Approximation Algorithms for D-hop Dominating Set Problem

Alina Campan, Traian Marius Truta, and Matthew Beckerich

***Abstract*—Social networks are increasingly used today and their influence competes with well-established media outlets such as television and radio. The problem that we address in this paper is determining efficiently a good approximation of the network minimum *d*-hop dominating set. For this, we extend three existing dominating set algorithms to the context of d-hop dominating set problem and we perform a series of experiments on both real and synthetic data using these algorithms. For algorithms extension, we generate the d-closure of the network before applying the existing algorithms to determine good approximations to the *d*-hop dominating set problem. For experiments, we use several real networks datasets made available by the Stanford Network Analysis Project and we also generate synthetic networks using both random and power-law models. Our experiments show that the selected algorithms can efficiently find quality approximations for the minimum-size *d*-hop dominating set problem. The best algorithm choice is dependent on the network characteristics and the order of importance between the size of the *d*-hop dominating set and the time required to determine such a set.**

I. Introduction

SOCIAL networks are increasingly used today and their influence competes with well-established media outlets such as television and radio. For instance, social networks are used extensively for political campaigns such as United States 2016 republican and democratic primaries [20]. Even in authoritarian regimes social networks have an increasing influence over the population and they are used to increase the political awareness of users in such countries [18]. As a result, social networks are extensively studied in order to find methods to maximize such influences.

One problem that was extensively studied in the literature is determining efficiently the network minimum dominating set. We formalize this problem as follows. For $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges, a set of nodes $\mathcal{DS} \subseteq \mathcal{N}$ is called a ***dominating set*** in $\mathcal{G}$ if every node $x \in \mathcal{N}$ is either in $\mathcal{DS}$ or is adjacent to a node in $\mathcal{DS}$ [3]. A dominating set with the smallest size is called ***minimum dominating set*** ($\mathcal{MDS}$). This problem has been shown to be an NP-complete problem [10], and various approximation algorithms have been proposed in order to obtain dominating sets close in size to the minimum dominating set [6, 9, 14, 16, 17].

In some networks dominating sets tend to be large and it will be enough if every node is not by the immediate neighbor but by node that lies at a distance of at most *d* from the original node. This modified domination problem was already introduced in the literature in the context of wireless networks [1]. Formally, a set $\mathcal{HDS}$ of nodes from $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ is called a ***d-hop dominating set*** if every node $x \in \mathcal{N}$ is either in $\mathcal{HDS}$ or there is a path between *x* and a node in $\mathcal{HDS}$ of size at most *d*. A *d*-hop dominating set with the smallest size is called ***minimum d-hop dominating set*** ($\mathcal{MHDS}$). Similarly to the original dominating set problem, determining $\mathcal{MHDS}$ is an NP-complete problem [1].

While the general minimum dominating set has received a lot of attention, there are fewer researchers that addressed the *d*-hop dominating set problems. As already mentioned before, this problem was introduced in [1] in order to determine the clusterheads in a wireless network. A local heuristic algorithm has been introduced to determine such clusterheads based on local cooperation between wireless nodes [1]. Newer distributed algorithms for producing a *d*-hop dominating set in a wireless network were analyzed in the wireless networks community [13, 19]. Outside wireless networking field, this problem is also analyzed from a specific graph model (directed graph with indegree bounded by one) in [2]. In the social networking context the closest work we are aware of is the problem of viral marketing where a *d*-hop dominating set can be seen as the subset of vertices that can activate all vertices in the network within at most *d* rounds [8]. Variants of *d*-hop dominating set problem also exists such as *d*-hop *k*-dominating set, connected *d*-hop dominating set, and connected *d*-hop *k*-dominating set [21].

Our contributions to this paper are to extend three dominating set algorithms from [6] to the context of *d*-hop dominating set problem and to perform a series of experiments on both real and synthetic data. For the first contribution, we use the *d*-closure of the social network [19] (see complete presentation in Section II) in order to apply the existing algorithms to determine good approximations to the dominating set problem. For experiments, we used several real networks datasets made available by the Stanford Network Analysis Project [10] and we also generated synthetic networks using both random and power-law models.

The remaining of this paper is structured as follows. Section II presents the three algorithms that are used in this paper. Section III describes the social networks used in our experiments the results of our experiments. Section IV presents conclusions and future work directions.

