entropy

MDPI

Article

Calculation of the Connected Dominating Set Considering Vertex Importance Metrics

Francisco Vazquez-Araujo [1], Adriana Dapena [1,*], María José Souto-Salorio [2] and Paula M. Castro [1]

[1] Department of Computer Engineering, Universidade da Coruña, Campus de Elviña, 15071 A Coruña, Spain; fjvazquez@udc.es (F.V.-A.); paula.castro@udc.es (P.M.C.)

[2] Department of Computation, Universidade da Coruña, Campus de Elviña, 15071 A Coruña, Spain; maria.souto.salorio@udc.es

* Correspondence: adriana.dapena@udc.es; Tel.: +34-981-167-000

Received: 21 December 2017; Accepted: 25 January 2018; Published: 28 January 2018

Abstract: The computation of a set constituted by few vertices to define a virtual backbone supporting information interchange is a problem that arises in many areas when analysing networks of different natures, like wireless, brain, or social networks. Recent papers propose obtaining such a set of vertices by computing the connected dominating set (CDS) of a graph. In recent works, the CDS has been obtained by considering that all vertices exhibit similar characteristics. However, that assumption is not valid for complex networks in which their vertices can play different roles. Therefore, we propose finding the CDS by taking into account several metrics which measure the importance of each network vertex e.g., error probability, entropy, or entropy variation (EV).

Keywords: connected dominating set; complex networks; graph entropy; vertex importance

1. Introduction

Complex networks are playing an increasingly important role in a large number of areas (for instance, biology, physics, Social Science, etc.) [1,2]. The identification of the most relevant set of vertices in such networks allows us to better control the spread of disease [3], design marketing strategies [4], optimize limited resource allocation [5], and so on. In particular, connected dominating sets (CDSs) are natural candidates for vertices to be used for information interchange in any kind of network. They can also be used as a virtual backbone infrastructure in ad hoc wireless networks [6–9].

A CDS is a subset of vertices constituting a connected induced subgraph such that every vertex in the network is either in the CDS or has a neighbour in it [8,9]. For the effectiveness of a virtual backbone, the underlying CDS must be small in size. Since the problem of finding a minimum-sized CDS has been shown to be non-deterministic polynomial-time hardness (NP-hard) [8], the design of approximation algorithms has become an important issue for the study of CDSs. Thus, for the last twenty years, many researchers have explored approximation algorithms distinct to that of Guha and Khuller [10]. The heuristics for obtaining the CDS can be divided into two groups: the first is focused on evolving a CDS by growing a small trivial CDS [10], and the second group strives to find disconnected independent sets of vertices which are joined through a minimum spanning tree or Steiner tree [6,11,12]. Our approach is based on the algorithm presented in [11], which constructs the CDS in three phases: firstly, the dominating sets are determined by iteratively identifying the maximum degree of vertices to discover the highest cover vertices; secondly, they are connected through a Steiner tree; and finally, the algorithm prunes this tree to form the CDS without redundant vertices.

In practice, it is natural to assume that the vertices of the graph have some positive weights. In the context of wireless ad hoc networks, these weights usually reflect residual energy or capabilities

of a node for a specific task. Thus, the computation of a CDS with a minimum number of nodes, also referred to as a minimum CDS (MCDS), can be extended to a minimum weighted CDS (MWCDS), by means of incorporating weights into each node with the objective of finding the CDS that minimizes the cost function of the total weight. Experimental literature is mainly focused on benchmarking and applications of algorithms for (weighted) connected dominating set problems. Ambühl and Erlebach [13] were the first to design a constant factor approximation algorithm for the MWCDS on a unit disk graph (UDG). They divided the region in square partitions and used the topological characteristics of the UDG to determine the vertices that covered each partition. Later, Huang et al. [14] proposed a strategy to reduce the computational cost by partitioning the whole plane into squares, and forming them into blocks. The MWCDS for each block was computed first, and then combined together to find the MWCDS of the graph. More recently in [15] the authors proposed including a new condition to the MWCDS of a UDG: all vertices in the MWDS must be connected with k vertices to guarantee redundancy. All these algorithms were designed for the UDG without taking into account the condition of minimizing the dominant size.

Unlike previous approaches, our work is focused on finding an MWCDS for any graph, without considering the topological characteristics. In the first stage, a set of dominating sets (DSs) is constructed taking into account the weight and the degree, and they are subsequently connected. Since we are also interested in reducing the dominant size, we include a prune stage to reduce the number of vertices in the MWCDS.

In addition, we will explain the relationship between the MWCDS and some theoretical concepts of metrics like error probability, entropy, and entropy variation. In particular, the study of the entropy to measure the information in a graph is a relevant topic in many areas. For instance, Rashevsky [16] proposed the concept of graph entropy to study the relationships between the topological properties of graphs and the information content in an organism. Mowshowitz and Dehmer [17,18] contextualized various entropy-based measures proposed from Rashevsky's work. In [19], Kajdanowic and Morzy presented several simulation results oriented towards examining the usefulness of the entropy concept for different graph models. Moreover, in a recent paper, Ai [20] introduced the concept of entropy variation as a measurement of the influence of each vertex in the graph. In this sense, our study is oriented towards finding an MWCDS considering that graph information.

This paper is organized as follows. Section 2 includes some definitions for the understanding of this work. Section 3 explains the computation of a CDS considering vertex importance metrics. Some results are shown in Section 4, and some concluding remarks are briefly made in Section 5.

2. Previous Definitions

A graph $G = (V; E)$ consists of a set of vertices, known as a *vertex set* and denoted by V, and of a set of edges, called the *edge set* and denoted by E. The vertices correspond to the objects to be modelled, while the edges indicate some relationship between pairs of these objects. For instance, in the case of social networks, the individuals of the population and the friendships among them are respectively represented by vertices and edges.

In our settings, the graphs are usually undirected i.e., if u is directly connected to v, then also v is directly connected to u.

Definition 1 (**Connected Dominating Set (CDS)).** *A subset D of V dominates in G if every vertex of $V - D$ has at least one neighbour in D. A subset D of V is connected and dominates if D dominates and the subgraph induced by D is connected.*

This definition states that the CDS is a subset of vertices such that any pair of vertices can be joined by a path in the network and any vertex in the network either belongs to the CDS (CDS vertex) or has a neighbour in the CDS (non-CDS vertex).

In the following, we focus on graphs whose vertices have positive weights. For example, in the context of wireless ad hoc networks, these weights usually reflect capabilities of a node for a specific task.

For this purpose, we assume that a function of vertex reliability, denoted as f, is given.

