
An optimization algorithm for the minimum k-connected m-dominating set problem in wireless sensor networks

---

Namsu Ahn And Sungsoo Park

Technical Report

June, 2014

Department of Industrial and Systems Engineering

Korea Advanced Institute of Science and Technology (KAIST)

An optimization algorithm for the minimum *k*-connected *m*-dominating set problem in wireless sensor networks

This article has been submitted to [Wireless Networks].

* Contact Address :

Sungsoo Park, Professor,

Department of Industrial and Systems Engineering

Korea Advanced Institute of Science and Technology (KAIST)

291 Daehak-ro, Yuseong-gu, Daejeon, 305-701, Republic of Korea

PHONE : +82-42-350-3121

FAX : +82-42-350-3110

E-mail : sspark@kaist.ac.kr

An optimization algorithm for the minimum k-connected m-dominating set problem in wireless sensor networks

Namsu Ahn
Defense Agency for Technology and Quality, 275-18, Boksu-dong, Seo-gu, Daejeon, 302-840
Republic of Korea, namsu.ahn@gmail.com

Sungsoo Park
Department of Industrial and Systems Engineering, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 305-701
Republic of Korea, sspark@kaist.ac.kr

In wireless sensor networks, virtual backbone has been proposed as the routing infra-structure and connected dominating set has been widely adopted as virtual backbone. However, since the sensors in wireless sensor networks are prone to failures, recent studies suggest that it is also important to maintain a certain degree of redundancy in the backbone. To construct a robust backbone, so called k-connected m-dominating set has been proposed. In this research, we propose a mathematical formulation and an optimal algorithm for the minimum k-connected m-dominating set problem. To the best of our knowledge, this is the first mathematical formulation for the problem, and extensive computational results show that our optimal algorithm is capable of finding a solution within a reasonable amount of time.

Key words: wireless sensor networks; robust connected dominating set; integer programming; optimal algorithm
History:

---

1. Introduction

Typically, wireless sensor networks (WSN) is featured by no fixed infrastructure, multi-hop communication and limited resources (battery capacity and bandwidth). Therefore, although the simplest routing method to send a message is flooding, this not only devastates batteries of the sensors but gives negative effect on the throughput of the whole network as indicated in [29]. Furthermore, if we use flooding for broadcasting purpose, it probably causes *broadcasting storm problem* as shown in [14]. Therefore recent broadcasting, activity scheduling and area monitoring algorithms are based on the concept of the virtual network infrastructure, so called backbone. Backbone can be defined as a subset of sensors that can perform data communication tasks, and serves sensors that are not included in the backbone [20]. This concept is frequently used to simplify the network and improve the efficiency of the routing. Previous work [18] has shown that the backbone can reduce the routing overhead dramatically. There is a vast amount of literature about the backbone construction and a summary can be found in [19].

An important concept used in the literature for backbone structure is a connected dominating set (CDS) [30]. A connected graph $G = (V, E)$ is used to represent the network, where V and E indicate the set of vertices and the set of edges, respectively. Each vertex $v(\in V)$ represents a sensor, and there is an edge e $(=uv, \in E)$ which denotes sensor u is within sensor v's communication range and vice versa. A set of vertices is called a dominating set if each of the vertices in the graph is either in this set or has a neighbor (vertex u is neighbor of vertex v if there exists an edge between the two vertices) in the set. Typically, the vertices which belong to the dominating set are called dominators, and the other vertices which are not included in the set are called dominatees. A dominating set is referred to as CDS if the subgraph induced by the dominators is connected. It is known that the connectivity of vertices in a dominating set is required for proper routing of signals as indicated in [1] and [9].

If CDS is used as a backbone, the following good characteristics of the network can be obtained.

- Routing overhead can be reduced because only the sensors belonging to a CDS need to maintain the routing information. Thus, if a dominatee wants to send a packet to another dominatee, it sends the packet to its dominator. Then the dominator will deliver the packet to the destination dominatee.

- Energy efficient area monitoring is possible. Since CDS is a good approximation of an area, dominators belonging to a CDS can take over the dominatees' sensing task. Therefore, while the dominators are actively performing the sensing task, all the other dominatees probably enter into a low-battery sleep state to save energy for future use.

Usefulness of CDS in WSN has been demonstrated in many communication protocols such as media access coordination, unicast, multicast/broadcast, location-based routing, energy conservation and topology control [4].

However, since the sensors in WSN are prone to failure due to accidental damages or battery depletion, recent researches ([2], [6], [11] and [12]) indicate that it is also important to maintain a certain degree of redundancy in CDS.