A. Campan (e-mail: campana1@nku.edu phone: +1-859-572-5776; fax: +1-859-572-5398,), T. M. Truta (e-mail: trutat1@nku.edu), and M. Beckerich (e-mail: beckerichm1@mymail.nku.edu) are with the Department of Computer Science, Northern Kentucky University, Nunn Drive, Highland Heights, KY 41099, USA.

II. D-HOP DOMINATING SETS ALGORITHMS

For a social network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges, we construct the network $\mathcal{G}_d = (\mathcal{N}, \mathcal{E}_d)$, where $\mathcal{E}_d = \{(x, y) \mid \delta(x, y) \le d \text{ and } x \ne y\}$ and $\delta(x, y)$ represents the distance from the original network $\mathcal{G}$ between vertices x and y. The network $\mathcal{G}_d$ is called the *d*-closure of $\mathcal{G}$ [19]. We illustrate in Fig. 1 a network and its 2-closure. The newly added edges are represented by red dashed lines.

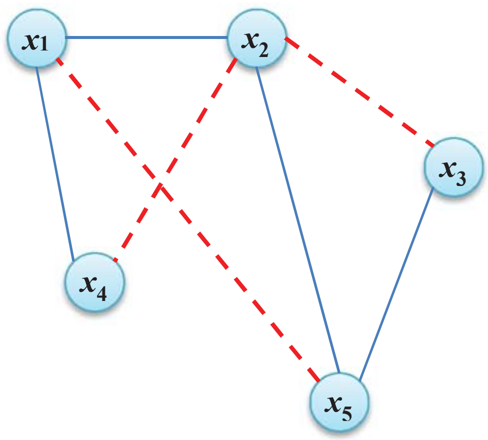

Fig. 1. A network $\mathcal{G}$ and its 2-closure, $\mathcal{G}_2$. The original edges from $\mathcal{G}$ are dark blue and the newly added edges in $\mathcal{G}_2$ are dashed red.

Based on the above construction, it is trivial to notice that any dominating set in $\mathcal{G}_d$ is a *d*-hop dominating set in $\mathcal{G}$, thus we can determine a *d*-hop dominating set in two easy steps. First, we construct the *d*-closure of the original network, and second we apply any existing dominating set algorithm to the newly created $\mathcal{G}_d$ network.

We use in this paper three algorithms for efficiently finding a close approximation to the minimum *d*-hop dominating set problem. These algorithms were presented in [6] for the general dominating set problem and we briefly describe them in the following paragraphs. We also include in these algorithms the creation of the *d*-closure network.

After the generation of the $\mathcal{G}_d$ network, the first algorithm (***Alg. 1***) selects in a for loop one node that covers the maximum number of currently uncovered nodes in the network. To determine if the nodes are in the dominating set, covered by a node in the dominating set, or not yet covered, we use a coloring scheme. Namely, nodes in the dominating set are colored as *black*, nodes covered by existing nodes in the dominating set are *gray*, and the remaining nodes are *white*. $\mathcal{W}(x)$ denotes the set of white nodes among the direct neighbors of x, including x itself.

```
Alg. 1 (G, HDS) is
  Input:   G = (N, E) - a social network;
           d - the d-hop parameter;
  Output:  HDS - a d-hop dominating set for G;
  Create Gd = (N, Ed) - the d-closure of G.
  HDS = ∅;
  while □ white nodes in Gd do
    choose v □ {x □ N | w(x) = max u □ N |W(u)|};
    HDS:= HDS ∪ { v };
    For all neighbors x of v in Gd
      Color x as gray;
    End for;
    // Delete the vertex v and its adjacent edges.
    Gd = (N\{v}, Ed\{(x, y) □ Ed| x = v or y = v)});
  end while;
end Alg. 1.
```