Definition 2 (Graph Entropy). *Let $f : V \to \mathbb{R}$ be an arbitrary function representing the vertex reliability in a graph G. For a vertex v we define*

$$p(v) = \frac{f(v)}{\sum_{v \in V} f(v)}. \tag{1}$$

Since $\sum_{v \in V} p(v) = 1$, $p(v)$ can be interpreted as a probability mass function so that the entropy of a graph G can be defined as follows

$$\begin{aligned} I_f(G) &= -\sum_{v \in V} p(v) \log_2(p(v)) = -\sum_{v \in V} \frac{f(v)}{\sum_{v \in V} f(v)} \log_2 \left(\frac{f(v)}{\sum_{v \in V} f(v)} \right) \\ &= \log_2 \left(\sum_{v \in V} f(v) \right) - \sum_{v \in V} \frac{f(v)}{\sum_{v \in V} f(v)} \log_2 f(v). \end{aligned} \tag{2}$$

This definition corresponds to the concept of entropy of a discrete random variable introduced by Shannon in [21]. In [20], the entropy defined as in Equation (2) is interpreted as a measure of the amount of information encoded in the network structure, although it is not used as a metric of the vertex influence, also referred to as *vertex importance*. Thus, the entropy variation is introduced by the authors in [20] to give an idea of such an influence.

Definition 3 (Entropy Variation (EV)). *For a reliability function f, the entropy variation produced by removing the vertex v is defined by*

$$EV(v) \quad = \quad I_f(G) - I_f(G_v), \tag{3}$$

where $G_v = (V', E')$ denotes the subgraph of G with a vertex set given by $V' = V - \{v\}$ and whose edge set E' verifies that $e = \{u, w\} \in E'$ if and only if $u \neq v$ and $w \neq v$.

3. CDS Computation Based on Vertex Importance

Taking into account that in real networks the vertices represent objects with different characteristics, we propose finding a CDS of a graph $G = (V; E)$ satisfying two conditions: (1) the CDS must have few vertices; and (2) the CDS must maximize a metric related to the vertex importance. We will consider a general importance function which computes the importance of any vertex in the network according to some metrics and present several examples, including the entropy variation.

In general, for a fixed reliability function, f, and its associated probability function, p, we consider that an importance function is any function from V to $\mathbb{R}$ in a graph $G = (V; E)$. We will denote it by T. For each vertex $v \in V$, the real number $T(v)$ denotes the importance of the vertex v.

For a fixed reliability function f in a graph $G = (V; E)$ and any importance function T, we denote T-CDS as the CDS verifying the following conditions,

- firstly, there is no other CDS of the G with a lower number of vertices;
- secondly, given several CDSs with the same number of vertices, the T-CDS maximizes a cost function given by

$$J(D) \quad = \quad \sum_{v \in D} T(v), \tag{4}$$

where D is the set of vertices of the CDS.

For some graphs with a regular structure, such as those briefly described in the following examples, it is possible to propose a simple procedure that guarantees the computation of the optimum T-CDS. However, the computation of an MCDS for a random graph is an NP-hard problem [10] and the algorithms proposed in the literature give only a CDS with a reduced number of nodes without guaranteeing the condition of minimum size. As a consequence, the computation of the T-CDS is also an NP-hard problem. Section 3.2 presents the generalized algorithm proposed in this paper for the construction of a T-CDS in a suboptimal way.

Example 1 (Bipartite Graph). *Let G be a bipartite graph of N vertices where each vertex is defined by an importance function T. The CDS with minimum size is formed by two vertices connected by an edge. Computing the value $t_{uv} = T(u) + T(v)$ for each pair of connected vertices u and v, the cost function J for any transformation T is maximized when the T-CDS consists of vertices u and v with maximum value of t_{uv}.*

Example 2 (Cycle Graph). *Let G be a cycle graph of N vertices where each vertex is defined by an importance function T. The CDS with minimum size is formed by a set of $N-2$ connected vertices. Computing the value $t_{uv} = T(u) + T(v)$ for each pair of connected vertices u and v, the cost function J for any transformation T is maximized when the T-CDS is formed by all the vertices excluding those u and v vertices with minimum values of t_{uv}.*

3.1. Selection of the Importance Function

The T-CDS defined above can be particularized using any importance function T. In particular, we will consider the following vertex importance metrics given by the definitions

$$T_1(v) = \log_2 p(v), \tag{5}$$
$$T_2(v) = -p(v)\log_2(p(v)), \tag{6}$$
$$T_3(v) = EV(v) = I_f(G) - I_f(G_v). \tag{7}$$

Example 3. *Given a graph $G = (V; E)$ and the importance function $T_1(v) = \log_2 p(v)$, we denote T_1-CDS as the CDS maximizing the following cost function:*

$$J(D) = \sum_{v \in D} \log_2 p(v). \tag{8}$$

Note that this function is related to the concept of error probability. Considering the vertices in the CDS i.e., $v \in D$, the error probability associated to the transmitted message through those vertices with such importances is given by

$$P_E(D) = 1 - \prod_{v \in D} p(v). \tag{9}$$

Note that

$$\min P_E(D) \equiv \max \prod_{v \in D} p(v) \equiv \max \log_2 \left(\prod_{v \in D} p(v) \right) \equiv \max \sum_{v \in D} \log_2 p(v). \tag{10}$$

Example 4. *Given a graph $G = (V; E)$ and the importance function $T_2(v) = -p(v)\log_2 p(v)$, we denote T_2-CDS as the CDS maximizing the following cost function (entropy),*

$$J(D) = -\sum_{v \in D} p(v)\log_2 p(v). \tag{11}$$

Example 5. *Given a graph $G = (V; E)$ and the importance function $T_3(v) = EV(v) = I_f(G) - I_f(G_v)$, we denote T_3-CDS as the CDS maximizing the following cost function,*

$$J(D) = \sum_{v \in D} \left(I_f(G) - I_f(G_v) \right), \tag{12}$$

where $I_f(G) - I_f(G_v)$ is the entropy variation defined in Equation (3).

3.2. Algorithm

Since the computation of an MCDS is an NP-hard problem [11], several approaches have been proposed in the literature [6,7,11]. In particular, the suboptimal algorithm proposed in [11] consists of three phases: the first one computes a disconnected DS; the second one connects the different DS subgraphs to build an initial CDS; and finally, the third phase prunes the resulting CDS so that the number of vertices is minimized. The problem presented by us in this paper is more complex because the optimum solutions would require the computation of all possible MCDSs and selection of the best according to the metric to be maximized. To reduce the computational cost we propose an algorithm which includes the metric at each step in order to determine the node to include or prune when there are several alternatives.

Figure 1 shows a flowchart of the proposed algorithm. Each of the three phases uses a different metric to find the node to add to or remove from the CDS. The following sections explain each phase in detail.