Suppose that we have a dominating set such that every dominatee has at least m adjacent dominators. Then even if (m-1) adjacent dominators of a dominatee failed, the dominatee still can be connected to the dominating set. We say that paths joining two distinct vertices s and t of G are internally disjoint if they have no internal vertices in common. Then the maximum number of internally disjoint paths connecting s and t is denoted by $p(s,t)$ and we say that G is k-connected if $p(s,t) \geq k$ for any two distinct vertices s and t. The maximum value of k for which G is k-connected is called the vertex connectivity of G and denoted by $\kappa(G)$. If the subgraph of G induced by a dominating set is k-connected, the dominating set still maintains connectivity even though (k-1) dominators failed.

This type of the connected dominating set can enhance the fault tolerance capability and the routing flexibility of the network, and it is referred to as the k-connected m-dominating set (=(k, m)-CDS).

Since the number of sensors constituting the virtual backbone needs to be as small as possible to decrease the protocol overhead and energy consumption, it is desirable to obtain a minimum sized (k, m)-CDS, and this is referred to as the minimum (k, m)-CDS problem. Then, the minimum (k, m)-CDS problem can be defined on graph as follows. Given a graph $G = (V, E)$ with two natural numbers k and m, find a subset $S \subset V$ of minimal size such that every vertex in $V \setminus S$ is adjacent to at least m vertices in S. Also, the subgraph induced by S is k-connected. For example, the induced subgraph by the dominators in Figure 1 constructs (2, 2)-CDS.

Many researches have been devoted to construct (k, m)-CDS. However, some are heuristics ([7] and [28]) which cannot guarantee the quality of the obtained solution, and others are approximation algorithms ([10], [13], [16], [17], [22], [24], [25], [26], [29], [31] and [32]) whose performance ratios are weak. Also, since most of the algorithms are designed for special type of graphs such as unit disk graphs, these algorithms cannot be applied to WSN which is represented on general graphs.

In this research, we present an exact algorithm for the minimum (k, m)-CDS problem using an integer programming (IP) formulation which can be applied to general graphs. Although obtaining an optimal solution of the problem is an important research goal, to the best of our knowledge, no other exact algorithm has been developed for the problem (note

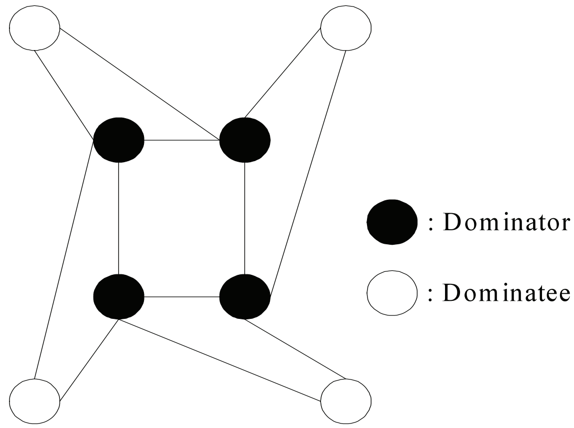

Figure 1: Example of the network which shows (2, 2)-CDS.

that, when k=m=1, the problem is a well-known NP-hard problem as shown in [8].).

One of the main characteristics of IP is that, we can obtain some information on the lower bound. If we relax the integrality condition of the IP formulation, the optimal value of the linear programming (LP) relaxation provides a lower bound on the optimal value of IP. Note that, the obtained lower bound can be used to measure the quality of the solution obtained from the heuristic and the approximation algorithms.

The remainder of this paper is organized as follows. In Section 2, we present a mathematical formulation for the minimum (k, m)-CDS problem, and introduce additional constraints to reduce the computational time to solve the problem. In Section 3, we suggest an optimal algorithm for the problem and discuss how the optimal value of the LP relaxation can be obtained. In Section 4, we compare the performance of the algorithm with the heuristic (CGA) which is developed in [28]. Finally, in Section 5, we conclude this paper and provide some suggestions for the future researches.

2. Mathematical formulation

This section presents a mathematical formulation for the minimum (k, m)-CDS problem. Before introducing our formulation, we review the objective and restrictions of the problem in detail. The objective is clear. When a graph is given with two natural numbers, k and m, we want to minimize the number of vertices which will be chosen to be included in the dominating set. Two restrictions exist for the dominating set. First restriction is, if a certain vertex is chosen to be a dominatee, then it has at least m adjacent dominators. Second restriction is that the induced subgraph which is constructed by the chosen dominators is k-connected.

To solve the problem using a mathematical formulation, binary decision variable x_i is defined as follows.

$$x_i = \begin{cases} 1, & \text{if vertex } i \text{ is chosen to be a dominator,} \\ 0, & \text{otherwise (=vertex } i \text{ is selected to be a dominatee.).} \end{cases}$$

Then the objective can be represented as follows.

$$minimize \sum_{i \in V} x_i \tag{1}$$

To represent the first restriction, suppose that vertex i is chosen to be a dominatee (x_i=0) and let the set of neighbors of vertex i be denoted by $N(i)$. Then, there should exist at least m dominators among the neighbors of the vertex i.