In the second algorithm (***Alg. 2***) after the *d*-closure generation $\mathcal{G}_d$, we aim to improve the running time of ***Alg. 1*** by removing both the black and gray nodes from the remaining network. In this algorithm the use of node coloring is no longer useful. The degree function represents the number of adjacent nodes in the *d*-closure network. Its pseudocode is shown below:

```
Alg. 2 (G, HDS) is
  Input:   G = (N, E) - a social network;
           d - the d-hop parameter;
  Output:  HDS - a d-hop dominating set for G;
  Create Gd = (N, Ed) - the d-closure of G.
  HDS = ∅;
  while N ≠ ∅ do
    choose v □ {x □ N |
              degree(x) = max u □ N (degree(u))};
    HDS:= HDS ∪ { v };
    // Delete v, Neighbors({v}), and their edges.
    // The remaining edges are labeled E'
    Gd = (N\{v}\Neighbors({v},E');
  end while;
end Alg. 2.
```

In the last algorithm (***Alg. 3***) after the *d*-closure generation $\mathcal{G}_d$, we consider the nodes in $\mathcal{N}$ in non-increasing order of the degree (again from the *d*-closure network) $v_1, v_2, \ldots, v_{|\mathcal{N}|}$, with $degree(v_1) \ge degree(v_2) \ge \ldots \ge degree(v_{|\mathcal{N}|})$. We pick the smallest i such that $|\cup_{j\le i} Neighbors\ (v_j)| = |\mathcal{N}|$ and we take the subset $\{v_1, v_2, \ldots, v_i\}$ to be the *d*-hop dominating set of the graph. From this set we exclude those v_i nodes for which $Neighbors\ (v_i) \subseteq \cup_{j<i} Neighbors\ (v_j)$. The pseudocode is presented next:

```
Alg. 3 (G, HDS) is
  Input:   G = (N, E) - a social network;
           d - the d-hop parameter;
  Output:  HDS - a d-hop dominating set for G;
  Create Gd = (N, Ed) - the d-closure of G.
  HDS = ∅;
  Order vertices in N = {v1, v2, … v|N|)}
    such that degree(v1) ≥ degree(v2) ≥ … ≥ degree(v|N|).
  i = 1;
  while (|∪j≤i Neighbors (vj)| < |N|) do
  if (not (Neighbors (vi) ⊆ ∪j<i Neighbors (vj)))
      HDS = HDS ∪ { vi};
    i++;
end Alg. 3.
```

III. Performance Evaluation of the Algorithms

A. Real and Synthetic Datasets

For our experiments, we used as initial networks both real and synthetic network datasets using a similar approach as in [6].

We selected seven real networks from the SNAP datasets website [11]. A short description of these seven datasets is shown in Table 1.

We generated our synthetic networks using two distribution models, namely, *Erdos-Renyi random graph model* [4] and the *configuration model* [5, 15], which generates a scale free network [7]. These types of networks are referred as ***ERNetworks*** and ***ConfNetworks*** thought this section.

TABLE I
Real Networks' Characteristics

Name	$\|\mathcal{N}\|$	$\|\mathcal{E}\|$	Description
ego-Facebook	4,039	88,234	Social circles from Facebook
email-Enron	36,692	183,831	Email communication network from Enron
ca-AstroPh	18,772	198,110	Collaboration network of Arxiv Astro Physics
ca-CondMat	23,133	93,497	Collaboration network of Arxiv Condensed Matter
ca-GrQc	5,242	14,496	Collaboration network of Arxiv General Relativity
ca-HepPh	12,008	118,521	Collaboration network of Arxiv High Energy Physics
ca-HepTh	9,877	25,998	Collaboration network of Arxiv High Energy Physics Theory

For *ERNetworks* generation we chose the following parameters. Firstly, we fixed the size of the network, $|\mathcal{N}|$, to 5,000, and we generated corresponding networks for 10 different average degree values (AVG = 2, 4, 6, ..., 20). Secondly, we selected a fixed AVG value (AVG = 10) and we varied the size of the network ($|\mathcal{N}|$ = 1K, 2K, …, 10K). For each combination of $|\mathcal{N}|$ and AVG values, we generated 5 distinct networks with those parameters.

For *ConfNetworks* generation we chose the following parameters. Firstly, we fixed the size of the network, $|\mathcal{N}|$, to 5,000, and we generated corresponding networks for 10 different power exponent values (γ = 1.7, 1.8, …, 2.6). Secondly, we selected a fixed γ value (γ = 2) and we varied the size of the network ($|\mathcal{N}|$ = 1K, 2K, …, 10K). For each combination of $|\mathcal{N}|$ and γ values, we generated 5 distinct networks.