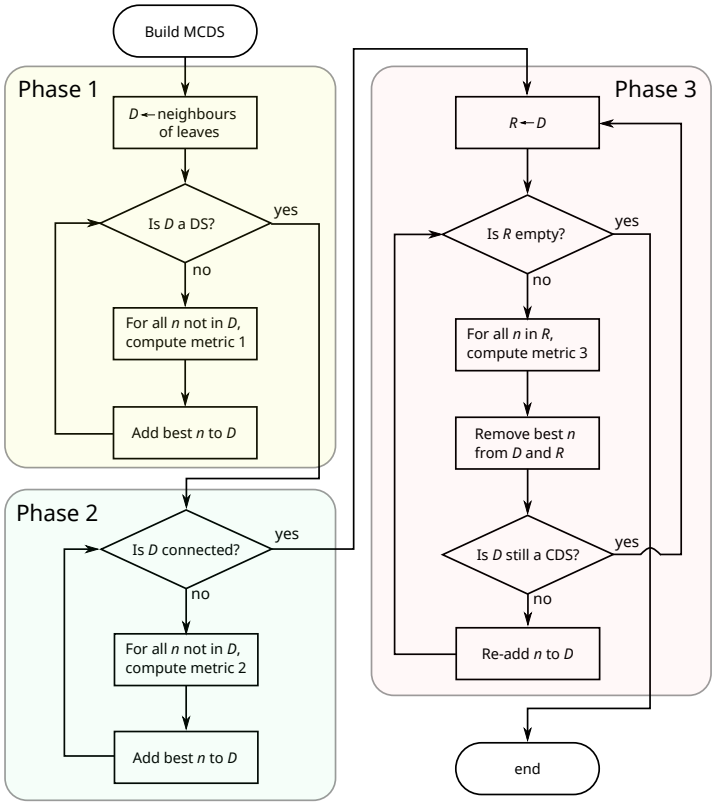

Figure 1. Flowchart for the computation of the minimum connected dominating set (MCDS).

3.2.1. Phase 1: Find the Disconnected DS

The DS is initialized with the nodes that are neighbours of leaf nodes, since they must be in the final CDS. Then it is constructed by adding vertices until all of them are either part of the DS or neighbours of it. The vertices are chosen in order of highest value for the metric 1 indicated in the

flowchart. We define this metric for each vertex as the number of vertices covered by the DS together with the vertex. In case there are more than one vertex with the same number, the vertex with the highest value of the importance function will be chosen.

The pseudo-code of the algorithm implemented for this first phase is detailed in Algorithm 1. In this pseudo-code, note that

$$\text{ARGMAX}A, C = \arg\max_{n \in A} C(n). \tag{13}$$

Algorithm 1 Algorithm for phase 1: Disconnected DS.

```
procedure PHASE 1(G)
    DS ← NEIGHBORS(LEAVES(G))
    while DS ∪ NEIGHBORS(DS) ≠ G do
        R ← G − DS
        for all vertex ∈ R do
            DS′ ← DS ∪ {vertex}
            N(vertex) ← SIZE(DS′ ∪ NEIGHBORS(DS′))
        end for
        L1 ← ARGMAX(R, N)
        vertex ← FIRST(ARGMAX(L1, J))
        DS ← DS ∪ {vertex}
    end while
end procedure
```

3.2.2. Phase 2: Compute Initial CDS

The CDS is constructed by adding vertices which connect the different disconnected subgraphs of the DS resulting from the previous phase. The order in which they are added (metric 2 in the flowchart) is: first, those that connect a higher number of disconnected DS subgraphs; then, in the case of several vertices connecting the same number of subgraphs, those with the highest degree; and finally, if there are multiple vertices of equal degree, that with the highest value of the importance function.

In this phase we need to make use of an auxiliary algorithm for finding the number of disconnected subgraphs, as shown in Algorithm 2.

Algorithm 2 Auxiliary algorithm for phase 2.

```
function N_SUBGRAPHS(G)
    n ← 0
    R ← G
    while R ≠ ∅ do
        S ← {FIRST(R)}
        while S ≠ S ∪ NEIGHBORS(S) do
            S ← S ∪ NEIGHBORS(S)
        end while
        R ← R − S
        n ← n + 1
    end while
    return n
end function
```

The pseudo-code of the algorithm used for connecting the CDS is shown in Algorithm 3. Again, note that

$$\text{ARGMIN}A, C = \arg\min_{n \in A} C(n). \tag{14}$$

Algorithm 3 Algorithm for phase 2: CDS.

```
1: procedure PHASE 2(G, DS)
2:     CDS ← DS
3:     R ← G − CDS
4:     while IS_DISCONNECTED(CDS) do
5:         for all vertex ∈ R do
6:             N(vertex) ← N_SUBGRAPHS(CDS ∪ {vertex})
7:         end for
8:         L1 ← ARGMIN(R, N)
9:         L2 ← ARGMAX(L1, DEGREE)
10:        vertex ← FIRST(ARGMAX(L2, J))
11:        CDS ← CDS ∪ {vertex}
12:        R ← R − {vertex}
13:    end while
14: end procedure
```

3.2.3. Phase 3: Prune Vertices

The final CDS results from pruning the CDS obtained from the previous phase, since there can be vertices added in the first phase that are no longer necessary after the second phase. The vertices are removed according to metric 3 in order of the lowest degree; in the case of several vertices with the same degree, those with lowest degree to nodes in the CDS are chosen. If there are several such nodes, that with a lowest value of the importance function is chosen. Note that every time a vertex is pruned the process must be restarted.

The pseudo-code of the algorithm used for this third phase is shown in Algorithm 4.

Algorithm 4 Algorithm for phase 3: Pruning.

```
1: procedure PHASE 3(G, CDS)
2:     R ← CDS
3:     while R ≠ ∅ do
4:         L1 ← ARGMIN(R, DEGREE)
5:         L2 ← ARGMIN(L1, DEGREE(CDS))
6:         vertex ← FIRST(ARGMIN(L2, J))
7:         TCDS ← CDS − {vertex}
8:         if CONNECTED(TCDS) & TCDS ∪ NEIGHBORS(TCDS) = G then
9:             CDS ← TCDS
10:            PHASE 3(G, CDS)
11:        else
12:            R ← R − {vertex}
13:        end if
14:    end while
15: end procedure
```

4. Results and Discussion

In this section we perform several simulations to verify that the proposed algorithm allows us to compute a CDS formed by a reduced number of vertices with a good performance in terms of maximization of importance functions.

In the literature, we can find several theoretical graph models proposed to construct graphs that would display certain properties frequently appearing in empirical graphs (see, for instance the review in [19]). In particular, we will consider the UDG [8] and the small-world model [19].

4.1. Unit Disk Graph

We have considered an ad hoc wireless network which is a decentralized type of wireless network characterized by a lack of fixed communication infrastructure, so that the selection of vertices forwarding data is dynamically made by considering the current network connectivity. Several researchers have proposed using the CDS as a virtual backbone in these networks as an alternative to the fixed routing infrastructure in classical wired networks [6,7]. The virtual backbone represents the "skeleton" of the entire network and is used to frequently exchange routing information (traffic conditions, neighbourhood information, etc.) and broadcast a message in the network.