The first restriction can be expressed as follows (if vertex i is chosen to be a dominator (x_i=1), constraint (2) holds trivially.).

$$\sum_{j \in N(i)} x_j \geq m(1 - x_i), \ \forall i \in V. \tag{2}$$

To represent the second restriction, we use $Menger's$ theorem. Given two nonadjacent vertices s and t of G, an st-vertex-cut is $S \subseteq V \setminus \{s, t\}$ such that the removal of S from G disconnectes s and t. We let $c(s, t)$ denote the minimum size vertex cut separating s and t. Then the maximum number of internally disjoint st-paths for any two distinct and nonadjacent nodes s and t can be identified using st-vertex-cut.

Theorem 1 *[5] (Menger's Theorem) In any graph G with two distinct and nonadjacent vertices s and t,*

$$p(s,\ t) = c(s,\ t).$$

Moreover, the result can be extended for any two distinct, not necessarily nonadjacent, vertices.

Theorem 2 *[5] If G has at least one pair of nonadjacent vertices,*

$$\kappa(G) = min\ \{\ p(s,\ t) : s,\ t \in V,\ s \neq t,\ st \notin E\ \}.$$

Therefore, when the graph G has at least one pair of nonadjacent vertices, the connectivity of a graph G is equal to the size of a minimum vertex cut of G.

In this research, we assume that the subgraph induced by the dominators has at least one-pair of nonadjacent vertices because the complete graphs (=any two vertices are adjacent) rarely occur in (k, m)-CDS problem instances.

For the given two nonadjacent vertices s and t of G, we let C_{st} denote the set of all st-vertex-cuts. Then the second restriction can be represented as the following.

$$\sum_{i \in c_{st}} x_i \geq k(x_s + x_t - 1), \ \forall \ s, t \in V, s \neq t, st \notin E, \forall \ c_{st} \in C_{st}. \tag{3}$$

Constraints (3) indicate that, if two nonadjacent vertices, s and t are chosen to be dominators ($x_s = x_t = 1$), then the size of every st-vertex-cut which is composed of the selected dominators should be greater than or equal to k. On the other hand, suppose that if at least one of s and t is selected to be a dominatee (this means, at least one of x_s and x_t is equal to 0), then constraint (3) holds trivially. Since constraint (3) holds for all nonadjacent pairs of vertices of V, the existence of k independent paths among the vertices which will be included in the dominating set are guaranteed.

The mathematical formulation for the minimum (k, m)-CDS problem is complete with constraints (2) and (3), but we also introduce two additional constraints. They may strengthen the formulation, which will reduce the computational burden in the branch-and-bound algorithm to solve the formulation.

Assume that, it is guaranteed that the number of vertices selected in a (k, m)-CDS is greater than or equal to two. Then, the following first additional constraints can be added to the formulation.

$$\sum_{j \in N(i)} x_j \ \geq kx_i, \ \forall \ i \in V. \tag{4}$$

Constraints (4) imply that, if vertex i is chosen to be a dominator (x_i=1), then it needs to have at least k dominators among the vertices which are adjacent to vertex i due to the k independent paths requirement.

Note that, constraints (4) and (2) can be combined into one constraint as follows.

$$\sum_{j \in N(i)} x_j \ \geq kx_i + m(1 - x_i), \ \forall \ i \in V. \tag{5}$$

Before introducing the second constraints, we consider the following two cases for the given two natural numbers, k and m.

The first case is when m is greater than or equal to k ($m \geq k$). Then, since the vertex connectivity for the nonadjacent pairs of dominators is at least k and every dominatee has at least m dominators as neighbors, there exist k independent paths for all pairs of vertices according to *Lemma* 1.

Lemma 1 *[27] If G is a k-connected graph, and G' is obtained from G by adding a new node x with at least k neighbors in G, then G' is also a k-connected graph.*

Therefore the following constraints (6) can be added to the formulation (let c be a vertex cut whose removal disconnects the graph and C be the set of all vertex cuts.).

$$\sum_{i \in c} x_i \geq k, \ \forall c \in C. \tag{6}$$

Now consider the second case when m is smaller than k ($m < k$). Similarly, the following constraints (7) can be added to the formulation.

$$\sum_{i \in c} x_i \geq m, \ \forall c \in C. \tag{7}$$

In conclusion, when a graph is given with two numbers, k and m, we can add the following second additional constraints (8) to the formulation.

$$\sum_{i \in c} x_i \geq min(m, k), \ \forall c \in C. \tag{8}$$

3. Optimal algorithm

In this section, we first propose an optimal algorithm for the minimum (k, m)-CDS problem, then we discuss how the optimal value of the LP relaxation of the mathematical formulation can be obtained. The optimal value of the LP relaxation can be used estimate the lower bound of the optimal value of the minimum (k, m)-CDS problem when the optimal algorithm fails to obtain the optimal solution in a reasonable amount of time.

