# Assurance of the Maximum Destruction in Battlefield using Cost-Effective Approximation Techniques

Fariha Tasmin Jaigirdar
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
Email: farihajaigirdar@yahoo.com

Mohammad Mahfuzul Islam
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
Email: mahfuzcse.buet@gmail.com

***Abstract*— Military Applications of Wireless Sensor Network in domains of maximizing security and gaining maximum benefits while attacking the opponent is a challenging and prominent area of research now-a-days. A commander's goal in a battle field is not limited by securing his troops and the country but also to deliver proper commands to assault the enemies using the minimum number of resources. In this paper, we propose two efficient and low cost approximation algorithms—the maximum clique analysis and the maximum degree analysis techniques. Both of the techniques find the strategies of maximizing the destruction in a battlefield to defeat the opponent by utilizing limited resources. Experimental results show the effectiveness of the proposed algorithms in the prescribed areas of applications. Gaining the cost-effectiveness of the algorithms are also major concerns of this research. A comparative study explaining the number of resources required for commencing required level of destruction made to the opponents has been provided in this paper. The studies show that the maximum degree analysis technique is able to perform more destruction than the maximum clique analysis technique using same number of resources and requires relatively less computational complexity as well.**

***Index Terms*—maximum destruction, military application, maximum degree analysis, maximum clique analysis, minimum resources, wireless sensor network.**

## I. INTRODUCTION

Over the last several years, wireless sensor networks (WSNs) have emerged as a vital research area in networking. This network tightly merges sensing, computing and wireless communications for the first time and advances the wireless communication era to a great extent. WSN is a wireless network consisting of spatially distributed autonomous devices embedding sensors capable of cooperatively monitoring one or more physical or environmental conditions, such as temperature, vibration, pressure, motion or pollutants at different locations [1]. Applications of WSN include ocean and wildlife monitoring, industrial process monitoring, home automation, traffic control, healthcare applications, building safety and earthquake monitoring, and many military applications [2][3].

Different applications of sensor networks are spread over different friendly environments as well as unfriendly or hostile environments. When the question arises concerning military application of WSN, it is almost obvious that the sensor deployment maintains unfriendly environment, may be from aircraft or if necessary under the sea. The term sensor refers to mine, tank, bomb etc in battlefield. Because sensor networks in military application may exchange sensitive data and/or operate in antagonistic unattended environments, it is imperative that security is a prime concern in this networking era. But while maintaining security, a commander's main goal focuses on destroying the opponent's area. So, the major task of the soldiers in a battlefield is making *maximum destruction of the opponents* with minimum resources which guarantees the cost effectiveness and this is one of the required goals of a commander. Maximum destruction means to destroy or corrupt opponent's area (soldiers or tanks or bombs, i.e., necessary valuables in opponent's end) in the highest scale.

Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. Due to the inherent broadcast nature, wireless networks are vulnerable to several kinds of security attacks like eavesdropping, jamming, malicious association, denial of service (DoS) and so on. Moreover, radio jamming makes the multi-hop wireless networks more vulnerable to DoS attacks. So, while considering different actions in battlefield, these are needed to draw attention apparently. As technology is getting popular, foes are planning new ideas to harm the network setup of a battlefield and make the network quality down for having control over the network. To ensure maximum destruction in the battlefield, in this paper, we focus on making the *Security Breaking Cost (SBC)* lower for the foes or jammers which help them to destroy opponent's area in a large scale, as, if the SBC is lower than the benefit obtained by breaking the security, we can say that the system is vulnerable. A commander's goal in a battlefield is to


doi:10.4304/jnw.7.12.1967-1977

make his opponent's network vulnerable so that he can get enough opportunity to assail. As a result, while we are focusing on maximum destruction in the battlefield with minimum resources, SBC estimation is a major concern.

Efficient and cost-effective deployment of active sensor nodes as well as finding the optimum location is one of the key problems in battlefield. A cost-effective deployment of the sensor nodes (tank, mine, bomb etc) can guarantee the minimum or nearly minimum number of these resources needed to destruct the opponent's area. To the best of our knowledge, there is no existing technique that ensures maximum destruction of opponents in battlefields. In this paper, we propose two sensor deployment strategies along with their locations, by which all other opponent's units (represent soldiers or opponent's valuables in the battlefield) of a network can be covered and the number of the resources needed is guaranteed to be optimum. Using these two techniques a commander can make maximum destruction of his opponents with minimum resources assuring the cost effectiveness of the proposed methods. Fig. 1 shows a scenario in the battlefield where resources (here tanks) are placed in most dense region of the network, and here, we can see that with only two tanks all the sensor units have been covered and thus, can be destroyed. One more consideration that can be added here is that, as we will determine the minimum locations in the sensor network with which overall network can be covered, a commander can place some watch towers in those positions to frequently watch the overall battlefield which is a spying strategy in opponent's ground as well.

Finding the minimum location of military resources is an NP-hard problem. Therefore, our solution focuses on finding out the best, which we say nearly minimum or optimum location of placing different resources by exploiting and merging the techniques of maximum clique analysis [4], maximum degree analysis, minimum set cover problem [5] and the greedy approach [6].

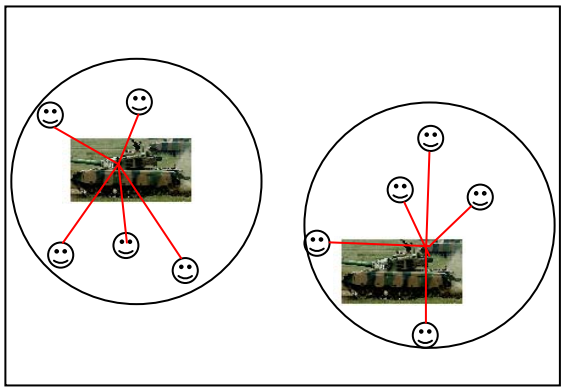


Figure 1. Destroying all the sensor units in a battlefield by placing tanks in the most dense region

Fig. 2 shows the actual scenario of the sensor network after placing the sensor nodes along with every nodes transmission range. In the figure, red nodes represent the malicious nodes, i.e., these are the node positions where by placing different resources (mine, tank, bomb etc) maximum destruction of opponents is possible. The blue nodes represent the opponent's unit in the figure. Thus, more generally, red nodes are those deployed minimum number of nodes by which all the sensor nodes in the network can be covered.