Using the above approach we generated a total of 190 synthetic networks (95-*ERNetworks* and 95-*ConfNetworks*).

B. Experimental Results

We implemented the three algorithms described in Section II as well as the synthetic graph generators using the SNAP framework [12]. For all the experiments we used an Intel® Xeon® E5430@2.66 GHz dual CPU machine with 4 GB memory running on 32-bit Windows Server 2007 operating system. For the synthetic networks, the results shown in this section are the average results of the five synthetic datasets generated with the same parameters.

Fig. 2 – 4 present some of the results computed for real network datasets. From Fig 2 we notice that in all presented cases ***Alg. 1*** outperforms the other two algorithms in terms of 2-hop dominating size. ***Alg. 2*** and ***3*** have similar results taking turns as the second best algorithm. In terms of running time, ***Alg. 3*** is the fastest algorithm, ***Alg. 1*** is usually the second fastest, and ***Alg. 3*** is with one exception (for *ca-CondMath*) the slowest algorithm. In Fig. 3, the *d*-hop dominating set sizes are presented for selected networks when *d* parameter varies. In all instances ***Alg. 1*** determines the smallest *d*-hop dominating set. In general ***Alg. 3*** is the second algorithm using this comparison criterion. We do notice that the difference in dominating set sizes is greater for d = 1, and decreases for d = 2 and 3. In terms of the running time (Fig. 4), ***Alg. 3*** is the clear winner. With few exceptions, ***Alg. 1*** is also faster than ***Alg. 2***.

Fig. 5 and 6 show some of the results for *ERNetworks*. In Fig 5 the size of the initial network is kept constant (5,000), while in Fig. 6 the average degree is fixed at 10. The results for the *d*-hop dominating sets sizes are very similar between the two types of *ERNetworks*. In all cases the smallest dominating set size is determined by ***Alg. 1***, followed by ***Alg. 2***, and ***Alg. 3***. In terms of running time, ***Alg. 3*** is the fastest algorithm. However, ***Alg. 1*** outperforms ***Alg**. **2*** for network size constant, while ***Alg. 2*** performs better when the average degree increases. Based on the fact that when the degree increases the distance between running time of ***Alg. 1*** and ***Alg. 2*** increases we conclude that average degree is the best indicator of which algorithm between ***Alg. 1*** and ***Alg. 2*** is the fastest, and ***Alg. 2*** is faster for smaller average degrees, while ***Alg. 1*** is faster for larger average degrees. Since all algorithms are applied to the *d*-closure network, note that the average degree of *d*-closure is the determining factor. Of course this value is influenced by the initial average degree.

Fig. 7 and 8 show some of the results for *ConfNetworks*. In Fig 7 the size of the initial network is kept constant (5,000), while in Fig. 6 the power exponent is fixed at 2.0. The results for the *d*-hop dominating sets sizes are very similar between the two types of *ERNetworks*. In terms of size, ***Alg. 1*** is followed closely by ***Alg. 3***, while ***Alg. 2*** is the clear looser. In terms of running time, ***Alg 3*** outperforms ***Alg. 2*** and ***Alg. 1***. Between the trailing algorithms, ***Alg. 1*** is faster.

Overall, ***Alg. 1*** is the best algorithm in terms of minimizing the *d*-hop dominating set size. ***Alg. 3*** is usually the second best algorithm in this category. The order in reversed for running time, ***Alg. 3*** having the edge over ***Alg. 1***. With several exceptions pointed out earlier, ***Alg. 2*** performs worse that ***Alg. 1*** and ***Alg. 3*** in terms of both dominating set size and running time. In conclusion, both ***Alg. 1*** and ***Alg. 3*** are good candidates for determining fast an acceptable approximation of the minimum *d*-hop dominating set. Depending of which criterion is more important (size of the dominating set or running time) one of these algorithms can be selected as the best choice.

IV. Conclusions

In this paper we presented three algorithms that determine an approximation of the minimum d-hop dominating set for a social network. We performed extensive experiments using these three algorithms on both synthetic and real networks. We concluded that the best choice between the three algorithms is dependent of the network characteristics and which criterion is more important (size of the dominating set or running time) for the person that wants to determine d-hop dominating sets approximations. Regardless of these factors, the choice is between ***Alg. 1*** and ***3***, and in most cases, ***Alg. 1*** should be used when the size of the d-hop dominating sets is the major factor, while ***Alg. 3*** should be used when the running time is the most important factor.