For the UDG model the network is defined by $G = (V; E)$, where the vertices in V are embedded in the Euclidean plane. We assume that the maximum transmission range is the same for all the vertices in the network and it is unit scaled. There exists an edge $\{u, v\} \in E$ if u and v are in the maximum transmission range of each other i.e., the Euclidean distance is $d(u, v) \leq 1$. Figure 2 shows an example of a UDG of 50 vertices with the coverage radius of each vertex. Figure 3 shows the values of $T_1(v) = \log_2(p(v))$, $T_2(v) = -p(v)\log_2(p(v))$, and $T_3(v) = EV(v)$, obtained generating f according to a uniform distribution in the interval $(0, 1]$. It is interesting to observe that $T_1(v) = \log_2(p(v))$ and $T_2(v) = -p(v)\log_2(p(v))$ have the same trend but $T_3(v) = EV(v)$ presents some differences which are marked in red in the figure.

In Figures 4–7 four different CDS can be observed: a CDS without using a vertex importance metric (called 1-CDS), the T_1-CDS, the T_2-CDS, and the T_3-CDS. Note that there are variations between the four configurations although all of them are constituted by the same number of vertices (19 out of initial 50 vertices).

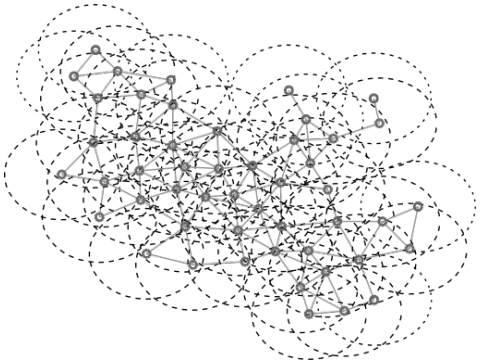

Figure 2. Example of a unit disk graph (UDG).

We have considered graphs with 20, 50, and 80 vertices with randomly generated connections. The function f of each vertex follows a standard uniform distribution. We have generated 1 000 realizations of different graphs for each one of those sizes. The CDS corresponding to each approach above depicted is computed so that its size is minimal and in the case of vertex importance metrics, the respective cost function given by Equations (8), (11), or (12), must be maximized for the obtained CDS. For all these CDSs, we have calculated the maximum value of every importance function, denoted by b, and its deviation with respect to that maximum so that we can obtain the parameter

$$\gamma = \frac{(r-b)^2}{b^2}, \tag{15}$$

where r is the value of the importance function obtained by any CDS.

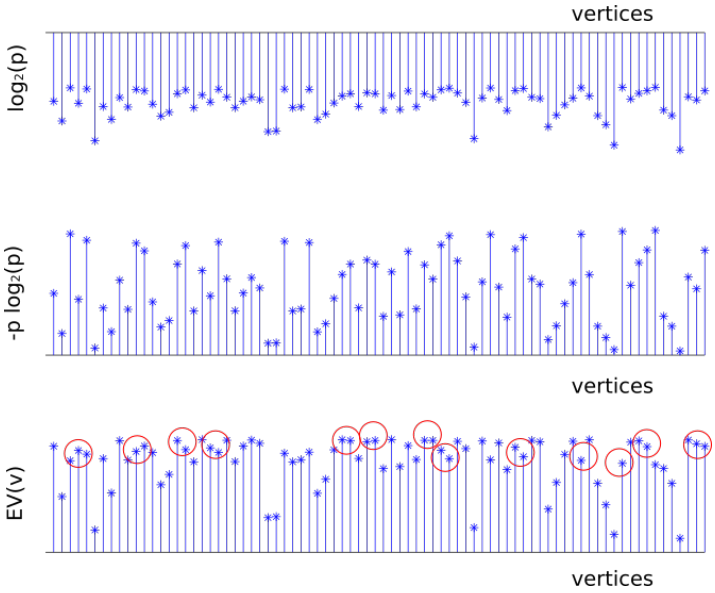

Figure 3. Importance functions T. It is interesting to observe that T_1 and T_2 have the same trend but T_3 presents some differences which are marked in red.

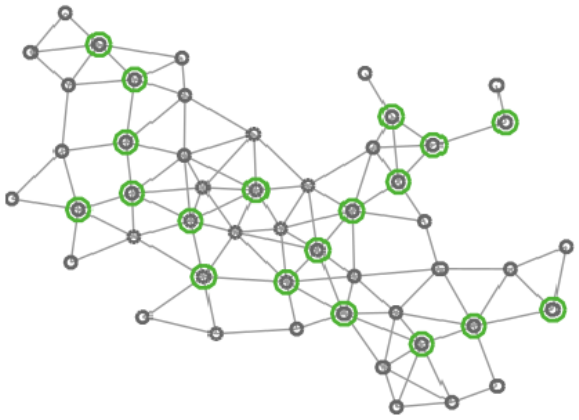

Figure 4. Graph of 50 vertices. Vertices in the 1-CDS are marked in green.

Figure 5. Graph of 50 vertices. Vertices in the T_1-CDS are marked in red.

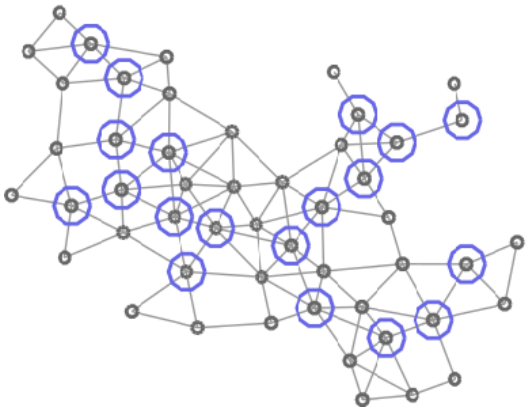

Figure 6. Graph of 50 vertices. Vertices in the T_2-CDS are marked in blue.

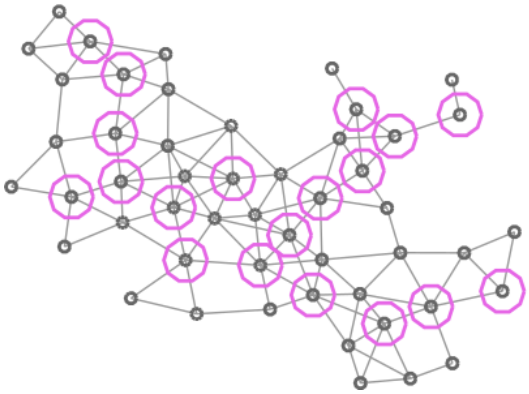

Figure 7. Graph of 50 vertices. Vertices in the T_3-CDS are marked in pink.