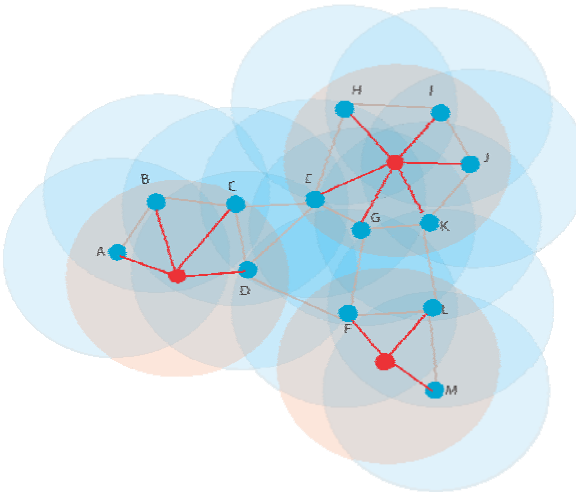



Figure 2. Actual scenario of the sensor network after placing malicious node to corrupt all the sensor nodes

To reach our desired goal of finding minimum or optimum number of resources that will corrupt all the sensor nodes, i.e., opponent's unit in the network, here we have come forward with two approximation techniques. The first approximation technique solves the problem by finding the maximum clique in the network, whereas the second one deals with the maximum degree node. The experimental results clearly reveal that, by setting appropriate parameters, the proposed solutions can efficiently find out the best locations of different military resources. The reason behind using two approximation techniques simply reveals a more efficient technique in respect of both cost-effectiveness and computational complexity. It also can dictate us which strategy needs to follow in which scenario.

The rest of the paper is organized as follows: Section II discusses the background and related works of the issue, while the details of the proposed maximum clique analysis approximation have been given in Section III. Section IV explains the details of the second approximation algorithm, i.e., maximum degree analysis. The comparison of the two approaches with different experimental setup has been given in Section V. Some concluding remarks and directions on future works are given in Section VI.

## II. Literature Review

The idea discussed in this paper seems close to that of the coverage problem, but in reality they are totally different. The coverage problem schemes are mainly of two types: area coverage and target or point coverage. The area coverage problem explores the solution to cover the entire area of a WSN, while point coverage problem, a special case of area coverage problem, focuses on determining the exact position of sensor nodes to provide efficient coverage application for a limited number of targets [7]. Another problem which is a bit similar to our

problem is Minimum Enclosing Circle (MEC), where the aim is to find out the smallest area enclosing circle in a given region to cover all the nodes of the network.

### A. Area Coverage Problem

The most studied coverage problem is the area coverage problem, where the goal of the sensor network is to cover a target area in such a way that there should be no such a point that is not monitored by an observer [14]. As an important research issue many researchers have studied comprehensively on this topic and different sensor network applications have revealed a new era to solve their area coverage problem in different scenarios by varying design choices and other factors. Our problem may seem closer to area coverage problem, but in real sense, it is just the opposite. Area coverage problem works by covering the entire area but our one focuses on covering all the sensor nodes in the network.

### B. Point Coverage Problem

In the point coverage problem, the objective is to cover a set of points or targets [8]. The point coverage scheme, a special case of area coverage problem, focuses on determining the exact position of sensor nodes to provide efficient coverage application for a limited number of targets [9].

A main consideration factor of point coverage problem is the limited number of target or points that need to be covered, leaving the sensor nodes' number out of account. However, this is the difference parameter of point coverage scheme with/which our paper work described already, as our goal is to cover all the sensing nodes with minimum number of foes or malicious nodes.

### C. Minimum Enclosing Circle (MEC)

The Minimum Enclosing Circle (MEC) [10] is a mathematical problem of computing the smallest circle that contains all of a given set of points in the Euclidean plane. Here, the Euclidian plane can be a targeted area or a region, where the size of these does not matter. Fig. 3 shows the minimum enclosing circle of set of points (solid black line).

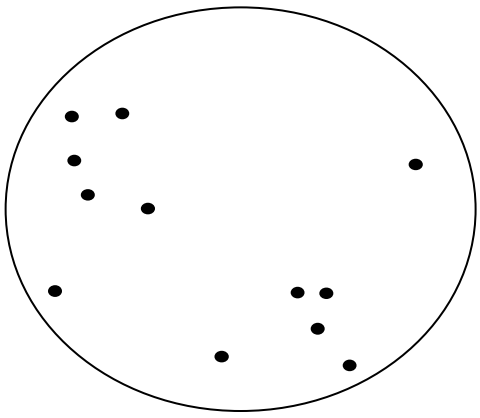


Figure 3. Minimum enclosing circle of a set of points

The well known "Bomb Problem" is actually another name of it [15]. Our problem may seem related to minimum enclosing circle or bomb problem, but actually it is not. Here, in MEC problem, the radiuses of the circles are arbitrary, whereas, the deployed sensor's transmission ranges in research have fixed range R.

## III. Maximum Clique Analysis Approximation

Our first proposed technique is maximum clique analysis approximation. Here, we assume that the overall topology is known to us. Our objective is to strategically place different resources in the battlefield instead of deploying them blindly to destroy all the sensor nodes which are different units of opponent's in the network. Here, the word "*strategically*" means the way of resource deployment that we have shown here to be the best. By our proposed deployment strategy, it would be possible for us to find out the best locations of deployment as well as the minimum or nearly minimum amount of resources to destroy or jam all the sensor nodes in the network. By this, we would gain the efficiency and cost effectiveness of the deployment methodology and reach the required goal as well. Maximum clique formulation is a well known problem and before moving forward to that topic, we first emphasize to clique and its members. We will discuss these in the following sub-sections. Afterwards, we will go ahead in the core of the methodology of this scheme.