Since the number of constraints (3) can be exponential, constructing the formulation in full is impractical. Therefore, the constraints need to be treated implicitly rather than explicitly. The core idea of handling the exponential number of constraints is generating the constraints only when needed as shown in [3].

We first solve the formulation optimally which is composed of constraints (5) with binary integer restrictions on variables x_i ($i \in V$). Then we construct an induced subgraph G' of G as follows.

Procedure to construct induced subgraph G' of G:

Step 1. Identify vertices whose corresponding value of decision variable x_i is equal to 0.

Step 2. Delete the vertices which are identified in **Step** 1 from G, and remove the edges which are incident to the deleted nodes.

Step 3. Construct the induced subgraph G' of G with remaining vertices V' and edges E'.

Then we check whether the induced subgraph G' is a (k, m)-CDS or not. This can be checked easily as follows. Since the constraints (2) represent the first restriction, any generated integer solution obtained from solving the formulation satisfies the m dominance requirement. Also, whether the induced subgraph satisfies the k connectivity requirement or not can be checked in polynomial time.

Note that, for the given natural number k, complexity of computing the connectivity of a graph G is $O(|E|^{1/2}|V|^2)$ as shown in [23]. Also, much efficient algorithms have been developed for a specific value of k. For example, when k=2, complexity of computing the vertex connectivity is $O(m+n)$ as shown in [21], where m is the number of edges and n is the number of vertices of the graph G.

If the connectivity of the induced subgraph is greater than or equal to k, we obtained a minimum (k, m)-CDS and stop. Otherwise, we identify some of the constraints (3) which are violated by the current solution, and add them to the formulation. Then, we optimally solve the enlarged formulation again, and this process is continued until a (k, m)-CDS is obtained.

In this research, when there exist several vertex cuts which are violated by the current solution for the given two nonadjacent vertices s and t of G, we use the minimum st-vertex-cut. This may tighten the formulation the most, and it probably reduces the computation time to solve the problem.

We first explain the idea of obtaining the minimum st-vertex-cut, then our optimal algorithm which uses the minimum cut will be discussed.

Let $G = (V, E)$ and suppose that two nonadjacent vertices s and t $(\in V)$ are given. Then the minimum st-vertex-cut can be obtained using the maximum flow algorithm [15].

Procedure to obtain the minimum st-vertex-cut:

Step 1. Replace each edge uv $(\in E)$ by arcs uv and vu with infinite capacity.

Step 2. Replace each vertex v $(\in V, \neq s, \neq t)$ by two vertices $v^{'}$ and $v^{''}$ and add an arc $v^{'}v^{''}$ with unit capacity.

Connect all the arcs that were coming to v to $v^{'}$, and similarly, connect all the arcs that were going out of v to $v^{''}$.

Step 3. Calculate the maximum flow from s to t using the maximum flow algorithm and identify a minimum st-cut.

Step 4. Take the vertices which correspond to the arcs in the minimum st-cut as the vertices of a minimum st-vertex-cut.

Step 2 can be illustrated as shown in Figure 2. Note that the arcs with infinite capacity cannot be included in a minimum st-cut identified in **Step 3**.

Optimal algorithm for the minimum (k, m)-CDS problem which uses the **Procedure to construct induced subgraph $G^{'}$ of G** and the **Procedure to obtain the minimum st-vertex-cut** can be described as follows (Note that, our optimal algorithm is designed for a general natural value of k.).

Procedure to find a minimum (k, m)-CDS:

Step 1. Construct the formulation which is composed of constraints (5) with binary integer restrictions on x_v $(v \in V)$.

Step 2. Solve the formulation optimally and construct an induced subgraph $G^{'}$ using the **Procedure to construct induced subgraph $G^{'}$ of G.**

Step 3. Assign a very large positive value to $\kappa(G^{'})$.

Step 4. **For All** nonadjacent pairs of the vertices in $V^{'}$ **Do**

Step 4.1. Let the first and the second vertex be s and t, respectively.

Step 4.2. Identify a minimum number of st-vertex-cut using the **Procedure to obtain the minimum st-vertex-cut**.

Step 4.3. If the minimum number of the cut is greater than or equal to $\kappa(G^{'})$, proceed to the next pair of vertices.