Table 1 shows the number of times, expressed as a percentage, that the four CDS achieve the minimum size. Data shown in the table demonstrate that all the CDS are similar in terms of size. This table also shows the mean deviation obtained by averaging the results of evaluating Equation (15) throughout 1 000 realizations, with b being the minimum value of the four CDS sizes. This deviation is very small because of difference in size is less than two vertices for all the network sizes. Note also that T_1-CDS and T_2-CDS give exactly same results.

Table 1. Size of the CDSs for the UDG.

CDS	Graph Size					
	20 Vertices		50 Vertices		80 Vertices	
	Percentage	Mean Deviation	Percentage	Mean Deviation	Percentage	Mean Deviation
1-CDS	95	0.0007	82	0.0005	70	0.0005
T_1 and T_2-CDS	92	0.0012	78	0.0006	67	0.0005
T_3-CDS	92	0.0012	79	0.0006	65	0.0006

Now, we wish to verify that T_1-CDS reduces the error probability. For this purpose, we evaluated the importance function shown in Equation (8) for the four CDSs. The table included in Figure 8 shows the result percentages in terms of number of times the maximum value of the importance function is achieved, i.e., $\gamma = 0$. We see that T_1-CDS and T_2-CDS exhibit the same performance, at 86.5% for 20 vertices. The difference with respect to 1-CDS and T_3-CDS is also remarkable for 50 and 80 vertices. Figure 8 also depicts the cumulative distribution function (CDF) of the function γ given in Equation (15) (curves of T_1-CDS and T_2-CDS are represented in the same line). We observe that the

difference appears depending on the applied method reduces with the graph size. Note that 1-CDS shows poor performance since the number of times it achieves the maximum value of the importance function is lower than that exhibited by the other methods.

Following the computer experiments, we compared the entropy of the computed CDS. For that purpose, we have evaluated Equation (11) to calculate the percentage of times in which each CDS achieves the maximum value of the importance function. The table included in Figure 9 shows the new results. It is apparent that T_1-CDS and T_2-CDS achieve the best performances with a considerable gap with respect to the rest of the algorithms. The same observation can be made if we see the CDF in Figure 9: T_1-CDS and T_2-CDS have a high probability regardless of the network size, while the other methods present a considerable error. Again, 1-CDS provides the worst results.

Percentage of Times Where $\gamma = 0$			
CDS	**20 Vertices**	**50 Vertices**	**80 Vertices**
1-CDS	53.6	24.6	17.6
T_1 and T_2-CDS	86.5	66.6	59.4
T_3-CDS	65.6	43.6	34.2

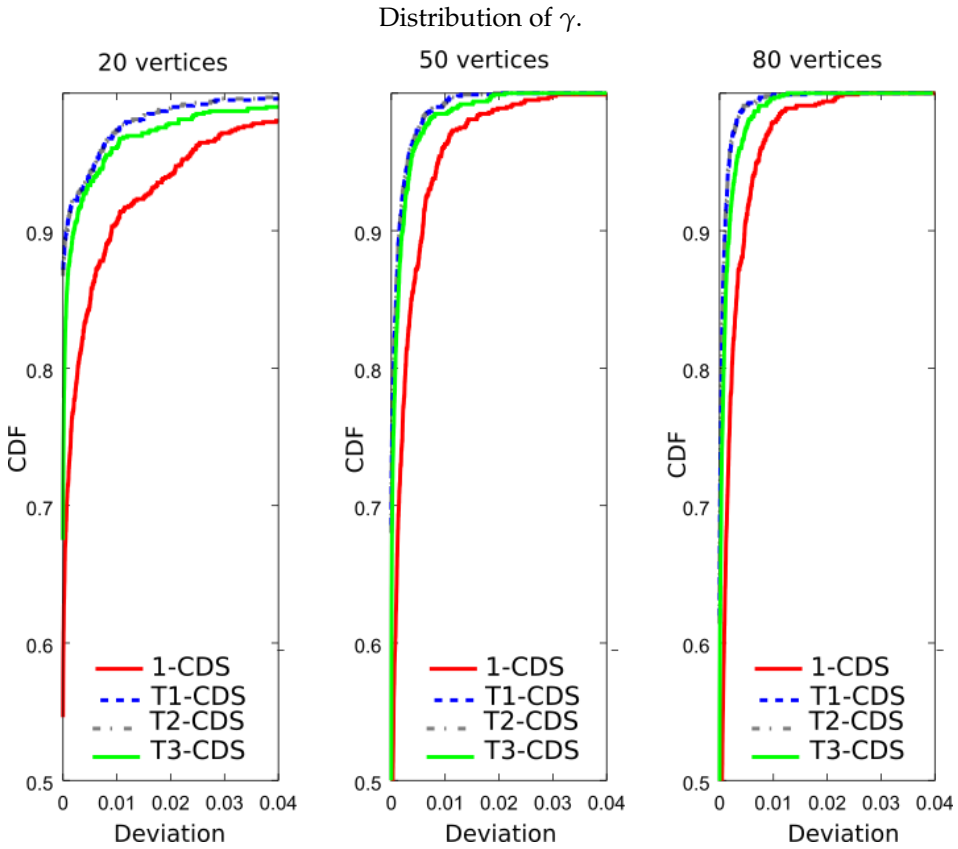

Figure 8. Analysis of the UDG: percentage of times where $\gamma = 0$ and cumulative distribution function (CDF) of γ for a metric based on the error probability, $J(D) = \sum_{v \in D} \log_2(p(v))$.

Percentage of Times Where $\gamma = 0$			
CDS	**20 vertices**	**50 vertices**	**80 vertices**
1-CDS	49.4	19.8	11.1
T_1 and T_2-CDS	87.5	70.7	66.5
T_3-CDS	65.7	42.2	32.8