### A. Clique and It Members

In an undirected graph, a clique is a subset of its vertices such that every two vertices are connected by an edge [4]. A vertex $v$ is connected to another one, if it resides within the transmission range, R of that vertex, and yet is connected with each other. A vertex, alone, can be said one sized clique, as it is connected to itself. Maximum clique in a graph is that clique which has maximum connection with all its vertices in a set of vertices, i.e., which is of highest sized clique among all the cliques in the graph.

### B. Finding the Maximum Clique Node and Placing the Jamming Node

Here we consider the entire networking region of the battlefield as a graph to come forward with the clique concept. To achieve multi-hop communication each node in the wireless network acts as a router, forwarding data packets for other nodes. In addition, we assume that each node has a low power Global Position System (GPS) receiver, which provides the position information of the node itself. If GPS is not available, the distance between neighbouring nodes can be estimated on the basis of incoming signal strength. Suppose a sensor network consisting of $n$ sensor nodes is deployed in a two-dimensional field of size $N \times N$ with transmission range $R$. To jam or corrupt a wireless node $v$, a jamming or malicious node $j$ must be placed so that $\|v, j\| < R$, i.e. j should be placed inside an $R$-radius circle centered at $v$.

The first task here is to find out the distance of all vertices from the entire vertices to determine the edges among them and by this, we can determine every vertex's neighbour to find out the clique, and finally the maximum clique for obtaining our desired goal. In the Fig. 4, all the vertices have been plotted randomly over the networking

region (these can be considered as opponents unit in a battlefield) which forms the graph here. Maximum clique from the graph can be identified by the square shaped block.

We can see from Fig. 4, the indicated maximum clique is of size four, because these four nodes are connected with each other and so, they are the members of the maximum clique. The degree of every member has been shown also in the figure. From the figure, we can see that vertex 1, 2, 3 and 4 have created the maximum clique and among them, vertex 1 has the maximum degree, which is six. As a result, we can easily understand that maximum destruction is possible by placing a malicious node (which is the node position where different military resources as tank, mine etc need to be plotted) in that vertex.

Thus the maximum clique analysis strategy will place the first malicious node in vertex 1 and by this, vertex 2, 3, 4, 7, 8, 9 can be destroyed. Actually these are the opponent's positions that were needed to be destroyed by the military commander.

After placing the first malicious node in the highest degree member of the maximum clique, this strategy will remove all the vertices from the total list of vertices as they are already shattered. So, the strategy will move forward with minimum set cover concept to find out the rest of deployment positions to cover the remaining sensors. With this goal this strategy will be executed again for finding out the next malicious node position to reach the desired goal.

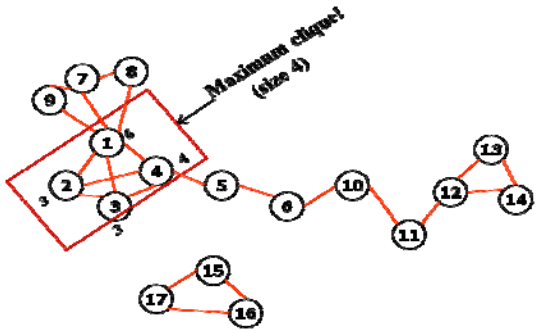



Figure 4. Vertices are placed over the deployment area and maximum clique along with their degree value have been determined

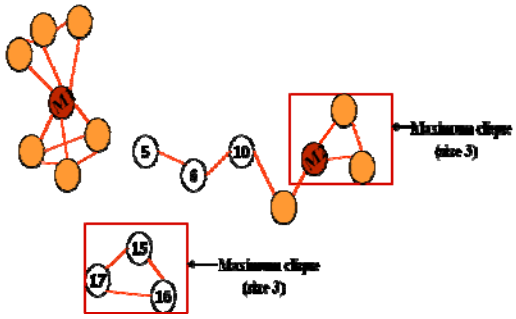



Figure 5. Placing the malicious node

Fig. 5 shows the after effects of placing the first and second malicious nodes in the most dense region accordingly. From the figure we can see that after placing the first malicious node the strategy will search for the maximum clique from the rest of the vertices and find two three sized cliques. So it will place the malicious node any of them and in the figure we can see that it has chosen the clique created by vertices 12, 13 and 14. Further the strategy needs to be reapplied for finding out next malicious node.

Fig. 6 shows the final snapshot of the strategy. The red coloured circle represents the malicious node (different resources in the battlefield) and they are indicated in the figure as M1, M2, M3 and M4 respectively. On the other hand, the orange coloured circle represents the sensor nodes that have been jammed by those malicious nodes.

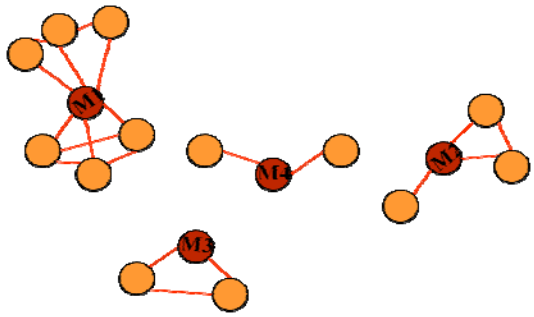



Fig. 6 All the vertices have been jammed by four malicious nodes

So, we can understand that if a commander applies this strategy, he can place tanks or bombs or mines in M1, M2, M3 and M4 positions, and can destroy his opponents unit with only four resources that guarantee the cost effectiveness of the strategy. Moreover, the commander can use this strategy to deploy some watch towers in these three positions which will help him and his team spying on opponent's region more effectively and efficiently.

### *C. Algorithm*

The algorithm first searches for the maximum clique in the given graph. Upon finding the maximum clique, the algorithm will place the malicious node to the member having the highest degree of the graph and then continue with proceedings till destroying all the sensor nodes in the network.

**Algorithm Maximum_Clique_Analysis (G, R)**

//num_unaffected=number of unaffected nodes, //high_freq=highest number of nodes covered, //high_freq_x,high_freq_y = represents the coordinates of //highest frequency point, R=transmission range, //unaffected_nodes=the nodes that are still unaffected. //Here, at first from the given graph the maximum clique //will be determined, and then the malicious node will be //placed in the maximum clique to destroy all the nodes in //the network.