Step 4.4. Otherwise, update $\kappa(G^{'})$ with the minimum number of the cut and

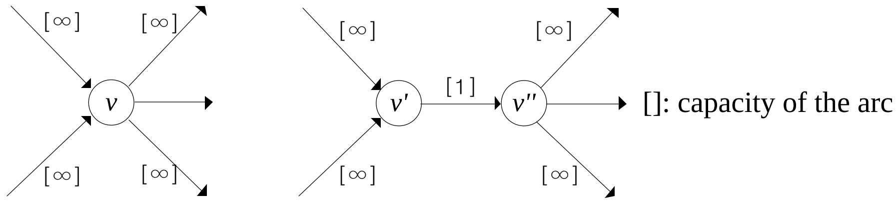

Figure 2: Splitting vertex v into two vertices $v^{'}$ and $v^{''}$.

construct the violated constraint (3) using the cut with the pair of vertices s and t, then add it to the formulation.

Step 4.5. If the minimum number of the st-vertex-cut is less than $min(m, k)$, construct the violated constraint (8) using the cut, and add it to the formulation.

End For

Step 5. If $\kappa(G^{'})$ is greater than or equal to k, we obtained an optimal solution. Stop the procedure.

Step 6. Otherwise, return to **Step 2**.

In this section, we also discuss how the optimal value of the LP relaxation can be obtained. The optimal value of the LP relaxation can be obtained similar to the **Procedure to find a minimum (k, m)-CDS** except for several differences. The first difference is that the integrality restriction is dropped when we construct the formulation. The second difference is that the value of x_v obtained from solving the formulation is used as the capacity of arc $v^{'}v^{''}$ when we find the minimum st-vertex-cut. Lastly, the **Procedure to construct induced subgraph $G^{'}$ of G** is not used, and the stopping criterion is changed from the k-connectivity requirement to no violated constraint (3). The detailed procedure of finding the optimal value of LP relaxation can be described as follows.

Procedure to find an optimal value of LP relaxation:

Step 1. Construct the formulation which is composed of constraints (5) with $0 \leq x_v \leq 1$ ($v \in V$).

Step 2. Solve the formulation optimally, and let the obtained value of x_v as $\bar{x}_v$, $v \in V$.

Step 3. **For All** nonadjacent pairs of vertices in V **Do**

Step 3.1. Let the first and the second vertex be s and t, respectively.

Step 3.2. Construct a directed graph G' using **Step** 1 and 2 in **Procedure to obtain the minimum st-vertex-cut**, and replace capacity of arc $v'v''$ with $\bar{x}_v$.

Step 3.3. Find a minimum st-cut using **Step** 3 in **Procedure to obtain the minimum st-vertex-cut**.

Step 3.4. If the size of cut is greater than or equal to $k(\bar{x}_s + \bar{x}_t - 1)$, proceed to the next pair of vertices.

Step 3.5. Otherwise, apply **Step** 4 in **Procedure to obtain the minimum st-vertex-cut** to identify the vertices of a minimum st-vertex-cut, and construct the violated constraint (3) with the pair of vertices s and t. Then add it to the formulation.

Step 3.6. If the size of the cut is less than $min(m, k)$, construct the violated constraint (8) using the cut, and add it to the formulation.

End For

Step 4. If no constraint is generated in **Step** 4, we obtained an optimal solution. Stop the procedure.

Step 5. Otherwise, return to **Step** 2.

Since the maximum number of possible pairs of nodes among dominators is $(|V| \times (|V| - 1))/2$ and LP relaxation can be solved in polynomial time, we also can solve the LP relaxation of the formulation in polynomial time as discussed in [3].

4. Computational results

In this section, we report the performance of the optimization algorithm which is proposed in Section 3 to construct (k, m)-CDS. To evaluate the performance of the algorithm, we implemented both the heuristic (CGA) developed in [28] and our optimal algorithm, then we compared the obtained (k, m)-CDS for the same instance. For each approach, the run time and the size of the (k, m)-CDS are reported.

Two approaches were implemented in C++ and ILOG CPLEX 12.1 was used as an optimization software. All experiments were run on Intel Pentium Dual Core (2.20 GHz) with 3GB RAM, and the running time of the algorithm is given in seconds.

To simulate the wireless sensor networks, we generated the sensors which were placed at random within a 1000 meters $\times$ 1000 meters area, and the sensors have the same transmission range of 350 meters (The size of the area and the number of sensors used in this paper are the same as the ones which were used in [28]. Although two transmission ranges, 250 meters and 350 meters, were used in [28], we fix the transmission range as 350 meters for simplicity.).

In this experiment, there exist two parameters, k and m. We performed the experiments for some combinations of the fixed two parameters. For each type of the combinations, we changed the number of vertices from 30 to 150 with an increment of 10. Tables 1, 2 and 3 consider the cases when the value of k is 1, 2 and 3, respectively (the value of m varies from 1 to 3 for each table).

Here, $|V|$ denotes the number of the vertices of the given graph, $|(k,m)$-CDS$|$ indicates the size of the obtained (k,m)-CDS from each approach, Opt. represents the optimal algorithm and $\lceil Z_{LP} \rceil$ means the next highest integer by rounding up the optimal value of the LP relaxation of our formulation.