As part of our future work, we intend to investigate how the proposed new algorithms can be used for more specific d-hop dominating set problems, such as connected d-hop dominating set, d-hop k-dominating set, and connected d-hop k-dominating set problems [21].

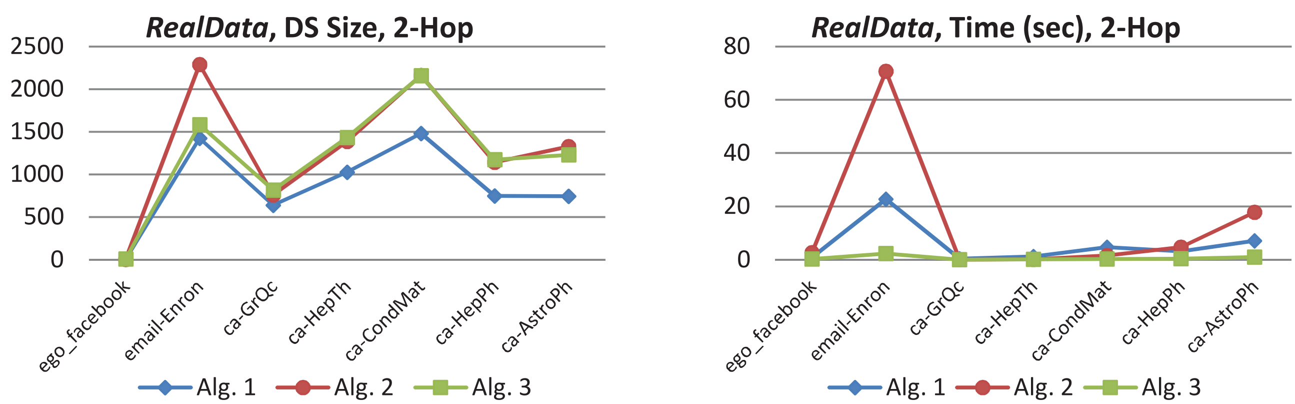

Fig. 2. The 2-hop dominating set results for real data networks. On the left the sizes of the determined 2-hop dominating sets are reported for all three algorithms. On the right, the running time of these algorithms is shown.

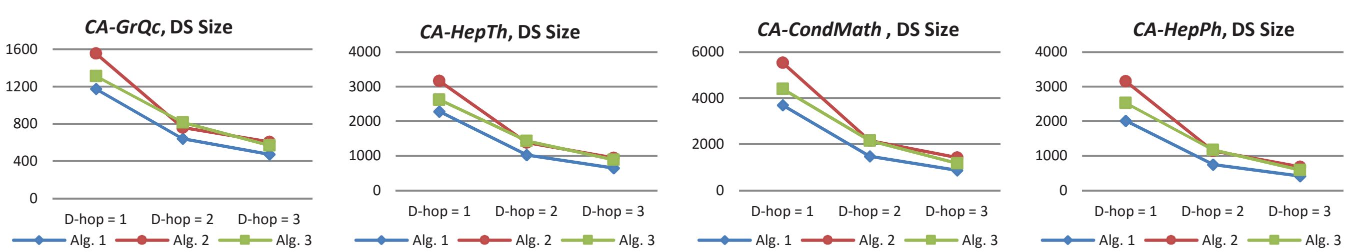

Fig. 3. The sizes of the d-hop dominating sets are shown for *CA-GrQc*, *CA-HepTh*, *CA-CondMath*, and *CA-HepPh* networks.