1. MAXCLQ(G,R)
2. while num_unaffected>0 do
3. high_freq:=0
4. Graph graph = Graph(num_unaffected, nodes, unaffected_nodes, R)
5. high_freq:=graph.CF
6. high_freq_x=nodes[unaffected_nodes[graph.get

```
7.  MalNode()]][0];
8.  high_freq_y=nodes[unaffected_nodes[graph.get
9.  MalNode()]][1];
10.          i=0;
11.    for j := 0 to num_unaffected
12.       if(distance(high_freq_x, high_freq_y,
13. nodes[unaffected_nodes[j]][0],nodes[unaffected
14. _nodes[j]][1])> R)
15.          temp[i++] = unaffected_nodes[j];
16.        endif
17.    endfor
18.    for j:=0 to i
19.      unaffected_nodes[j] = temp[j];
20.    endfor
21.  endwhile
```

**Algorithm MAXCLQ (G,R)**

```
//The input is a Graph, G = (V, E), where V =vertices,
//i.e, number of nodes and E is a set of edges.
//GraphArray is a two dimensional array whose 1st
//element is the index of  vertex and second element
//shows the index of other nodes with which it is
//connected and it is related to another array, Degree [i]
//that show the degree of that vertex. Degree [] returns the
//number of connection with a node, Malnode is the node
//where the malicious node need to be placed. CF =
//Clique Found
1.  for each u in V[G]
/Going to find the node which has maximum no of
//neighbours, means which degree is highest, then we
//will proceed from that node to find the maximum
//clique in the Graph.
2.  IsvertValid[u]:=0
3.  degreecount:=0
4.    for each v in V[G]
5.      if u ≠ v and distance (nodes[un[u]][0],
6.  nodes[un[u]][1],nodes[un[v]][0],
7.  nodes[un[v]][1])≤R
8.         GraphArray[u][degreecount++]:=v
9.      endif
10.    endfor
11. Degree[u]:=degreecount
12.   if(Degree[u]>Degree[HighestDegreeNode])
13.      HighestDegreeNode:=u
14.    endif
15. endfor
16. CF:=HighestDegreeNode:=0
17.    if V>0
18.      CF++
19.       Malnode:=HighestDegreeNode
20.   endif
21. while HighestDegreeNode>0
22. Tree[0]:=HighestDegreeNodee
23.  TreeLevel:=0
24.  EXPLORE()
25. IsvertValid[Tree[0]]++
26. HighestDegreeNode:=BRANCHING_RULE
27. endwhile
28.  return CF
```

**Algorithm EXPLORE(G)**

```
//Tree[i] represents every node in the tree, vertex
//represents the current node, CountValidChild
//represents the number of valid child in a tree.
1.  for i:=TreeLevel to 0 step -1
2.    if there is not a connection between Tree[i]
3.         and vertex
4.       Root:=i+1
5.     endif
6.   endfor
7.  if CF<TreeLevel-i
8.     CF:=TreeLevel-i
9.   endif
10. for i:=0 to Degree[vertex v]
//Here considering all the degrees that is children of the
//current vertex v.
11.    if v is a member of current clique or v is
12.            already taken or Degree of v<CF
13.       children[i]:=0
14.     else
15.       children[i]:=1
16.       CountValidChild++
17.     endif
18.   endfor
19.  for each valid child
20.     if CountValidChild+(TreeLevel+1-
21.                          Root)>=CF
//Here (TreeLevel+1-Root) represents current clique
//size.
22.       TreeLevel++
23.     Tree[TreeLevel]:=GraphArray[vertex][i]
24.       EXPLORE(G)
25.       TreeLevel--
26.       CountValidChild--
27.     endif
28.   endfor
```

**Algorithm BRANCHING_RULE (G)**

```
// If there are some vertices left that has not considered or
//missed anyhow.
1.  Hgvtnode:=0, flag:=0
2.  for i:=0 to V
3.    if IsVertValid[i] is not considered
4.     if Degree[i]>Degree[HgVtNode]
5.       HgVtNode:=i
6.        flag++
7.     endif
8.     endif
9.     if flag>0
10.     return HgVtNode+1
11.    return 0
12.    endif
13.  endfor
```

### *D. Experimental Results*

Different network scenarios with changing network parameters have shown in this section to better understand the strategy as well as the effectiveness of the algorithm. As our paper work deals with different network topologies, the changing parameters used here are transmission range R, number of sensing nodes N and total networking area or dimension, D. Here, different

network scenarios are presented to clarify in what scenario our proposed approximation will work better, i.e., how maximum destruction is possible with minimum number of resources.

The first scenario keeps the number of nodes to be fixed which is 100 and changes the other two parameters, i.e., transmission range, R and dimension, D. To maintain three levels of values, transmission range values have been chosen as 25, 40 and 70 and dimension values have been changed from 50 to 200.

Fig. 7 shows the number of malicious nodes (different resources as already stated) needed for such a scenario. From the result it can be observed that as the value of transmission range increase from 25 to 70, the number of malicious node needed to jam or corrupt the network decreases accordingly. The reason is, with the higher value of the transmission range, more nodes can be in the range of the maximum clique member and thus, number of resources decrease accordingly.

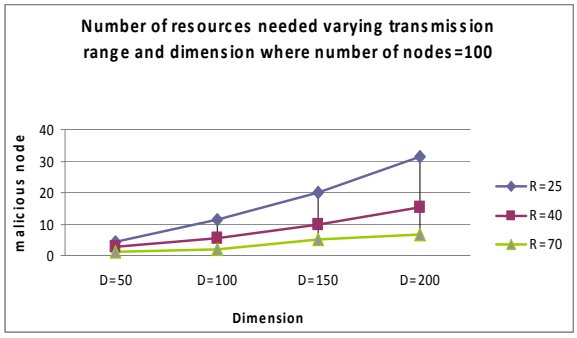



Figure 7. Results by changing transmission range and dimension

In the second scenario, the transmission range remains fixed while other two parameters change accordingly. Fig. 8 shows the reflection of above statement clearly. Here comes another vital parameter of the network, number of sensing nodes, N. In this scenario, we have changed the values of number of nodes to 70,100,150,200. In Fig. 8 we can see the impact of changing values of number of nodes in the network. In most of the cases, as number of nodes increase, the number of malicious nodes increases accordingly. This is because, with higher number of nodes, more malicious nodes are necessary to embrace them meeting the corresponding criteria.