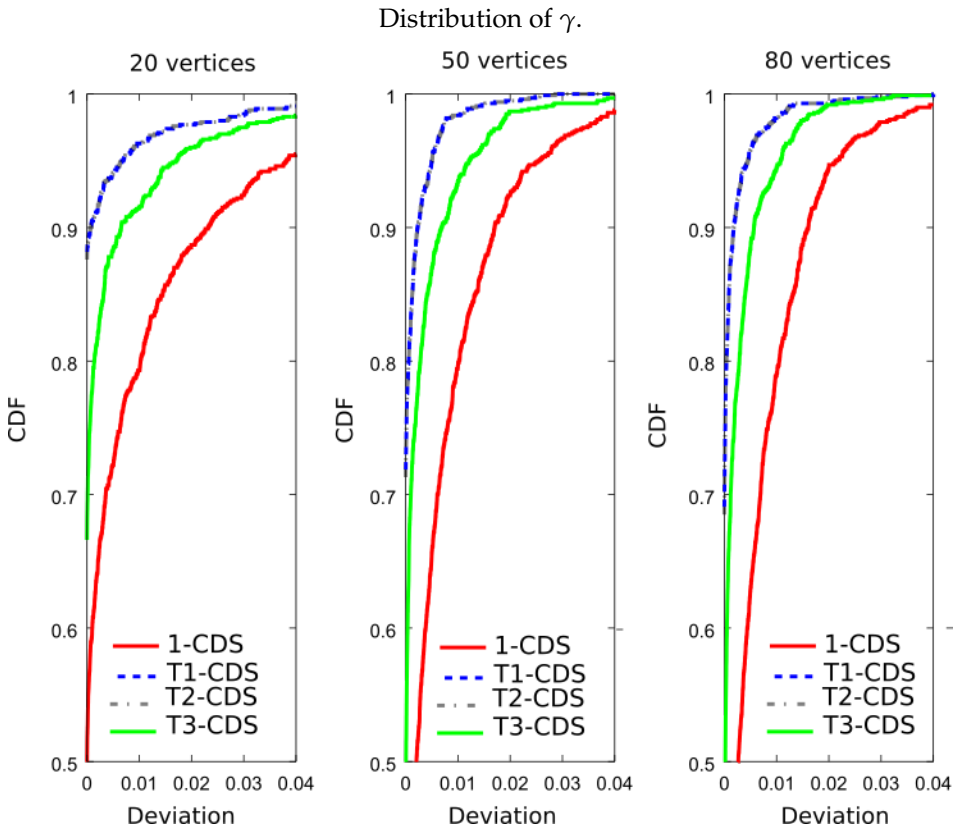

Figure 9. Analysis of the UDG: percentage of times where $\gamma = 0$ and CDF of γ for a metric based on the entropy, $J(D) = -\sum_{v \in D} p(v) \log_2(p(v))$.

Finally, we compared the CDS in terms of the sum of entropy variation. We evaluated Equation (12) for the vertices of the obtained CDS. Figure 10 shows that T_3-CDS gives the best performances in terms of the percentages above explained although the differences in CDF compared to the T_1-CDS and T_2-CDS are negligible.

Therefore, it can be said that the algorithms proposed in this paper are correctly working in the sense of maximizing their cost function using a reduced number of vertices, and that the metrics defined in Equations (8) and (11) show same performances, while the metric of entropy variation (see Equation (12)) presents differences that we will try to analyse. 1-CDS provides the worst results for all the defined metrics since the algorithm only considers the vertex degree.

Percentage of Times Where $\gamma = 0$			
CDS	**20 vertices**	**50 vertices**	**80 vertices**
CDS	52.10	22.7	15.3
T_1 and T_2-CDS	65.6	41.6	34.8
T_3-CDS	86.3	67.4	60.3

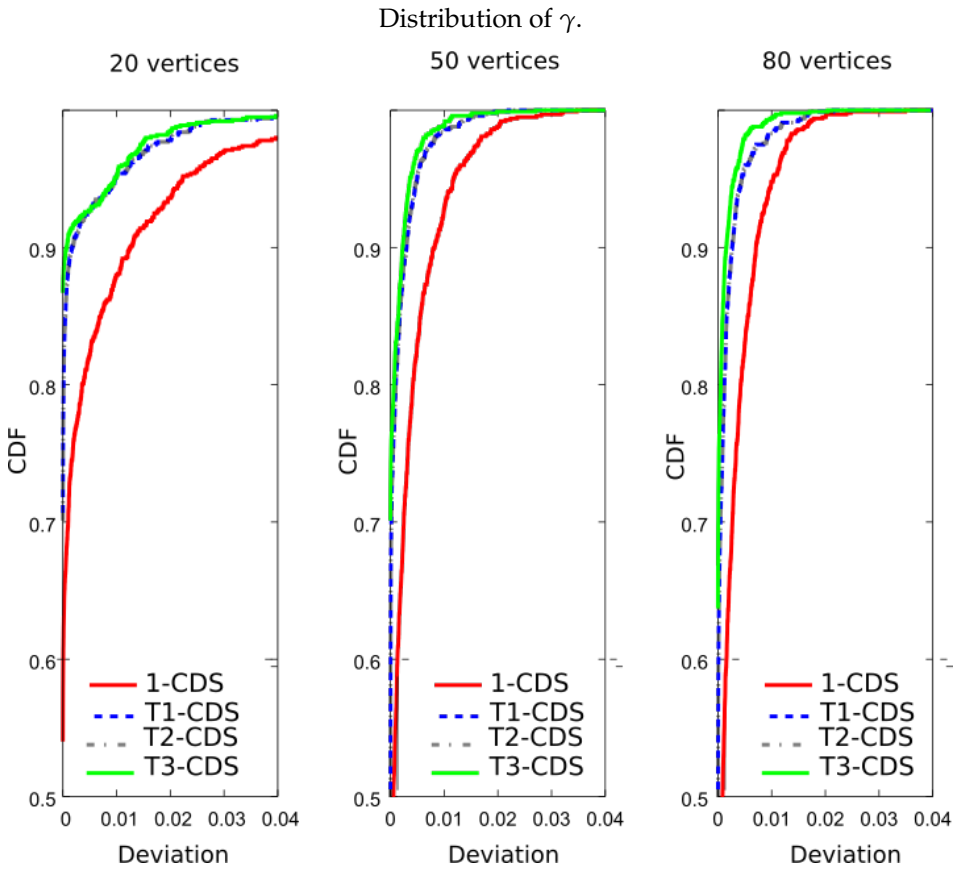

Figure 10. Analysis of the UDG: percentage of times where $\gamma = 0$ and CDF of γ for a metric based on the sum of entropy variation, $J(D) = \sum_{v \in D} EV(v)$.

4.2. Small-World Model

The small-world model was introduced in [22]. According with this model, a set of N vertices is organized into a regular circular graph where each vertex is directly connected to a mean number of K-nearest neighbours. For each edge in the graph, the target node with probability β is rewired. When $\beta = 1$, the small world graph becomes the random graph. In our simulations, we generated 1 000 realizations of graphs with 20, 50, and 80 vertices with $\beta = 0.5$ and $K = N/2$ (i.e, a mean number of 10, 25, and 40 connections for any vertex). The function f follows a uniform distribution in the interval $(0, 1]$.

Table 2 shows the number of times, expressed as a percentage, where the CDS achieved the minimum size. We can see that, as occurs in the UDG, there is no a remarkable difference between the four CDSs. However, the deviation with respect to the optimum value is considerably higher than for the UDG graph. This means that in those occasions where the CDS does not achieve the best size, the CDS size differs in more than two vertices from the optimum.

We evaluated the deviation of Equation (15) in order to verify the correct behavior of the proposed algorithm. Tables 3–5 show the result percentages in terms of number of times achieving the maximum

value of the importance functions in Equations (8), (11), and (12), respectively. We can see that each T-CDS gives the best result for the corresponding importance metric. In general, the results are similar to those obtained with UDG.