In some cases, **Step** 3 in **Procedure to find an optimal value of LP relaxation** does not significantly increases the value of $\lceil Z_{LP} \rceil$. Therefore, if the value of LP relaxation does not improves even after adding some number of cuts, we stopped the adding the cut and reported the terminated value of LP relaxation.

Computational results show that, our optimal approach outperformed the heuristic (CGA) in terms of the quality of the solution, and the optimal solution can be obtained within a reasonable amount of time. Furthermore, we can observe that the gap between our optimal approach and the value of the LP relaxation of our formulation is very tight (The run time to obtain the value of $\lceil Z_{LP} \rceil$ is omitted in Tables 1, 2, 3 and 4, because it requires very small amount of time.).

Additionally, we tested the performance of the two approaches when the number of vertices is huge. The values of k and m are set as one, and we changed the number of vertices from 200 to 1000 with an increment of 100. In some cases, our optimal algorithm failed to obtain the solution in 3600 seconds and these cases are denoted by * symbols. However, since information on lower bound (the value of LP relaxation) is provided, the quality of the solutions obtained from the heuristic (CGA) can be evaluated.

Table 1: Comparison of the two approaches when k=1 and m=1 $\sim$ 3.

$\|V\|$	k=1 and m=1					k=1 and m=2					k=1 and m=3				
	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$
	CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.	
30	0.000	0.016	6	6	5	0.000	0.016	9	9	9	0.000	0.031	14	13	12
40	0.016	0.109	6	6	5	0.016	0.047	10	10	10	0.000	0.063	15	13	13
50	0.000	0.109	7	6	5	0.000	0.047	9	8	8	0.000	0.031	15	12	12
60	0.000	0.078	6	5	5	0.016	0.172	11	10	10	0.016	0.125	16	12	12
70	0.016	0.078	6	5	5	0.016	0.063	10	9	9	0.000	0.188	13	12	12
80	0.016	0.172	5	5	5	0.016	0.266	10	9	9	0.000	0.094	14	13	13
90	0.016	0.688	6	5	5	0.016	0.391	10	9	9	0.016	0.078	14	13	13
100	0.016	1.234	7	6	5	0.016	0.313	12	10	10	0.016	0.281	14	13	13
110	0.016	0.266	6	5	5	0.031	0.984	11	10	9	0.016	0.203	15	12	12
120	0.016	1.484	6	6	5	0.016	0.219	11	9	9	0.031	0.234	14	12	12
130	0.016	2.250	7	6	5	0.031	0.547	10	9	9	0.016	0.328	17	13	13
140	0.031	0.547	6	5	5	0.047	0.797	11	10	10	0.047	0.469	14	13	12
150	0.047	2.375	7	6	5	0.000	0.672	10	9	9	0.047	0.734	16	13	13

Table 2: Comparison of the two approaches when k=2 and m=1 ~ 3.

$\|V\|$	k=2 and m=1					k=2 and m=2					k=2 and m=3				
	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$	Run time(s)		$\|(k,m)\text{-CDS}\|$		$\lceil Z_{LP} \rceil$
	CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.	
30	0.000	0.016	6	6	5	0.000	0.000	11	10	10	0.016	0.047	16	15	15
40	0.000	0.016	7	6	5	0.016	0.000	10	10	10	0.000	0.016	16	13	13
50	0.016	0.047	7	7	5	0.000	0.016	11	9	9	0.016	0.031	18	13	13
60	0.031	0.047	7	7	5	0.078	0.125	12	11	11	0.031	0.031	16	14	14
70	0.016	0.047	6	6	5	0.031	0.016	9	9	9	0.094	0.063	17	14	14
80	0.063	0.078	9	8	5	0.063	0.016	13	9	9	0.141	0.094	17	13	13
90	0.109	0.109	9	8	5	0.047	0.016	10	9	9	0.078	0.109	16	13	13
100	0.063	0.266	9	7	5	0.313	0.016	14	10	10	0.188	0.281	18	14	14
110	0.188	0.297	9	7	5	0.063	0.063	13	9	9	0.141	0.234	16	13	13
120	0.141	0.250	9	7	5	0.063	0.250	11	10	9	0.719	0.250	14	13	13
130	0.281	0.172	8	7	5	0.156	0.203	11	10	10	0.172	0.328	15	13	13
140	0.172	0.344	8	7	5	0.391	0.063	12	9	9	0.609	0.578	16	14	14
150	0.438	0.516	9	7	5	0.578	0.141	12	10	9	1.203	0.438	17	14	14

Table 3: Comparison of the two approaches when k=3 and m=1 ∼ 3.