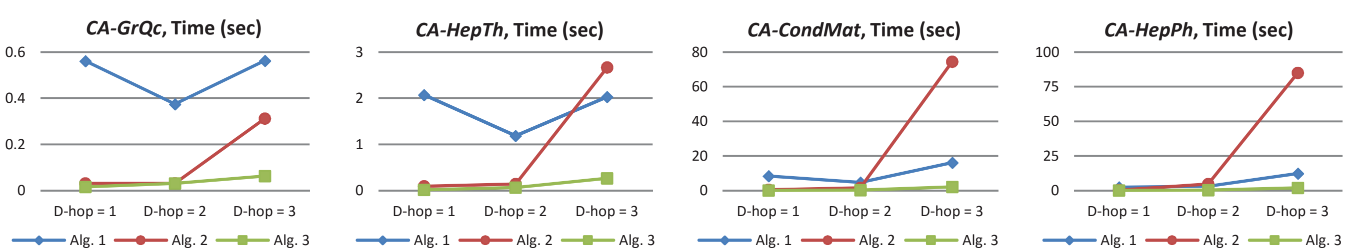

Fig. 4. The running time of running the three algorithms to determine the size of the d-hop dominating set for *CA-GrQc*, *CA-HepTh*, *CA-CondMath*, and *CA-HepPh* networks is shown.

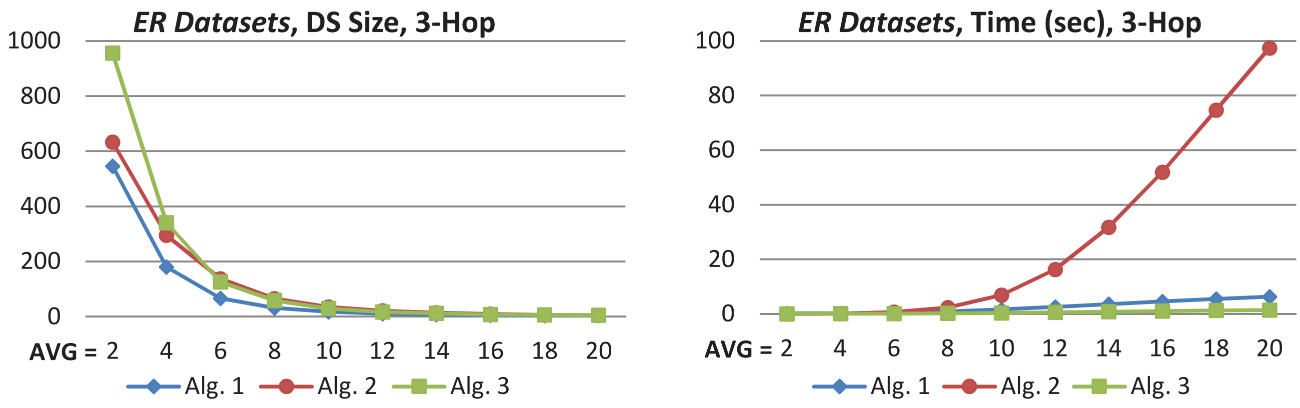

Fig. 5. The 3-hop dominating set results for *ERNetworks* when the number of nodes is constant ($\mathcal{N}$=5,000). On the left the sizes of the determined 3-hop dominating sets are reported for all three algorithms. On the right, the running time of these algorithms is shown.

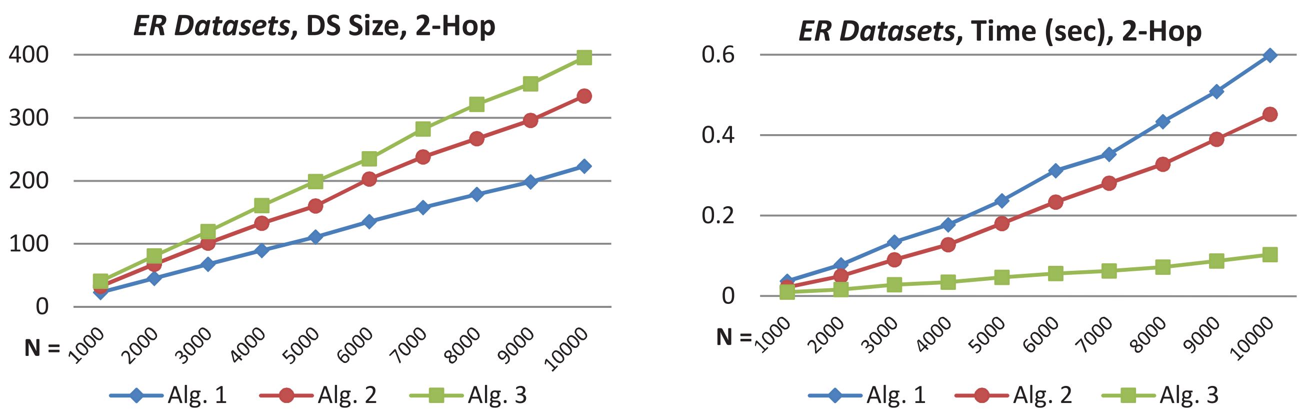

Fig. 6. The 2-hop dominating set results for *ERNetworks* when the average degree is constant (AVG = 10). On the left the sizes of the determined 2-hop dominating sets are reported for all three algorithms. On the right, the running time of these algorithms is shown.