Table 2. Size of CDS for the small-world model.

CDS	Graph Size					
	20 Vertices		50 Vertices		80 Vertices	
	Perc.	Mean Deviation	Perc.	Mean Deviation	Perc.	Mean Deviation
1-CDS	94	0.013	85	0.016	88	0.009
T_1 and T_2-CDS	92	0.016	86	0.014	88	0.008
T_3-CDS	94	0.010	85	0.017	87	0.010

Table 3. Analysis of the small-world model for a metric based on the error probability, $J(D) = \sum_{v \in D} \log_2(p(v))$.

Percentage of Times Where $\gamma = 0$			
CDS	**20 Vertices**	**50 Vertices**	**80 Vertices**
1-CDS	53.1	44.5	45.0
T_1 and T_2-CDS	82.0	73.9	73.9
T_3-CDS	64.3	55.2	51.5

Table 4. Analysis of the small-world model for a metric based on the entropy, $J(D) = -\sum_{v \in D} p(v) \log_2(p(v))$.

Percentage of Times Where $\gamma = 0$			
CDS	**20 Vertices**	**50 Vertices**	**80 Vertices**
1-CDS	51.7	42.3	41.5
T_1 and T_2-CDS	84.6	75.3	76.7
T_3-CDS	62.7	54.8	50.5

Table 5. Analysis of the small-world model for a metric based on the sum of entropy variation, $J(D) = \sum_{v \in D} EV(v)$.

Percentage of Times Where $\gamma = 0$			
CDS	**20 Vertices**	**50 Vertices**	**80 Vertices**
CDS	52.3	44.0	43.7
T_1 and T_2-CDS	65.7	52.4	48.6
T_3-CDS	80.6	75.0	74.4

4.3. Importance Function Comparison

In order to compare the three importance functions, we will consider a graph formed by N nodes, denoted by v_i, with $f(v_i) = f_i = i/N$, where $i = 1, 2, ..., N$. Figure 11 shows the three importance functions for $N = 20$, 50, and 80. From the top figure, we can observe that the

$T_1(v_i) = \log_2(p(v_i))$ function is increasing with respect to f. In fact, for our discrete distribution, we can find the analytical expression

$$T_1(v_i) = \log_2(p_i) = \log_2(f_i) - \log_2\left(\sum_{k=1}^{N} f_k\right)$$
$$= \log_2\left(\frac{i}{N}\right) - \log_2\left(\sum_{k=1}^{N} \frac{k}{N}\right) = \log_2\left(\frac{i}{N}\right) - \log_2\left(\frac{N+1}{2}\right). \quad (16)$$

In Figure 11, we can see that the curves converge for large number of vertices ($N = 50$ and $N = 80$).

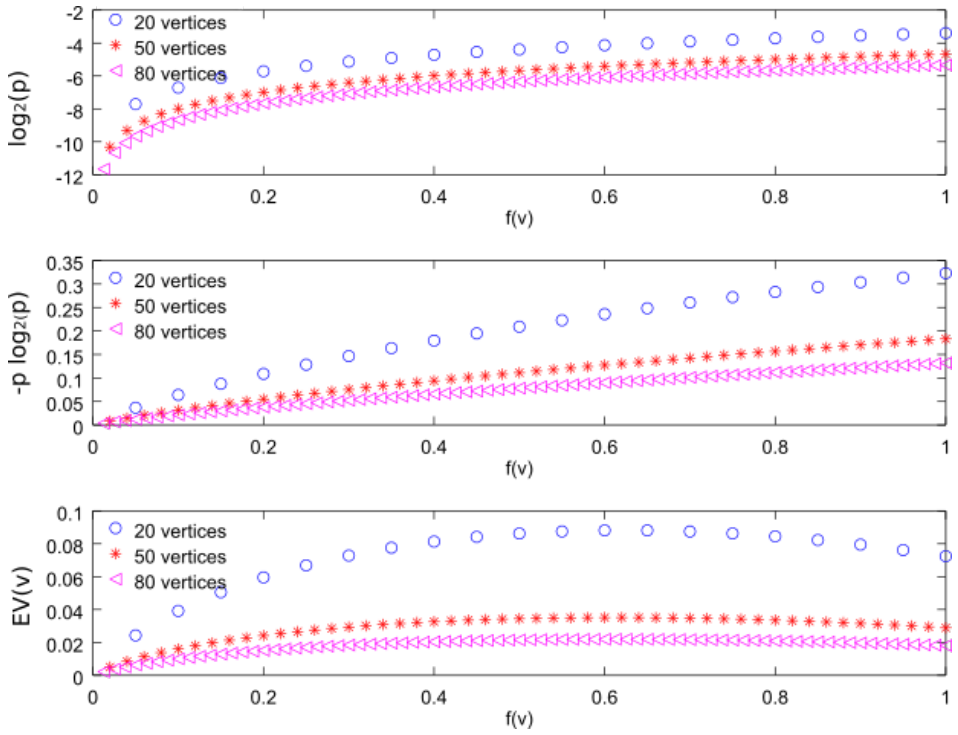

Figure 11. Importance functions for values of $f(v) = i/N$ with $i = 1, 2, ..., N$ and $N = 20, 50$, and 80.

The second importance function represented in the figure is $T_2(v_i) = -p(v_i)\log_2(p(v_i))$. This metric allows us to maximize the entropy. The analytical expression for $f(v_i) = f_i = i/N$ is given by

$$T_2(v_i) = -p_i \log_2(p_i) = -\frac{f_i}{\sum_{k=1}^{N} f_k}\left(\log_2(f_i) - \log_2\left(\sum_{k=1}^{N} f_k\right)\right)$$
$$= \frac{2i}{N^2+N}\left(\log_2\left(\frac{N+1}{2}\right) - \log_2\left(\frac{i}{N}\right)\right). \quad (17)$$

Note that the N^2 term has an important influence on the curve values, as it can be seen in Figure 11, but again the curves converge for large number of vertices. The importance function $T_2(v_i) = -p(v_i)\log_2(p(v_i))$ increases with respect to f, as happens with $T_1(v_i) = \log_2(p(v_i))$, and, for this reason, T_1-CDS and T_2-CDS give the same results in the simulation figure above presented.