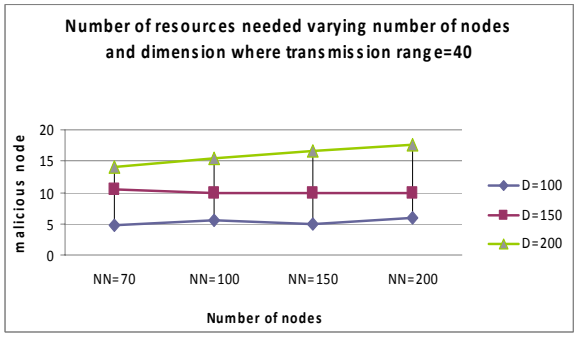



Figure 8. Results by changing dimension and number of nodes

### *E. Complexity*

The maximum clique problem is computationally equivalent on arbitrary graphs. Maximum clique finding is a well known NP-complete problem [11]. Furthermore, for the maximum clique problem, the complexity of approximating remained an open question until recently. Our maximum clique analysis strategy works for finding the maximum clique in the network first, and then moves for placing malicious node. So the total complexity remains NP-complete.

## IV. Maximum Degree Analysis Approximation

Our second approximation scheme is discussed in this section. Here, the new concept works with the degree of a vertex. By the term vertex, we mean the sensor node of the network. In this strategy, again, we assume the network placement as a graph and its sensor nodes as vertices of the graph [12]. Our aim is to place the malicious node strategically in most dense region of the network to find out the best locations of resource deployment and obtain our required goal as well. To fulfil the aim, the main consideration factor is, actually, what strategy we should follow in. In the last section, we showed maximum clique analysis approach, which worked by finding out the maximum clique in the networking region. In that case, we have described the strategy and the result with the complexity as well. The complexity of maximum clique analysis approximation is NP-complete. So, we need a better approximation technique which decreases the computational complexity as well as number of resources needed to destroy opponent's unit in the battlefield. In this section, the new strategy works by placing the malicious node in that vertex whose degree [13] is the maximum. By our proposed deployment strategy, it would be possible for us to find out the best locations as well as the minimum or nearly minimum amount of malicious nodes needed to jam or destruct all the opponent's valuables in the network and thus, fulfil our goal of maximum destruction in the battlefield in a more efficient manner.

For understanding the maximum degree strategy, we have to learn about the graph formulation of the network topology as well as the degree of a vertex. These will be discussed in the next sub-sections.

### *A. Graph and Degree*

A graph is a set of points (we call them vertices or nodes) connected by lines (edges or arcs) [12]. Here, all the sensor nodes are deployed randomly, and they are connected with each other by their transmission range. That is, every node, which is called vertex, can transmit or receive information from the other sensor node which is within the transmission range of that node, and here, the concept of edges (the connecting line with two or more nodes) comes.

The degree (or valency) of a vertex of a graph is the number of edges incident to the vertex, with loops counted twice [12]. The degree of a vertex $v$ is denoted deg ($v$). The maximum degree of a graph $G$, denoted by $\Delta$

(*G*), and the minimum degree of a graph, denoted by δ (*G*), is the maximum and minimum degree of its vertices.

### *B. Finding the Highest Degree Node and Placing the Jamming Node*

Here we consider each node as a vertex and the connectivity among the vertices as edges. The first task here is to find out the distance of all vertices other vertices to determine the edges among them and by this, we can find out the degree of a vertex. Here, to determine the degree of each vertex, the strategy will search for all nodes *u*, *v* such that dist (*u*, *v*) ≤ *R*, where *R* is the transmission range of the vertices as well the malicious node's. A table is maintained periodically to store the vertex position, its neighbours and the degree of each vertex. Then from all the degree that is stored in the table, greedily, this strategy will find out the highest degree vertex in the graph. In Fig. 9, sensor nodes, i.e., vertices (small circle) are randomly placed over the deployment area and *n1* node's transmission area is shown in larger shaded circle. By this *n1* node's degree can be understood, which is three here, as *n1* is connected to *n2*, *n3* and *n1* itself.

In this way, degree of every vertex is counted and stored in a table for further implementation. In Fig. 10, all vertices connectivity is shown to better understand every vertex degree. Here, the graph of the network topology along with their vertices and edges are shown. All these information will be stored in the table. Actually, upon finding first node that is the highest degree vertex in the graph, we will place our first malicious node in that position and with that malicious node, the vertices, those are the degree or member of that highest degree vertex, will be destroyed. Then the jammed vertices will be removed from the entire set of vertices as affected and the strategy will search again in other unaffected vertices by updating the table. Thus, processing as minimum set cover problem, the result can be found, i.e., the final form of the table will show the minimum number of malicious nodes or resources needed to destruct all the sensor nodes in the network.

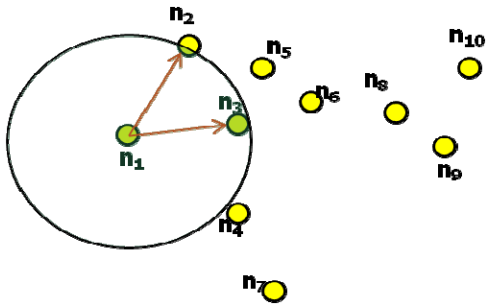



Figure 9. n1 vertex's degree along with its transmission range

In Table 1, every vertex along with its neighbors' and degree is shown. Preliminarily, the table is built with all these information as shown in the table, i.e., with every vertex's degree and neighbours. Then, from this table, as stated previously, the maximum degree vertex will be chosen, which is *n3* here and which is also the position of the first malicious node. Upon placing the first malicious node in *n3*, it will destroy six of its members, *n1*, *n2*, *n3*, *n4*, *n5* and *n6*. Upon placing it, the strategy will remove all the vertices from the network and reapply the strategy on the unaffected vertices to reach the desired goal. In this way, the table will be updated periodically, and finally it will end by finding out the minimum or nearly minimum number of malicious nodes needed to corrupt all the sensors in the network. In this way, the minimum number of resources can be identified for maximum destruction in the battlefield. Fig. 11shows the final update of the table, where *n3* is the first malicious node, whereas the second and third malicious nodes are *n8* and *n4* respectively. In the figure, the red nodes represent the malicious nodes which are positions of the resources to corrupt the opponents unit in the battlefield. The affected sensor nodes are shown in Fig. 11 in orange colour. Actually, maximum degree analysis scheme works better than the previous strategy as in this technique connectivity among nodes maintain the highest priority.

