li, “Constructing Minimum Connected Dominating Sets with Bounded Diameters in Wireless Networks”, *IEEE Trans. Parallel Distrib. Syst.*, Vol. 20, No. 2, pp. 147–157, 2009.

[19] P. J. Wan, L. Wang, and F. Yao, “Two-phased approximation algorithms for minimum CDS in wireless ad hoc networks”, In: *Proc. of the 28th Int. Conf. Distrib. Comput. Syst. ICDCS 2008*, pp. 337–344, 2008.

[20] S. Bouamama, C. Blum, and J. G. Fages, “An algorithm based on ant colony optimization for the minimum connected dominating set problem”, *Appl. Soft Comput. J.*, Vol. 80, pp. 672–686, 2019.

[21] A. R. Hedar, R. Ismail, G. A. El-Sayed, and K. M. J. Khayyat, “Two Meta-Heuristics Designed to Solve the Minimum Connected Dominating Set Problem for Wireless Networks Design and Management”, *Journal of Network and Systems Management*, Vol. 27, No. 3, 2019.

[22] G. V. Shaamili Varsa and D. Sridharan, “A balanced energy efficient virtual backbone construction algorithm in wireless sensor

networks", *AEU - Int. J. Electron. Commun.*, Vol. 107, pp. 110–124, 2019.

[23] C. Luo, J. Yu, D. Li, H. Chen, Y. Hong, and L. Ni, "A Novel Distributed algorithm for constructing virtual backbones in wireless sensor networks", *Comput. Networks*, Vol. 146, pp. 104–114, 2018.

[24] T. Gaspar and Z. Tarnai, "Upper bound of density for packing of equal circles in special domains in the plane", *Period. Polytech. Ser. Civ. Eng*, Vol. 44, No. 1, pp. 13–32, 2000.