Finally, the bottom figure represents the third importance function considered in this work i.e., $T_3(v_i) = EV(v_i)$. We can observe that it is an increasing function with f, similarly to the first two

functions, although for higher f it decreases with a smaller slope. By evaluating Equation (2) for $f(v_i) = f_i = i/N$, we can express this importance function as follows:

$$\begin{aligned} T_3(v_i) &= I_f(G) - I_f(G_{v_i}) \\ &= I_f(G) - \log_2\left(\frac{N+1}{2} - \frac{i}{N}\right) + \frac{1}{\frac{N+1}{2} - \frac{i}{N}}\left(\sum_{k=1}^{N}\frac{k}{N}\log_2\left(\frac{k}{N}\right) - \frac{i}{N}\log_2\left(\frac{i}{N}\right)\right) \\ &= I_f(G) - \log_2\left(\frac{N^2+N-2i}{2N}\right) + \frac{2N}{N^2+N-2i}\left(\sum_{k=1}^{N}\frac{k}{N}\log_2\left(\frac{k}{N}\right) - \frac{i}{N}\log_2\left(\frac{i}{N}\right)\right), \end{aligned} \quad (18)$$

where $I_f(G)$ is constant for a given graph. As can be seen in Figure 11, we can directly observe that the maximum values are close to 0.60 regardless of the vertex number.

Using simulations we confirmed that the value 0.60 obtained for $f(v_i) = f_i = i/N$ is also valid for random samples of a uniform distribution. For that, we generated 1 000 samples of a uniform distribution and computed $EV(v)$ using Equation (12). We found that the maximum values for $N = 20$, 50, and 80 vertices are, respectively, 0.6055, 0.6023, and 0.6059.

Therefore, we can conclude that the importance function $EV(v)$ behaves in a similar way to the other two when $f(v) < 0.60$. For this reason, the T_3-CDS does not maximize the error probability or the entropy.

5. Conclusions

In this paper we proposed selecting the CDS of a graph by incorporating vertex importance metrics defined in order to maximize a desired cost function such as error probability, entropy, or entropy variation. We have shown that finding the optimum CDS is very simple for the bipartite graph and the cycle graph. For the general case, the computation of such a CDS is an 1/N-hard problem and we proposed an algorithm which selects the vertices in the CDS taking into account the defined importance metric and the vertex degree. Several simulation results show that the proposed algorithm allows us to find a CDS formed by a reduced number of nodes, similarly to previous methods, with the advantage of maximizing the objective metric.

Acknowledgments: This work has been funded by the Xunta de Galicia (ED431C 2016-045, ED341D R2016/012), the Agencia Estatal de Investigación of Spain (TEC2015-69648-REDC, TEC2016-75067-C4-1-R, TIN2017-85160-C2-1-R) and ERDF funds of the EU (AEI/FEDER, UE).

Author Contributions: All the authors have equally contributed to this work.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CDS	connected dominating set
CDF	cumulative distribution function
DS	dominating set
EV	entropy variation
MCDS	minimum connected dominating set
MWCDS	minimum weighted connected dominating set
NP-hard	non-deterministic polynomial-time hardness
UDG	unit disk graph

References

1. Stam, J.S.; Reijneveld, J. Graph theoretical analysis of complex networks in the brain. *Nonlinear Biomed. Phys.* **2007**, *1*, doi.10.1186/1753-4631-1-3.
2. Strogatz, S. Exploring complex networks. *Nature* **2001**, *410*, 268–276.

3. Kitsak, M.; Gallos, L.K.; Havlin, S.; Liljeros, F.; Muchnik, L.; Stanley, H.E.; Makse, H.A. Identification of influential spreaders in complex networks. *Nat. Phys.* **2010**, *6*, 888–893, doi:10.1038/nphys1746.
4. Richardson, M.; Domingos, P. Mining knowledge-sharing sites for viral marketing. In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02), New York, NY, USA, 23–26 July 2002; pp. 61–70.
5. Csermely, P.; Korcsmáros, T.; Kiss, H.J.; London, G.; Nussinov, R. Structure and dynamics of molecular networks: A novel paradigm of drug discovery. *Pharmacol. Ther.* **2013**, *138*, 333–408, doi:10.1016/j.pharmthera.2013.01.016.
6. Alzoubi, K.M.; Wan, P.J.; Frieder, O. Distributed heuristics for connected dominating set in wireless ad hoc networks. *IEEE J. Commun. Netw.* **2002**, *4*, 22–29, doi:10.1109/JCN.2002.6596929.
7. Butenko, S.; Cheng, S.; Du, D.; Pardolos, P.M. On the construction of virtual backbone for ad hoc wireless network. In *Cooperative Control: Models, Applications and Algorithms*; Springer: Massachusetts, MA, USA, 2003; pp. 43–54.
8. Clark, B.; Colbourn, C.; Johnson, D. Unit disk graphs. *Discrete Math.* **1990**, *86*, 165–177.
9. Du, D.-Z.; Wan, P.-J. *Connected dominating set: theory and applications*; Springer Optimization and its Applications: New York, NY, USA, 2013; ISSN: 1931-6828.
10. Guha, S.; Khuller, S. Approximation algorithms for connected dominating set. *Algorithmica* **1998**, *20*, 374–387.
11. Rai, M.; Verma, S.; Tapaswi, S. A Power aware minimum connected dominating set for wireless sensor networks. *J. Netw.* **2009**, *4*, 511–519.
12. Gandhi, R.; Parthasarathy, S. Distributed algorithms for connected domination in wireless networks. *J. Parallel Distrib. Comput.* **2007**, *67*, 848–862.
13. Ambühl, C.; Erlebach, T.; Mihalák, M.; Nunkesser, M. Constant-factor approximation for minimum-weight (connect) dominating sets in unit disk graphs. *Lect. Notes Comput. Sci.* **2006**, *4110*, 3–14.
14. Huang, Y.C.; Gao, X.F.; Zhang, Z.; Wu, W.L. A better constant-factor approximation for weighted dominating set in unit disk graph. *J. Comb. Optim.* **2008**, *18*, 179-194.
15. Shi, Y.; Zhang, Z.; Du, D.Z. Approximation algorithm for minimum weight (k,m)-CDS problem in unit disk graph. *IEEE/ACM Trans. Netw.* **2017**, *25*, 925–933, doi:10.1109/TNET.2016.2607723.
16. Rashevsky, N. Life, information theory, and topology. *Bull. Math. Biophys.* **1955**, *17*, 229–235.
17. Mowshowitz, A. Entropy and the complexity of graphs: I. An index of the relative complexity of a graph. *Bull. Math. Biophys.* **1968**, *30*, 175–204.
18. Mowshowitz, A.; Dehmer, M. Entropy and complexity of graphs revisited. *Entropy* **2012**, *14*, 559–570.
19. Kajdanowicz, T.; MMorz, M. Using graph and vertex entropy to compare empirical graphs with theoretical graph models. *Entropy* **2016**, *18*, 320, doi:10.3390/e18090320.
20. Ai, X. Node importance ranking of complex networks with entropy variation. *Entropy* **2017**, *19*, 303, doi:10.3390/e19070303.
21. Shannon, C.E. A Mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423, 623–656.
22. Watts, D.J.; Strogatz, S.H. Collective dynamics of 'small-world' networks. *Nature* **1998**, *393*, 440–442.