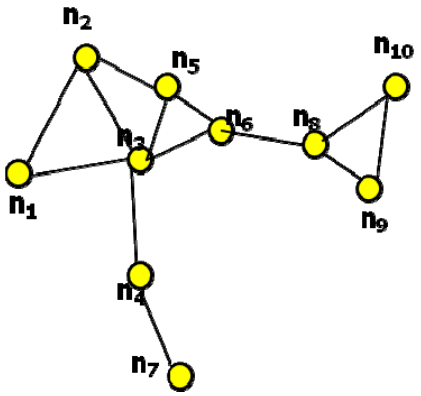


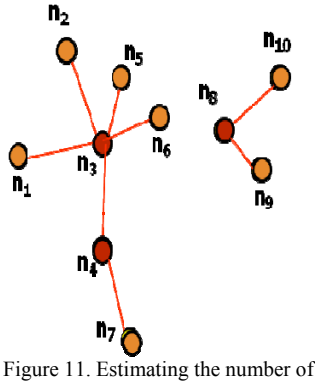

Figure 10. Degrees of the vertices

TABLE I.
HIGHEST DEGREE VERTEX IN THE GRAPH

| Vertex | Nodes Covered | Degree |
|---|---|---|
| *n1* | *n1, n2, n3* | 3 |
| *n2* | *n1, n2, n3, n5* | 4 |
| *n3* | *n1, n2, n3, n4, n5, n6* | 6 |
| *n4* | *n3, n4, n7* | 3 |
| *n5* | *n2, n3, n5, n6* | 4 |
| *n6* | *n3, n5, n6, n8* | 4 |
| *n7* | *n4, n7* | 2 |

| | | |
|---|---|---|
| *n8* | *n6, n8, n9, n10* | 4 |
| *n9* | *n8, n9, n10* | 3 |
| *n10* | *n8, n9, n10* | 3 |

The final snapshot of the strategy reflects in the following figure.




| Vertex | Nodes covered | Degree |
|---|---|---|
| M1 | $n_1, n_2, n_3, n_4, n_5, n_6$ | 6 |
| M3 | $n_7$ | 1 |
| M2 | $n_8, n_9, n_{10}$ | 3 |

Figure 11. Estimating the number of malicious nodes needed to destroy all the vertices in the graph

### C. Algorithm

The algorithm consists of two parts. At first it finds out the maximum degree vertex and then place the malicious node in that position for making maximum destruction. Finally it finds out the minimum number of resources or malicious node needed to completely jam the network. Here *Algorithm GetHighestDegree ()* performs the first part, and the later part is done by *Algorithm Maximum_Degree_Analysis ()*.

**Algorithm Maximum_Degree_Analysis (num_unaffected, R)**

//num_unaffected= number of nodes unaffected, //high_freq=highest number of nodes covered within the //transmission range(R), high_freq_x,high_freq_y //represents the coordinates of highest frequency point, //i.e., co-ordinate of maximum degree vertex , //R=transmission range

```
1.  while num_unaffected > 0 do
2.        high_freq:=0
3.        HighestDegreeNode:= 0
4.   high_freq=GetHighestDegree(num_unaffected,
5.                        nodes, unaffected_nodes,  R)
6.         high_freq_x := nodes [unaffected_nodes
7.                        [HighestDegreeNode]] [0]
8.         high_freq_y :=  nodes [unaffected_nodes
9.                        [HighestDegreeNode]] [1]
10.              for j := 0 to num_unaffected
11.       if distance ( high_freq_x, high_freq_y,
12.          nodes[unaffected_nodes[j]][0],
13.          nodes [unaffected_nodes[j] ][1]) > R
//update the list of unaffected nodes.
14.                 temp [i++] := unaffected_nodes [j]
15.                  endif
16.               endfor
17.                for j := 0 to i
18.                 unaffected_nodes [j] := temp [j]
19.              endfor
20.  endwhile
```

**Algorithm GetHighestDegree (n, nodes, un, R)**

//R=transmission range, un=unaffected_nodes, //HighestDegreeNode=The vertex that connects //maximum number of nodes, n=Total number of nodes

```
1.  n := num_unaffected
2.  for i := 0 to n
3.      degreecount := 0
4.      for j := 0 to n
5.         ifdistance(nodes[un[i]][0], nodes[un[i]][1],
6.           nodes[un[j]][0],  nodes [un[j]][1]) <= R
7.              degreecount++
8.         endif
9.       endfor
10.   Degree[i] := degreecount
11.       if Degree[i]>Degree[HighestDegreeNode]
12.            HighestDegreeNode := i
13.       endif
14.  endfor
```

### D. Experimental Results

As used in the previous approximation scheme, this strategy deals with the changing parameters of different network topologies and shows the experimental result for understanding the algorithms effectiveness. The network parameters are same as before. We have shown two scenarios to better understand the strategy. From the Fig. 12, it can be noted that with the increasing value of transmission range, the number of resources needed to corrupt the network decreases accordingly. The reason is the same as stated in the previous approximation scheme.