$\|V\|$	k=3 and m=1					k=3 and m=2					k=3 and m=3				
	Run time(s)		$\|(k,m)$-CDS$\|$		Z_{LP}	Run time(s)		$\|(k,m)$-CDS$\|$		$\lceil Z_{LP} \rceil$	Run time(s)		$\|(k,m)$-CDS$\|$		$\lceil Z_{LP} \rceil$
	CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.		CGA	Opt.	CGA	Opt.	
30	0.000	0.031	7	7	6	0.031	0.031	11	11	10	0.125	0.000	16	15	15
40	0.281	0.063	9	9	8	0.047	0.078	12	11	10	0.203	0.078	17	17	17
50	0.438	0.438	13	12	6	0.750	0.375	15	14	13	0.422	0.047	17	16	16
60	1.828	0.438	12	10	6	2.797	0.563	15	14	14	1.703	0.156	16	15	15
70	1.391	0.125	10	8	5	1.531	1.672	12	12	11	3.906	0.234	20	18	18
80	3.594	0.547	10	9	5	1.063	0.453	12	11	9	3.484	0.250	16	14	14
90	3.781	2.188	11	10	6	2.328	4.375	14	13	10	10.172	1.547	19	17	17
100	2.563	0.328	10	9	5	6.141	1.500	15	12	11	11.438	1.734	19	17	17
110	9.953	3.922	10	9	5	10.469	0.969	15	12	10	5.297	3.219	16	15	15
120	7.531	1.516	11	9	5	2.688	1.125	14	11	9	23.953	2.094	16	13	13
130	20.516	7.734	12	9	5	21.156	7.844	15	13	10	12.156	1.203	17	14	14
140	52.766	1.250	12	9	5	21.391	6.359	16	13	10	14.313	7.563	18	15	14
150	0.656	2.531	10	8	5	11.266	7.469	14	12	9	51.984	4.281	18	15	13

Table 4: Comparison of the two approaches when the number of vertices is huge.

$\|V\|$	Run time(s)		$\|(k,m)$-CDS$\|$		$\lceil Z_{LP} \rceil$
	CGA	Opt.	CGA	Opt.	
200	0.06	3.61	6	6	5
300	0.22	22.30	6	6	5
400	0.38	99.34	7	6	5
500	0.84	599.23	6	6	5
600	1.19	365.77	7	6	5
700	1.98	383.86	7	6	5
800	2.52	1391.48	7	6	5
900	3.67	*	7	*	5
1000	4.89	*	7	*	5

5. Conclusions

In this research, we considered the minimum k-connected m-dominating set problem to guarantee the routing flexibility and the fault tolerance which are hard to achieve in connected dominating set. We first proposed a mathematical formulation for the problem. Since the number of constraints of the formulation can be exponential, we developed an optimal algorithm which handles the constraints implicitly rather than explicitly.

To illustrate the performance of the suggested optimal algorithm, we implemented a previously developed heuristic and compared the sizes of the obtained (k, m)-CDS with our optimal algorithm. Computational results indicated that, our algorithm obtained the optimal solution in a reasonable amount of time except for some cases and outperformed the heuristic in terms of the obtained quality of the solution.

Furthermore, when our optimal algorithm failed to obtain an optimal solution within a reasonable amount of time, the tight lower bound from the LP relaxation of our formulation can provide information about the quality of the solution obtained from the existing heuristic algorithm. Therefore, improving the lower bound can be a good research topic for the future research. Also, developing a heuristic algorithm which uses the value of the LP relaxation can be an another good research topic.

References

[1] K.M. Alzoubi, P. J. Wan, O. Frieder, Message-Optimal Connected Dominating Sets in Mobile Ad Hoc Networks, 3^{rd} *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ACM, 157-164, Lausanne, Switzerland, 2002.

[2] P. Basu and J. Redi, Movement control algorithms for realization of fault tolerant ad hoc robot networks, *IEEE Networks*, 18(4):36-44, 2004.

[3] D. Bertsimas and R. Weismantel, *Optimization over Integers*, Dynamic Ideas, Massachusetts, 2005.

[4] J. Blum, M. Ding, A. Thaeler and X. Cheng, *Connected dominating set in sensor networks and MANETs*, Handbook of Combinatorial Optimization, Kluwer Acacemic Publishers, Massachusetts, 2004.

[5] J. A. Bondy and U. S. R. Murty, *Graph Theory*, Springer, 2008.

[6] J. L. Bredin, E. D. Demaine, M. Hajiaghayi and D. Rus, Deploying sensor networks with guaranteed capacity and fault tolerance, 6^{th} *ACM international symposium on mobile ad hoc networking and computing*, ACM SIGMOBILE, 309-319, Urbana-Champaign, Illinois, 2005.

[7] F. Dai and J. Wu, On Constructing k-Connected k-Dominating Set in Wireless Network, *Journal of parallel and distributed computing*, 66:947-958, 2006.

[8] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, San Francisco, 1979.