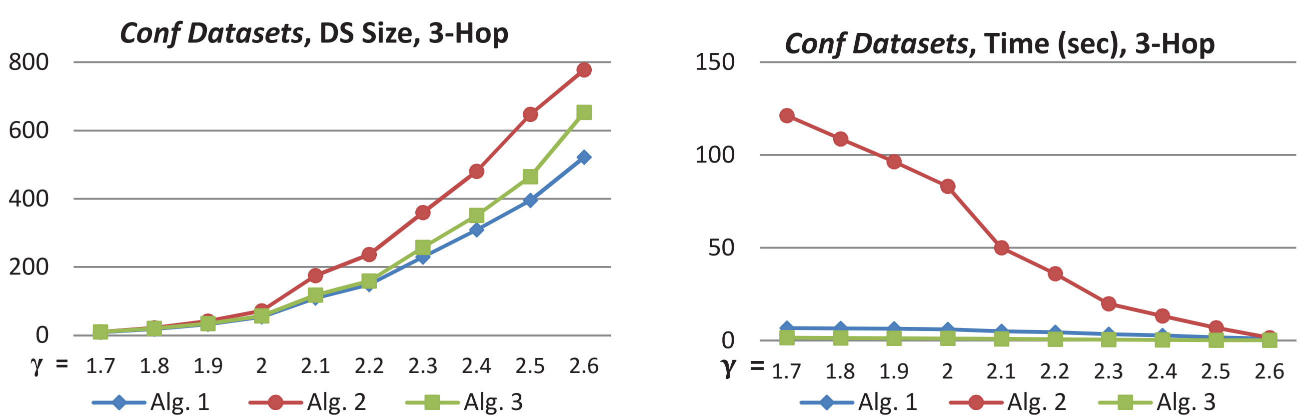

Fig. 7. The 3-hop dominating set results for *ConfNetworks* when the number of nodes is constant ($\mathcal{N}$=5,000). On the left the size of the determined 3-hop dominating set is reported for all three algorithms. On the right, the running time of these algorithms is shown.

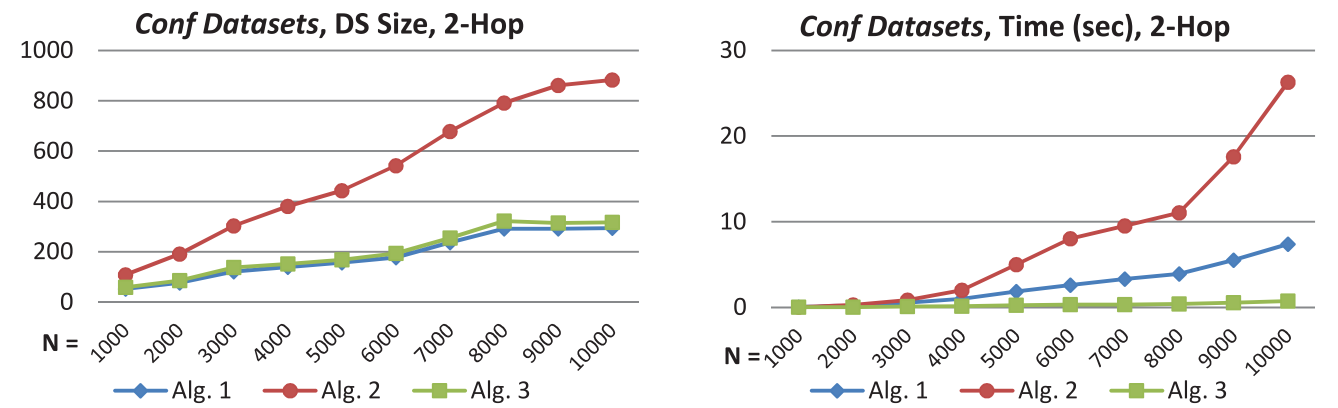

Fig. 8. The 2-hop dominating set results for *ConfNetworks* when the power exponent is constant ($\gamma = 2.0$). On the left the size of the determined 2-hop dominating set is reported for all three algorithms. On the right, the running time of these algorithms is shown.