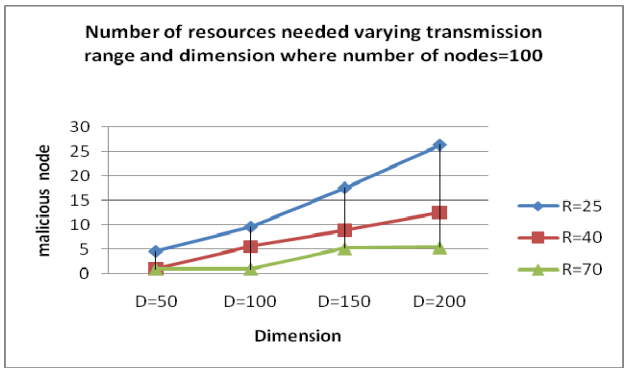



Figure 12. Results by changing transmission range and dimension

The second scenario comes with making the transmission range fixed, 40 and changing other two parameters, dimension and number of nodes. Here comes another important parameter of the network, number of sensing nodes, N. In this scenario, we have changed the values of number of nodes as/to 70,100,150,200. In most of the cases, we need a sensor network with maximum

number of nodes deployed in the network for establishing WSNs different kinds of applications. If we have a look in the Fig. 13, we can see that, in most of the cases, as number of nodes increase, the number of malicious nodes requires increase accordingly. This is because, with higher number of nodes, more malicious nodes are necessary to embrace them meeting the corresponding criteria.

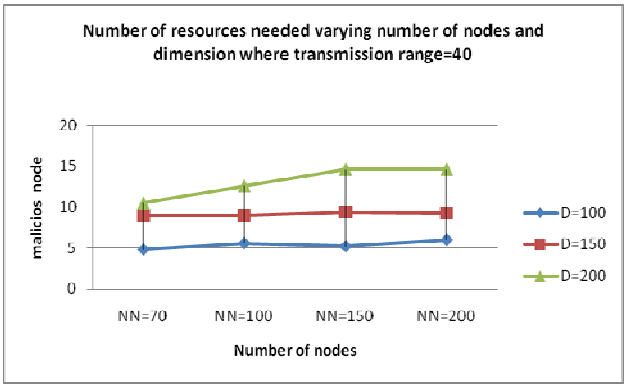



Figure 13. Results by changing dimension and number of nodes

### E. Complexity

We considered Number of sensors = $N$, $R$ = Transmission range, $C_D$ = Cost for calculating the distance and $D$= Dimension of the network.

*Step 1*: The first task here is to determine the degree of each node. For this, distance from every sensor node to the all other sensor nodes need to be calculated first. So distance calculation for this step is $N \times N = N^2 C_D$.

*Step 2*: To determine each node's degree, the strategy will search for all nodes $u$, $v$ such that dist $(u, v) \leq R$, where $R$ is the transmission range of the malicious nodes. Average number of degree of a sensor node can be calculated by dividing the area of a sensing node by the total dimension of the network.

So, the average number of degree for a sensor node will be

$$\frac{\Pi R^2}{D^2} \times (N\text{-}1)$$

Thus total degree for N nodes is

$$\frac{\Pi R^2}{D^2} \times (N\text{-}1) \times N \times C_D$$

Other processing costs are negligible in comparison with the distance calculation and that is why needed not to be added while finding out the total complexity.

*Finally*, total complexity for this strategy is $N^2 C_D + [(\Pi R^2/D^2) \times (N\text{-}1) \times N \times C_D$

$$= \boldsymbol{C_D\,[N^2 + (\Pi R^2/D^2) \times (N\text{-}1) \times N]}$$

## V. Comparison Of The Two Approaches And Finding Out The Best One

Comparative analysis of the proposed two approximations will help us to know the pros and cons of the approaches. Moreover, it will also guide us to better understand every scenario and also show us which approach is good in which scenario. Moreover, it would reveal our interest for further research. Our goal here is to find out the minimum number of resources with minimum time complexity. So, number of the resources as well as time complexity measurement is the main consideration of comparing both approximations. For clearly comparing different strategies, different scenarios with changing values of different network parameters have been shown here that would be helpful to better understand every approach's efficiency as well as working area.

### A. Effect after Changing the Transmiision Range

Transmission range is a very important consideration issue while designing such a sensor network where node covering is a prime concern. If we can place a malicious node within the transmission range of a sensor node, we can jam that sensor node easily. With higher valued transmission range, many sensor nodes can be covered and thus, by placing a malicious or jamming node in such a position maximum destruction is possible.

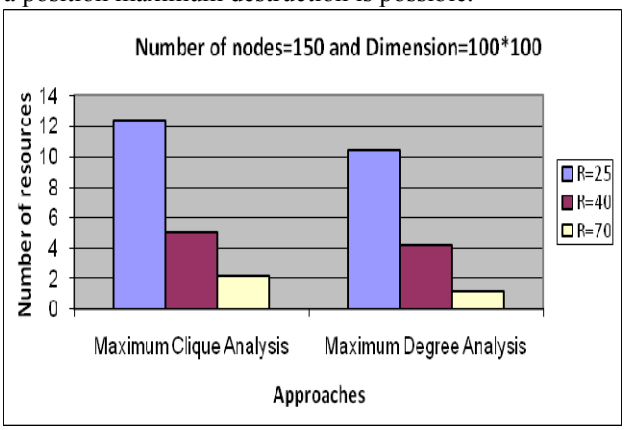



Figure 14. Different approximation result by changing the transmission range with number of nodes 150 and dimension 100

So, we can easily understand that, as the value of transmission range increases, the number of resources needed to destruct all the sensor nodes in a network decrease accordingly. This statement's clarification can be easily understood from Fig. 14. From the figure, we also can notice that number of resources needed to completely destroy the battlefield is fewer in maximum degree analysis approximation than in maximum clique analysis approximation. The reason is very obvious. Maximum clique formulation is more expensive than maximum degree formulation, because, to form a clique all the sensor nodes in the clique need to be connected to each other. So, from this perspective, we can say that maximum degree analysis performs better than the maximum clique analysis algorithm.

### B. Effect after Changing the Number of Nodes

Another important issue is the number of nodes in a network. In most of the cases, as the number of sensor nodes in a network increases, the resources needed to corrupt all the nodes increase accordingly. This is because, with the greater value of number of nodes, more

malicious nodes are necessary to embrace them with necessary criterion. We can see the reflection of this statement in Fig. 15. Here, we have kept the transmission range and dimension value is fixed to 40 and 150 respectively. By changing the value of number of nodes, different results have been stored. We can see in the figure, for the number of nodes 70 and 100; the resources needed to corrupt all the sensor nodes are minimum for maximum degree analysis scheme compared to maximum clique analysis scheme. But when the number of nodes value have been increased to 150 and 200, the result shows the opposite, that is, maximum clique analysis approximation works better then maximum degree analysis approximation. The reason can be understood from the clique formulation behavior. To form a clique the entire sensor nodes connectivity is a must, and this number can increase if there are more sensor nodes in the network. So, for such a scenario, maximum clique analysis algorithm works better than maximum degree analysis algorithm.