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

[10] D. Kim, W. Wang, X. Li, Z. Zhang and W. Wu, A New Constant Factor Approximation for Computing 3-connected m-dominating Sets in Homogeneous Wireless Networks, *INFOCOM*, IEEE, 1-9, San Diego, CA, 2010.

[11] H. Koskinen and J. Karvo, On improving connectivity of static ad-hoc networks by adding nodes, 4^{th} *annual Mediterranean workshop on ad hoc networks*, IEEE Communications Society, 169-178, Porquerolles, France, 2005.

[12] F. Kuhn, T. Moscibroda and R. Wattenhofer, Fault-tolerant clustering in ad hoc and sensor networks, 26^{th} *international conference on distributed computing systems*, IEEE Computer Society, Lisboa, Portugal, 2006.

[13] Y. Li, Y. Wu and C. Ai and R. Neyah, On the construction of k-connected m-dominating sets in wireless networks, *Journal of combinatorial optimization*, 23:118-139, 2012.

[14] S. Y. Ni, Y. C. Tseng, Y. S. Chen and J. P. Sheu, The broadcast storm problem in a mobile ad hoc network, *Wireless networks*, 8(2-3):153-167, 2002.

[15] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Dover Publications, N.Y., 1998.

[16] W. Shang, F. Yao, P. Wan and X. Hu, Algorithms for minimum m-Connected k-Dominating Set Problem, *Lecture Notes in Computer Science*, 4616:182-190, 2007.

[17] W. Shang, F. Yao, P. Wan and X. Hu, On minimum m-connected k-dominating set problem in unit disc graphs, *Journal of combinatorial optimization*, 16(2):99-106, 2008.

[18] P. Sinha, R. Sivakumar and V. Bharghavan, Enhancing ad hoc routing with dynamic virtual infrastructures, 20^{th} *annual joint conference of the IEEE computer and communications societies*, Technical Committees on Computer Communications (TCCC) of the Societies, 3:1763-1772, Anchorage, Alaska, 2001.

[19] I. Stojmenović and J. Wu, *Mobile ad hoc networking*, John Wiley & Sons, Inc. 2005.

[20] D. S. Ryl, I. Stojmenović, J. Wu, *Energy-Efficient Backbone Construction, Broadcasting, and Area Coverage in Sensor Networks*, Handbook of sensor networks, I. Stojmenović(Editor), 343-380, John Wiley & Sons, Inc. 2005.

[21] R. E. Tarjan, Depth first search and linear graph algorithms, *Siam Journal on Computing*, 146-160, 1972.

[22] M. T. Thai, N. Zhang, R. Tiwari and X. Xu, On approximation algorithms of k-connected m-dominating sets in disk graphs, *Theoretical Computer Science*, 385(1-3):49-59, 2007.

[23] K. Thulasiraman and M.N.S. Swamy, *Graphs: Theory And Algorithms*, Wiley, New York, 1992.

[24] R. Tiwari and M. T. Thai, On Enhancing Fault Tolerance of Virtual Backbone in a Wireless Sensor Network with Unidirectional Links, *Sensors: Theory, Algorithms, and Applications*, 61:3-18, 2012.

[25] F. Wang, M. T. Thai and D. Z. Du, On the construction of 2-connected Virtual Backbone in Wireless Network, *IEEE Transactions on Wireless Communications*, 8(3):1230-1237, 2009.

[26] W. Wang, D. Kim, M. K. An, W. Gao, X. Li, Z. Zhang and W. Wu, On Construction of Quality Fault-Tolerant Virtual Backbone in Wireless Networks, *IEEE/ACM Transactions on Networking*, Early Access Articles, 2012.

[27] D. B. West, *Introduction to Graph Theory*, Prentice-Hall, 2001.

[28] Y. Wu, F. Wang, M. T. Thai and Y. Li, Constructing k-connected m-dominating sets in wireless sensor networks, *Military Communications Conference*, Orlando, FL, 2007.

[29] Y. Wu and Y. Li, Construction algorithms for k-connected m-dominating sets in wireless sensor networks, 9^{th} *ACM international symposium on mobile ad hoc networking and computing*, ACM SIGMOBILE, May:83-90, Hong Kong, 2008.

[30] J. Yu, N, Wang, G. Wang and D. Yu, Connected dominating sets in wireless ad hoc and sensor networks - A comprehensive survey, *Computer Communications*, 36(2): 121-134, 2013.

[31] J. Zhang, X. Gao and W. Wu, Algorithms for connected set cover problem and fault-tolerant connected set cover problem, *Theoretical Computer Science*, 410:812-817, 2009.

[32] J. Zhou, Z. Zhang, W. Wu and K. Xing, A greedy algorithm for the fault-tolerant connected dominating set in a general graph, *Journal of Combinatorial Optimization*, June:1-10, 2013.