REFERENCES

[1] A. D. Amis, R. Prakash, T. H. P. Vuong, and D. T. Huynh, "Max-min d-cluster formation in wireless ad hoc networks," In *Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Vol. 1, pp. 32-41, 2000.

[2] J. Banerjee, A. Das, and A. Sen, "Analysis of d-hop dominating set problem for directed graph with indegree bounded by one," arXiv preprint arXiv:1404.6890, 2014.

[3] C. Berge, "Theory of graphs and its applications" Methuen, London, 1962.

[4] B. Bollobás, "Random graphs (2nd ed.)," Cambridge University Press. ISBN 0-521-79722-5, 2001.

[5] T. Britton, M. Deijfen, and A. Martin-Lof, "Generating simple random graphs with prescribed degree distribution," Journal of Statistical Physics, Vol. 124, Issue 6, 1377-1397, 2006.

[6] A. Campan, T. M. Truta, and M. Beckerich, "Fast dominating set algorithms for social networks," In *Proc. of the Modern Artificial Intelligence and Cognitive Science Conference (MAICS)*, pp. 55-62, 2015.

[7] M. Catanzaro, M. Boguna, and R. Pastor-Satorras, "Generation of uncorrelated random scale-free networks," Physical Review, E 71, 027103, 2005.

[8] T. N. Dinh, D. T. Nguyen, and M. T. Thai, "Cheap, easy, and massively effective viral marketing in social networks: truth or fiction?," In *Proc. of the 23rd ACM conference on Hypertext and social media*, 2012.

[9] S. Eubank, V. S. Anil Kumar, M.Marathe, A. Srinivasan, and N. Wang, "Structural and algorithmic aspects of massive social networks," In *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, pp. 718–727, 2004.

[10] M. R Garey and D. S. Johnson, "Computers and intractability: A guide to the theory of NP-completeness," W.H. Freeman, ISBN 0-7167-1045-5, page 190, 1979.

[11] J. Leskovec and A. Krevl, "SNAP datasets: Stanford large network dataset collection," 2014, Available at: http://snap.stanford.edu/data.

[12] J. Leskovec and R. Sosic, "SNAP: A general purpose network analysis and graph mining library in C++," 2014, Available at: http://snap.stanford.edu/snap.

[13] X. Li, X. Gao, and C. Zhao, "A new greedy algorithm for D-hop connected dominating set," In *Proc. of the 10th International Conference on M obile Ad-hoc and Sensor Networks (MSN)*, 2014.

[14] L. Lovasz, "On the ratio of optimal integral and fractional covers," Discrete Mathematics, Vol. 13, 383–390, 1975.

[15] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," Random Structures and Algorithms, Vol. 6, 161-180, 1995.

[16] F. Molnar, S. Sreenivasan K. Szymanski, and G. Korniss, "Minimum dominating sets in scale-free network ensembles," Scientific Reports, No. 3, Article number 1736, Nature Publishing Group, 2013.

[17] J. C. Nacher and T. Akutsu, "Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control," New Journal of Physics, Vol. 14, 2012.

[18] O. J. Reuter and D. Szakonyi, "Online social media and political awareness in authoritarian regimes," Working Paper, Series: Political Science, WP BRP 10/PS/2012, National Research University, 2012, Available online at: https://iims.hse.ru/data/2013/01/18/1305921529/10PS2012.pdf .

[19] M. Q. Rieck, S. Pai, and S.Dhar, "Distributed routing algorithms for multi-hop ad hoc networks using d-hop connected d-dominating sets," Computer Networks, Vol. 47, No. 6, pp. 785-799, 2005.

[20] S. Whitten, "Can 140 characters affect the 2016 presidential election?," in CNBC, 2015, Available online at: http://www.cnbc.com/2015/11/05/can-140-characters-affect-the-2016-presidential-election.html.

[21] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks – A comprehensive survey," Computer Communications, Vol. 36, No. 2, pp. 121-134, 2013.