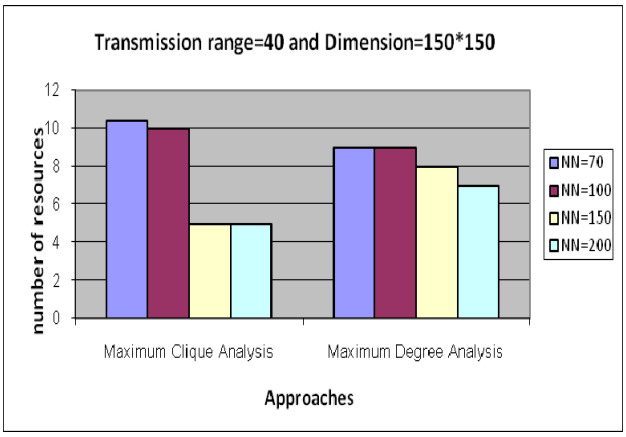



Figure 15. Different approximation result by changing the number of nodes with transmission range 40 and dimension 150

### C. Complexity Analysis

Whenever we are presenting an algorithm, the time complexity is a very important contemplation. For maximum clique analysis algorithm we have seen that it is a NP-Complete problem. So, we need to go through a different approximation strategy to increase the complexity as well. Maximum degree analysis scheme's complexity has proved to be quadratic, which is better than that of maximum clique analysis algorithm. So, from this perspective, we can say that maximum degree analysis approximation is better.

## VI. Conclusions

While securing own troops and resources, a commander in battlefield is required to plan for destroying the same of the opponent. However, every commander is required to work with limited resources and to deploy the resources in a planned way for maximizing harms to the enemies. The approximation algorithms proposed in this paper are able to guide battlefield resource planning through finding the better places for resource deployment and exploiting the concept of wireless sensor network. The algorithms also able to find out the minimum number of resources required to fight against the opponent in an effective way. Experiemntal results shows that the number of resources required for generating plan using the concept of maximum degree analysis algorithm is less compared with the plan generated using the concept of maximum clique analysis algorithm for the same strength of opponent. The computational complexity of the maximum degree analysis algorthm is also less compared with the maximum clique analysis algorithm. The algorithms proposed in this paper are suitable to be applied in battlefield effectively.

## References

[1] Y. Mioa, "Applications of sensor networks," Seminar on Wireless Self-Organization networks, 2005.
[2] Wireless sensor networks, *http:// en.wikipedia.org/wiki/ Wireless_sensor network*, last visited 20th May, 2011.
[3] C.–Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities and challenges," *Proceedings of IEEE*, vol. 91, pp. 1247-1256, 2003.
[4] E. Tomita and T. kameda, "An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments," *Journal of Global optimization*, vol. 37, pp. 95-111, 2007.
[5] E. Tomita and T. kameda, "An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments," *Journal of Global optimization*, vol. 37, pp. 95-111, 2007.
[6] Greedy algorithm, *http:// en.wikipedia.org/wiki/ Greedy_algorithm*, last visited 6th June, 2011.
[7] F. T. Jaigirdar, M. M. Islam and S. R. Huq, "Grid approximation based inductive charger deployment technique in wireless sensor networks," *International Journal of Advanced Computer Science and applications (IJACSA),* vol. 2, pp.30-37, 2011.
[8] M. Cardei and J. Wu, "Coverage in wireless sensor networks," *Handbook of Sensor Networks*, M.Ilyas and I. Mahgoub (eds.) Publishers, CRC Press, 2004.
[9] F. Gaojun and J. Shiyao, "Coverage problem in wireless sensor network: A survey," *Journal of Networks*, vol. 5, pp. 1033-1040, 2010.
[10] Minimum enclosing circle problem, *http://www.cs. mcgill.ca/~cs507/projects/1998/jacob/problem.html*, last visited 27th June, 2011.
[11] M. Marathe, H. Breu, H. H. Iii, S. S. Ravi and D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, pp. 59–68, 1995.
[12] Wilson, R. J., Introduction to Graph Theory, Pearson education, Singapore, 1972.
[13] Degree of a graph, http://en.wikipedia.org/wiki/Degree_ %28graph_theory%29, last visited 9th May, 2011.
[14] M. Marengoni, B. A. Draper, A. Hanson and R. Sitaraman, "A system to place observers on polyhedral terrain in polynomial time," *Image and Vision Computing*, pp. 773-780, 2000.
[15] Bomb Problem, http://mathworld.wolfram.com/Bomb Problem.html, last visited 4th December, 2011.

**Fariha Tasmin Jaigirdar** completed her B. Sc. Engineering in Computer Science and Engineering degree from the Department of Computer Science and Engineering (CSE) of Chittagong

University of Engineering and Technology (CUET) in 2007 and M. Sc. Engineering in Computer Science and Engineering degree from the Department of CSE of Bangladesh University of Engineering and Technology (BUET) in 2011. She is an Assistant Professor in the Department of CSE of Stamford University Bangladesh. Her research interest includes network security, wireless network and digital signal processing. She has published more than 10 articles in renowned peer-reviewed international journals and conferences.

**Dr. Mohammad Mahfuzul Islam** obtained his PhD degree from the Gippsland School of Information Technology, Monash University, Australia. He obtained his B. Sc. Engg. and M. Sc. Engg. degrees in Computer Science and Engineering (CSE) from the Bangladesh University of Engineering and Technology (BUET). Dr. Islam published more than 50 articles in internationally reputed peer-reviewed journals and conferences. His research interest includes network security, wireless communications, wireless sensor network, e-health and cloud computing. Dr. Islam is working as a professor in the Department of CSE, BUET. He is a fellow of Bangladesh Computer Society, a fellow of Institute of Engineers Bangladesh and a member of IEEE.